

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA DE RECONHECIMENTO DE GESTOS DA
MÃO

RODRIGO BAMBINETI

BLUMENAU
2009

2008/2-24

RODRIGO BAMBINETI

**FERRAMENTA DE RECONHECIMENTO DE GESTOS DA
MÃO**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Dalton Solano dos Reis, M.Sc. - Orientador

**BLUMENAU
2009**

2008/2-24

FERRAMENTA DE RECONHECIMENTO DE GESTOS DA MÃO

Por

RODRIGO BAMBINETI

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Dalton Solano dos Reis, M.Sc. – Orientador, FURB

Membro: _____
Prof. Paulo Cesar Rodacki Gomes, Dr. – FURB

Membro: _____
Prof. Roberto Heinzle, M.Sc. – FURB

Blumenau, 12 de fevereiro de 2009

Dedico este trabalho a sociedade e a todos os amigos e familiares que estiveram comigo durante a realização deste, incentivando e apoiando.

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

À minha família que sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, Dalton Solano dos Reis, por ter contribuído e acreditado na conclusão deste trabalho.

RESUMO

Este trabalho apresenta o desenvolvimento de uma ferramenta para realizar o reconhecimento de gestos da mão. Pesquisas mostram que informações visuais são responsáveis pela metade do conhecimento adquirido e as expressões corporais representam 80% das impressões durante uma iteração. O trabalho é dividido em duas fases, o tratamento de imagens e reconhecimento. Dentre as técnicas de tratamento de imagens aplicadas, operações morfológicas e a esqueletonização são destacadas. As operações morfológicas realizam o embelezamento do objeto da imagem. A esqueletonização é baseada no algoritmo de Zhang-Suen, onde o esqueleto do gesto da mão é obtido e submetido para a fase de reconhecimento. Uma adaptação do cálculo da distância Baddeley compara duas imagens informando um valor de aproximação. A combinação dessas técnicas possibilitou o desenvolvimento da ferramenta de reconhecimento de gestos, baseando-se em ambientes controlados.

Palavras-chave: Segmentação da região da mão. Visão computacional. Histograma de orientação. Esqueletonização.

ABSTRACT

This paper presents the development of a tool to achieve the recognition of hand gestures. Research shows that visual information are responsible for half of the knowledge and bodily expressions represent 80% of impressions during an iteration. The paper is divided into two phases, the Image Processing and Recognition. Among the image processing techniques applied, morphological operations and skeletonization are highlighted. The morphological operations perform the beautification of the object's image. The skeletonization is based on the Zhang-Suen's algorithm, where the skeleton of the hand gesture is made and submitted to the fase of Recognition. An adaptation of the calculation of the Baddeley distance compares two images entering a value in approximation. The combination of these techniques enabled the development of the tool for recognition of gestures, based on controlled environments.

Keywords: Hand region segmentation. Computer vision. Orientation histogram. Skeletonization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo do término da palavra igreja em LIBRAS	15
Figura 2 – Imagens em tons de cinza e seus respectivos histogramas.....	18
Figura 3 – Intervalos obtidos a partir do limiar T.....	19
Figura 4 – Intervalos obtidos a partir de dois limiares T_1 e T_2	19
Figura 5 – Extração do esqueleto.....	21
Figura 6 – Demonstração de tamanho de diferentes esqueletos.....	21
Figura 7 – Localização, vizinhança e conectividade de pixel	22
Figura 8 – Diagrama do processo de reconhecimento de gestos.....	25
Figura 9 – Extração da região da mão.....	26
Figura 10 – Taxa de reconhecimento de gestos estáticos	27
Figura 11 – Diagrama de casos de uso.....	29
Quadro 1 – Caso de uso treinar amostras.....	30
Quadro 2 – Caso de uso reconhecer gestos	31
Quadro 3 – Caso de uso tratar imagens.....	32
Figura 12 – Diagrama de classes da interface	33
Figura 13 – Diagrama de classes das funções	34
Quadro 4 – Processamento da imagem capturada	36
Quadro 5 – Cálculo da orientação local.....	39
Quadro 6 – Cálculo do histograma de orientação - magnitude e orientação.....	40
Quadro 7 – Esqueletonização de Zhang-Suen.....	41
Quadro 8 – Reconhecimento	43
Quadro 9 – Distância de Baddeley	45
Figura 14 – Equação de Baddeley	45
Figura 15 – Ferramenta durante a fase de treinamento.....	46
Figura 16 – Ferramenta durante a fase de reconhecimento.....	47
Figura 17 – Imagem representando a letra V e cada passo do tratamento de imagem.....	51
Figura 18 – Imagem representando a letra B exposta a diferentes pontos de luz.....	52
Figura 19 – Aplicação de operações morfológicas das imagens da Figura 18.....	53
Figura 20 – Esqueletos representando das imagens da Figura 18	53
Figura 21 – Histogramas de orientação das imagens da Figura 18	54

LISTA DE TABELAS

Tabela 1 – Relação de vizinhança e conectividade.....	23
---	----

LISTA DE SIGLAS

2D – *2 Dimensions*

GPL – *General Public License*

LIBRAS – Língua BRAsileira de Sinais

RGB – Acrônimo de vermelho (*Red*), verde (*Green*) e azul (*Blue*).

RF – Requisitos Funcionais

RNA – Rede Neural Artificial

RNF – Requisitos Não-Funcionais

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS DO TRABALHO.....	13
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 DICIONÁRIO DE LIBRAS	15
2.2 VISÃO COMPUTACIONAL	16
2.2.1 Processamento de imagens.....	16
2.2.2 Análise de imagens	17
2.3 SEGMENTAÇÃO	17
2.3.1 Histograma	18
2.3.2 Limiarização	18
2.4 OPERAÇÕES MORFOLÓGICAS.....	19
2.5 EXTRAÇÃO DO ESQUELETO.....	20
2.5.1 Algoritmo de Zhang-Suen.....	22
2.6 RECONHECIMENTO DE GESTOS DA MÃO USANDO O ESQUELETO.....	23
2.6.1 Pré-processamento das imagens da extração de feições	25
2.6.2 Extração de feições de gestos estáticos.....	26
2.6.3 Treinamento e classificação	27
3 DESENVOLVIMENTO DA FERRAMENTA.....	28
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO	28
3.2 ESPECIFICAÇÃO	28
3.3 IMPLEMENTAÇÃO	35
3.3.1 Técnicas e ferramentas utilizadas	35
3.3.1.1 Cálculo do histograma de orientação local	38
3.3.1.2 Esqueletonização	40
3.3.1.3 Classificação	42
3.3.2 Operacionalidade da implementação	45
3.3.2.1 Treinamento	45
3.3.2.2 Reconhecimento	46
3.4 RESULTADOS E DISCUSSÃO	48
3.4.1 Implementação.....	48

3.4.1.1 Requisitos.....	51
3.4.2 Testes	52
4 CONCLUSÕES	56
4.1 EXTENSÕES	57
REFERÊNCIAS BIBLIOGRÁFICAS	58

1 INTRODUÇÃO

A fala é sem dúvida a maneira mais prática de comunicação entre pessoas. Algumas pessoas possuem limitações nas cordas vocais, sejam elas oriundas de doenças naturais, deficiências auditivas ou causadas por acidentes. Tais pessoas necessitam aprender outra forma de comunicação, que é a língua de sinais.

Segundo Viader, Pertusa e Vinardell (1999, p. 54), a língua de sinais é a “língua própria das pessoas surdas, usando sua estrutura, sintaxes e gramáticas próprias, sem o uso simultâneo e alternativo da língua falada. Respeita-se seu *status* lingüístico como língua. Se expressa com elementos prosódicos e reflexões próprias”. No Brasil a língua de sinais é conhecida como LIBRAS.

Uma justificativa para o empenho em aprender a língua de sinais, é que a linguagem corporal é responsável por 80% das nossas impressões durante uma interação, e que as pessoas aprendem mais da metade do que sabem a partir de informação visual (FISCHER, 2000, p. 18). É comum deparar-se na rua com uma pessoa fazendo gestos para outro indivíduo, indicando localizações, quantidades, sinais de confirmação e até mesmo certas ofensas. Alguns destes sinais possuem o mesmo significado expressado em LIBRAS, outros não.

Ocorrem situações em que uma pessoa que fala normalmente e um libriano¹ são postos para conversar e apesar da similaridade dos comportamentos, o diálogo provavelmente não ocorre. O libriano com o passar do tempo adquire a capacidade de interpretar a fala, apenas pelos movimentos de boca e lábios. O contrário não se aplica, sendo que uma pessoa que fala normalmente não tem o hábito de ler os lábios, pois a fala não causa a necessidade de observar e interpretar a leitura dos lábios.

O desafio proposto é utilizar um ambiente controlado para reconhecer os gestos de uma mão como os gestos de um libriano. Estes gestos devem ser processados através de imagens digitais, expressos através das configurações dos dedos, mãos e braços sem o uso de marcações especiais. O reconhecimento dos gestos pode ser feito inicialmente com imagens contendo apenas a região da mão, o que minimiza esforços na localização das partes do corpo humano, a fim de encontrar a região da mão. Esta forma de determinação é uma tarefa difícil, pois o ator pode assumir diferentes posturas, expressar-se em diferentes ambientes, estar

¹ Para melhor compreensão, chama-se assim, a pessoa que se expressa em LIBRAS.

usando roupas semelhantes ao tom de pele, entre outros fatores que dificultem o tratamento das imagens para posterior reconhecimento de gestos. Os gestos em LIBRAS constituem um artefato para validar este trabalho, podendo ser aplicado a um futuro software que auxilie os librianos.

Técnicas de tratamento de imagens são utilizadas para permitir isolar ruídos e separar informações utilizadas no reconhecimento dos gestos. Mesmo com a separação destas informações, o reconhecimento não é uma tarefa trivial, muitas vezes exigindo-se a utilização de técnicas de inteligência artificial, por exemplo, Rede Neural Artificial (RNA), principalmente quando o ambiente não é controlado (imagem de fundo constante).

A manipulação sobre o ambiente torna-se uma premissa para viabilizar o reconhecimento do gesto da mão do corpo humano. Logo, ambientes variados, onde existem grande concentração de ruídos e variação de luminosidade tornariam o trabalho ainda mais complexo.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é reconhecer os gestos de um libriano expressando-se, através da submissão de uma imagem digital a uma ferramenta própria.

Os objetivos específicos do trabalho são:

- a) implementar técnicas de tratamento de imagens sem o uso de bibliotecas de terceiros;
- b) reconhecer gestos da região da mão sem a utilização de RNAs;
- c) reconhecer o gesto em tempo real usando imagens 320 x 240 pixels;
- d) realizar o reconhecimento de gestos em LIBRAS.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em quatro capítulos. O segundo capítulo contém a fundamentação teórica necessária para o entendimento do trabalho. Nele são discutidos tópicos relacionados sobre LIBRAS, visão computacional, processamento de imagens,

operações morfológicas, extração do esqueleto e por último o artigo de maior relevância para a construção da ferramenta.

O terceiro capítulo comenta sobre o desenvolvimento da ferramenta, onde são explanados os requisitos principais do problema trabalhado, a especificação contendo diagramas de casos de uso e classes. Também são feitos comentários sobre a implementação abrangendo as técnicas e ferramentas utilizadas, operacionalidade e por fim são comentados os resultados e discussão.

O quarto capítulo refere-se às conclusões do presente trabalho bem como e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os assuntos e técnicas utilizadas para o desenvolvimento da ferramenta de reconhecimento de gestos da mão. Na seção 2.1 são apresentados dicionários de libras, onde é possível consultar inúmeros gestos. Na seção 2.2 são apresentados conceitos e técnicas da visão computacional. Na seção 2.3 são apresentados os conceitos de histograma e limiarização de imagens digitais. Na seção 2.4 são apresentadas técnicas de eliminação de ruídos através de operações morfológicas. Na seção 2.5 são apresentados o conceito e a técnica de esqueletonização utilizada. Na seção 2.6 é apresentado o artigo que viabilizou a realização deste trabalho.

2.1 DICIONÁRIO DE LIBRAS

Alguns dicionários de LIBRAS estão disponíveis na internet, como por exemplo os presentes nos *sites*: Acessobrasil (2008), Ines (2008) e o Dicionariolibras (2008). Além de possuir o significado das palavras, estes dicionários mostram imagens ou vídeos com duração de alguns segundos demonstrando a expressão em LIBRAS das palavras pesquisadas. Abaixo segue um exemplo do *site* Acessobrasil (Figura 1).

The screenshot shows the interface of the 'LIBRAS Dicionário da Língua Brasileira de Sinais' website. At the top, it says 'versão 2.0 - web - 2008'. Below the header, there are navigation tabs for 'Ordem' (Alfabetica, Por Assunto, Mão) and a search bar. A horizontal menu lists letters from # to Z. The main content area is divided into four columns: 'Assuntos', 'Palavras', 'Acepção', and 'Vídeo'. The 'Palavras' column has a dropdown menu with 'IGREJA' selected. The 'Acepção' column shows the definition: 'Templo cristão; local de culto, cerimônias religiosas e orações.'. The 'Vídeo' column shows a woman performing the sign for 'IGREJA'. Below the main content, there are sections for 'Exemplo' (Eu frequento a igreja todos os domingos.), 'Exemplo Libras' (CASA^CRUZ EU FREQUÊNTAR DOMINGO TOD@.), 'Classe Gramatical' (SUBSTANTIVO), and 'Origem' (nacional). The footer includes the 'Acessibilidade Brasil' logo and website URL, along with credits for 'concepção e metodologia' and 'libras em cd'.

Fonte: Acessobrasil (2008).

Figura 1 – Exemplo do término da palavra igreja em LIBRAS

2.2 VISÃO COMPUTACIONAL

Pedrini e Schwartz (2008, p. 1) explicam que a visão computacional procura auxiliar a resolução de problemas complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas na imagem. Entretanto a decisão a ser tomada é condicionada pela seqüência dos passos relevantes que envolvem o processamento e a análise de imagens, sendo muitas vezes realizada por um operador humano que detém o conhecimento ou a experiência sobre o domínio da aplicação.

Para simplificar o estudo das tarefas envolvidas em visão computacional, dois níveis de abstração são normalmente estabelecidos: processamento de imagens (baixo nível) e análise de imagens (alto nível).

2.2.1 Processamento de imagens

“O processamento de imagens é a ramo da computação que transforma as imagens, assim como o processamento de dados é ramo da computação que transforma os dados” (CONCI, AZEVEDO e LETA, 2008, p. 4). O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador. O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas (Pedrini e Schwartz, 2008, p. 1-2).

Pedrini e Schwartz (2008, p. 1-2) afirmam que os métodos de baixo nível geralmente utilizam pouco conhecimento sobre o conteúdo ou a semântica das imagens. Esses métodos envolvem operações como a redução de ruído, o aumento de contraste, a extração de bordas e a compreensão de imagens. Esse processamento utiliza imagens capturadas por equipamentos bastante variados, como câmeras de vídeo, câmeras de fotográficas, tomógrafos médicos, satélites e outros. Para que a imagem possa ser processada em um computador, primeiro ela deve ser digitalizada e convertida em formato adequado.

2.2.2 Análise de imagens

Para Pedrini e Schwartz (2008, p. 1-2) a “Análise de imagens é, tipicamente, baseada na forma, na textura, nos níveis de cinza ou nas cores dos objetos presentes nas imagens”. O processo de análise de imagens traz consigo uma dificuldade, o caráter multidisciplinar, em que diversos domínios de conhecimento são mutuamente necessários para solucionar satisfatoriamente o problema, bem como geometria computacional, visualização científica, psicofísica, estatística, teoria da informação e muitas outras.

Pedrini e Schwartz (2008, p. 1-2) afirmam que os métodos de alto nível envolvem tarefas como segmentação de imagens em regiões ou objetos de interesse, descrição desses objetos de modo a reduzi-los a uma forma mais apropriada para representar o conteúdo da imagem e reconhecimento ou classificação desses objetos. Características comuns utilizadas para descrever objetos incluem contornos ou bordas, dimensões do objeto, relacionamento entre os objetos da imagem, cores, texturas e muitas outras.

2.3 SEGMENTAÇÃO

Um processo de segmentação que identifique corretamente a localização, a topologia e a forma dos objetos é um requisito de fundamental importância para que as informações resultantes de um sistema de análise de imagens sejam confiáveis (PEDRINI e SCHWARTZ, 2008, p. 152). As abordagens convencionais para a segmentação de imagens são normalmente baseadas nas propriedades básicas dos níveis de cinza da imagem, buscando detectar descontinuidades ou similaridades na imagem.

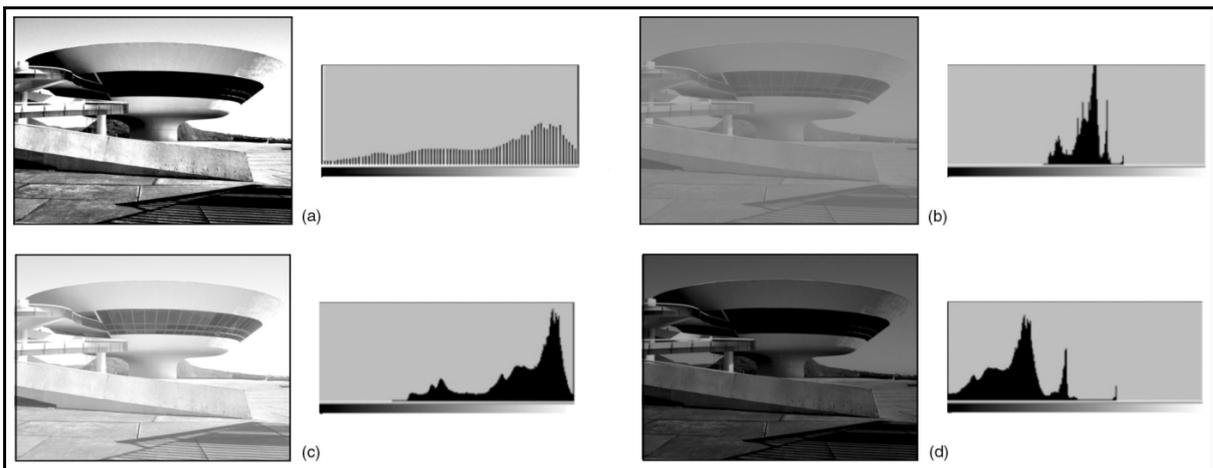
As técnicas de realce são uma subárea da segmentação e visam acentuar ou melhorar a aparência de determinadas características da imagem, como o segmento dos objetos, tornando-a mais adequada à aplicação em questão.

Um das informações mais relevantes em uma imagem é o seu histograma (seção 2.3.1), pois a partir dele é possível realizar segmentação da imagem de maneira simples.

2.3.1 Histograma

Conci, Azevedo e Leta (2008, p. 77-81) definem histograma como um conjunto de números indicando o percentual de pixels de uma imagem representando o nível de cinza (ou cor). Esses valores são normalmente representados por um gráfico de barras que fornece, para cada nível de cinza, o número ou percentual de pixels correspondentes na imagem. Em outras palavras é um gráfico cartesiano no qual o eixo horizontal reflete os valores das tonalidades de cinza que a imagem possui, e o eixo vertical, a frequência com que essa tonalidade ocorre.

Além de ser instrumento para realizar a limiarização (ver seção 2.3.2), existem indicadores de qualidade (ou predominância) quanto ao nível de contraste e quanto ao nível de luminosidade média, como demonstrados na Figura 2. No item (a) a imagem possui alto contraste enquanto o item (b) esta com baixo contraste, o item (c) e (d) possuem alta e baixa luminosidade respectivamente.



Fonte: adaptado de Conci, Azevedo e Leta (2008, p. 78).

Figura 2 – Imagens em tons de cinza e seus respectivos histogramas

2.3.2 Limiarização

Para Conci, Azevedo e Leta (2008, p. 81) a limiarização baseia-se muitas vezes na utilização do histograma. O exemplo mais simples consiste em uma imagem, na qual o objeto a ser segmentado apresenta uma tonalidade bem diferente do fundo da imagem. Se a aparência do histograma desta imagem for bimodal, será possível achar um limiar entre as duas tonalidades, que segmentará as regiões do objeto e do fundo.

Neste contexto simples, após encontrar um limiar τ é possível extrair os objetos do

fundo realizando a varredura em cada ponto da imagem (PEDRINI e SCHWARTZ, 2008, p. 182-183). Para cada ponto (x, y) tal que $f(x, y) > T$, é denominado um ponto do objeto, do contrário, é denominado um ponto do fundo. A Figura 3 mostra a equação que realiza esta separação.

$$g(x,y) = \begin{cases} 0, & \text{se } f(x,y) \leq T \\ 1, & \text{se } f(x,y) > T \end{cases}$$

Fonte: Pedrini e Schwartz (2008, p. 183).

Figura 3 – Intervalos obtidos a partir do limiar T

A limiarização da imagem pode ser definida de modo geral tal que, para cada intervalo, é especificado um nível de cinza corresponde T conforme mostrado na Figura 4.

$$g(x,y) = \begin{cases} I_1, & \text{se } f(x,y) \leq T_1 \\ I_2, & \text{se } T_1 < f(x,y) \leq T_2 \\ I_3, & \text{se } f(x,y) > T_2 \end{cases}$$

Fonte: Pedrini e Schwartz (2008, p. 183).

Figura 4 – Intervalos obtidos a partir de dois limiares T_1 e T_2

Quando é possível realizar um único corte para separar objetos do fundo, este processo é conhecido como limiarização global ou binarização. Ou seja, imagem resultante possui apenas duas cores o preto (0) e o branco (1). Porém este não é o cenário mais comum, devido a fatores como iluminação não-uniforme, ruídos, processo de aquisição de imagem não uniforme e muitos outros. Para estas outras situações utilizasse a limiarização local, que consiste em encontrar limiares dentro de subáreas da imagem.

2.4 OPERAÇÕES MORFOLÓGICAS

Wangenheim (2001) relata que a morfologia é a forma e estrutura de um objeto ou os arranjos e inter-relacionamentos entre as partes de um objeto. Os mais antigos usos desta palavra estão relacionados com linguagem e biologia. Em biologia, por exemplo, a morfologia esta relacionada mais diretamente à forma de um organismo.

Já a morfologia digital, é um caminho para descrever ou analisar a forma de um objeto digital. A morfologia digital é uma ciência relativamente recente, pois só os computadores digitais permitiram seu uso na prática. Por outro lado os matemáticos consideram a

morfologia como uma subárea da teoria de conjuntos que é uma ciência mais antiga, portanto mais conhecida e melhor estudada.

A idéia de morfologia digital é que uma imagem consiste de um conjunto de pixels que são reunidos em grupos tendo uma estrutura bidimensional (forma). Certas operações matemáticas em conjuntos de pixels podem ser usadas para ressaltar aspectos específicos das formas permitindo que sejam contadas ou reconhecidas.

A base da morfologia consiste em extrair as informações relativas a geometria e a topologia de um conjunto desconhecido (no caso uma imagem) pela transformação através de outro conjunto bem-definido, chamado elemento estruturante (máscara). Com isso torna importante ao contexto a utilização de teoria dos conjuntos, pois esta é a base utilizada na morfologia.

As operações básicas da morfologia digital são a erosão, em que pixels que não atendem a um dado padrão são apagados da imagem; e dilatação, em que uma pequena área relacionada a um pixel é alterada para um dado padrão. Todavia, dependendo do tipo de imagem sendo processada (preto e branco, tons de cinza ou colorida) a definição destas operações muda, assim cada tipo deve ser considerado separadamente. A partir destas duas operações outras duas podem ser realizadas, a abertura e o fechamento.

A operação de abertura em geral suaviza o contorno de uma imagem, quebra estreitos e elimina proeminências delgadas. A operação de abertura é usada também para remover ruídos da imagem. A obtenção da abertura é realizada por uma erosão entre a imagem e elemento estruturante, seguida de uma dilatação.

A operação de fechamento funde pequenas quebras e alargas golfos estreitos eliminando pequenos orifícios. Se uma abertura cria pequenos vazios na imagem, um fechamento irá preencher ou fechar os vazios, permitindo esta operação remover muitos dos pixels brancos que representam ruídos. A operação de fechamento é constituída pelas mesmas operações aplicadas a operação de abertura, porém na ordem inversa.

2.5 EXTRAÇÃO DO ESQUELETO

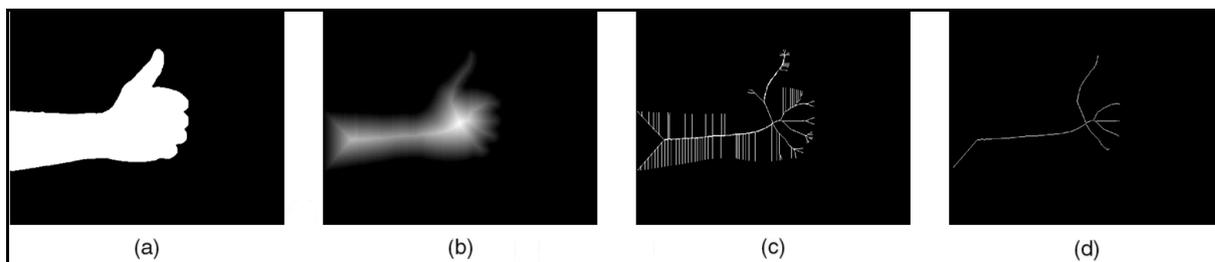
A técnica de esqueletonização (PLOTZE e BRUNO, 2004, p. 59) é freqüentemente utilizada para obter o esqueleto de uma região através do seu afinamento. Afinamento é o processo de redução de uma forma para uma versão simplificada que ainda retém as

características essenciais do objeto original. A versão afinada da forma é chamada de esqueleto.

Outra definição é dada por Ionescu et al (2005, p. 2105) que expõe um esqueleto como uma representação compacta de um objeto. As propriedades desejadas do esqueleto são a preservação da topologia do objeto (o mesmo número de componentes conectados e mesmo número de orifícios), robustez perante deslocamentos, rotação e alterações de escala, e espessura (não mais de dois pixels de largura).

A extração do esqueleto da mão é realizada através de um método de transformação baseado em distâncias, denominada transformação pela chanfradura² da distância. Este consiste em calcular as distâncias para extrair os eixos medianos da região. Em seguida, o conjunto resultante é ligado, a fim de preservar a topologia do objeto.

A Figura 5 mostra diferentes passos de extração do esqueleto da região da mão.



Fonte: Ionescu et al (2005, p. 2105).

Figura 5 – Extração do esqueleto

Onde (a) é imagem binária da região da mão, (b) a chanfradura da distância (branco corresponde à maior distância), (c) esqueleto obtido após ligação dos centros de discos máximos, e (d) o esqueleto obtido depois de preencher falsos buracos, podar, e “embelezar” o esqueleto.

Os esqueletos da região da mão podem variar significativamente como apresentado na Figura 6, onde as imagens (a e c) produzem esqueletos (b e d) correspondentes a região da mão. É possível notar que existe uma diferença considerável no tamanho entre os dois esqueletos.



Figura 6 – Demonstração de tamanho de diferentes esqueletos

²Chanfrar (FERREIRA, 2008, p. 454): cortar em semicírculo. Desfazer com a plaina as arestas ou quinas.

Plotze e Bruno (2004, p. 59-64) realizaram comparativos entre os algoritmos de esqueletização. Os algoritmos foram analisados e implementados a fim de obter as vantagens e desvantagens de cada algoritmo. Os algoritmos utilizados foram o de Hilditch, Stentiford, Zhang-Suen, Holt e Multi-Escala. Ao término dos experimentos concluíram que os algoritmos de esqueletização são específicos para alguns tipos de imagens. Assim, a escolha do algoritmo deve ser adequada ao problema e ao tipo de imagem.

Na seção a seguir será descrito o algoritmo de Zhang-Suen, que fora adotado empiricamente para a implementação da etapa de esquelotização.

2.5.1 Algoritmo de Zhang-Suen

Um dos algoritmos que realiza a extração do esqueleto é o de Zhang-Suen (Pedrini e Schwartz, 2008, p. 183). Nele os pixels são analisados um a um ao longo de varias iterações. Cada iteração aplica regras para determinar se os pixels serão ou não removidos. Quando não houver mais pixels a serem removidos o algoritmo é finalizado, resultando um esqueleto que não possui mais de 2 pixels quadrados de espessura.

Para entender melhor as regras aplicadas no algoritmo de Zhang-Suen o conceito de vizinhança e de conectividade são ilustrados na Figura 7.

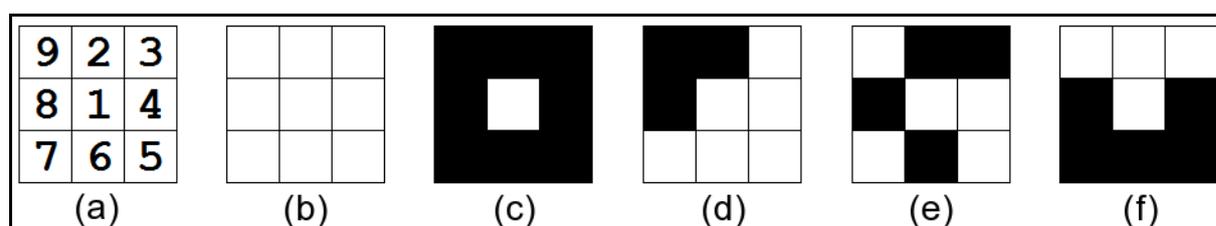


Figura 7 – Localização, vizinhança e conectividade de pixel

O item (a) da Figura 7 possui a localização dos pixels, ao redor do pixel central (sempre marcado como branco). Para os demais itens a Tabela 1 mostra respectivamente o número de vizinhos e conectividade.

O item (b) está rodeado de pixels brancos indicando que todos eles são seus vizinhos, ou seja, o número máximo de vizinhos. Em contrapartida o item (c) não possui nenhum vizinho, pois todos os pixels são pretos. Porém tanto o item (b) como o (c) possuem conectividade 0 pois não há alteração de pixels brancos e pretos ao redor do pixel central.

Já o item (e) possui uma configuração bem diferente dos itens explicados anteriormente. Como existem 4 pixels brancos (sem contar o pixel central) a vizinhança para o pixel central é também 4. Para entender a conectividade coloca-se os valores dos pixels de 2

à 9 (os vizinhos) repetindo o valor do pixel 2 (0, 0, 1, 1, 0, 1, 0, 1, 0); conta-se cada alteração de 0 para 1 e de 1 para 0. O número de alterações dividido por 2, resulta no valor da conectividade. Neste caso o resultado da “conta” é 6, e a conectividade é 3.

Tabela 1 – Relação de vizinhança e conectividade

Relação de vizinhança e conectividade		
Item	Vizinhos	Conectivos
b	8	0
c	0	0
d	5	1
e	4	3
f	3	1

O algoritmo de Zhang-Suen é considerado um algoritmo paralelo porque durante a varredura da imagem, todos os pixels podem ser conferidos ao mesmo tempo e com isso é decidido se um pixel deve ou não ser removido. A exclusão na n-ésima iteração depende apenas dos pixels da iteração n-1. Por isso, todos os pixels podem ser analisados independentemente, de forma paralela a cada iteração.

Existem duas iterações que são repetidas até que nenhum pixel seja marcado para exclusão. Estas iterações possuem 4 regras cada, sendo as duas primeiras comuns às iterações. Cada pixel do objeto deve satisfazer 4 condições para ser excluído dentro das duas iterações abrangentes. As 4 regras a seguir são aplicadas a cada pixel correspondente a região da mão:

- a) regra 1: ter pelo menos 2 pixels vizinhos e não mais que 6;
- b) regra 2: possuir conectividade igual a 1;
- c) regra 3: possuir algum dos pixels 2, 4 ou 6 como fundo;
- d) regra 4: possuir algum dos pixels 4, 6 ou 8 como fundo.

Para a 2ª iteração as regras 3 e 4 são substituídas por:

- a) regra 3: possuir algum dos pixels 2, 4 ou 8 como fundo;
- b) regra 4: possuir algum dos pixels 2, 6 ou 8 como fundo.

2.6 RECONHECIMENTO DE GESTOS DA MÃO USANDO O ESQUELETO

Ionescu et al (2005, p. 2101-2109) propuseram uma técnica de reconhecimento dos gestos das mãos através do esqueleto da mesma. Esta técnica tinha o objetivo de que os gestos das mãos fossem interpretados por robôs através de uma simples câmera digital de modo

genérico e rápido. A partir da câmera digital o robô recebe instruções de movimento, que na verdade são gestos como: atrás, pegar, lançar, esquerda, direita, devagar, parar, girar e ir.

O processo de reconhecimento de gestos da mão pode ser dividido em duas tarefas principais. A primeira é a extração de feições que contém informações de baixo-nível (seção 2.2.1) provenientes da imagem original, cujo objetivo é analisar e produzir informações de alto-nível (seção 2.2.2) com significado. Com as feições coletadas, entra a tarefa de classificação que usa as informações coletadas para realizar a detecção do gesto.

O algoritmo proposto baseia-se na visão 2D, já que a entrada da informação ocorre exclusivamente através de uma câmera digital (2D), e são processadas por um computador. As imagens são convertidas na escala de cinza e possuem um contexto restrito:

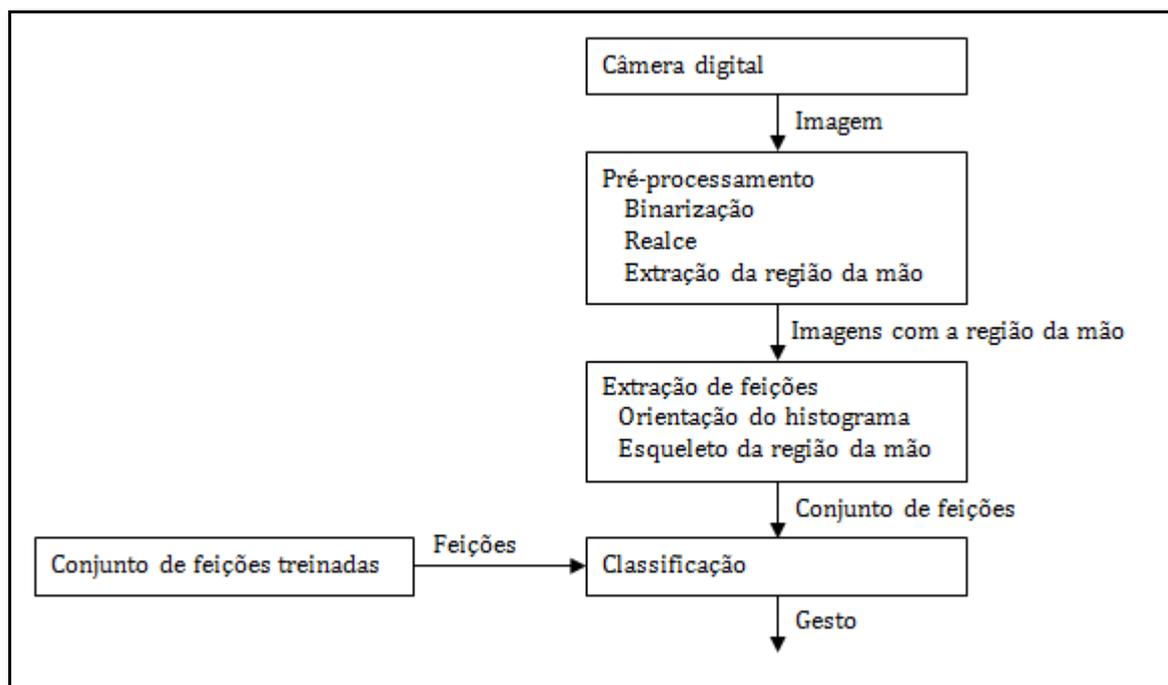
- e) os gestos da mão são realizados em um fundo mais uniforme possível;
- f) a distância entre a câmera e a mão é aproximadamente constante e o gesto é realizado em uma região de espaço já conhecida;
- g) o alfabeto de gestos deve ser conhecido.

Devido à eficiência relativa ao tempo computacional, simples técnicas de extração de feições são usadas para extrair as feições dos gestos (no caso o esqueleto). Mesmo trabalhando intensamente com os pixels tem-se um resultado de custo computacional muito menor do que treinar alguma rede neural ou extrair estatísticas do gesto devido à simplicidade das técnicas utilizadas. Como resultado desta fase, as feições extraídas mantêm as propriedades físicas e espaciais do gesto, o esqueleto.

A última fase é a de classificação que requer um breve estágio de treinamento formado de exemplos para cada gesto. Quanto maior o número de exemplos para cada gesto, mais preciso tende-se a tornar esta etapa.

Para um melhor entendimento do processo de reconhecimento dos gestos da mão é necessário definir previamente o que são gestos. Um gesto é um movimento caracterizado por um padrão que ocorre no espaço e no tempo. Os gestos de mão podem ainda serem divididos em estáticos e dinâmicos. Estáticos são representados por uma única imagem da mão, ou seja, não se desloca no espaço durante o tempo, enquanto os gestos dinâmicos são constituídos de uma seqüência de imagens onde há deslocamento no espaço ao decorrer do tempo, denominado também de movimento da mão.

O diagrama da Figura 8 mostra o fluxo completo da técnica de reconhecimento de Ionescu et al (2005), desde a captura da imagem até o reconhecimento do gesto, nos quais os principais passos são descritos nas seções seguintes.



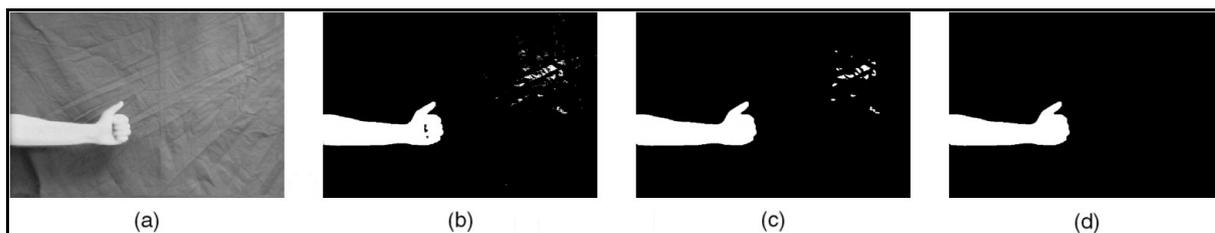
Fonte: Ionescu et al (2005, p. 2103).

Figura 8 – Diagrama do processo de reconhecimento de gestos

2.6.1 Pré-processamento das imagens da extração de feições

A fim de extrair as feições necessárias para a etapa de classificação do processo de reconhecimento do gesto, tem-se que pré-processar as imagens fornecidas pela câmera digital. A principal meta é isolar e reforçar a região da mão que é constituída pelas seguintes operações: extração da imagem binária, aplicação de operações morfológicas (realce) e extração da região da mão.

As imagens devem possuir um fundo mais uniforme possível e mais escuro do que a região da mão. Com este cenário controlado é possível realizar o tratamento de imagens. O tratamento de imagem com operações morfológicas aperfeiçoam a imagem removendo ruídos. Este aperfeiçoamento é feito com a aplicação das operações morfológicas abertura, fechamento e enchimento de buraco. A etapa seguinte é a detecção da maior região, equivalente a área com pixels brancos na imagem binária, resultando na região da mão. O resultado da detecção da região da mão pode ser visto na Figura 9, onde (a) é imagem original em tons de cinza, (b) a imagem binária após segmentação, (c) a imagem binária após operações morfológicas, e (d) a região da mão.



Fonte: Ionescu et al (2005, p. 2104).

Figura 9 – Extração da região da mão

2.6.2 Extração de feições de gestos estáticos

Um gesto de mão estático é representado por uma imagem em tons de cinza com uma determinada configuração de dedos, polegar, palma da mão e braços³. A extração da orientação do gesto estático é realizada através de histogramas de orientação local.

A partir do pré-processamento descrito anteriormente (seção 2.6.1) obtém-se a região da mão. Esta por sua vez, é substituída pelos pixels da imagem original resultando em uma imagem em tons de cinza sem fundo (descartado para os futuros manuseios).

O cálculo dos histogramas de orientação local é uma técnica simples e rápida com baixo custo computacional, além de ser robusta a variação de iluminação. Histogramas de orientação local são baseados na orientação local dos pixels que são descritos pelas direções dos gradientes locais. As amplitudes de gradiente e orientações são obtidas usando operadores de Sobel somente para os pixels na região da mão (nova imagem em tons de cinza).

A idéia de usar histogramas de orientação é de que cada gesto tem um histograma de orientação local, e isto pode ser usado para distinguir entre imagens de partida e parada⁴ nos gestos dinâmicos.

Como o pré-processamento e a análise sobre os histogramas de orientação local a etapa de extração do esqueleto da região (seção 2.5) é executada, obtendo-se uma representação compacta (esqueleto) da região da mão. Esta etapa é fundamental para o resultado que será obtido com a fase de classificação e treinamento.

³ Considerar braços como o antebraço e braço, evitando confusão em relação aos dois membros superiores.

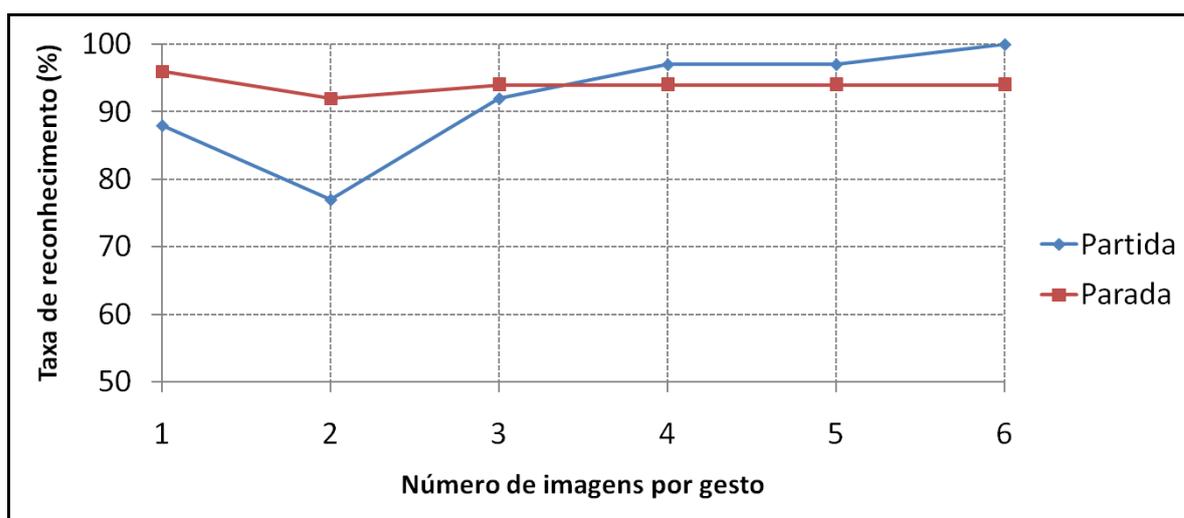
⁴ Gestos dinâmicos possuem uma imagem de partida, quando se inicia o gesto, e outra de parada, quando se termina o gesto. A definição completa dos gestos dinâmicos está no artigo de Ionescu et al (2005, p. 2105).

2.6.3 Treinamento e classificação

A etapa de treinamento para a fase de Classificação do gesto estático da mão consiste em gravar a orientação obtida com o histograma de orientação, a fim de obter um conjunto de amostras para cada gesto dentro do alfabeto de gestos. Cada novo gesto produz um novo histograma de orientação que é calculado e comparado, usando a distância Euclidiana, com todos os histogramas do conjunto de amostra. A menor distância euclidiana indica o gesto estático reconhecido.

Ionescu et al (2005) testou o método para um pequeno alfabeto de gestos da mão contendo dez diferentes gestos, a fim de determinar o seu desempenho. Trinta e seis seqüências foram utilizadas para os testes, contendo no máximo 6 imagens para um mesmo gesto. Os conjuntos de amostras dos histogramas de orientação local foram calculados utilizando a média de três amostras de cada gesto, a fim de aumentar a robustez do processo de reconhecimento.

A Figura 10 mostra a taxa de reconhecimento de gestos estáticos versus o número de treinos compostos de imagens diferentes para um mesmo gesto. A taxa de reconhecimento se mantém entre 77% e 100%. No entanto, estes erros consistem em algumas confusões (ruídos, posição de mãos irregulares) entre as imagens de partida e parada (no caso dos gestos dinâmicos). Analisando o gráfico Ionescu et al (2005) concluem que um número maior de seqüências no treinamento conduz a uma taxa de reconhecimento maior.



Fonte: Ionescu et al (2005, p. 2105).

Figura 10 – Taxa de reconhecimento de gestos estáticos

3 DESENVOLVIMENTO DA FERRAMENTA

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A ferramenta tem como Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF):

- a) capturar as imagens da mão de um libriano para serem reconhecidas (RF);
- b) tratar as imagens capturadas, aplicando algoritmos que destaquem (segmentação) as partes do corpo utilizadas na interpretação (RF);
- c) encontrar o limiar que realiza a binarização de forma automática tirando a responsabilidade do usuário (RF);
- d) construir o histograma de orientação local das imagens (RF);
- e) extrair o ângulo predominante do histograma de orientação local (RF);
- f) extrair o esqueleto do braços, mão e dedos (RF);
- g) criar uma conjunto de treinamento representando gestos em LIBRAS (RF);
- h) utilizar o cálculo da distância de Baddeley para realizar o reconhecimento de gestos da mão (RF);
- i) ser implementada utilizando como ambiente principal de desenvolvimento o Eclipse 3.4, na linguagem de programação Java (RNF);
- j) realizar o reconhecimento de uma letra ou palavra em LIBRAS que permita uma comunicação em tempo real (RNF).

3.2 ESPECIFICAÇÃO

A especificação do presente trabalho foi desenvolvida utilizando a notação *Unified Modeling Language* (GUEDES, 2008) juntamente com a ferramenta Enterprise Architect (SPARX SYSTEMS, 2008). São explanados diagrama de casos de uso e classes. Os diagramas de classes estão em sua forma resumida para melhor visualização, mas não comprometendo o entendimento do trabalho.

Na Figura 11 é demonstrado o diagrama de casos de uso com as principais interações

do usuário com a ferramenta.

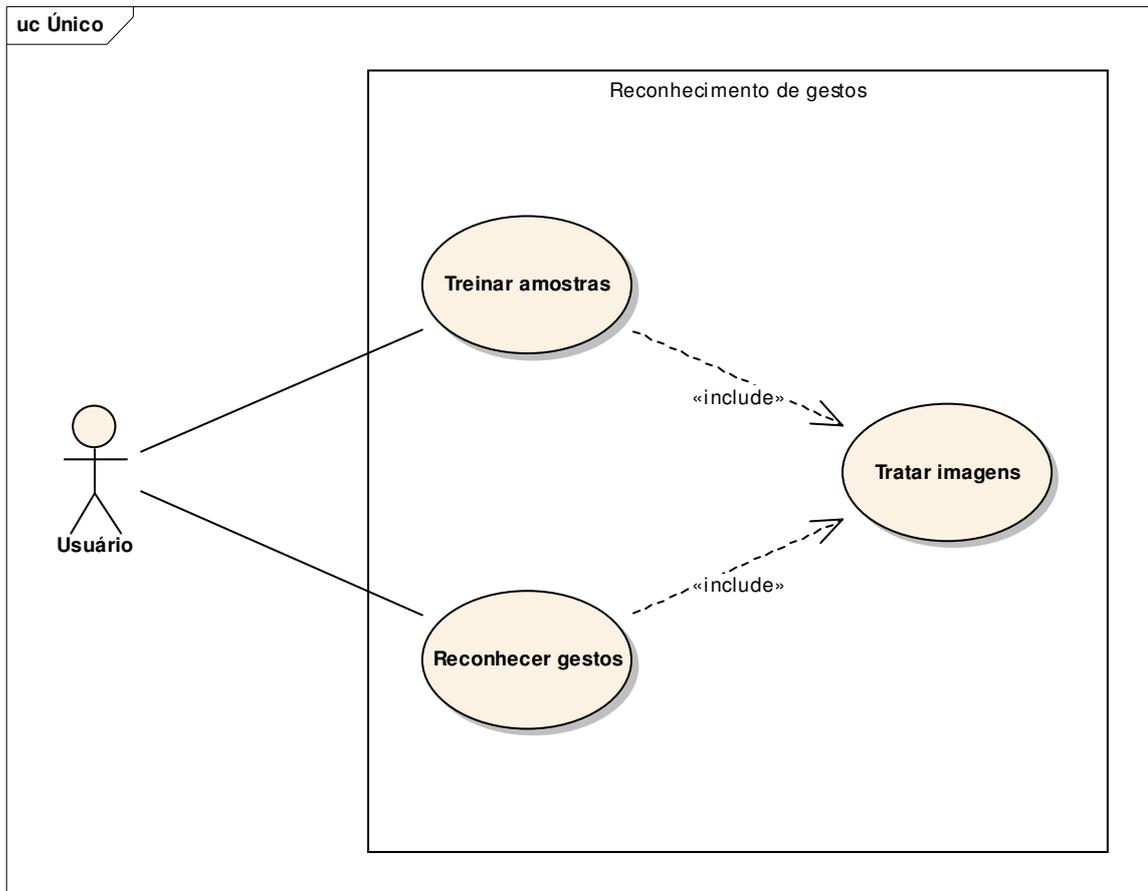


Figura 11 – Diagrama de casos de uso

O caso de uso *Treinar amostras* descreve como o usuário poderá iniciar o processo de treinamento das amostras, sendo imprescindível para realizar o reconhecimento das imagens capturadas em gestos. Este caso de uso possui um cenário principal e dois cenários alternativos não havendo cenários de exceção como mostra o Quadro 1.

Treinar amostras: possibilita o usuário selecionar as imagens que serão tratadas como amostras, a fim de compor o banco de amostras.	
Pré-condição	Haver alguma imagem(s) válida(s) com a região da mão disponível no sistema de arquivos.
Cenário principal	1) Usuário marca o marcador Treinamento. 2) Usuário seleciona a imagem para tratamento. 3) Ferramenta realiza o tratamento da imagem. 4) Ferramenta salva a amostra no banco de amostras. 5) Usuário opta por continuar o treinamento voltando ao passo 2 ou encerra o tratamento desmarcando o marcador Treinamento.
Cenário alternativo 1	No passo 2 caso o usuário selecione uma imagem inválida, a ferramenta desconsidera-a e solicita outra imagem até que uma imagem válida seja selecionada.
Cenário alternativo 2	No passo 2 o usuário pode selecionar um diretório com várias imagens para treinar. Neste caso o passo 3 e 4 são realizados implicitamente para cada imagem válida contida no diretório.
Pós-condição	O banco de amostras contém amostras.

Quadro 2 – Caso de uso treinar amostras

O caso de uso Reconhecer gestos descreve como o usuário poderá submeter uma imagem para ser reconhecida pela ferramenta com base no banco de amostras. Este caso de uso possui um cenário principal e três cenários alternativos não havendo cenários de exceção conforme apresentado Quadro 3.

Reconhecer gestos: possibilita o usuário selecionar uma imagem por vez para que a ferramenta retorne as amostras mais próximas.	
Pré-condição	Haver imagens com a região da mão disponíveis no sistema de arquivos e haver ao menos uma amostra no banco de amostras.
Cenário principal	<ol style="list-style-type: none"> 1) Ferramenta carrega as amostras do banco de amostras. 2) Usuário seleciona a imagem para reconhecimento. 3) Ferramenta realiza o tratamento da imagem e mantém em memória a amostra da imagem. 4) Ferramenta realiza pré-filtragem das amostras comparando a orientação da imagem com a orientação de cada amostra. 5) Ferramenta realização a classificação das amostras. 6) Ferramenta exibe as amostras reconhecidas ao usuário.
Cenário alternativo 1	No passo 2, caso o usuário selecione uma imagem inválida, a ferramenta desconsidera-a e solicita outra imagem até que uma imagem válida seja selecionada.
Cenário alternativo 2	No passo 4 a orientação da imagem pode ter variação superior a 5 graus em relação a todas as amostras. Portanto, caso o marcador Filtrar orientação estiver desmarcado nenhuma amostra será reconhecida.
Cenário alternativo 3	No passo 6, caso a imagem selecionada possua a cor da região da mão mais escura do que o cor do fundo, a classificação ficará destorcida. Neste caso o usuário poderá marcar o marcador Inverter cores e re-selecionar a imagem.
Pós-condição	O banco de amostras contém amostras.

Quadro 3 – Caso de uso reconhecer gestos

O caso de uso Tratar imagem descreve é usado implicitamente nos casos de uso anteriores realizando todos os passos do tratamento da imagem. Este caso de uso possui um cenário principal e dois de exceção não havendo cenários alternativos de acordo com o Quadro 4.

Tratar imagens: realiza o tratamento das imagens para uso interno da ferramenta e disponibiliza cada passo deste tratamento de forma visual ao usuário.	
Pré-condição	Execução dos casos de uso Reconhecer gestos ou Treinar amostras que devem fornecer a imagem a ser tratada.
Cenário principal	<ol style="list-style-type: none"> 1) Ferramenta transforma imagem na escala de cinza. 2) Ferramenta encontra o limiar entre o fundo e a região da mão. 3) Ferramenta realiza a binarização da imagem. 4) Ferramenta aplica operações morfológicas na imagem binarizada. 5) Ferramenta descobre a região da mão. 6) Ferramenta obtém imagem com a região da mão original em escala de cinza e sem o fundo. 7) Ferramenta encontra a orientação predominante da imagem. 8) Ferramenta constrói o histograma de orientação da imagem. 9) Ferramenta obtém o esqueleto da região da mão. 10) Ferramenta calcula distância dos pixels do esqueleto e obtém a amostra.
Cenário de exceção 1	No passo 1, caso a imagem possua a intensidade de pixels diferente de 8 ou 24 bits, a ferramenta gerará uma exceção notificando o usuário que a imagem é inválida.
Cenário de exceção 2	No passo 2, caso não haja a distinção do fundo e da região da mão, a ferramenta notificará o usuário que não é possível encontrar o limiar.
Pós-condição	Imagem tratada e cada passo disponível para visualização.

Quadro 4 – Caso de uso tratar imagens

Na Figura 12 é exibido o diagrama de classes usado na interface da ferramenta.

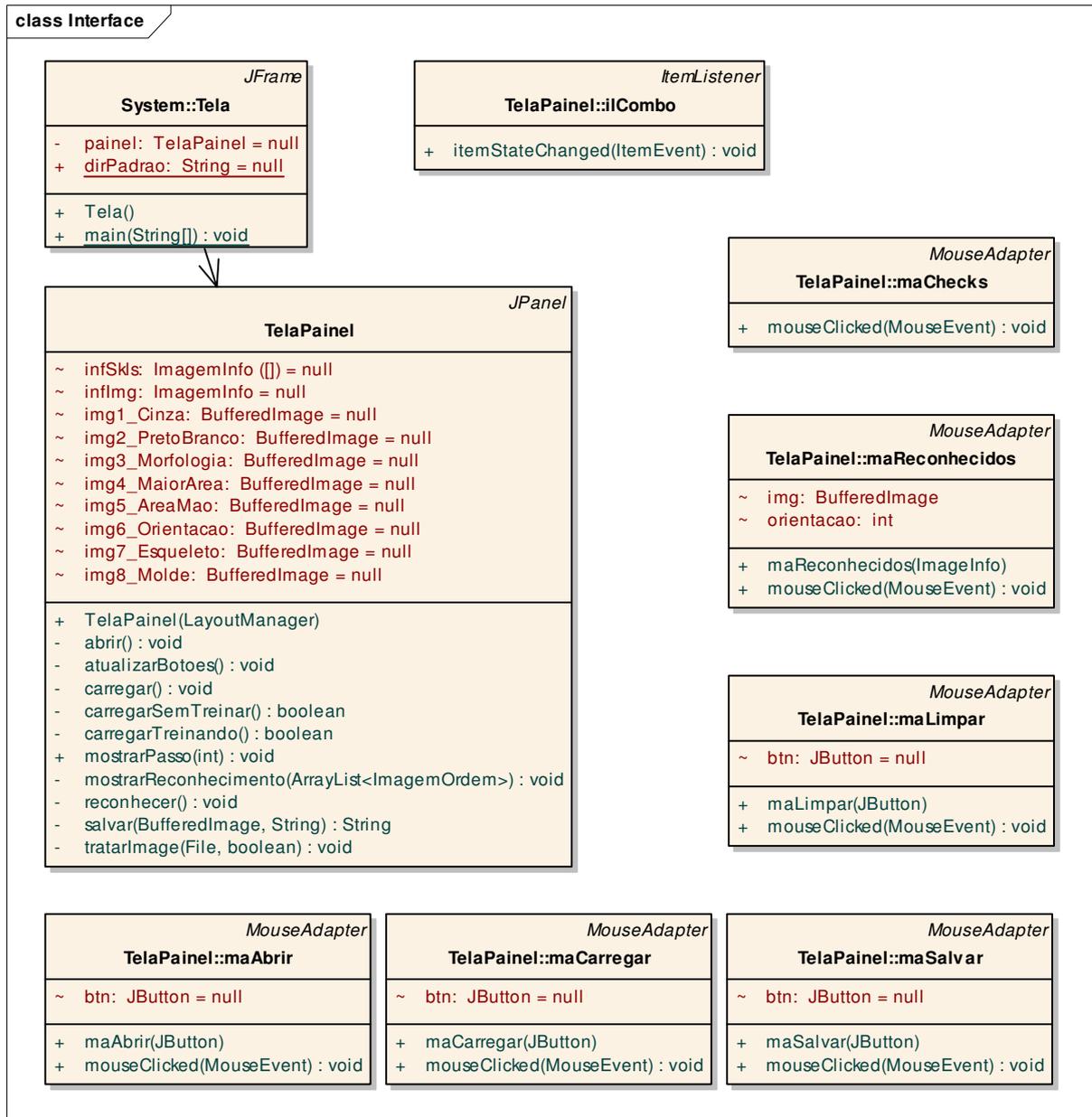


Figura 12 – Diagrama de classes da interface

As classes de interface são responsáveis por exibir informações e atender a necessidade do usuário. A execução da ferramenta deve ser feita através método estático `main` para a classe `Tela`. A classe `Tela` possui um painel principal substituído pela classe `TelaPainel`. Os botões, *labels* e demais objetos da tela se encontram dentro deste painel. As demais classes exibidas na Figura 12 são os eventos dos objetos da `TelaPainel`, como por exemplo, a classe `maSalvar` que realiza a execução das classes de funções para salvar as amostras treinadas.

O diagrama de classes das funcionalidades da ferramenta está disponível na Figura 13. Neste diagrama procurou-se demonstrar as classes com apenas os atributos e métodos mais significantes.

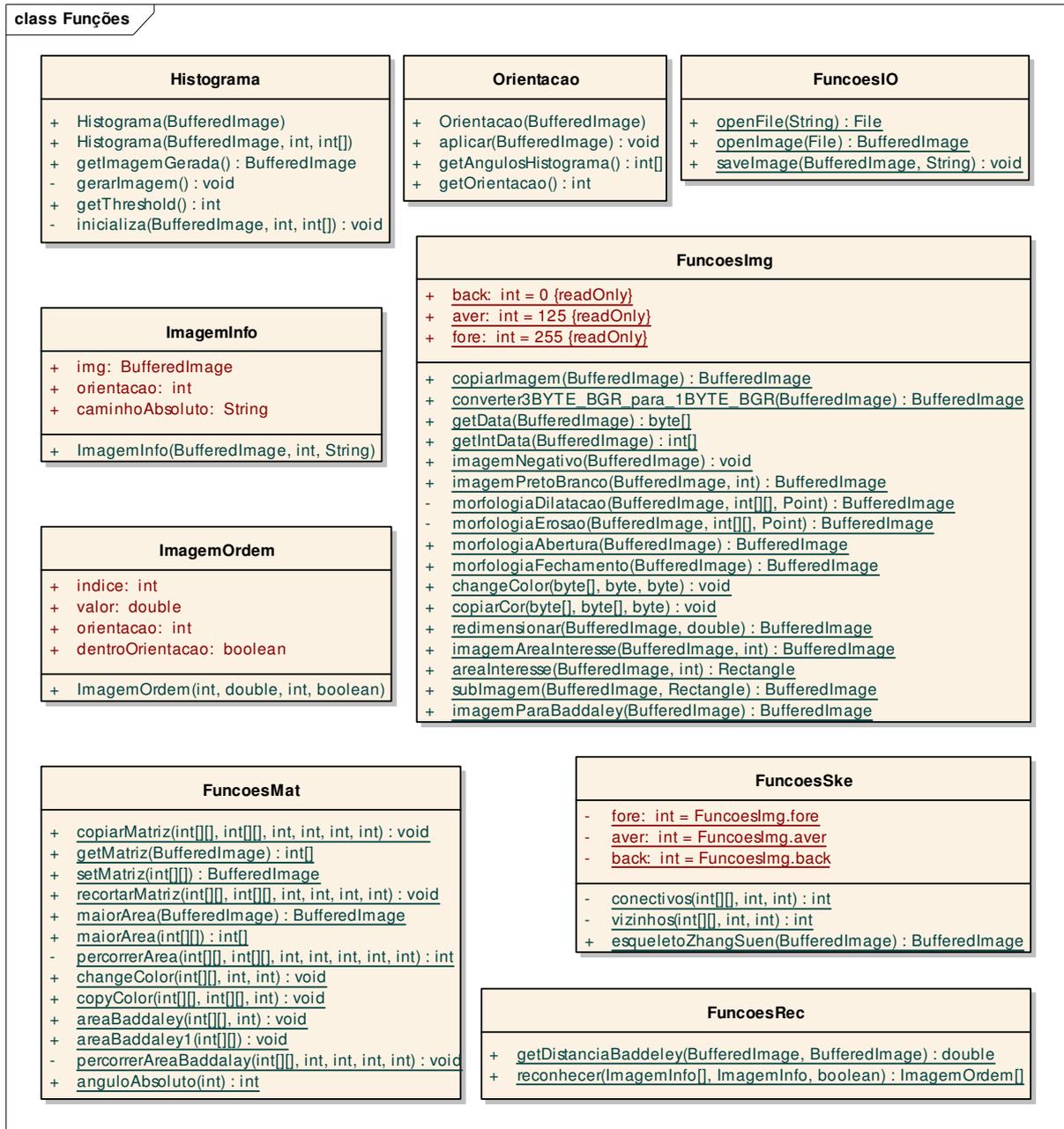


Figura 13 – Diagrama de classes das funções

As classes de funções podem ser divididas em dois grupos, as classes estáticas e as não estáticas. As classes não-estáticas são instanciadas com frequência, pois guardam informações ou realizam várias funcionalidades. A classe `Histograma` calcula o limiar e é responsável pela geração da imagem do histograma. A classe `Orientação` é responsável por identificar a orientação do gesto em uma imagem. A classe `ImagemInfo` guarda as informações das amostras. Já a classe `ImagemOrdem` é utilizada na etapa de classificação para ordenação das amostras.

As classes estáticas são instanciadas uma única vez para evitar a perda de tempo com alocação de memória. Os métodos contidos nestas classes realizam funções específicas. A

classe `FuncoesIO` realizada operações com o sistema operacional (leitura e escrita). A classe `FuncoesImg` contém de forma geral as funções de tratamento de imagens. A classe `FuncoesSke` implementa o algoritmo de Zhang-Suen. Na classe `FuncoesMat` estão os métodos que manipulam os dados das imagens no plano 2D. E finalmente a classe `FuncoesRec` é a encarregada de calcular a distância Baddeley e reconhecimento dos gestos da mão.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para a implementação da ferramenta na linguagem Java 1.6, foi utilizado o ambiente de desenvolvimento Eclipse na versão 3.4.

Existem no mercado bibliotecas que provêm a funcionalidade dos algoritmos adotados, algumas delas são gratuitas ou possuem licença GPL (*General Public License*). Entretanto estas bibliotecas não foram utilizadas a fim de elevar o nível de entendimento em cada passo necessário no reconhecimento dos gestos.

A ferramenta possui duas funções principais, que são o treinamento das amostras e o reconhecimento dos gestos. Ambas compartilham a maior parte da implementação que se baseia em funções que manipulam diretamente os pixels das imagens, constituídas pelo tratamento de imagens.

O Quadro 5 mostra a função onde se realiza todo o tratamento de imagens, invocando as chamadas das principais funções da ferramenta. Neste trecho é demonstrado cada passo do tratamento a partir do recebimento da imagem original até a construção da amostra usada na fase de treinamento e na fase de reconhecimento.

```

void tratarImage(BufferedImage img) {

    //A - Transformar para nível de cinza (1 Byte);
    if (img.getType() != BufferedImage.TYPE_BYTE_GRAY)
        img = FuncoesImg.converter3BYTE_BGR_para_BYTE(img);
    img1_Cinza = FuncoesImg.copiarImagem(img);

    //B - Descobrir o threshold/limiar
    Histograma histograma = new Histograma(img1_Cinza);
    int limiar = histograma.getThreshold();

    //C - Passar imagem para preto e branco
    img2_PretoBranco = FuncoesImg.imagemPretoBranco(img1_Cinza, limiar);

    //D - Aplicar fechamento e abertura
    img3_Morfologia = FuncoesImg.morfologiaFechamento(img2_PretoBranco);
    img3_Morfologia = FuncoesImg.morfologiaAbertura(img3_Morfologia);

    //E - Obter a maior área (Mão e braço, excluindo possíveis ruídos)
    img4_MaiorArea = FuncoesMat.maiorArea(img3_Morfologia);

    //F - Substituir maior área por fundo original
    img5_AreaMao = FuncoesImg.copiarImagem(img1_Cinza);
    FuncoesImg.copiarCor(img4_MaiorArea, img5_AreaMao, (byte) 0);

    //G - Orientação do histograma local
    Orientacao orientacao = new Orientacao(img5_AreaMao);
    orientacao.aplicar(img4_MaiorArea);
    direcaoOrientacaoHistograma = orientacao.getOrientacao();

    int[] angulo = orientacao.getAngulosHistograma(); //360 = nº angulos
    histograma = new Histograma(img5_AreaMao, angulo.length, angulo);
    img6_Orientacao = histograma.getImagemGerada();

    //H - Esqueleto
    img7_Esqueleto = FuncoesSke.esqueletoZhangSuen(img4_MaiorArea);

    //I - Amostra
    img8_Amostra = FuncoesImg.imagemParaBaddeley(img7_Esqueleto);
    img8_Amostra = FuncoesImg.imagemAreaInteresse(img8_Amostra);
}

```

Quadro 5 – Processamento da imagem capturada

A partir da imagem original se obtém a imagem em nível de cinza e é feita a transformação do espaço de cores *Red*, *Green* e *Blue* (RGB) para o espaço binário, que é realizada pelo método `converter3BYTE_BGR_para_BYTE` (A) da classe estática `FuncoesImg`. Trabalhar com um byte facilita a implementação dos algoritmos e se reduz o custo computacional ao invés de 3 bytes do espaço RGB.

Para gerar uma imagem binária é necessário haver um limiar que divida a imagem de tal forma que o objeto mais claro seja separado do fundo mais escuro. Em uma imagem em nível de cinza esse valor pode variar de 1 a 254, pois uma divisão em 0 ou em 255 não geraria dois conjuntos de pixels, sendo um deles o fundo e outro o objeto. A descoberta do limiar que possibilite a binarização é feita através da classe `Histograma` com o método `getThreshold`

(B). Esta classe também é responsável por gerar a imagem de um histograma pelo método `getImagemGerada`.

Com o limiar encontrado é possível efetuar a limiarização realizada através da classe estática `FuncoesImg` com o método `imagemPretoBranco` (C).

A extração da mão, o objeto mais claro, muitas vezes possui pontas, cumes e buracos. Estes ruídos são causados pela proximidade de regiões do objeto e do fundo, luminosidade, sombra e problemas na captura da imagem como: baixa resolução e compactação nativa das câmeras digitais. Para minimizar estes problemas as operações morfológicas de fechamento e abertura foram aplicadas a partir dos métodos `morfologiaFechamento` e `morfologiaAbertura` (D) respectivamente.

Com a imagem da mão devidamente tratada é necessário processá-la para descobrir a região da mão. A limiarização não garante que a imagem da mão seja o único elemento destacado do fundo, como demonstrada na Figura 9 (item b). Ruídos do fundo ou até mesmo outros pequenos objetos contidos na imagem devem ser descartados, para permitir alcançar o objetivo de reconhecimento do gesto da mão. O método `maiorArea` (E) da classe estática `FuncoesMat` realiza a extração da maior área branca contida na imagem. É importante enfatizar que o maior objeto na imagem deve ser a mão e braços, de acordo com os requisitos definidos para a imagem de entrada (descritos a seguir, seção 3.4).

Na seqüência calcula-se o histograma de orientação local, que indica a direção em graus predominante da imagem original da região da mão. Entretanto este cálculo deve ser aplicado sobre os pixels originais da região da mão onde o fundo deve ser desprezado afim de não interferir na orientação a se obtida. O passo anterior (E) resultou em uma imagem com a região da mão em cor branca e com o fundo na cor preta. Aplicando o fundo desta imagem sobre a imagem original se obtém a imagem desejada. Esta operação é feita com o método `copiarCor` (F) da classe estática `FuncoesImg`.

A partir desta nova imagem (obtida na etapa F), a classe `Orientacao` realiza cálculos que são explicados a seguir na seção 3.3.1.1, e retorna com o método `getOrientacao` (G) o ângulo que representa a direção da região da mão. Este ângulo é usado na etapa de classificação descartando as imagens com ângulos que possuam variações superiores a 5 graus, minimizando o processamento realizado nesta fase. Para fins ilustrativos uma instância da classe `Histograma` é criada para confeccionar uma imagem do histograma da orientação local com o método `getImagemGerada`.

Voltando a imagem da região da mão em preto e branco (F) se faz necessário a

extração do esqueleto desta imagem. A extração do esqueleto é a principal rotina existente na fase de tratamento da imagem, pois ela é a base utilizada na etapa de classificação. O método `esqueletoZhangSuen` (H) da classe estática `FuncoesSke` recebe uma imagem contendo a região da mão (em branco) e retorna outra imagem com a representação do esqueleto (descrito em detalhes na seção 3.3.1.2).

A última etapa da fase de Tratamento de imagem visa melhorar a fase de Classificação realizando dois tratamentos. Na classe estática `FuncoesImg` existe o método `imagemParaBaddeley`. Nesta função a imagem com o esqueleto adquirido recebe camadas, em tom mais escuro a cada nível, que o contornam. Cada nível de camada que se afasta do esqueleto sinaliza a distância que possui do esqueleto. Estas camadas servem para evitar que o cálculo da distância dos pixels seja realizado durante a fase de Classificação, embutindo na (demonstrado no item i da Figura 17) amostra o valor da distância. O segundo tratamento encontra a região que contém o esqueleto e suas camadas, em outras palavras, pixels superiores a 0 (preto). O método `imagemAreaInteresse` (I) é o responsável por recortar a imagem resultando nesta região.

3.3.1.1 Cálculo do histograma de orientação local

O cálculo do histograma de orientação é demonstrado em detalhes no Quadro 6 a classe `Orientacao`. Após a criação da instância passando a imagem a ser calculada (mostrada anteriormente no Quadro 5 item G) o método `aplicar` é chamado obtendo como parâmetro a imagem da região da mão.

```

float[] mask = { 1, 0, -1, 2, 0, -2, 1, 0, -1 };

void aplicar(byte[] areaMao) {
    direcaoOrientacao = -1;

    histograma = new int[360]; //ângulos possíveis
    float[] GY = new float[w * h];
    float[] GX = new float[w * h];
    int[] magnitude = new int[w * h];

    int x1, y1, x2, y2;
    float somaY, somaX;
    int angulo;

    //Soma os gradientes para cada pixel, retira 1 pixel das extremidades
    for (x = 1; x < w - 1; x++) {
        for (y = 1; y < h - 1; y++) {
            somaY = 0;
            somaX = 0;

            for (x1 = 0; x1 < maskTam; x1++) {
                for (y1 = 0; y1 < maskTam; y1++) {
                    x2 = (x - 1 + x1);
                    y2 = (y - 1 + y1);

                    //Somatória dos gradientes em Y
                    somaY += data[y2 * w + x2] *
                        mask[y1 * maskTam + x1];

                    //Somatória dos gradientes em X
                    somaX += data[y2 * w + x2] *
                        mask[x1 * maskTam + y1];
                }
            }

            GY[y * w + x] += somaY;
            GX[y * w + x] += somaX;
        }
    }
    ...
}

```

Quadro 6 – Cálculo da orientação local

A orientação local é a obtida com a utilização do filtro de Sobel equivalente a máscara `mask` (variável global no Quadro 6). O filtro de Sobel consiste em calcular o gradiente para cada pixel da imagem em relação aos pixels vizinhos. Note que o gradiente precisa ser calculado separadamente no eixo X e no eixo Y. Esta divisão no cálculo dos gradientes por eixo permite que a magnitude e a orientação sejam calculados. O Quadro 7 mostra a continuação da função `aplicar` onde são obtidas a magnitude e a orientação.

```

...
for (x = 0; x < w; x++) {
    for (y = 0; y < h; y++) {
        //Calcular uma vez o ponto = evita várias multiplicações
        p = y * w + x;

        //Só calcula para a região da mão
        if (areaMao[p] == -1) { // -1 (byte) == 255 (int)
            //Calcular magnitude/amplitude do pixel
            magnitude[p] = Math.sqrt(GX[p] * GX[p]
                + GY[p] * GY[p]);
            //Filtra magnitude inferior a 1.5
            if (magnitude[p] > 1.5) {
                //Ângulo/direção para o gradiente do pixel
                angulo = (Math.atan2(GX[p], GY[p]) + Math.PI)
                    * 180 / Math.PI - 0.50;
                histograma[angulo]++;
            }
        }
    }
}
}
}

```

Quadro 7 – Cálculo do histograma de orientação - magnitude e orientação

Nesta etapa cada pixel da imagem que corresponde a um pixel da região da mão é submetido ao cálculo de magnitude do gradiente. A magnitude, ou amplitude, do gradiente corresponde à intensidade em que a orientação é definida. Magnitudes inferiores a 1.5 são desconsideradas (IONESCU, 2005, p. 2104), pois não influem significativamente na orientação. A orientação, ou direção, é acumulada em forma de histograma (recebendo a orientação de cada pixel no respectivo ângulo) que pode variar de 0 a 359, ou seja, os ângulos possíveis. A maior frequência de ângulos acumulados (`angulo`) resultará na orientação da imagem.

3.3.1.2 Esqueletonização

O processo de esqueletonização demonstrado no Quadro 8 está contido na classe estática `FuncoesSk1` através do método `esqueletoZhangSuen`. O algoritmo adotado é o de Zhang-Suen e foi criado em 1984, por este fato outros autores propuseram novas implementações do mesmo, visando à qualidade do esqueleto e a melhora no desempenho. Apesar de haver várias versões para este algoritmo, a idéia básica do procedimento firmasse nas bordas do objeto na imagem, que recebem sucessivas remoções de pixels visando o afinamento deste objeto. Como o código do algoritmo é extenso optou-se em apresentar o seu pseudo-código (Quadro 8).

```

while (excluindo) {
    // 1ª Iteração
    excluindo = false;
    para cada pixel na região da mão {

        //1 - Ter entre 2 a 6 vizinhos
        nrViz = getVizinhos(pixel);
        if (nrViz >= 2 && nrViz <= 6) {

            //2 - Ter 1 de conectividade
            con = getConectividade(pixel);
            if (con == 1) {

                //3 - Vizinhos 2 ou 4 ou 6 = 0
                if (getViz2 == 0 || getViz4 == 0 || getViz6 == 0) {

                    //4 - Vizinhos 4 ou 6 ou 8 = 0
                    if (getViz4==0 || getViz6==0 || getViz8==0) {
                        marcarPixel(pixel);
                        excluindo = true;
                    }
                }
            }
        }
    }

    //Excluir pixels marcados
    if (excluindo) {
        excluirPixelsMarcados();
    }

    // 2ª Iteração
    excluindo = false;
    para cada pixel na região da mão {

        //Repete os passos 1 e 2
        ...
        //3 - Vizinhos 2 ou 4 ou 8 = 0
        if (getViz2 == 0 || getViz4 == 0 || getViz8 == 0) {

            //4 - Vizinhos 2 ou 6 ou 8 = 0
            if (getViz2==0 || getViz6==0 || getViz8==0) {
                marcarPixel(pixel);
                excluindo = true;
            }
        }
    }
}

//Excluir pixels marcados
if (excluindo) {
    excluirPixelsMarcados();
}
}

```

Quadro 8 – Esqueletonização de Zhang-Suen

Com a representação do esqueleto pronta as amostras podem ser construídas para a fase de classificação. As amostras são guardadas em um diretório e as principais informações são a orientação e o esqueleto, muito embora receba sucessivas camadas em degradê

(distância dos pixels do esqueleto) ao redor do esqueleto para otimização na fase de Classificação.

3.3.1.3 Classificação

A fase de classificação procura encontrar para a imagem capturada a amostra com a região da mão mais próxima. Inicialmente a imagem recebe o mesmo tratamento de imagens aplicado as amostras (Quadro 5), garantindo que a imagem possua um esqueleto e uma orientação. O Quadro 9 mostra um código adaptado da função de reconhecimento através da função `reconhecer` encontrada na classe estática `FuncoesRec`.

```

public ArrayList reconhecer(BufferedImage img) {
    ArrayList lista = new ArrayList(amostras.length);

    //Calcula range dos angulos/orientações/dir(eções)
    int oriImg = FuncoesMat.anguloAbsoluto(oriacaoImagem);
    int oriAmostra;

    //Calcula limites
    int variacao = 5; //5 Graus
    int menor = oriImg - variacao;
    int maior = oriImg + variacao;

    double db; //Distancia Baddeley
    boolean dentroOri; //Se a amostra esta dentro da orientação

    //Percorre guardando cada cálculo
    for (int i = 0; i < amostras.length; i++) {
        BufferedImage amostra = amostras [i];
        oriAmostra = FuncoesMat.anguloAbsoluto(oriacaoAmostra[i]);

        //A - Verifica limites normal
        if (oriAmostra >= menor && oriAmostra <= maior) {
            dentroOri = true;
        } else {
            //B - Verifica limites quando a variação exceder (180)
            if (oriAmostra >= (180 + menor)) {
                dentroOri = true;
            } else {
                //C - Verifica limites quando variação abaixo de 0
                if (oriAmostra <= (maior - 180)) {
                    dentroOri = true;
                } else {
                    dentroOri = false;
                }
            }
        }

        //Se o orientação da amostra não bater com a da imagem
        if (!dentroOri) {
            continue;
        }

        //Calcula a distância Baddeley se a orientação for próxima
        db = FuncoesRec.getDistanciaBaddeley5(amostra, img);

        //Guarda informações para ordenar a menor distância e outros
        imgOrd = new ImagemOrdem(i, db, oriAmostra, dentroOri);
        lista.add(imgOrd);
    }

    //Ordena lista pela distância encontrada, quanto menor mais próximo
    ordenarLista(lista);

    return lista;
}

```

Quadro 9 – Reconhecimento

A primeira etapa da fase da Classificação é calcular se a variação de orientação esta dentro de um limite aceitável. Assim as orientações das amostras que não se aproximam da orientação da imagem a ser reconhecida, ou seja, não estejam dentro do limite aceitável, são

descartadas da etapa seguinte. É importante ressaltar que existe um espelhamento na orientação das imagens onde graus superiores ou iguais a 180 são convertidos para o grau oposto (-180) entre 0 e 179. Esta conversão é realizada, pois o cálculo do histograma de orientação local não reflete a direção que percebemos através da visão. Por exemplo, uma imagem pode conter uma mão entendida na horizontal da esquerda para direita, mas orientação resultando pode ser tanto de 90° como de 270°. A variação adotada recebeu o valor 5, tanto para uma lado como para outro. Vale lembrar que graus de 0 a 4 somados (diminuídos implicitamente) a variação possuem limites entre 175 e 179, e vice-versa B e C (Quadro 9). Por exemplo o grau 2 varia de 2 – 5 até 2 + 5, ou seja, de 177 até 7. De forma semelhante o grau 175 varia de 175 – 5 até a 175 + 5, que representa de 170 até 0.

Caso as orientações das imagens estejam em limites aceitáveis, o cálculo da distância de Baddeley é aplicado. A validação dos limites da orientação ocorre por que o custo computacional do cálculo do histograma de orientação local é muito menor do que o cálculo da distância de Baddeley. Logo calcular o histograma de orientação pode eliminar várias amostras, evitando que cálculos da distância de Baddeley sejam realizados desnecessariamente.

Uma adaptação da distância de Baddeley é calculada entre duas imagens, no caso a amostra e a imagem a ser reconhecida, para encontrar o grau de diferença entre as mesmas. O cálculo aplicado na ferramenta é exibido no Quadro 10.

```
double getDistanciaBaddeley(BufferedImage img, BufferedImage amostra) {
    para cada pixel da amostra {
        //Obtém a distância do pixel ao objeto
        if (se pixel estiver dentro da imagem) {
            d = imagem[p] - amostra[p]
        }
        //Se pixel esta fora da imagem
        else {
            d = 255 - amostra[p];
        }
        d = d * d; //Eleva distância a 2
        soma = soma + d;
    }

    //1 sobre largura * altura
    double tam = img.getWidth() * img.getHeight(); //distancia calculada
    bd = new BigDecimal(1);
    bd = bd.divide(new BigDecimal(tam), 10, RoundingMode.HALF_EVEN);
    //Multiplica pela soma das distâncias
    bd = bd.multiply(new BigDecimal(soma));
    distancia = bd.doubleValue();
    //Tira raiz
    distancia = Math.sqrt(d);

    return distancia;
}
```

Quadro 10 – Distância de Baddeley

O código adaptado acima representa a seguinte equação demonstrada na Figura 14, na qual duas imagens são comparadas I_1 e I_2 .

$$d_B(I_1, I_2) = \left[\frac{1}{N \times M} \sum_{p \in S} |d_{I_1}(p) - d_{I_2}(p)|^q \right]^{1/q}$$

Fonte: Ionescu et al (2005, p. 2106).

Figura 14 – Equação de Baddeley

Onde $N \times M$ é número total de pixel na imagem, $d_{I_1}(p)$ é a distância do pixel entre a $d_{I_2}(p)$, q é um expoente controlando o peso da magnitude da distância. A variável q foi atribuída no código com o valor 2, de acordo com o definido em Ionescu (2005, p. 2106).

3.3.2 Operacionalidade da implementação

A execução da ferramenta é feita através da criação de uma instância da classe `Tela` contida no diretório `Interface`. As imagens utilizadas deverão obedecer os seguintes pré-requisitos:

- a) possuir cor de fundo mais escura do que a toda a região mão, ou o contrário;
- b) manter a distância entre a mão e o dispositivo de captura relativamente constante;
- c) método de captura deve fornecer imagens no formato PNG ou JPEG;
- d) possuir a intensidade de bits por pixel de 8 ou 24 bits.

O funcionamento da ferramenta é dividido em duas fases, a de treinamento e a de reconhecimento.

3.3.2.1 Treinamento

A fase de treinamento tem com objetivo criar uma base de amostras para possibilitar o reconhecimento. Para tanto se faz necessário o treinamento das amostras, que deve ser sinalizado através do marcador `Treinamento`, e realizado através de duas formas:

- a) através do botão `Abrir`: clicando neste botão um diálogo será exibido solicitando a imagem que será transformada em amostra. Ao selecionar a imagem o processamento é realizado disponibilizando a respectiva amostra para o

armazenamento na base de amostras. Para salvar a amostra é necessário informar o nome do campo `Nome`, que inicialmente é carregado com o nome da imagem, e clicar no botão `Salvar`;

- b) através do botão `Carregar`: clicando neste botão um diálogo será exibido solicitando o diretório de imagens que serão transformados em amostras. Ao selecionar o diretório o processamento para cada imagem é realizado. Diferentemente do botão `Abrir` as amostras geradas já são salvas com o nome da imagem original na base de amostras.

Toda amostra é salva em um diretório denominado `Banco` que fica acima do diretório de execução. Caso seja necessário limpar as amostras treinadas, esta ação deve ser feita pelo botão `Limpar`. A Figura 15 mostra a ferramenta após a fase de treinamento.

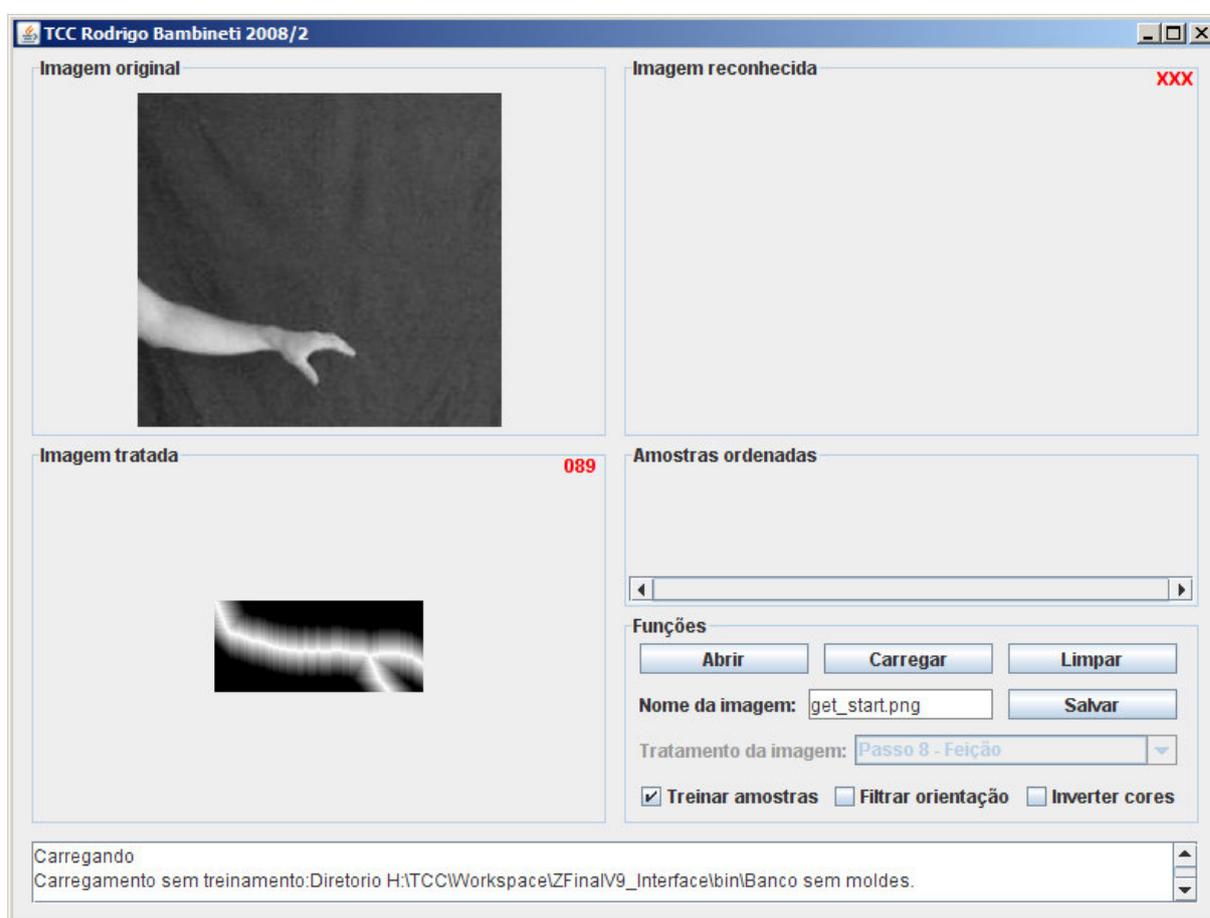


Figura 15 – Ferramenta durante a fase de treinamento

3.3.2.2 Reconhecimento

Na fase de reconhecimento, ativada com o botão `Treinar amostras` desmarcado, as

amostras são comparadas com uma imagem selecionada. Para selecionar uma imagem basta clicar no botão *Abrir* e escolher a imagem a ser reconhecida. A imagem selecionada é processada da mesma maneira que as amostras foram na fase de treinamento, entretanto nesta fase as imagens não são adicionadas a base de amostras. As amostras são comparadas uma a uma com a amostra obtida da imagem selecionada e são ordenadas da mais próxima para a menos próxima no grupo *Amostras ordenadas*, da esquerda para a direita respectivamente. A Figura 16 mostra a ferramenta em execução durante a fase de reconhecimento.



Figura 16 – Ferramenta durante a fase de reconhecimento

Algumas funcionalidades foram desenvolvidas para facilitar o uso da ferramenta e para auxiliar na compreensão das informações das imagens. A área inferior da tela (1) é destinada ao *log* dos eventos que ocorrem como informações de carregamento de amostra, dados do reconhecimento, imagens em formato inválido entre outros. A orientação da imagem (2) ou amostra (3) exibem a *Imagem tratada* e *Imagem reconhecida*. No grupo de *Amostras ordenadas* é possível selecionar outras amostras além da reconhecida, esta seleção altera tanto a imagem como a sua orientação da área informações *Imagem reconhecida*.

O marcador *Filtrar orientação* inibe o reconhecimento de ângulos com variação superior a 5 graus. Quando este marcador estiver desmarcado as amostras com ângulos que

possuam variação maior do que 5 graus são contornadas de vermelho na área `Amostras ordenadas`, enquanto as demais são contornadas com verde. No campo `Tratamento da imagem` é possível selecionar os passos do tratamento da imagem que é exibido na área `Imagem tratada`.

Existe também a possibilidade de abrir uma imagem onde a cor de fundo é mais clara do que a região da mão. Nesse caso o marcador `Inverter cores` deve ser marcado, indicando que a imagem esta na cor inversa do perfil desejado.

3.4 RESULTADOS E DISCUSSÃO

Neste trabalho uma ferramenta de visão computacional fora inspecionada em nível de código, onde várias etapas puderam ser conhecidas de forma específica. Dentre estas etapas as que mais se destacam pela sua complexidade foram, a orientação do histograma local e a esqueletonização. Apesar da complexidade destas etapas identificou-se a viabilidade da utilização de técnicas de visão computacional para construir não apenas uma ferramenta, mas também um sistema de reconhecimento de gestos (proposto na seção 4.1).

Na seção 3.4.1 serão apresentados os problemas encontrados durante o desenvolvimento da ferramenta, bem como os pontos positivos das técnicas utilizadas. Os testes realizados na aplicação serão mostrados na seção 3.4.2, relatando a eficácia dos algoritmos aplicados e pontos a melhorar.

3.4.1 Implementação

Na implementação foram desenvolvidos os algoritmos de tratamento de imagens básicos, como a manipulação do canal de cores RGB, histograma de cores, e obtenção do negativo da imagem. Este primeiro passo teve o intuito de conhecer a estrutura de armazenamento dos pixels. A manipulação desta estrutura trouxe a necessidade de conversão entre os tipos primitivos `byte` e `int`, pois as formas de armazenamento de dados e a de manuseio são diferentes.

As técnicas de limiarização foram desenvolvidas em seqüência, onde se encontrou o

primeiro problema da extração da região da mão. O problema é que independente da escolha de um limiar, a imagem em questão pode contém outros objetos (ruídos) não pertencentes à região da mão (demonstra na Figura 9 no item b).

A partir da constatação destes ruídos técnicas de manipulações de matrizes começaram a ser implementadas. Este fato forçou um rápido estudo nas técnicas básicas como conversão de um vetor para uma matriz e o encontro da maior região conexa da matriz. O algoritmo para encontrar a maior região fora desenvolvido de maneira recursiva, obtendo êxito no seu funcionamento. Porém quando imagens com dimensões maiores foram submetidas ao algoritmo, o mesmo gerou o erro de `StackOverflow` (247 níveis de profundidade). Ou seja, a recursividade tomava proporções em que a falta de memória impedia que a região fosse encontrada. Este problema forçou o algoritmo a ser convertido para o modo iterativo, apesar de otimizações terem sido aplicadas no algoritmo em sua forma recursiva para evitar esta conversão.

Em seguida algoritmos de operações morfológicas foram estudados e desenvolvidos, na sua forma mais simples entre outras técnicas que utilizam vários algoritmos simples. Sendo que alguns algoritmos implementados foram dispensados na construção da ferramenta, como a detecção de bordas, onde constatou-se inicialmente que seria utilizado para as etapas seguintes, o que não ocorreu.

A maior parte dos artigos pesquisados sugeria a criação de uma RNA para realizar a fase de Reconhecimento. Logo a criação de uma RNA foi iniciada para realizar a tarefa de reconhecimentos dos gestos da mão. Esboços de RNAs foram estudados e criados a fim de determinar a melhor RNA a ser aplicada. Em paralelo a leitura do artigo de Ionescu (IONESCU, 2005) chamou a atenção por realizar o reconhecimento sem utilizar a construção de alguma RNA. Este artigo fazia uso do histograma de orientação local e cálculos sobre o esqueleto da região da mão. O reconhecimento através de cálculos sobre o esqueleto da região mão foi escolhido, pois seria um conjunto de técnicas a ser explorada.

A orientação do histograma foi a técnica mais difícil de ser implementada, pois dentre a literatura pesquisada muito pouco era explicado das etapas constituintes. Vários algoritmos foram construídos baseando-se no conteúdo encontrado, sendo que os resultados obtidos não batiam com as imagens de orientação de histograma encontrada em livros e artigos. A leitura da dissertação de Medeiros (2008) enfatizou o uso do filtro de Sobel para o cálculo dos gradientes horizontais e verticais na obtenção da direção das bordas. Neste ponto foi possível compreender que o cálculo dos gradientes dos pixels de toda a região da mão resultaria na direção da imagem. O algoritmo foi desenvolvido e aplicado resultando em histogramas de

orientação satisfatórios.

Dentre a pesquisa realiza percebeu-se que várias literaturas como (FREEMAN; ROTH, 1994) e (ZHOU; LIN; HUANG, 2004) conseguiram realizar o reconhecimento dos gestos da mão apenas com o uso do histograma de orientação, aplicando a distância Euclidiana entre os histogramas de orientação e outras técnicas. Porém não foi possível realizar a reconhecimento apenas com esta etapa desenvolvida, pois não se encontrou uma forma clara o suficiente para realizar o reconhecimento dos gestos da mão.

As tentativas iniciais de esqueletonização produziram esboços de esqueleto com formas indefinidas e muito instáveis, onde a disposição da região da mão dentro da imagem interferia no resultado do esqueleto. O desenvolvimento do algoritmo de Zhang-Suen (seção 2.5.1) produziu o esqueleto da imagem, onde foram obtidos resultados além do esperado (em relação à etapa de tratamento). A implementação desta etapa foi a segunda em questão de dificuldade, porém destacou-se pela qualidade dos resultados obtidos com o algoritmo de Zhang-Suen.

Ao fim da implementação do trabalho 8 passos distintos foram obtidos. A Figura 17 – Imagem representando a letra V e cada passo do tratamento de imagem. Figura 17 ilustra cada um destes passos.

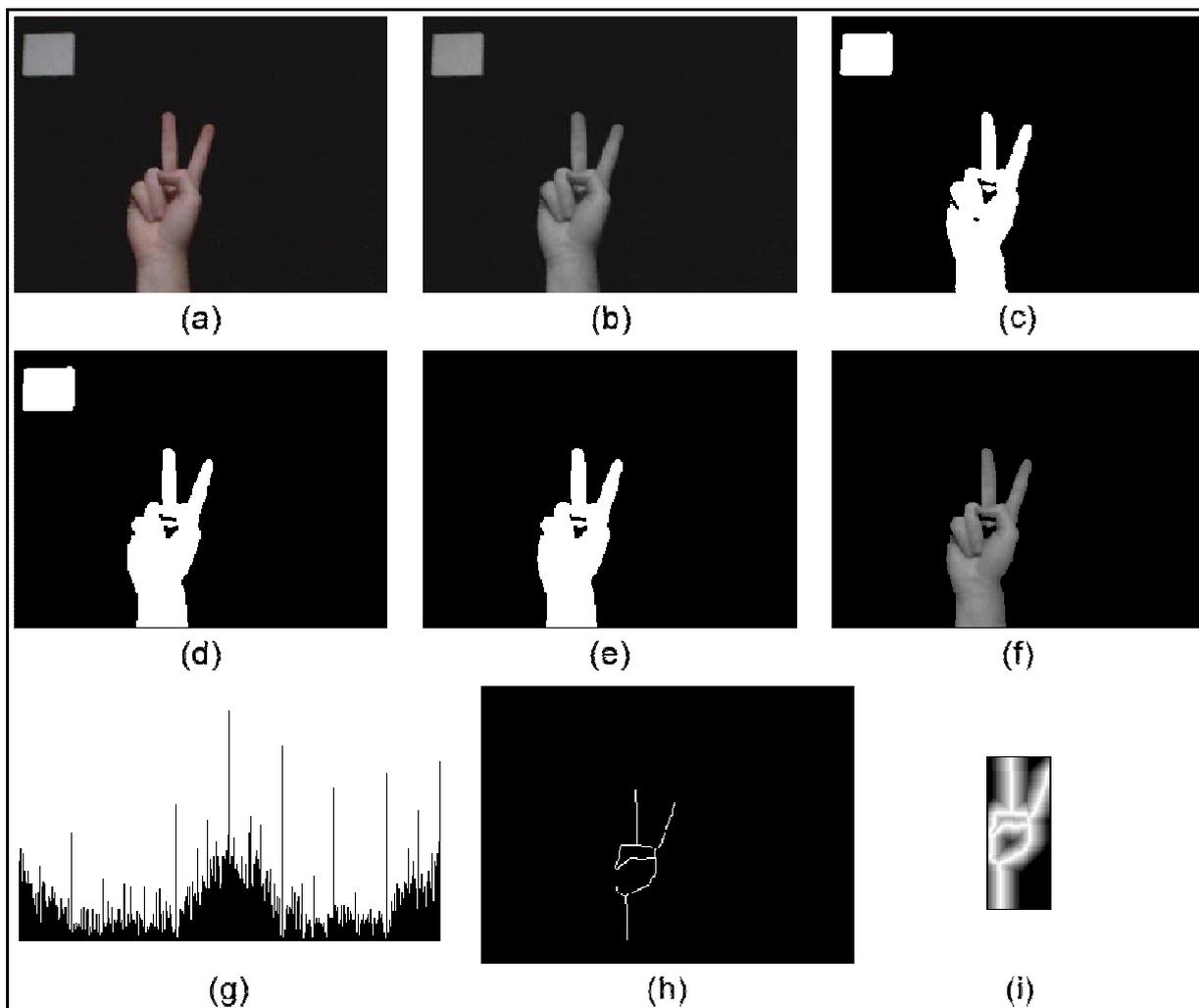


Figura 17 – Imagem representando a letra V e cada passo do tratamento de imagem

A imagem original (a) é convertida para tons de cinza (b) e binarizada (c). As operações morfológicas (d) são aplicadas e região da mão (e) é extraída. A etapa seguinte consiste em restaurar a região da mão com os pixels em tons de cinza (f), para depois poder gerar o histograma de orientação (g). A esqueletonização (h) é realizada gerando a respectiva amostra (i).

3.4.1.1 Requisitos

Ao término da implementação constatou-se que alguns requisitos deveriam ser seguidos para que o reconhecimento torna-se possível. As imagens submetidas ao treinamento e reconhecimento devem cumprir os seguintes requisitos:

- a) o fundo deve ser uniforme e possuir cor mais escura do que a cor da região da mão (preferencial) ou cores inversas;

- b) manter a distância entre a mão e o dispositivo de captura aproximadamente constante;
- c) a mão não deve ser o maior objeto da imagem, bem como não se aproximar de outros objetos existentes;
- d) a luminosidade do ambiente deve permitir a definição da mão, pois a penumbra e o excesso de luz podem atrapalhar o tratamento e o reconhecimento.

Os requisitos descritos relatam dificuldades e pontos a ser melhorados. Cada etapa que possui algum problema poderá ser substituída ou aperfeiçoada com outras técnicas e algoritmos.

3.4.2 Testes

Dentre os testes aplicados ainda na fase de implementação destacam-se os referentes ao histograma de orientação e de esqueletonização. O histograma de orientação local recebeu testes de luminosidade, onde imagens de um mesmo gesto foram submetidas a diferentes pontos de luz. Dentre 12 gestos submetidos, um gesto com particularidades foi escolhido para ser demonstrado. A Figura 18 ilustra o gesto da letra B (LIBRAS) testado sobre dois ambientes de luz.

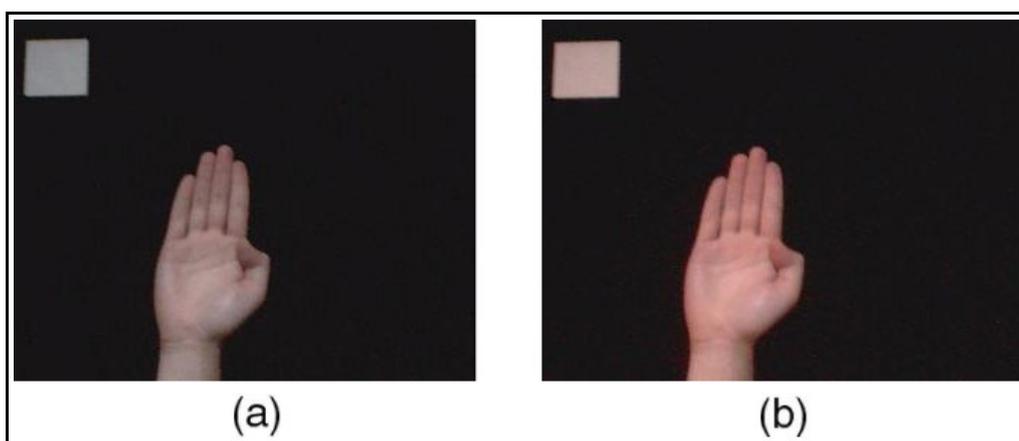


Figura 18 – Imagem representando a letra B exposta a diferentes pontos de luz

Ambas as imagens foram realizadas em um ambiente controlado, submetidas a luz fluorescente. Entretanto a imagem (b) recebeu além da iluminação fluorescente outra concentração de luz com o auxílio de uma lâmpada incandescente. Pode-se observar que imagem (b) ficou mais clara do que a imagem (a) devido a concentração de iluminação.

A aplicação de operações morfológicas das imagens da Figura 18 estão respectivamente demonstradas na Figura 19.



Figura 19 – Aplicação de operações morfológicas das imagens da Figura 18

Observando o resultado das operações morfológicas nas imagens (a) e (b) é possível verificar uma pequena diferença entre ambas. Nota-se o preenchimento de um buraco (preto dentro da região do gesto da mão). Este buraco foi fechado devido a concentração de iluminação, efetuando o clareamento da área. Os esqueletos das imagens contidas Figura 18 estão respectivamente apresentados na Figura 20.

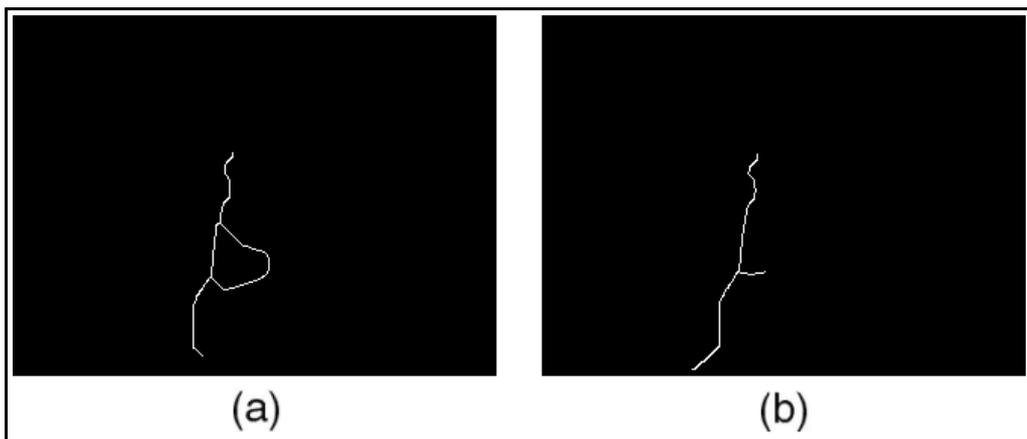


Figura 20 – Esqueletos representando das imagens da Figura 18

Analisando as imagens (a) e (b) é possível destacar a região poligonal. Esta região é formada a partir do buraco, pois o algoritmo de esqueletonização cria uma região poligonal para cada buraco contido na imagem. Os histogramas de orientação das imagens da Figura 18 estão respectivamente exibidos na Figura 21.

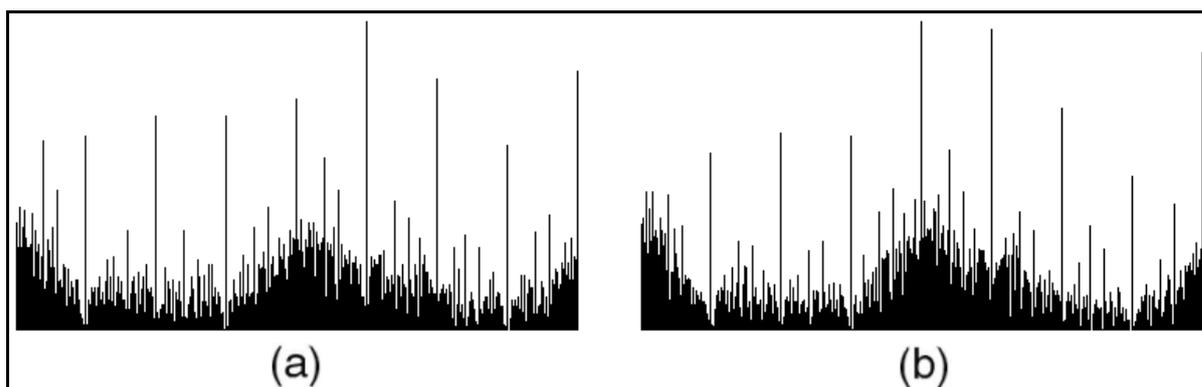


Figura 21 – Histogramas de orientação das imagens da Figura 18

É possível perceber que as imagens similares (a) e (b) geraram histogramas de orientação também similares, porém produzindo resultados diferentes. A imagem (a) possui orientação de 224 graus, enquanto a imagem (b) resulta na orientação de 179 graus.

Baseado nos testes de iluminação foi possível constatar dois problemas nos tratamentos aplicados. O excesso de iluminação provoca a perda das propriedades topológicas da imagem, sendo que de região da mão submetida ao histograma de orientação não retorna a orientação correta. Outro problema é criação de buracos devido a sombras que se formam dentro da região da mão. Estas sombras podem ser ocasionadas por quatro fatores, pela disposição do foco da luz, pela escassez de iluminação, pela escolha ruim de um limiar para a binarização ou pela proximidade das cores do objeto e do fundo.

O teste mais importante refere-se ao reconhecimento dos gestos aplicado sobre o esqueleto da região da mão. Após o treinamento realizado com algumas imagens contendo gestos, as mesmas imagens submetidas ao reconhecimento resultavam em 100% de acerto. Entretanto quando outras imagens dos mesmos gestos foram submetidas para o reconhecimento, a taxa de acerto ficava abaixo de 20%. Imagens muito parecidas não eram reconhecidas, demonstrando a imprecisão desta fase.

O cálculo da distância de Baddeley compara duas imagens, pegando os pixels de cima para baixo da esquerda para a direita. Desta forma outras possibilidades de cálculo surgiram como pegar os pixels dos demais sentidos, de cima para baixo da direita para a esquerda, de baixo para cima da esquerda para a direita e finalmente de baixo para cima da direita para a esquerda. Realizando estes quatro cálculos e selecionando o menor dos valores, a taxa de reconhecimento saltou para 40% de acerto para os gestos submetidos. Apesar do dobro na melhora do reconhecimento os resultados até então eram insatisfatórios.

Logo após a fase de Reconhecimento, calculada com a distância Baddeley, não produzia o resultado esperado. Constatou-se que devido à largura do esqueleto ser muito fina (no máximo 1 pixel de largura e comprimento, como definido na seção 2.5.1), o cálculo da

distância de Baddeley ficava prejudicado. Para resolver este problema foram criadas camadas ao redor do esqueleto, onde cada camada recebia um tom de branco mais escuro para representar a distância do esqueleto. O calculado da distância de Baddeley foi adaptado para considerar estas camadas que contornar o esqueleto. Assim esqueletos da região da mão para gestos parecidos não seriam iguais, mas ficariam muito próximo da amostra treinada, e para o reconhecimento o que vale é a amostra mais próxima da imagem submetida.

Com a fase de Reconhecimento refeita, os mesmos testes aplicados na fase de Reconhecimento inicial foram submetidos. Os resultados obtidos passaram para níveis satisfatórios de 90% de acerto. O custo computacional desta versão do reconhecimento é igual à versão que atingia 20% de acerto e inferior a versão que resultava em 40% de precisão.

De forma semelhante à melhoria aplicada na versão inicial de reconhecimento, onde a os pixels eram obtidas de cima para baixo da esquerda para a direita, uma nova melhoria poderá ser aplicada em versões futuras, o que provavelmente aumentará ainda mais a precisão do reconhecimento.

Outro teste realizado abrangeu diferentes tamanhos de imagens a ferramenta. A imagem padrão tem o tamanho de 320x240 pixels obtendo êxito no reconhecimento de 12 gestos. Imagens de dimensões de 160x120 pixels foram submetidas a testes, onde o custo computacional foi menor. Em contra partida a taxa de reconhecimento diminuiu, pois imagens reconhecidas no tamanho padrão não possuíam o mesmo resultado no tamanho menor. Uma imagem de tamanho maior, com 640x480 pixels, foi aplicada ao uso da ferramenta. Logo percebeu-se um custo computacional muito elevado, demorando 2 segundos para tratar e reconhecer um gesto, enquanto o reconhecimento se manteve acurado.

4 CONCLUSÕES

Com o intuito de auxiliar librianos a comunicar-se com as pessoas que não dominam LIBRAS, conseguiu-se desenvolver uma ferramenta que possibilite essa comunicação em futuras implementações. O uso de imagens é ainda uma maneira simples de realizar esse reconhecimento, sendo necessário evoluir a ferramenta ao ponto em que uma câmera digital possa ser usada.

A implementação da ferramenta contempla o processamento de imagem digital de uma maneira acurada. O passo inicial de binarização é fundamental para se obter a imagem da mão em boas condições, evitando que ruídos sejam acoplados a esta imagem. As operações morfológicas auxiliam no “embelezamento” da imagem, reduzindo os ruídos, buracos e removendo pontas indesejáveis. A obtenção da região da mão possibilita focar os esforços computacionais no objeto de interesse (mão), desprezando possíveis ruídos e outros objetos contidos na imagem. O histograma de orientação é utilizado como filtro de ângulos, proporcionando que a etapa seguinte seja realizada de forma otimizada. A esqueletização foi a implementação mais interessante na percepção visual, apesar de ser razoavelmente simples de se construir, a disponibilidade de literatura auxiliaria na obtenção de resultados ainda melhores.

O treinamento de imagens cria um banco de amostras a partir de imagens de gestos da mão, possibilitando o reconhecimento dos mesmos. O reconhecimento da imagem se dá pelo cálculo sobre as amostras treinadas com uma adaptação da distância de Baddeley, onde a menor diferença encontrada resulta no gesto reconhecido.

Durante o desenvolvimento da ferramenta duas fases distintas puderam ser construídas, o processamento de imagem e o reconhecimento de gestos. Apesar da distinção das fases existe a dependência em nível de código e em resultado, pois a qualidade do processamento da imagem implica diretamente na eficácia do reconhecimento dos gestos.

Dentre os principais fatores analisados durante os testes, constatou-se que a iluminação se mostrou impactante, podendo alterar significativamente a fase de reconhecimento. A qualidade da imagem possui impacto mínimo, uma vez que toda fase de testes recebera imagens em resolução baixa de 320x240 pixels. Em contrapartida imagens com tamanhos superiores consumiriam recurso computacionais exorbitantes, impossibilitando que o reconhecimento seja realizado em tempo real.

4.1 EXTENSÕES

Como extensões para este trabalho sugerem-se:

- a) reconhecer gestos em ambientes onde a existência de ruídos seja abrupta, como o corpo do indivíduo e até mesmo várias pessoas na imagem capturada;
- b) expandir o reconhecimento para gestos com ambas as mãos de um indivíduo;
- c) elaborar uma ferramenta de tradução escrita ou falada com os gestos reconhecidos;
- d) desenvolver um sistema para celular onde as fotos e vídeos capturados possam ser submetidos ao reconhecimento de gestos, levando em consideração a limitação de processamento;
- e) melhorar o problema com o tamanho da imagem capturadas, criando algum mecanismo para manter a proporção de uma mão tanto perto quanto longe da câmera;
- f) possibilitar o reconhecimento do gesto da mão em ambiente 3D, simulado com ao pelo menos um câmera em cada um dos três eixos geométricos;
- g) expandir o reconhecimento para gestos onde existem movimentos, ou seja, uma série de quadros compondo um gesto.

REFERÊNCIAS BIBLIOGRÁFICAS

ACESSOBRASIL. **Dicionário da língua brasileira de sinais**. [S.l.], 2008. Disponível em: <<http://www.acessobrasil.org.br/libras/>>. Acesso em: 18 set. 2008.

CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. **Computação gráfica: teoria e prática**. São Paulo: Campus/Elsevier, 2008.

DICIONARIOLIBRAS. **Dicionário libras**. Praia Grande, SP, 2008. Disponível em: <<http://www.dicionariolibras.com.br/>>. Acesso em: 19 set. 2008.

FERREIRA, Aurélio B. de H. **Aurélio século XXI: o dicionário da Língua Portuguesa**. 3 ed. Rio de Janeiro: Nova Fronteira, 1999.

FISCHER, Graciana S. **Um ambiente virtual multimídia de ensino na web, com transmissão ao vivo e interatividade**. 2000. 125 f. Dissertação (Mestrado em Ciências da Computação) – Instituto de Informática, Universidade Federal de Rio Grande do Sul, Porto Alegre.

FREEMAN, William T.; ROTH, Michal. Orientation histograms for hand gesture recognition. **International workshop on automatic face and gesture**. Zurich, Switzerland, v. 1995, n. 1, p. 296–301, 1995.

GUEDES, Gilleanes T. A. **UML: uma abordagem prática**. São Paulo: Novatec, 2008.

INES. **Vocabulário**. [S.l.], 2008. Disponível em: <<http://www.ines.gov.br/libras/index.htm>>. Acesso em: 18 set. 2008.

IONESCU, Bogdan et al. Dynamic hand gesture recognition using the skeleton of the hand. **EURASIP journal on applied signal processing**, [S.l.], v. 2005, n. 13, p. 2101–2109, 2005.

MEDEIROS, Israel M. **Construção de uma ferramenta voltada à medicina preventiva para diagnosticar casos de estrabismo**. 2008. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PEDRINI, Hélio; SCHWARTZ, William R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. São Paulo: Thomson Learning, 2008.

PLOTZE, Rodrigo O.; BRUNO, Odemir M. Estudo e comparação de algoritmos de esqueletonização para imagens binárias. In: CONGRESSO BRASILEIRO DE COMPUTAÇÃO, 4., 2004, Criciúma. **Anais...** Itajaí: [s.n.], 2004, p. 59-64.

SPARX SYSTEMS. **Enterprise Architect**. Creswick, Austria, 2008. Disponível em: <<http://www.sparxsystems.com.au/products/ea/index.html>>. Acesso em: 8 nov. 2008

VIADER, Maria P. F.; PERTUSA, Esther; VINARDELL, Marta. Importância das estratégias e recursos do educador surdo no processo de ensino/aprendizagem da língua escrita. In: SKLIAR, C. (Org.). **Atualidade da educação bilíngüe para surdos**. Porto Alegre: Mediação, 1999. p. 47-57.

WANGENHEIM, Aldo v. **Morfologia matemática**. Florianópolis, SC, 2001. Disponível em: <<http://www.inf.ufsc.br/~visao/morfologia.pdf>>. Acesso em: 11 out. 2008.

ZHOU, Hanning; LIN, Dennis L., HUANG, Thomas S. Static Hand Gesture Recognition based on Local Orientation Histogram Feature Distribution Model. **Conference on Computer Vision and Pattern Recognition Workshop**. Zurich, Switzerland, v. 2004, n. 10, p. 161, 2004.