

UNIVERSIDADE PAULISTA

DANILO DE OLIVEIRA DOROTHEU

DIEGO DA SILVA SANTANA

MARCIO FERNANDES CRUZ

THIAGO GUY MOZOL VICENTE

**UMA APLICAÇÃO PARA INTERPRETAR SINAIS DO ALFABETO
DIGITAL DE LIBRAS**

SÃO PAULO

2015

DANILO DE OLIVEIRA DOROTHEU

DIEGO DA SILVA SANTANA

MARCIO FERNANDES CRUZ

THIAGO GUY MOZOL VICENTE

**UMA APLICAÇÃO PARA INTERPRETAR SINAIS DO ALFABETO
DIGITAL DE LIBRAS**

Trabalho de conclusão de curso para
obtenção do título de graduação em Ciência
da Computação apresentado à Universidade
Paulista – UNIP.

Orientador: Prof. Me. Gley Fabiano C. Xavier

SÃO PAULO

2015

CIP – Catalogação na Publicação

UMA APLICAÇÃO PARA INTERPRETAR SINAIS DO ALFABETO
DIGITAL DE LIBRAS / Danilo de Oliveira Dorotheu...[et al]. -2015,
62 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao instituto
de Ciência Exatas e Tecnologia da Universidade Paulista, São Paulo,
2015.

Área de Concentração: Reconhecimento de Padrões.
Orientador: Prof. Me. Gley Fabiano Cardoso Xavier.

1. Reconhecimento de Padrões. 2. Visão Computacional. 3.
Processamento Digital de Imagens. I. Dorotheu, Danilo de Oliveira. II.
Xavier, Gley Fabiano Cardoso (orientador).

DANILO DE OLIVEIRA DOROTHEU

DIEGO DA SILVA SANTANA

MARCIO FERNANDES CRUZ

THIAGO GUY MOZOL VICENTE

**UMA APLICAÇÃO PARA INTERPRETAR SINAIS DO ALFABETO
DIGITAL DE LIBRAS**

Trabalho de conclusão de curso para
obtenção do título de graduação em Ciência
da Computação apresentado à Universidade
Paulista – UNIP.

Aprovado em:

BANCA EXAMINADORA

_____/____/____
Prof. Nome do professor
Universidade Paulista - UNIP

_____/____/____
Prof. Nome do professor
Universidade Paulista - UNIP

_____/____/____
Prof. Nome do professor
Universidade Paulista - UNIP

AGRADECIMENTOS

Danilo:

Agradeço a Deus pela oportunidade e por sempre estar comigo, dando-me força e saúde para superar as dificuldades. Aos meus pais, que me proporcionaram o melhor possível, apoiando cada escolha minha, mostrando que “O único bem de uma pessoa é seu conhecimento adquirido através do estudo”. Aos meus mestres que me conduziram pelos caminhos da pesquisa com paciência e dedicação. E por fim, às pessoas que contribuíram em minha vida, incentivando, auxiliando e acreditando no meu potencial.

Diego:

Agradeço a Deus por ter me dado saúde e força para superar as dificuldades. Aos meus pais, pelo amor, incentivo e apoio incondicional. Ao nosso Professor orientador Gley Fabiano Cardoso Xavier, que com paciência e sabedoria nos ajudou, nos guiou e mostrou os melhores caminhos. A todos que fizeram parte da trilha acadêmica e também àqueles que mesmo de fora contribuíram de alguma forma.

Marcio:

Aos meus colegas Danilo, Diego e Thiago por se dedicar a realização deste trabalho. A minha família que sempre acreditou em mim. Dedico também este trabalho ao meu filho Michel. Ele um dia vai ler este texto e espero ter orgulho de seu pai. Por último e mais importante agradeço a Deus e dedico a Ele toda a glória de minhas realizações.

Thiago:

Agradeço primeiramente a Deus por dar-me forças para segui em frente. Aos meus familiares que acreditam no meu sucesso. Aos meus mestres pelas orientações dadas para elaborar este trabalho, pois ajudou a desenvolver minhas ideias. E aos meus amigos que sempre estiveram me apoiando.

A primeira regra de qualquer tecnologia utilizada nos negócios é que a automação aplicada a uma operação eficiente aumentará a eficiência. A segunda é que a automação aplicada a uma operação ineficiente aumentará a ineficiência.

(Bill Gates)

RESUMO

Este trabalho apresenta um estudo de conceitos e técnicas de processamento de imagens que tem como objetivo o reconhecimento de padrões. Através de técnicas de Visão Computacional, tem o desafio de criar um dispositivo artificial que desempenhe a função da visão humana na problemática de reconhecimento de sinais de um alfabeto digital de Libras. Nesta questão, é delimitado o problema, que é especificadamente reconhecer gestos de um alfabeto digital desta língua de sinais. No final, é discutido o estudo e sugerido uma proposta para trabalhos futuros, que, no caso, damos como exemplo o reconhecimento da linguagem Libras em si e, não somente seu alfabeto digital.

Palavras-chave: Visão Computacional. Reconhecimento de padrões. Processamento de imagens.

ABSTRACT

This work demonstrates a study of concepts and image processing techniques in order to recognize patterns. Its objective is creating an artificial device which performs human vision function in signals recognition matter of a Libras digital alphabet. Thus, the problem is delimited, that is specifically recognize gestures of a digital alphabet of this signal language. Finally, a proposal study is discussed and suggested for further work, which, in this case, we give as an example the recognition of Libras language itself and not only its digital alphabet.

Keywords: Computer vision. Pattern Recognition. Image processing.

LISTA DE FIGURAS

Figura 1 - Aquisição da imagem	19
Figura 2 - Etapas de um sistema de VC genérico	21
Figura 3 - Esquerda original, direita filtro mediana com o kernel 9x9	28
Figura 4 - Esquerda original, direita filtro gaussiano com o kernel 9x9 e desvio padrão = 0	29
Figura 5 - Representação da modelagem de fundo através do período de treinamento	31
Figura 6 - Critério utilizado na máscara de segmentação da limiarização	32
Figura 7 - Limiarização do modelo baseado em similaridades, onde os pixels brancos são parte do objeto e os pretos, parte do fundo	32
Figura 8 - Ruídos gerados na limiarização por problemas de iluminação	33
Figura 9 - Esquerda operação de dilatação, direita operação de erosão com o kernel 9x9	35
Figura 10 - Esquerda operação de abertura, direita operação de fechamento com o kernel 9x9	36
Figura 11 - Esquerda sem transformações morfológicas, direita com operações de abertura e fechamento com o kernel 9x9	37
Figura 12 - Esquerda laplace, direita canny	39
Figura 13 – Detecção de contornos	40
Figura 14 - Problema de mais de 1 elemento em cena	41
Figura 15 - Resultado do algoritmo de seleção da silhueta da mão	41
Figura 16 - Resultado do algoritmo de simplificação de poligonais	43
Figura 17 – A fronteira do domínio D é uma curva fechada simples C	44

Figura 18 – Teorema de Green	45
Figura 19 - Resultado do cálculo do centro da área pelo Teorema de Green	45
Figura 20 - Resultado do algoritmo de Graham	47
Figura 21 – Contorno fecho e defeito convexo	48
Figura 22 - Resultado dos defeitos de convexidade	49
Figura 23 - Resultado da classificação de características da mão	50
Figura 24 - Resultado do reconhecimento de um gesto	51
Figura 25 – Ambiente de hardware e sistema operacional utilizado	52
Figura 26 – Objeto de captura da imagem	53
Figura 27 - Gesto reconhecido C	54
Figura 28 - Gesto reconhecido V	54
Figura 29 - Gesto reconhecido L	55
Figura 30 - Gesto reconhecido U	55

LISTA DE TABELAS

Tabela 1 - Resultados dos filtros mediana e gaussiano	30
Tabela 2 - Resultados sem filtro de suavização	56
Tabela 3 – Resultados sem filtros morfológicos	56
Tabela 4 – Resultados com filtros de suavização e morfológicos	56

LISTA DE ABREVIATURAS E SIGLAS

DCBD	Descoberta de Conhecimento em Banco de Dados.
IA	Inteligência Artificial
IRC	Índice de Reprodução de Cor
VC	Visão Computacional

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 Estrutura do trabalho.....	16
2 FUNDAMENTAÇÃO TEÓRICAS.....	17
2.1 Importância da delimitação do problema.....	17
2.2 A imagem digital.....	18
2.3 O conceito de pixel.....	19
2.4 Iluminação.....	20
2.5 Conceituando a Visão Computacional.....	21
2.5.1 Aquisição de imagens.....	22
2.5.2 Realce.....	22
2.5.3 Restauração.....	23
2.5.4 Segmentação.....	23
2.5.5 Extração de características.....	23
2.5.6 Classificação e Reconhecimento.....	24
2.5.7 Decisão.....	24
2.6 Vizinhança de um <i>pixel</i>	25
2.6.1 Conectividade e adjacência.....	25
2.6.2 Convolução com máscaras.....	26
3 ARQUITETURA.....	27
3.1 Filtros de suavização.....	27
3.1.1 Filtro da mediana.....	27
3.1.2 Filtro de Gauss.....	28
3.1.3 Comparativo.....	29
3.2 Segmentação.....	30
3.2.1 Subtração de fundo.....	30
3.2.2 Limiarização.....	31
3.2.3 Resultados.....	33
3.3 Transformações Morfológicas.....	33
3.3.1 Erosão e Dilatação.....	34
3.3.2 Abertura e fechamento.....	35

3.4 Detecção de borda	37
3.4.1 Laplaciano	38
3.4.2 <i>Canny</i>	38
3.4.3 Comparativo	39
3.5 Detecção de contorno	39
3.5.1 Lista de códigos	40
3.6 Seleção da silhueta da mão	40
3.7 Simplificação da silhueta da mão	42
3.7.1 Simplificação de poligonais com o método Douglas-Peucker	42
3.8 Detecção das características da mão	43
3.8.1 Cálculo do centro da área	44
3.8.2 Envoltório convexo	45
3.8.3 Defeitos de convexidade	47
3.9 Classificação das características da mão.....	49
3.9.1 Ponta dos dedos	49
3.10 Reconhecimento	50
3.10.1 Gestos do alfabeto digital de libras	51
4 COMPARATIVO DE RESULTADOS.....	52
4.1 Ambiente de <i>hardware</i> e sistema operacional.....	52
4.2 Equipamento para aquisição de imagens.....	53
4.3 Cenário controlado	53
4.4 Gestos de teste	53
4.5 Resultados finais sem filtro de suavização.....	56
4.6 Resultados finais sem filtros morfológicos.....	56
4.7 Resultados finais com aplicação de filtros de suavização e morfológicos.....	56
4.8 Conclusão sobre a análise dos resultados	57
5 CONCLUSÃO	58
5.1 Trabalhos Futuros	59
REFERÊNCIAS.....	60

1 INTRODUÇÃO

A linguagem Libras é a língua gestual utilizada pela comunidade surda no Brasil. Embora antiga e com garantias previstas em leis, seu uso não é planejado no país como é a Língua Portuguesa.

Art. 1º É reconhecida como meio legal de comunicação e expressão a Língua Brasileira de Sinais - Libras e outros recursos de expressão a ela associados. Parágrafo único. Entende-se como Língua Brasileira de Sinais - Libras a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil. (BRASIL, 2002, lei 10.436, p. 1).

Uma pessoa surda, conhecedora da linguagem de sinais consegue se comunicar bem em sua comunidade ou juntos de seus familiares. Entretanto sente dificuldade no convívio com pessoas ouvintes pelo fato destas últimas geralmente conhecer somente a Língua Portuguesa. Esta dificuldade é observada em ambientes sociais e no trabalho. O fato dos ouvintes não conhecerem Libras gera a exclusão social da pessoa surda. É uma questão social que deve ser combatida.

A proposta deste trabalho é criar uma aplicação que ajude a comunicação de pessoas surdas com pessoas ouvintes. Estas últimas verão em tela a letra da Língua Portuguesa que representa o gesto que o interlocutor em Libras fez em frente a uma câmera acoplada ao computador.

O objetivo do trabalho é criar um sistema facilitador, que faça o computador reconhecer um gesto em Libras, assim, exercendo parcialmente a função de um profissional intérprete. A função parcial de intérprete é dada pelo fato que o sistema irá se focar no reconhecimento de algumas letras do alfabeto digital¹ de Libras, assim, confirmando a hipótese que é possível criar, através de recurso de visão computacional um aplicativo para esta função.

O objetivo deste trabalho não é criar um aplicativo que substitua a função de um intérprete bilíngue em Libras, profissão esta amparada pela lei. Dado o nível de qualificação exigida para esta profissão, à lei é bem clara no artigo 17 “A formação do tradutor e intérprete de Libras - Língua Portuguesa deve efetivar-se por meio de curso superior de Tradução e Interpretação, com habilitação em Libras - Língua

¹ Alfabeto digital é uma tabela que contém as letras de A a Z e números de 0 a 9 e o gesto, único para cada caractere.

Portuguesa.” (BRASIL, Decreto 5.626 de 22 de dezembro de 2005).

Para criar um sistema computacional que reconheça gestos executados em Libras é necessário o computador “enxergar” o ambiente. Para isso, foi recorrido o estudo de Visão Computacional, para trazer conceitos e técnicas para direcionar a confirmação de hipótese, que é criar um sistema intérprete de Libras e Língua Portuguesa.

Sobre a Visão Computacional, Conci, Azevedo e Leta (2008, p. 5) nos diz: “[...] trata da extração de informações das imagens e da identificação e classificação de objetos presentes nessa imagem”.

1.1 Estrutura do trabalho

No capítulo 2 é apresentado a fundamentação teórica de Libras e Visão Computacional. É informado o porquê da delimitação do tema do trabalho, que é reconhecer somente algumas letras do alfabeto digital de Libras, bem como descrever conceitos principais das etapas comuns aos sistemas que utilizam a Visão Computacional.

No capítulo 3, é descrito a função dos métodos utilizados no protótipo desenvolvido neste trabalho, bem como descrito outros métodos equivalentes.

No capítulo 4 é apresentado um comparativo de resultados entre métodos que podem ser utilizados em um sistema de visão computacional do tema proposto do trabalho e, é descrito a análise dos métodos escolhidos, que possuem a melhor acurácia num menor tempo de processamento.

No capítulo 5 é descrito a conclusão da pesquisa realizada, bem como é apresentado uma proposta para Trabalhos Futuros.

2 FUNDAMENTAÇÃO TEÓRICAS

O capítulo inicia introduzindo a importância de se delimitar o escopo de atuação de um sistema de visão computacional e do porque em algumas aplicações de Visão Computacional é imprescindível utilizar em conjunto a Inteligência Artificial.

Continuando o capítulo é descrito o que é uma imagem digital e sua composição mais básica, o pixel.

Por fim, é descrito o que é e quais são as etapas comuns de sistemas de visão computacional.

2.1 Importância da delimitação do problema

Com relação ao uso da capacidade cerebral pela visão, é afirmado: “De todos os sentidos humanos, a visão é que fornece a maior quantidade de informação a ser processada pelo cérebro, estima-se que metade do processamento cerebral humano se destina à visão. [...]” (CONCI; AZEVEDO; LETA, 2008, p. 12).

Da mesma forma, um sistema de visão computacional ao receber estímulos de um sensor como uma câmera, deve atuar somente no domínio de problema a que foi programado. Por exemplo, se a razão de existência de determinado sistema for localizar aumento de áreas de desmatamento a partir de imagens de satélite, o algoritmo não deve gerar processamento para questões como demarcar cidades ou regiões fluviais. Para ser eficiente, o sistema deve atuar somente em função do domínio do problema a que foi designado.

Ainda segundo Conci, Azevedo e Leta (2008), o sistema de visão humana não é somente o olho, existe todo um contexto envolvendo informações armazenadas no cérebro como experiências anteriores e padrões. Da mesma forma, alguns sistemas de visão computacional podem recorrer a Inteligência Artificial para que desempenhe bem sua função, tornando a Visão Computacional uma área multidisciplinar.

Se a informação visual puder ser padronizada, o objetivo da análise pode ser seu reconhecimento ou classificação, considerando um catálogo de padrões possíveis [...] Essa forma particular é chamada de reconhecimento de padrões [...]. (CONCI; AZEVEDO; LETA, 2008, p. 5).

Para pretensões de novos projetos em visão computacional, o escopo e os mínimos detalhes de funcionamento devem ser bem definidos. Pois, qualquer item que seja intangível, pode inviabilizar o projeto. Por exemplo, se for necessário um sistema que necessite de respostas rápidas, próximas às humanas, hoje pode ser impossível.

A velocidade de resposta de um sistema humano é da ordem de 0,06 s (não se detectam mais de 30 quadros por segundo em animações). [...] Na visão computacional, a velocidade de aquisição depende diretamente do tamanho da imagem, do tempo de captura do equipamento e do tipo de câmera. Dependendo desses fatores, pode-se considerar que a aquisição de imagens em VC é cerca de 10 vezes maior que na visão humana. (CONCI; AZEVEDO; LETA, 2008, p. 61).

Entretanto, Conci, Azevedo e Leta (2008, p. 60) afirma que no aspecto da qualidade das medições de dados quantitativos, a visão computacional se comporta de forma bem superior à humana. “A visão humana pode discernir de 10 a 20 níveis de cinza, enquanto dispõe para a visão computacional facilmente de 256 tons de cinza”. (CONCI; AZEVEDO; LETA, 2008, p. 61). O autor cita as áreas médicas e espaciais como exemplos que se servem bem de aplicações de visão computacional devido aos requisitos de qualidade das medições.

Na próxima sessão é descrito o que é uma imagem digital, que, por fim, é o objeto de estudo da visão computacional.

2.2 A imagem digital

Segundo Scuri (2002), as imagens do mundo real são formadas por infinitas representações de valores de posição espacial, cores e outros atributos como brilho e resolução. O autor ainda ressalta que imagens digitais nunca são modelos fidedignos da realidade. Isto ocorre pelo fato que um computador possui arquitetura binária e possui impossibilidade técnica de apresentar números contínuos.

O computador só é capaz de armazenar bits, um valor que pode ser 0 ou 1. Para simplificar as coisas juntou-se 8 bits formando uma palavra chamada de byte. Um byte então pode assumir $2^8 = 256$ valores diferentes, variando de 0 a 255.[...].

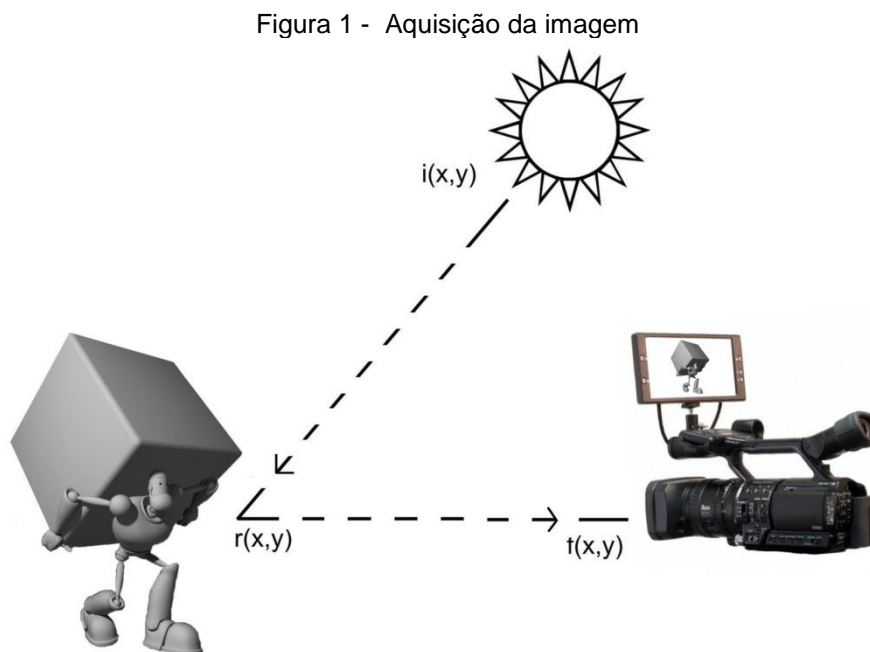
O computador trabalha sempre com números inteiros ou na melhor das hipóteses com uma aproximação de um número real, chamada de ponto flutuante (isso porque o número de bits dedicado para as casas decimais é flexível de acordo com o número). Por consequência, não é possível representar uma função contínua no computador. Podemos apenas simulá-la. (SCURI, 2002, p. 14).

A imagem digital é em essência uma simulação discreta da realidade. É definida por Gonzalez e Woods (2000) como uma matriz bidimensional, de função no formato $f(x,y)$. Cada ponto da imagem é chamado de pixel.

2.3 O conceito de pixel

Segundo Scuri (2002), pixel vem do inglês *Picture Element*. “O pixel pode ser definido como um elemento da imagem ou a menor unidade individual da imagem.” (CONCI; AZEVEDO; LETA, 2008, p. 69).

Mais importante do que explicar o que é um pixel é conceituar como é gerado o valor que o representa. Segundo Conci, Azevedo e Leta (2008), a formação do valor de um pixel é definida pela taxa de reflexão do objeto em sua posição real, bem como a taxa de intensidade luminosa do ambiente.



Fonte: Conci; Azevedo; Leta, 2008.

Conforme ilustrado na figura 1 e enfatizado por Conci, Azevedo e Leta (2008), o fator iluminação é algo importante para definir a qualidade da imagem.

2.4 Iluminação

Para um sistema de visão computacional ser eficiente deve-se primeiramente ter qualidade desde o início, que é o processo da aquisição de imagem.

Alguns sistemas estarão em operação ao ar livre, sob a influência de luz natural, outros, estarão sob a iluminação artificial.

Dependendo do domínio da aplicação, um sistema deve interpretar corretamente as cores, outros, simplesmente, extrair a forma de determinado objeto. Assim sendo, a iluminação é um fator importante no processo.

Para a iluminação artificial, existem vários modelos e marcas disponíveis. “Cada fabricante desenvolve modelos de lâmpadas segundo as necessidades de uso percebidas na sociedade.” (CONCI; AZEVEDO; LETA, 2008, p.28).

O foco deste trabalho não é descrever cada categoria ou modelo de lâmpada disponível, mas sim, demonstramos um conceito importante que é intrinsecamente ligado à questão de cor, que é o IRC (Índice de Reprodução de Cor).

Um conceito importante no estudo de iluminantes é o índice de reprodução de cor (IRC). O IRC é a variação que se observa a partir da visão da cor dos objetos sob o iluminante em estudo, em relação a um iluminante-padrão [...] O IRC é definido como 100% para lâmpadas incandescentes. (CONCI; AZEVEDO; LETA, 2008, p. 33).

Ainda segundo os autores citados acima, as lâmpadas artificiais no geral estão classificadas em dois tipos, incandescente e de descarga. Cada categoria possui subtipos que devem ser minuciosamente escolhidos em função da necessidade do sistema de visão computacional a ser desenvolvido. Por exemplo, existem tipos de lâmpadas de descarga que tem um índice de IRC variando de 10% a 98%.

Alguns modelos podem ser mais caros e econômicos, porém traz um índice de variação de cor mais próximo ao iluminante-padrão, que podem ser fontes naturais de iluminação ou lâmpadas incandescentes. Outros modelos podem ser mais baratos, mas, que trazem um custo energético alto, como é o caso de lâmpadas incandescentes, porém, trazem um alto índice de IRC.

2.5 Conceituando a Visão Computacional

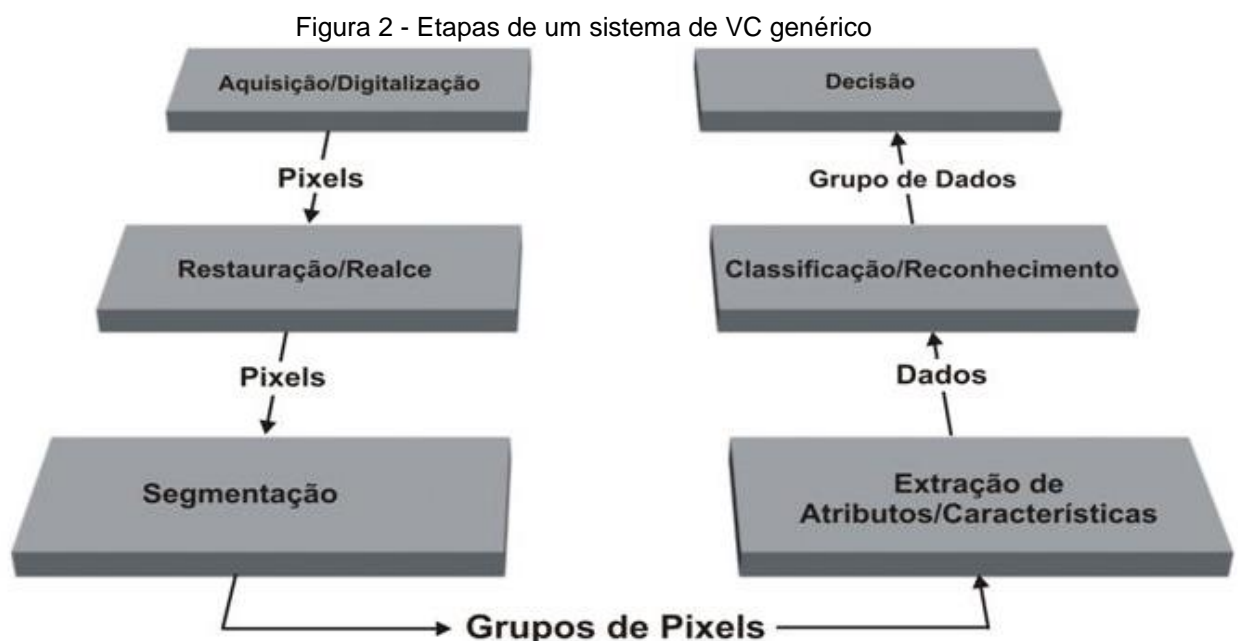
Conci, Azevedo e Leta (2008, p. 5) nos diz que a visão computacional (VC), é uma área dentro da computação gráfica relacionada à análise de imagens. “[...] trata da extração de informações das imagens e da identificação e classificação de objetos presentes nessa imagem”.

Sobre a abrangência de atuação, o autor cita como exemplo que essa área atua em reconhecimento de pessoas, objetos e peças fora do padrão de uma linha de produção, sensoriamento remoto, etc.

É citado por Conci, Azevedo e Leta (2008, p. 5) que a visão computacional pode atuar em conjunto com a área de Inteligência Artificial. Isso ocorre quando há necessidade de tomada de decisão a partir de informações extraídas de imagens. Nesse contexto, o autor expõe outra definição para a área.

[...] Nesse contexto, pode-se definir a visão computacional como o domínio da ciência da computação que estuda e aplica métodos que permitem aos computadores “compreenderem” o conteúdo de uma imagem. (CONCI; AZEVEDO; LETA, 2008, p. 5).

As etapas de processamento de imagem de um sistema de visão computacional serão explicadas nas próximas sessões. A figura 2 apresenta as etapas de um sistema de VC genérico.



Fonte: Conci; Azevedo; Leta, 2008.

Para um sistema de visão computacional ser assertivo em sua operação, deve aplicar corretamente uma sequência de técnicas de processamento de imagens digitais. Em cada domínio do problema, há várias possibilidades da escolha da sequência de técnicas e métodos. O desafio dos projetistas é ser assertivo na escolha da técnica tendo em vista sempre a eficiência geral do sistema. E, para ser assertivo, deve-se recorrer à literatura e executar testes de desempenho em cada etapa, escolhendo algumas técnicas propostas na literatura.

2.5.1 Aquisição de imagens

De acordo com Gonzalez e Woods (2002), para a etapa de aquisição ser realizada, são necessários dois componentes: dispositivo físico de captura e um digitalizador. “Digitalizador é um dispositivo para a conversão da saída elétrica de um dispositivo de sensoriamento físico para a forma digital”. (GONZALEZ; WOODS, 2002, p. 7).

2.5.2 Realce

Segundo Conci, Azevedo e Leta (2008, p. 55) “Realce tem por objetivo destacar detalhes da imagem que são de interesse para análise ou que tenham sofrido deterioração”.

Gonzalez e Woods (2000, p. 115) ressalta que as técnicas de processamento de imagens utilizadas para o realce, são divididas em duas categorias: métodos de domínio espacial e métodos no domínio de frequência.

O domínio espacial refere-se ao próprio plano da imagem, e as abordagens nesta categoria são baseadas na manipulação direta dos pixels das imagens. Técnicas de processamento no domínio da frequência são baseadas na modificação das transformadas de Fourier da imagem. (GONZALES; WOODS, 2002, p. 115).

Conci, Azevedo e Leta (2008, p. 117) chamam as operações de domínio de frequência de operações globais na imagem e, assim define: “Operações globais são operações em que um pixel da imagem resultante depende de um processamento realizado em todos os pixels da imagem original”.

2.5.3 Restauração

Segundo Gonzalez e Woods (2002, p. 180) a restauração estuda conceitos e técnicas que tem o objetivo de recuperar imagens degradadas, principalmente de obras de arte e fotografias antigas. Técnicas desta etapa de processamento não são utilizadas neste trabalho.

2.5.4 Segmentação

Gonzalez e Woods (2002, p. 295) diz que a segmentação é a primeira etapa do processo de análise de imagem. O objetivo principal é, a partir da definição do domínio do problema, separar os objetos de uma imagem para servir as etapas seguintes no processo geral de interpretação e reconhecimento.

Para imagens monocromáticas, segundo Gonzalez e Woods (2002, p. 205), as técnicas se baseiam em duas propriedades da imagem, que são a descontinuidade e similaridade de tons de cinza.

Para sistemas de visão computacional ainda existem técnicas de segmentação que se baseiam em algo que o sistema de visão humana utiliza para identificar um objeto, que é a movimentação. Se determinado objeto se mover, entre a leitura de uma imagem e outra, significa que aquele objeto é algo que pode ser segmentado.

O movimento é uma poderosa pista usada por seres humanos e animais para a extração de um objeto de interesse de um fundo de detalhes irrelevantes. Em aplicações de imageamento, o movimento é originado de um deslocamento relativo entre o sistema sensor e a cena sendo observada [...]. (GONZALES; WOODS, 2002, p. 331).

2.5.5 Extração de características

Segundo Conci, Azevedo e Leta (2008, p. 56) a extração de características é aplicada em função de imagens previamente segmentadas. Nesta fase, temos os dados binários da imagem e também informações relativas à imagem original, como luminosidade ou textura.

De acordo com o domínio do problema, deve-se verificar os atributos relevantes de cada objeto segmentado.

Segundo os autores citados acima, algumas técnicas de extração de características vão analisar os dados binários. Como exemplo é citado a extração de características como dimensão e geometria dos objetos segmentados.

Em alguns casos, a extração de características é executada diretamente na imagem original. É o caso da extração de informações de luminosidade e textura dos objetos.

Após a aplicação de técnicas nesta etapa serem concluída, o sistema de visão computacional alimenta a próxima etapa, que é a de classificação e reconhecimento dos objetos.

2.5.6 Classificação e Reconhecimento

Nesta etapa, o sistema de visão computacional, de posse de uma base de dados onde há padrões de objetos, busca nas propriedades extraídas na fase anterior um comparativo. Tentando conseguir uma classificação adequada ou aproximada para cada objeto.

Gonzalez e Woods (2002, p. 409) define padrão como: “[...] um padrão é uma descrição quantitativa ou estrutural de um objeto ou alguma outra entidade de interesse em uma imagem [...]. Em geral, um padrão é formado por um ou mais descritores”.

Segundo Conci; Azevedo e Leta (2008, p. 260), a etapa de classificação e reconhecimento de padrões, torna a visão computacional uma área multidisciplinar.

O termo padrão, no contexto de análise de imagem, visão computacional e inteligência artificial, se referem a qualquer aspecto que pode ser padronizado em uma estrutura armazenável [...].
O reconhecimento de padrões se relaciona não só com o processamento de imagem, mas também com a inteligência artificial [...]. (CONCI; AZEVEDO; LETA, 2008, p. 260).

2.5.7 Decisão

Esta etapa executa técnicas intrinsecamente relacionadas ao que um sistema de visão computacional está proposto a fazer. “O objetivo de um sistema de visão computacional é tomar decisões a partir da extração de informações do mundo real através de imagens” (CONCI; AZEVEDO; LETA, 2008, p. 57).

Quando o processamento chega nessa etapa, temos um padrão de um objeto e não o objeto em si. Sendo assim, devem-se usar técnicas específicas para tentar buscar o que é o objeto numa base de conhecimento preexistente.

A descoberta de conhecimento em banco de dados (DCBD) é o processo de identificar nos dados, as informações previamente desconhecidas, potencialmente útil e compreensível, visando melhorar o entendimento de um problema e os procedimentos de tomada de decisão [...]. A mineração de dados é a etapa em DCBD responsável pela seleção dos métodos a serem utilizados para localizar padrões nos dados [...]. (CONCI; AZEVEDO; LETA, 2008, p. 57).

2.6 Vizinhança de um *pixel*

Para a correta utilização ou mesmo futura criação de algoritmos de processamento digital de imagens, é necessário conhecer as propriedades de uma imagem digital, mais precisamente, saber analisar sua unidade mais básica, o *pixel*, bem como, a sua vizinhança.

Um *pixel* p , de coordenadas (x,y) , tem 4 vizinhos horizontais e verticais, cujas coordenadas são $(x+1, y)$, $(x-1, y)$, $(x, y+1)$ e $(x, y-1)$. Estes *pixels* formam a chamada "4-vizinhança" de p , que será designada $N_4(p)$. Os quatro vizinhos diagonais de p são os *pixels* de coordenadas $(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y-1)$ e $(x+1, y+1)$, que constituem o conjunto $N_d(p)$. (MARQUES FILHO; VIEIRA NETO, 1999, p. 25).

2.6.1 Conectividade e adjacência

Outro conceito importante é a conectividade de um *pixel*.

A conectividade entre pixels é um importante conceito usado para estabelecer limites de objetos e componentes de regiões em uma imagem. Para se estabelecer se dois pixels estão conectados, é necessário determinar se eles são adjacentes segundo algum critério e se seus níveis de cinza satisfazem a um determinado critério de similaridade. (MARQUES FILHO; VIEIRA NETO, 1999, p.25).

Ao conceito de conectividade, o autor cita a adjacência e o caminho. O primeiro, segundo o autor, um pixel é adjacente ao outro se ambos estiverem conectados. Quanto ao caminho, é uma sequência de *pixels* que possuem os mesmos critérios de adjacência.

2.6.2 Convolução com máscaras

As máscaras são matrizes bidimensionais que são aplicadas as imagens, a fim de servir de apoio a vários tipos de operações de processamento digital de imagens. Sobre convolução, são operações realizadas em um determinado pixel que dependem de uma máscara a aplicada a sua vizinhança.

As operações lógicas e aritméticas orientadas a vizinhança utilizam o conceito de convolução com máscaras (ou janelas ou templates)[...] As operações de convolução com máscaras são amplamente utilizadas no processamento de imagens. (MARQUES FILHO; VIEIRA NETO, 1999, p. 33).

3 ARQUITETURA

Neste capítulo serão demonstradas técnicas específicas para criar sistemas de visão computacional.

3.1 Filtros de suavização

As técnicas de filtragem fazem parte do pré-processamento e, utilizam máscaras, também conhecidas como *kernel*², para percorrer pixel a pixel a matriz de uma determinada imagem. Seu objetivo é realizar transformações na imagem a partir do elemento central e da intensidade de seus vizinhos presentes no *kernel*.

Os filtros são denominados 'passa-baixas' quando atenuam ou eliminam as componentes de alta frequência no domínio das transformadas de Fourier. Como as componentes de alta frequência correspondem a regiões de bordas e/ou detalhes finos na imagem, o efeito da filtragem passa-baixas é a suavização da imagem, provocando um leve borramento na mesma. (MARQUES FILHO; VIEIRA NETO, 1999, p.85).

3.1.1 Filtro da mediana

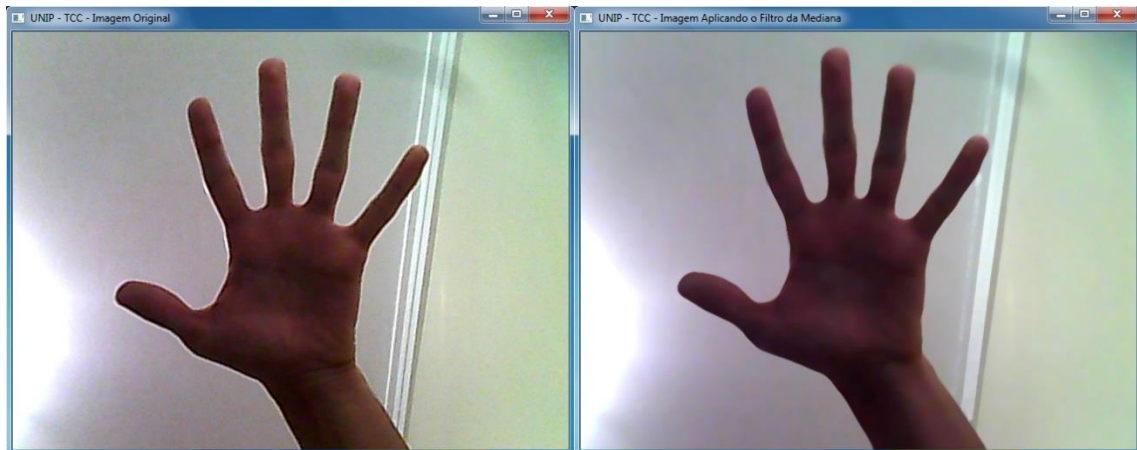
O filtro da mediana, bem como os filtros que serão abordados em seguida, é classificado como filtro passa-baixas, onde buscam melhorar a imagem através de técnicas de suavização que utilizam o *kernel*.

Dado uma imagem, o algoritmo deste filtro tem como entrada um número inteiro positivo sendo ele maior ou igual a três, mas pela ferramenta do *OpenCV* sempre deverá ser ímpar, pois a mediana requer o elemento central da distribuição dos dados e, seu custo será maior ao calcular a média de dois elementos centrais.

Seu objetivo é aplicar ao elemento central do *kernel* o valor da mediana de todos os pixels da região. Este algoritmo tem a vantagem de reduzir os ruídos na imagem, bem como preservar a borda, como mostra a imagem abaixo.

² *Kernel* é sinônimo de máscaras (máscaras são matrizes bidimensionais, conforme descrito no item Convolução com máscaras).

Figura 3 – Esquerda original, direita filtro mediana com o *kernel* 9x9



Fonte: Extraída do protótipo desenvolvido.

O grande problema é o custo de processamento, pois, para se encontrar a mediana de determinada sequência de números, é necessário ordenar os valores em ordem numérica. Isto ocasiona um custo de processamento, pois quanto maior for o *kernel*, mais custoso será para ordenar a lista de cada pixel da matriz percorrida.

A mediana m de um conjunto de n elementos é o valor tal que metade dos n elementos do conjunto situem-se abaixo de m e a outra metade acima de m . Quando n é ímpar, a mediana é o próprio elemento central do conjunto ordenado. Nos casos em que n é par, a mediana é calculada pela média aritmética dos dois elementos mais próximos do centro. A ordenação (sorting) constitui uma etapa de tempo de processamento relativamente alto, apesar de inúmeros métodos eficientes existentes na literatura. (FILHO; NETO, 1999, p.112).

3.1.2 Filtro de Gauss

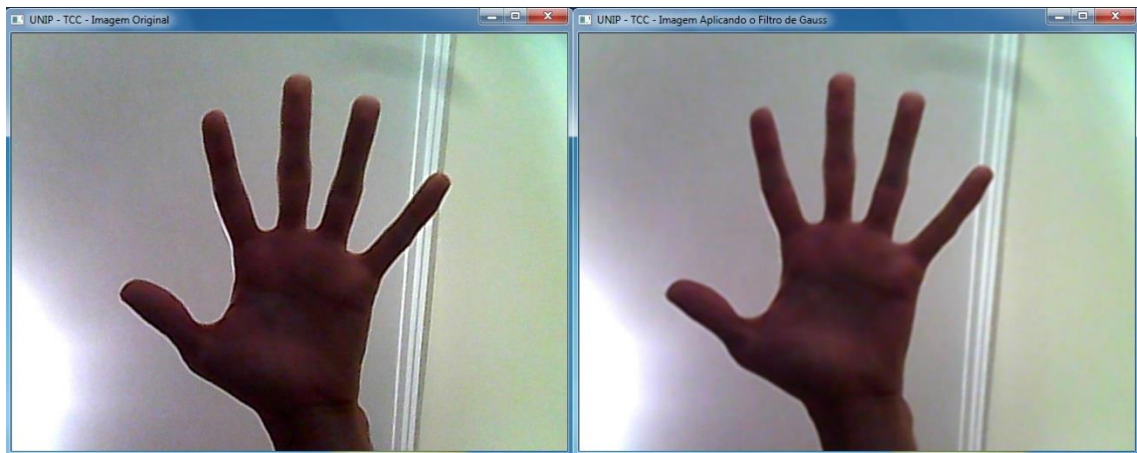
O filtro gaussiano se caracteriza por ser extremamente rápido, percorre a matriz pixel a pixel, assim como os demais filtros, porém diferente do filtro da mediana, ele possui a vantagem de não precisar ordenar as listas em ordem numérica.

O algoritmo deve ter como entrada dois números inteiros positivos e ímpares, sendo maior ou igual a 3 que, determinam a dimensão do *kernel*. Dois desvios padrão, sendo maior ou igual a 0, tanto para direção X como para direção Y que, quanto maior for seu desvio padrão, maior será o efeito de suavização. E por último, servindo como opcional, o tipo da borda em que será aplicado um algoritmo de

interpolação que visa preservar as bordas da imagem.

Sua atribuição ao elemento central do *kernel* está relacionada com a média de todos os pixels da região, levando em considerações seus pesos que são diferentes para cada vizinho, deixando os pesos maiores para os vizinhos mais próximos do centro. Este algoritmo também tem a vantagem de reduzir os ruídos e preservar a borda, como mostra a imagem abaixo.

Figura 4 - Esquerda original, direita filtro gaussiano com o *kernel* 9x9 e desvio padrão = 0



Fonte: Extraída do protótipo desenvolvido.

Repare que os detalhes presentes na imagem de agora não se encontram modificados.

O filtro Gaussiano é uma função que utiliza diferentes pesos para a vizinhança de um pixel de acordo com a distância do ponto vizinho para o ponto que receberá o novo valor. Dessa forma, o filtro Gaussiano suaviza a imagem considerando que os pixels mais próximos ao pixel de interesse devem ter maior influência sobre o novo valor do que os pixels mais distantes. (CARDIM, 2015, p.69).

3.1.3 Comparativo

Apesar do filtro da mediana se mostrar eficiente na redução de ruídos, que é um fator importante para favorecer um melhor reconhecimento, ela também prejudica o elemento na imagem, que no caso é uma mão, pois os dedos sendo finos e, próximos um do outro, uma mediana de *kernel* um pouco grande poderia ocasionar na junção ou, inversamente, na redução e, até mesmo na não aparição dos dedos. Além disso, seu tempo de ordenação resultou em tempo de processamento superior ao filtro gaussiano.

O filtro de Gauss costuma ser a solução de muitas aplicações, pois além de ser eficiente, se sobressai muito bem em relação ao tempo de processamento quando comparado aos demais filtros. Por esses motivos decidimos escolhê-lo para permitir uma menor taxa de erro na etapa do reconhecimento.

Tabela 1 – Resultados dos filtros mediana e gaussiano

Filtro	<i>kernel</i>	TEMPO DE PROCESSO
Mediana	9X9	70ms
Gaussiano	9X9	10ms

Fonte: Elaborada pelos autores.

3.2 Segmentação

Neste capítulo são apresentados algoritmos cujo quais são utilizados para a identificação e detecção de objetos em movimento em um fundo estático, bem como o impacto destes métodos sob influências externas, tais como mudança de iluminação, sombras e até mesmo, flutuações nas fontes de luz.

Segundo Gonzalez e Woods (2002, p. 295), a segmentação é a primeira etapa do processo de análise de imagem. O objetivo principal é, a partir da definição do domínio do problema, separar os objetos de uma imagem para servir as etapas seguintes no processo geral de interpretação e reconhecimento.

3.2.1 Subtração de fundo

Para que um objeto seja segmentado de uma imagem, é necessário que este se movimente em um fundo estático. Truyenque (2005, p.30) explica bem o conceito de segmentação por subtração de fundo:

Uma abordagem muito utilizada para a segmentação de objetos em movimento do fundo da cena é a subtração de fundo. A idéia da subtração de fundo é subtrair da imagem atual uma imagem de referência, a qual é adquirida e modelada a partir de um fundo estático durante um certo período de tempo, conhecido como período de treinamento. Assim, o fundo pode ser qualquer um, desde que permaneça razoavelmente estático (TRUYENQUE, 2005, pg. 35).

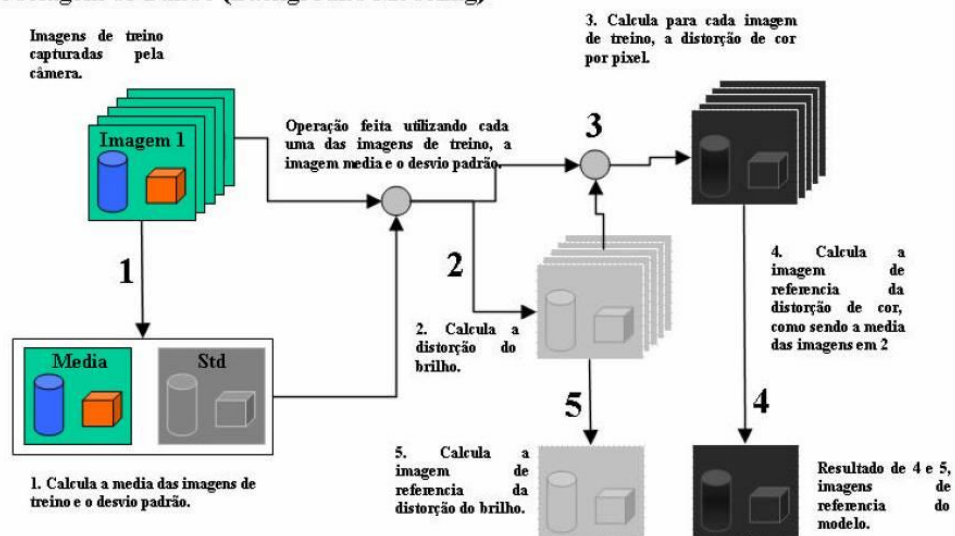
Portanto, a ideia da subtração do fundo é extrair o objeto de um fundo pré-estabelecido pelo período de treinamento. Truyenque (2005, p. 41) descreve

resumidamente, que o processo de treinamento é iniciado com a captura de um número de imagens de treino, a partir das quais se calcula uma imagem média e outra representando o desvio padrão. Então, para cada uma das imagens de referência calculam-se as imagens da distorção de brilho e da distorção de cor em relação à imagem média. Quando o processo de treino do algoritmo é finalizado, são obtidas 4 imagens de referência, sendo elas a imagem média, a imagem do desvio padrão médio, a imagem da distorção média de brilho e a imagem de distorção média de cor.

Quando um objeto entra na cena, São identificados os pixels que sofreram diferenciações na imagem média. Então, cálculos limiares definem estes pixels com uma estrutura binária cujo qual será abordada na próxima seção. A imagem seguinte representa todos estes passos descritos por Truyenque (2005).

Figura 5 - Representação da modelagem de fundo através do período de treinamento

Modelagem do Fundo (Background Modeling)



Fonte: SOUZA³

3.2.2 Limiarização

A operação de limiar é um exemplo de segmentação baseada em similaridade. Segundo Akio (2006, p. 16), a operação de limiarização é aquela em que os pontos são comparados como um limiar T , que pode ser função da posição (x, y) , da

³SOUZA. Disponível em: < [http://webserver2.tecgraf.puc-rio.br/~mgattas_s/ra/trb09/Guilherme_Visao Computacional% 20-%20Trabalho%201.htm](http://webserver2.tecgraf.puc-rio.br/~mgattas_s/ra/trb09/Guilherme_Visao_Computacional%20-%20Trabalho%201.htm)>. Acesso em: 04 dez. 2015.

intensidade $I(x, y)$ ou até mesmo a propriedade $P(x, y)$ calculada para aquela posição. Aos limiares T que dependem de propriedades locais $P(x, y)$, chamamos de Locais. Caso o limiar depender também da posição (x, y) , então ele é Dinâmico. Então, estas propriedades geram uma máscara de segmentação $R(x, y)$, tal que:

Figura 6 – Critério utilizado na máscara de segmentação da limiarização

$$R(x, y) = \begin{cases} 1, & I(x, y) > T \\ 0, & I(x, y) \leq T \end{cases}$$

Fonte: Akio, 2006.

Segundo Akio (2006, p.16), quando utilizada esta regra, é definida na máscara, 1 para pixels pertencentes ao objeto e 0 para pixels pertencentes ao fundo. Um exemplo de aplicação deste algoritmo pode ser visto na imagem seguinte.

Figura 7 - Limiarização do modelo baseado em similaridades, onde os pixels brancos são parte do objeto e os pretos, parte do fundo



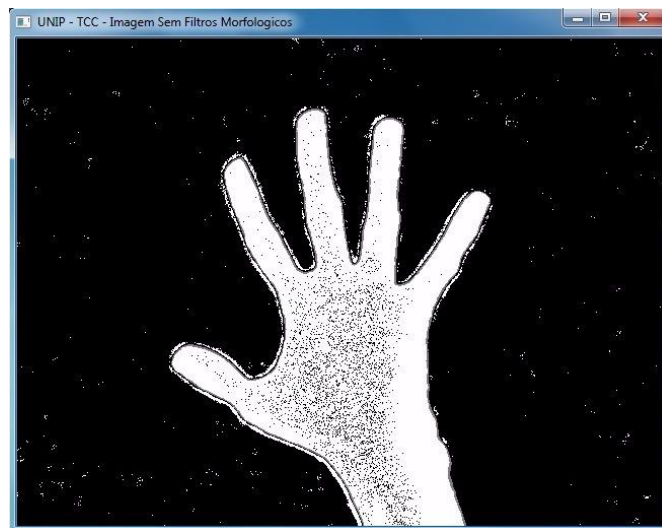
Fonte: Extraída do protótipo desenvolvido.

As condições de iluminação, sombras e variância de fontes de luz podem impactar diretamente na execução deste método. Outro problema causado pelas condições é a aparição de ruídos na imagem segmentada. Estes ruídos são gerados quando há uma grande diferenciação dos pixels da imagem média e os respectivos pixels da imagem atual, onde a diferenciação é o resultado das variações de iluminação ou do foco de luz.

3.2.3 Resultados

Utilizando a segmentação de imagem no projeto, foi obtido a extração da mão (objeto) e o fundo. Porém foram obtidos também, diversos ruídos tanto no objeto quanto no fundo, ruídos estes que podem ter sido gerados com as variâncias de iluminação, por exemplo. Na figura 8, é possível analisar que o algoritmo foi capaz de analisar o objeto em movimento e segmenta-lo, utilizando a limiarização como recurso para subtrair o objeto do fundo.

Figura 8 - Ruídos gerados na limiarização por problemas de iluminação



Fonte: Extraída do protótipo desenvolvido.

Portanto, será necessário utilizar algoritmos que tratem estes ruídos, como filtros de Dilatação e Erosão, por exemplo.

3.3 Transformações Morfológicas

Durante a subtração de fundo, foi verificado que sempre há mudanças mínimas no ambiente que causam ruídos na imagem. Tanto na cena como no objeto de interesse. Este fato levou a ser direcionado o desenvolvimento do projeto a buscar por métodos que eliminem esses pixels indesejáveis.

As transformações morfológicas são operações de processamento digital de imagens que utilizam conceitos da área de morfologia matemática. “A morfologia matemática pode ser aplicada em várias áreas de processamento e análise de

imagens, com objetivos tão distintos como realce, filtragem, segmentação, detecção de bordas, esqueletização, afinamento, dentre outras.” (MARQUES FILHO; VIEIRA NETO, 1999, p. 139). Assim sendo, quando aplicados, melhoram o aspecto da imagem.

Ainda segundo Marques Filho e Viera Neto (1999), a base da morfologia matemática é a teoria dos conjuntos e o seu princípio básico é conseguir criar funções para extrair informações relativas à geometria das regiões.

Outros autores definem esta área como um meio de construtor de operadores. “A morfologia matemática consiste em uma metodologia para análise de imagens que permite a construção de operadores úteis para a descrição de objetos em imagens.” (PEDRINI; SCHATZ, 2008, p. 327).

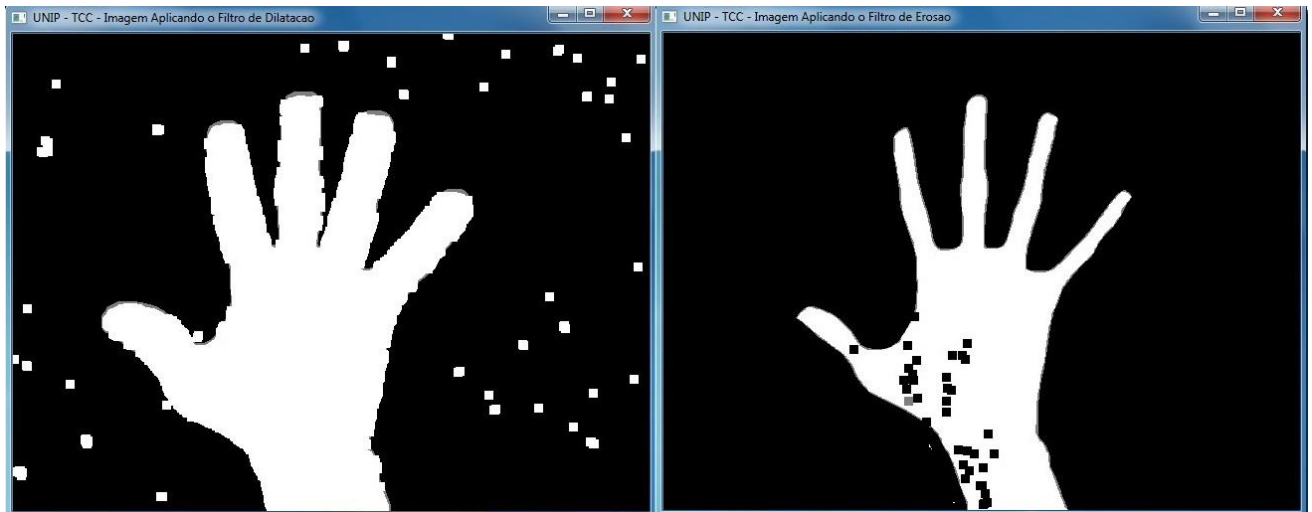
Duas operações básicas das transformações morfológicas são a erosão e a dilatação, conforme será visto na próxima sessão.

3.3.1 Erosão e Dilatação

Em termos de algoritmo, a erosão utiliza um *kernel* de pelo menos 3x3 de dimensão que, converte um pixel para preto, quando possui algum vizinho preto na imagem original enquanto que, a dilatação converte para branco quando tem algum branco na vizinhança.

Na prática, a aplicação da dilatação busca preencher o elemento na imagem por dentro e expandir suas bordas, enquanto a erosão reduz esses elementos ou até mesmo os elimina por completo, mas essas operações também podem expandir ruídos presentes no fundo e no elemento, como mostra as imagens abaixo.

Figura 9 - Esquerda operação de dilatação, direita operação de erosão com o *kernel* 9x9



Fonte: Extraída do protótipo desenvolvido.

[...] a dilatação expande uma imagem, enquanto a erosão a reduz. [...] (GONZALEZ; WOODS, 2000, p. 373).

3.3.2 Abertura e fechamento

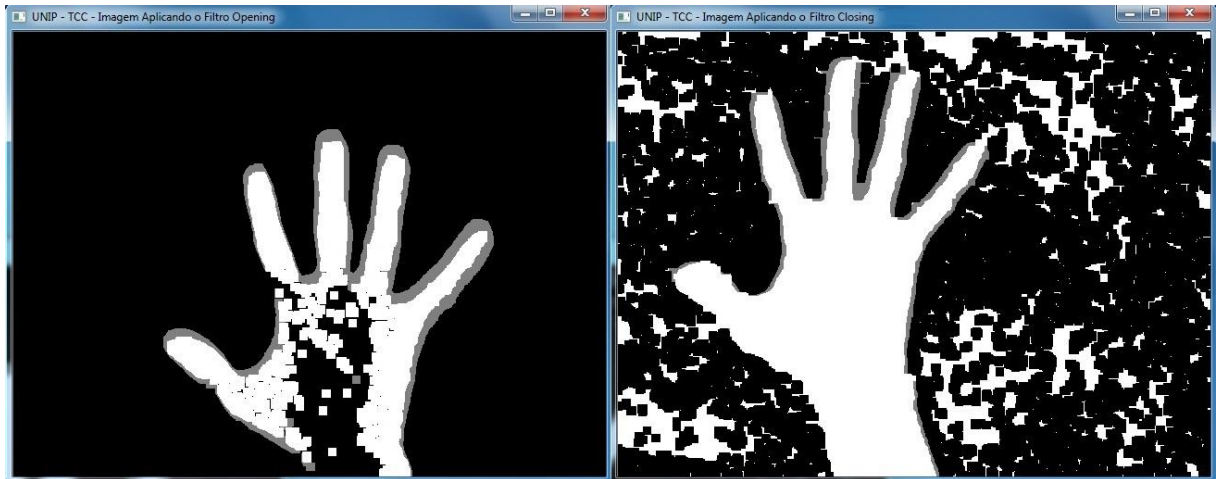
As operações básicas explicadas na sessão anterior não demonstraram um resultado satisfatório, pois tanto um como o outro nos leva a um determinado problema.

Grosso modo, a transformação morfológica de abertura é o resultado da aplicação de uma dilatação sobre uma imagem que sofreu erosão. Enquanto que a transformação de fechamento é o resultado da aplicação do método de erosão sobre uma imagem dilatada. (PEDRINI; SCHARTZ, 2008, p. 350). Na pratica, o objetivo é que a operação de abertura resulte primeiramente na eliminação de ruídos do fundo e, conseqüentemente, na redução do elemento presente na imagem. Para resolver o problema da redução do elemento na imagem, aplicamos a operação de dilatação em seguida, no qual o elemento retorna ao seu tamanho original.

No caso da operação de fechamento, começaríamos primeiramente pela dilatação, onde os ruídos presentes dentro do elemento na imagem devem ser eliminados e, conseqüentemente, o elemento expandido. A fim de resolver o mesmo problema de tamanho encontrado na operação de abertura, agora será necessária a

operação de erosão para que o elemento na imagem seja reduzido para o seu tamanho original, como mostra na imagem abaixo.

Figura 10 - Esquerda operação de abertura, direita operação de fechamento com o *kernel* 9x9



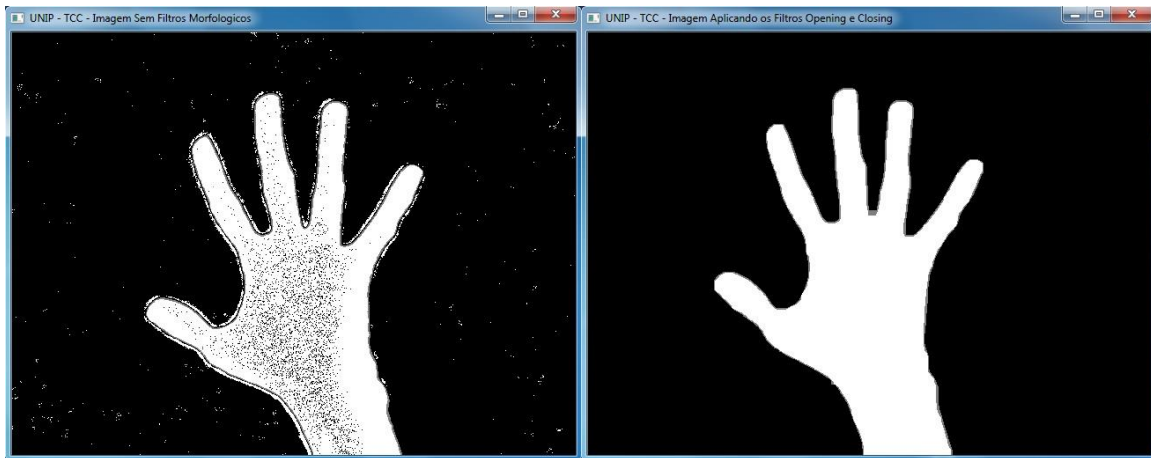
Fonte: Extraída do protótipo desenvolvido.

“A abertura em geral suaviza o contorno de uma imagem, quebra istmos estreitos e elimina proeminências delgadas. O fechamento, por sua vez, funde pequenas quebras e alarga golfos estreitos, elimina pequenos orifícios e preenche gaps no contorno.” (MARQUES FILHO; VIEIRA NETO, 1999, p. 143).

Essas operações se demonstraram bem eficientes quando queremos eliminar ruídos apenas do fundo ou apenas do elemento na imagem. Mas como podemos ver, ainda não foi satisfatório ao nosso objetivo, resolver um lado da moeda e prejudicaram gravemente o outro lado.

Analisando essas operações, seus nomes, abertura e fechamento, possuem alguma ligação, no qual se dizem ser uma sequência de ordem já definida, começando pela aplicação da operação de abertura e encerrando pela de fechamento, como mostra a imagem abaixo.

Figura 11 - Esquerda sem transformações morfológicas, direita com operações de abertura e fechamento com o *kernel* 9x9



Fonte: Extraída do protótipo desenvolvido.

Enfim, chega-se ao resultado esperado, tanto ruídos do fundo como os do elemento na imagem foram removidos com bastante precisão, resultando em uma imagem limpa de pixels desnecessários. Esta etapa foi de grande importância, pois sem ela, a aparição de muitos ruídos numa imagem prejudicaria a próxima etapa de detecção de bordas.

Encerrando a explanação deste capítulo, é salientado que tanto as operações básicas de erosão e dilatação, como as operações mais complexas de abertura e fechamento são demonstrados matematicamente e geometricamente sob demonstrações matemáticas da teoria do conjunto que não explanadas neste trabalho.

3.4 Detecção de borda

A imagem digital, objeto de estudo da visão computacional, contém regiões que se pretende extrair informações. Estas regiões são delimitadas por bordas que, quando observados em uma matriz de *pixels*, fazem parte de um conjunto de pontos, ligados um ao outro, formando uma linha única.

O tema 'detecção de bordas' (edge detection) vem desafiando os pesquisadores da área de Processamento de Imagens há muitos anos e sobre ele continuam sendo experimentadas novas técnicas, cujos resultados são publicados ainda hoje nos mais conceituados periódicos científicos mundiais. Trata-se, portanto, de um tema em aberto, a detecção de bordas em cena consideradas 'difíceis'. (MARQUES FILHO; VIEIRA NETO, 1999, p. 37).

Segundo Gonzalez e Woods (2002), uma das maneiras mais comum de se detectar bordas numa imagem é com a utilização de um *kernel* e operadores de derivação, percorrendo toda matriz, a fim de encontrar faixas claras em um fundo escuro, ou até mesmo o inverso.

Sendo assim, a detecção de bordas engloba um conjunto de técnicas com o objetivo de servir a etapa de segmentação de objetos de uma imagem, separando os objetos de interesse de uma imagem.

3.4.1 Laplaciano

É um filtro isotrópico que tem a característica de usar derivadas de segunda ordem. Sua característica mais marcante é a de trazer uma imagem de saída mais acentuada a detalhes, como exemplo, pontos isolados. “O Laplaciano de uma imagem é um operador escalar dado pela soma de suas segundas derivadas parciais com relação a x e a y .” (CAMPOS, 1998, P. 36).

Dado como exemplo uma máscara 3x3, pode haver duas possibilidades para o *pixel* central, sendo normalmente a cor preta como valor negativo ou o branco como positivo. O primeiro remove as bordas exteriores aos pixels onde a máscara é aplicada e, o positivo, remove as bordas interiores ou adjacentes ao *pixel* central.

Segundo Campos (1998), ele possui bom funcionamento quando a imagem a ser analisada não possui muitos ruídos, porém, se isso ocorrer, é necessário utilizar outro filtro de suavização, no caso, a função Gaussiana. Daí surge outro método de detecção de bordas, chamado Laplaciano de Gaussiana. “A sua implementação é obtida através da convolução da imagem com uma Gaussiana bidimensional de desvio padrão σ [...] O desvio padrão σ da Gaussiana indicará o grau de suavização da imagem, ou a escala a ser considerada”. (CAMPOS, 1998, p.38).

3.4.2 Canny

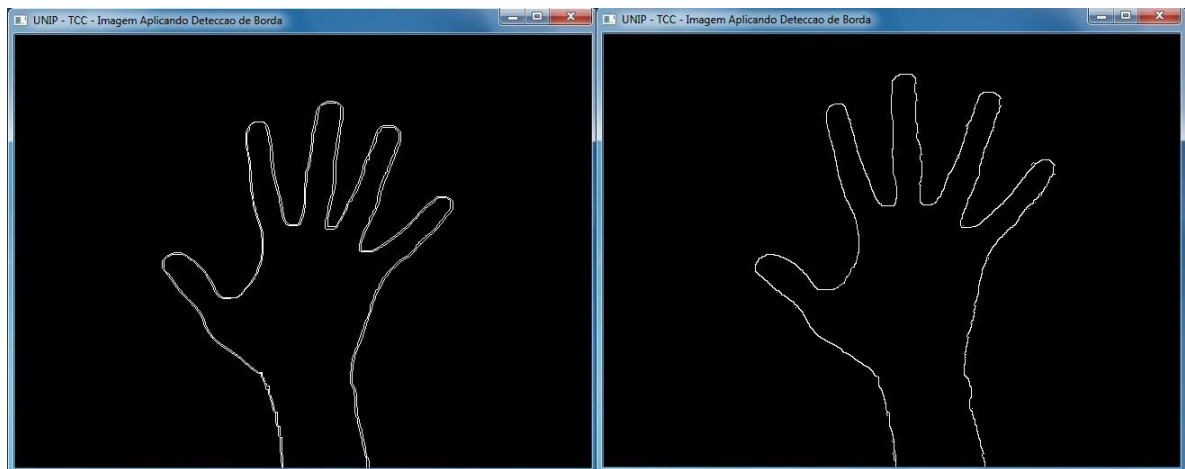
Este método foi desenvolvido em 1986 e, seu algoritmo possui duas etapas de processamento. Na primeira etapa, é aplicado um filtro Gaussiano, no qual lhe da com derivadas de primeira ordem. Quanto à segunda etapa, calcula-se a intensidade da borda e, sua direção em cada *pixel* adjacente do *pixel* central da máscara.

Além das etapas, possui 3 critérios de otimização para detectar os bordas.

O primeiro critério descrito é a baixa taxa de erro, onde bordas reais não podem ser descartadas e falsas bordas não podem ser encontradas. O segundo critério se refere à distância entre o centro da borda e os pontos ao seu redor, o algoritmo foca a minimização destas distâncias. O terceiro critério foi desenvolvido, pois os dois primeiros ainda não conseguiam eliminar a possibilidade de múltiplas respostas para uma mesma borda. Assim o terceiro critério estabelece que o algoritmo deve obter apenas uma resposta para uma única aresta. (COSTA, 2012, p. 35).

3.4.3 Comparativo

Figura 12 - Esquerda laplace, direita canny



Fonte: Extraída do protótipo desenvolvido.

A detecção de bordas pelo Laplace demonstrou que, a iluminação refletindo por trás do elemento na imagem, resultou em 2 contornos próximos. O que não é o caso do *Canny*, bem pelo fato de ser conhecido como o detector ideal de bordas que, devido aos seus 3 critérios, se sobressai de forma excelente. Além disso, de acordo com os testes de tempo de processo realizados, o *Canny* conseguiu se sobressair 25% mais rápido que o Laplace.

3.5 Detecção de contorno

Nesta etapa do processamento, é utilizado o contorno da imagem gerado pelo detector de bordas para representa-lo em uma estrutura de dados, a fim de possibilitar a extração de características do objeto de interesse.

As imagens mudam de ângulo e tamanho e, é necessária uma forma de representação relativa, sem depender do formato clássico do plano cartesiano.

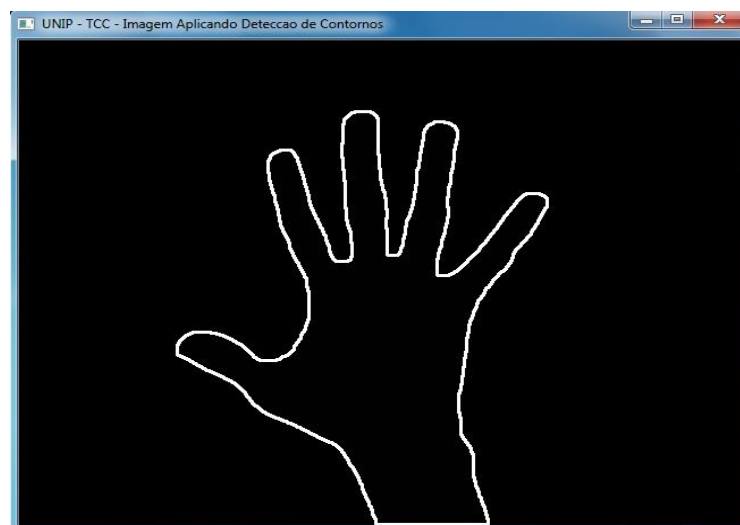
3.5.1 Lista de códigos

Freeman (1961, 1974) propôs um esquema conhecido como código da cadeia para representar a borda de um objeto. Ao invés de armazenar as coordenadas absolutas dos pixels, a representação pelo código da cadeia utiliza a posição relativa entre pixels consecutivos da borda. (PEDRINI; SCHATZ, 2008, p. 250).

Dado uma imagem que passou pela detecção de bordas, a lista de códigos atuará percorrendo pixel a pixel a busca de um ponto determinado como borda. Quando este for localizado, ela então iniciará o processo de armazenamento, no qual armazena o pixel atual, determina seu pixel antecessor como já visitado e, o posterior como pretensão de ser o próximo a visitar.

Nessa lista é absorvida uma sequência de números que representa a borda da mão. Esta representação é armazenada numa estrutura de dados chamada de lista de códigos. Os códigos armazenados são referentes aos contornos das pontas dos dedos, início da mão e base dos dedos, como mostra a imagem abaixo.

Figura 13 – Detecção de contornos



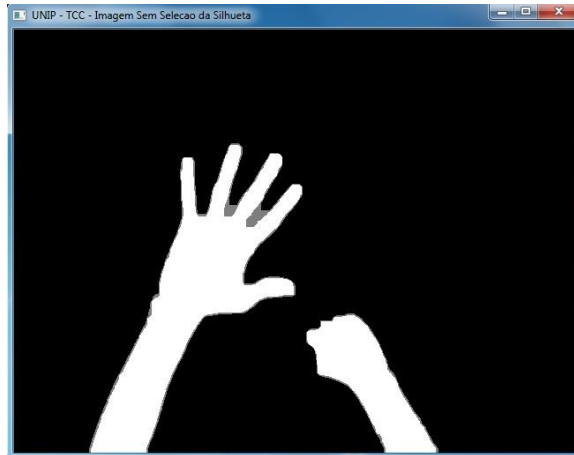
Fonte: Extraída do protótipo desenvolvido.

3.6 Seleção da silhueta da mão

Durante a etapa de detecção de contorno, foi verificado que o algoritmo varre a matriz inteira contornando bordas normalmente. Porém, ao ser analisado mais de um elemento (seja um ruído ou um objeto) em cena simultaneamente, não deve ser levado em consideração o segundo. Pois, o objetivo é extrair informação de apenas

um elemento que entrou em cena, conforme a figura de exemplo extraída.

Figura 14 - Problema de mais de 1 elemento em cena

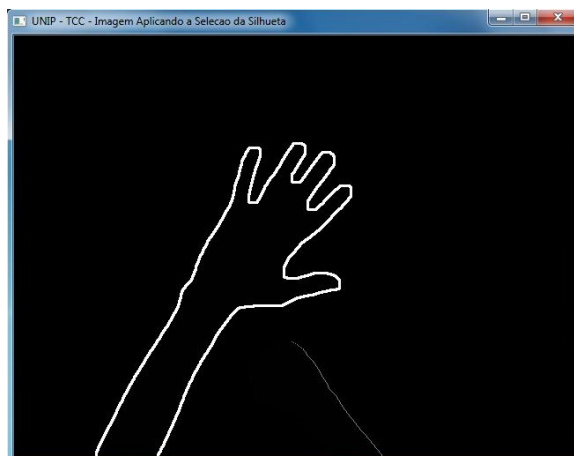


Fonte: Extraída do protótipo desenvolvido.

A fim de contornar este problema, foi aplicada a busca pelo maior contorno em cena, onde primeiramente é determinada a menor área possível que se identifique mais com a área de uma mão. Em seguida, é percorrido cada contorno já armazenado previamente pela lista de códigos e, caso encontre um contorno de área maior que o mínimo determinado, este é referenciado como o maior contorno até então. Para os demais contornos, continuarão a ser verificados, porém comparados ao novo maior contorno que foi determinado.

Ao final deste algoritmo, a silhueta da mão é selecionada e, apenas esta é levada adiante para as próximas etapas, como mostra a imagem abaixo.

Figura 15 - Resultado do algoritmo de seleção da silhueta da mão



Fonte: Extraída do protótipo desenvolvido.

3.7 Simplificação da silhueta da mão

Na sessão anterior, foi realizado procedimento para selecionar a silhueta da mão, mas ainda, há muitos pontos contidos em nossa lista de códigos. É preciso deixar mais simplificada com o objetivo de extrair as características com mais facilidade.

Existem métodos que possibilitam a simplificação de poligonais, cujo qual é formado por vértices que, possuem ligações entre pontos de coordenadas distintas. O método de Douglas-Peucker permite eliminar os vértices desnecessários, onde sua eliminação não degradará a informação que se deseja extrair.

3.7.1 Simplificação de poligonais com o método Douglas-Peucker

Segundo Davis (2000), eliminar os vértices desnecessários visa conquistar dois objetivos. O primeiro é que será necessário menos espaço de armazenamento para a estrutura de dados e, métodos de busca podem atuar mais rapidamente. “Foi proposto em 1973, e é até hoje reconhecidamente o melhor em termos de preservação das características da poligonal original, especialmente se utilizado com tolerâncias pequenas”. (DAVIS, 2000)⁴. Ainda sobre seu funcionamento, o autor afirma.

O algoritmo é recursivo, e a cada passo processa o intervalo de pontos contido entre um vértice inicial (chamado de âncora) e um vértice final (denominado flutuante). É estabelecido um corredor de largura igual ao dobro da tolerância, formando duas faixas paralelas ao segmento entre a âncora e o flutuante. (DAVIS, 2000)⁵.

O funcionamento do algoritmo se passa em um conjunto de pontos, que é o caso da lista de códigos. Os pontos que sempre permanecerão na nova lista são o primeiro, o último, e os decorrentes do algoritmo que se identificam como pontos de extremidade.

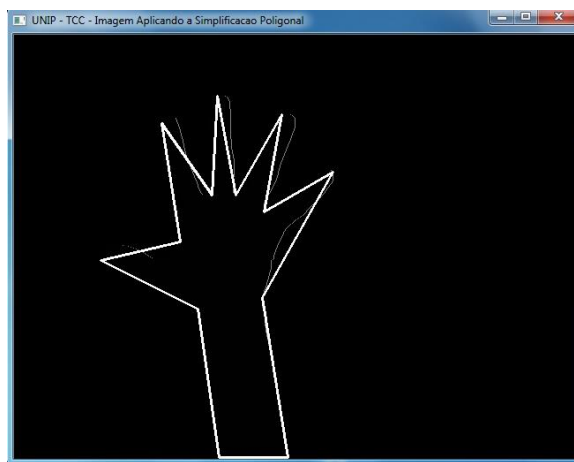
O primeiro passo acontece entre o primeiro e último ponto, identificando o ponto mais distante entre ambos. O segundo passo é dado pelo ponto identificado e

^{4, 5} (DAVIS, Clodoveu, 2000). Disponível em: <<http://mundogeo.com/blog/2000/06/01/simplificacao-de-poligonais/>>. Acesso em: 06 dez. 2015.

o primeiro ponto que, identifica o ponto mais distante entre ambos e, continua percorrendo todos os pontos entre eles, identificando os pontos mais distantes entre 3 ou mais vértices em que vão sendo encontrados. O terceiro passo tem o mesmo procedimento do segundo, porém ele se inicia pelo primeiro ponto identificado e o último.

Ao final deste algoritmo, tem-se uma nova lista com os pontos de extremidade que são de interesse para a próxima etapa, como vemos na imagem abaixo.

Figura 16 - Resultado do algoritmo de simplificação de poligonais



Fonte: Extraída do protótipo desenvolvido.

3.8 Detecção das características da mão

Segundo Conci, Azevedo e Leta (2008, p. 56) a extração de características é aplicada em função de imagens previamente segmentadas. Nesta fase, temos os dados binários da imagem e também informações relativas à imagem original, como luminosidade ou textura.

De acordo com o domínio do problema, devem-se verificar os atributos relevantes de cada objeto segmentado, porém queremos lhe dar apenas com um objeto.

Segundo os autores citados acima, algumas técnicas de extração de características vão analisar os dados binários. Como exemplo é citado à extração de características como dimensão e geometria dos objetos segmentados.

Em alguns casos, a extração de características é executada diretamente na imagem original. É o caso da extração de informações de luminosidade e textura dos

objetos.

Após a aplicação de técnicas nesta etapa serem concluída, o sistema de visão computacional alimenta a próxima etapa, que é a de classificação e reconhecimento dos objetos.

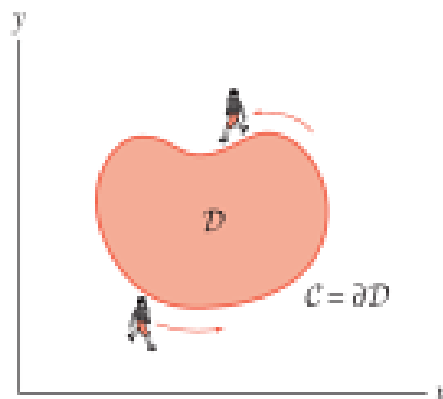
3.8.1 Cálculo do centro da área

A primeira característica que devemos nos preocupar em relação à mão, é o centro de sua área. No qual se pretende obter informações de angulação a partir do seu centro, como veremos mais a frente.

Quando falamos da área de uma curva, se lembramos da integral, e como temos um contorno que se prediz poder ser formado por 4 curvas, no qual se aplicam a uma curva fechada, recorremos ao Teorema de Green.

Antes de enunciar o Teorema de Green, Rogawski (2008) explica o que a curva fechada simples é uma curva que não se intersecta, conforme observado na imagem abaixo.

Figura 17 – A fronteira do domínio D é uma curva fechada simples C



Fonte: Rogawski, 2008.

Seja D um domínio cuja fronteira ∂D é uma curva fechada com orientação de fronteira. Se $P(x,y)$ e $Q(x,y)$ são diferenciáveis e têm derivadas parciais de primeira ordem contínuas, então:

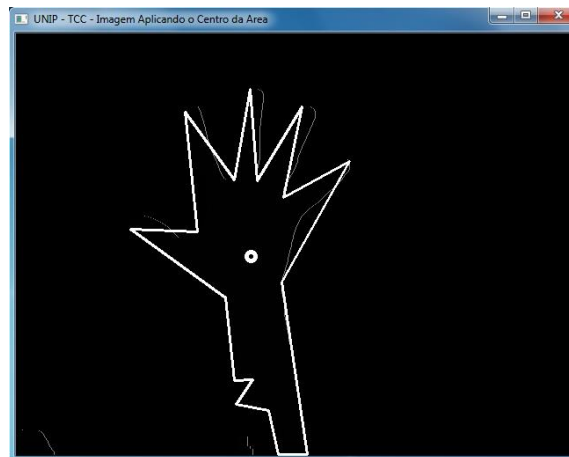
Figura 18 – Teorema de Green

$$\oint_{\partial D} P dx + Q dy = \iint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA$$

Fonte: Rogawski, 2008.

Ao calcular a diferença entre a curva à esquerda e a da direita, obtemos a coordenada x do centro da área da curva fechada. A coordenada central do eixo y se da pela diferença entre a curva a cima e a baixo. Ao final deste cálculo é obtida a coordenada do ponto central da área da mão, como mostra a imagem abaixo.

Figura 19 - Resultado do cálculo do centro da área pelo Teorema de Green



Fonte: Extraída do protótipo desenvolvido.

3.8.2 Envoltório convexo

Para muitos projetos de visão computacional, é pertinente armazenar estruturas de dados de características geométricas de regiões de uma figura. Na sessão da simplificação da silhueta da mão, foi possível reduzir o número de pontos irrelevantes e, a partir deste procedimento, são buscadas partes convexas do contorno.

Segundo Pradi (2012 apud QUEIROZ, 2005, p. 61). “O envoltório convexo de um conjunto de pontos S é o menor polígono convexo P que contém S. Esse envoltório fornece informações sobre o formato de um conjunto de pontos, servindo como uma primeira aproximação de um objeto mais complexo”.

Dado um conjunto de pontos da lista de código que foi previamente

simplificada, o algoritmo de Graham tem como objetivo armazenar em uma nova lista, apenas os pontos que formam um polígono convexo, no qual os pontos que estão na parte mais interna do contorno, sejam descartados.

Seu funcionamento é dado inicialmente por encontrar a menor coordenada do eixo y, mas caso haja outros pontos situados na mesma posição y, então deve ser considerado o com menor coordenada x. A partir do ponto inicial encontrado, diversos algoritmos de ordenação podem ser utilizados, levando em consideração seu grau de complexidade em relação à quantidade de pontos que devem ser computados. Ao escolher um algoritmo de ordenação, é calculada ponto a ponto, a tangente do ângulo em relação ao ponto inicial, para então ordena-los de forma crescente.

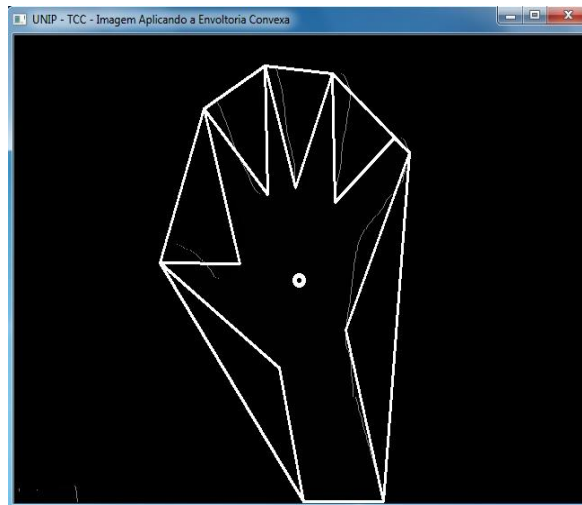
Em seguida, deve-se utilizar o conceito pilha para gerar a nova lista de pontos convexos. Os 2 pontos situados no início da lista são empilhados, sendo o primeiro sempre o ponto previamente encontrado. Partindo do ponto subsequente destes, é percorrida a lista até último ponto, onde o autor (FREITAS) considera os seguintes critérios:

Se o ponto X em questão está a esquerda do vetor penúltimo-último elemento da pilha (último elemento da pilha é o topo) então empilhamos e olhamos o próximo;
Se o ponto X em questão está à direita deste mesmo vetor, então desempilhamos e voltamos a analisar X. (FREITAS).⁶

Ao final obtemos um contorno convexo situado ao redor da silhueta, como mostra a imagem abaixo.

^{5,6} FREITAS. Disponível em: <<http://www.ime.usp.br/~freitas/gc/fecho.html>>. Acesso em: 03 dez. 2015.

Figura 20 - Resultado do algoritmo de Graham



Fonte: Extraída do protótipo desenvolvido.

3.8.3 Defeitos de convexidade

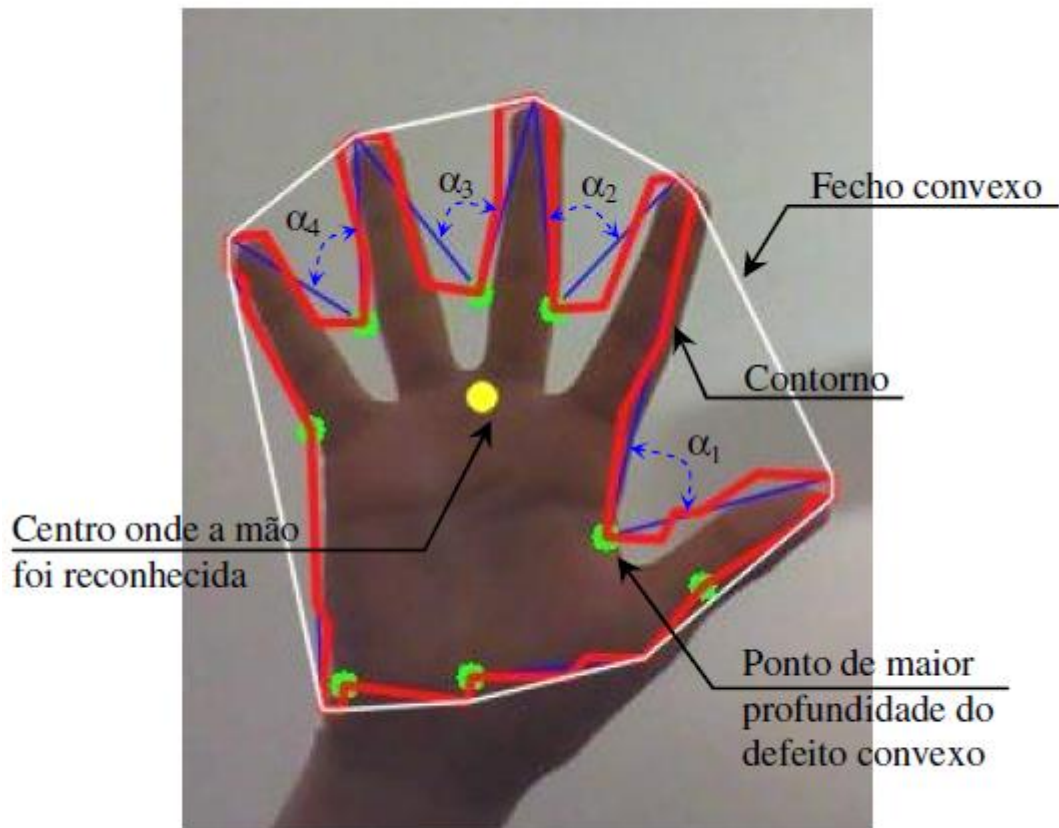
Uma forma de se identificar os dedos da mão é utilizando funções baseadas nos defeitos de convexidade. Conforme apresentado em seções anteriores, foi obtido a partir do objeto (mão), o menor contorno possível entre as extremidades.

Segundo Masutani et. al, o envoltório convexo é o menor polígono convexo que contém os pontos do objeto. O autor ainda define o os defeitos convexos como uma sequência de pontos entre dois vértices do envoltório convexo.

Portanto, o algoritmo é responsável por identificar a profundidade entre as curvaturas internas mais distantes e as extremidades geradas pelo algoritmo de Graham. Para cada uma destas curvaturas, é atribuído um vértice cujo qual constitui o defeito de convexidade.

Na imagem 21, é possível identificar os pontos distantes pelos vértices verdes, estes então serão utilizados para reconhecer os dedos.

Figura 21 – Contorno fecho e defeito convexo

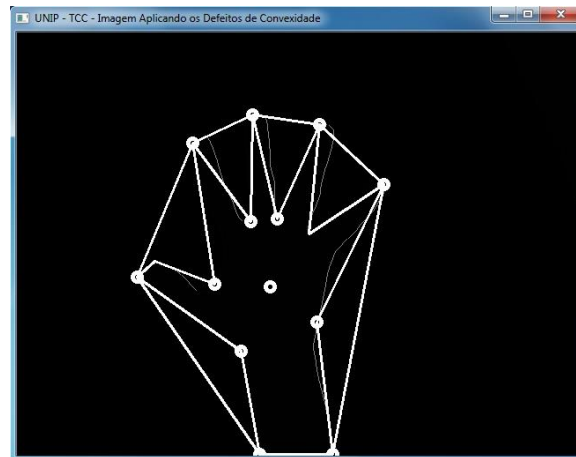


Fonte: Masutani et . al.

Gonçalves (pg.5) descreve que em alguns casos, variações sutis na topologia do contorno faz com que existam mais defeitos de convexidade do que o esperado. Para solucionar este problema, deve ser incluído um filtro que ignora as regiões de pequena profundidade e mantém as maiores, correspondendo aos dedos da mão.

Com a implementação deste algoritmo no protótipo, foi possível concluir que o sistema conseguia interpretar estes defeitos de convexidade, conforme segue representado pela imagem 22.

Figura 22 - Resultado dos defeitos de convexidade



Fonte: Extraída do protótipo desenvolvido.

Com a identificação dos dedos e da profundidade que estes possuem, é possível avançar mais um passo no processo de reconhecimento de imagens.

3.9 Classificação das características da mão

Na aplicação do último método para extração de características da mão, o sistema de visão computacional conseguiu uma lista repleta de pontos. Porém, isso, ainda não é considerado como uma informação precisa para o reconhecimento.

Nesta questão, é necessário classificar as características, para verificar se há sentido nestas.

3.9.1 Ponta dos dedos

A classificação é aplicada através de padrões definidos pelas características da mão, como as pontas dos dedos, por exemplo. Essas características poderiam então de alguma forma representar alguma informação.

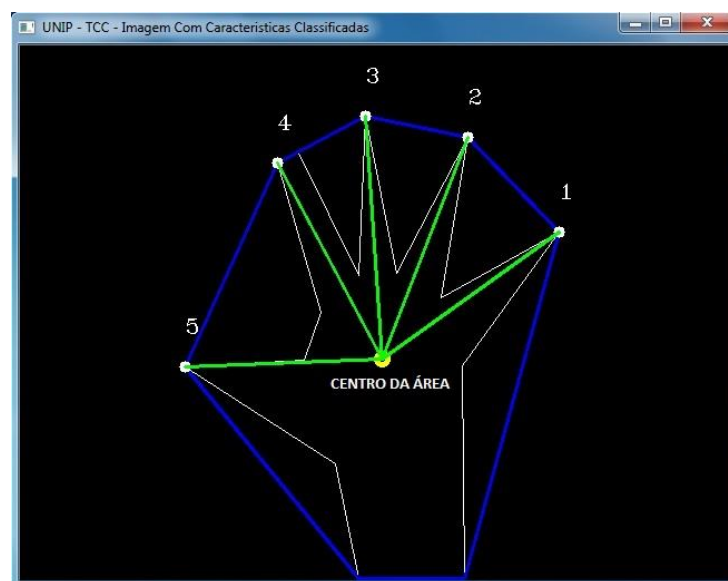
Gonzalez e Woods (2002, p. 409) define padrão como: “[...] um padrão é uma descrição quantitativa ou estrutural de um objeto ou alguma outra entidade de interesse em uma imagem [...]. Em geral, um padrão é formado por um ou mais descritores”.

Para o correto processamento em fases subsequentes, se é necessário verificar as características das pontas dos dedos. As características relevantes que

devem ser verificadas são duas. A primeira é se há um ponto muito próximo, fazendo o algoritmo interpretar ou não que seja um dedo. Outra característica é verificar algum ângulo fora do comum que descarta a possibilidade de ser uma imagem capturada de uma mão.

Caso o algoritmo consiga validar as duas características, têm-se informações relevantes sobre as cinco possíveis pontas dos dedos, bem como o centro da área, obtido anteriormente conforme imagem abaixo.

Figura 23 - Resultado da classificação de características da mão



Fonte: Extraída do protótipo desenvolvido.

3.10 Reconhecimento

A etapa de classificação de características possibilitou uma ampla visão do que é possível reconhecer, seja ele um número, um simples gesto ou até mesmo de um alfabeto digital de libras.

Existem 2 tipos de reconhecimentos: o estático e o dinâmico. O primeiro é o mais simples, pois envolve apenas uma imagem para cada gesto rotulado. No caso do dinâmico, deve-se levar em consideração referências de cada posição do gesto para cada imagem, a fim de rotular um único gesto.

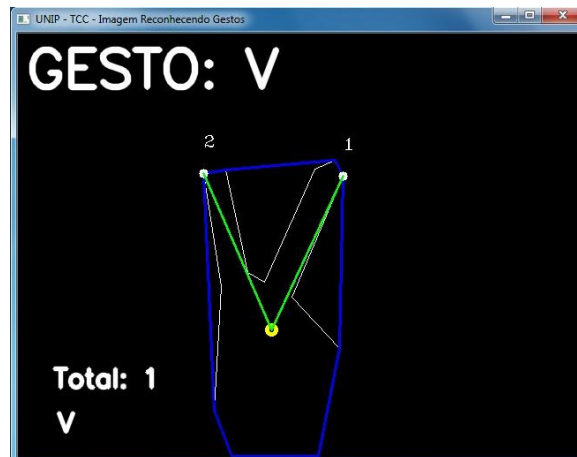
Este trabalho apresenta o reconhecimento de um gesto estático do alfabeto digital de libras e, portanto, será explicado a seguir seu procedimento final.

3.10.1 Gestos do alfabeto digital de libras

Através da classificação de características, foi possível reconhecer todos os dedos da mão, então seguindo neste contexto, podemos de início aumentar a possibilidade de reconhecer um gesto. Para isso é necessário apenas que verifique a quantidade de dedos que o gesto apresenta.

Após diminuir a quantidade de gestos possíveis, as características entre o ponto do centro e das pontas dos dedos que se deve considerar para um determinado gesto é, em relação ao ângulo, à distância, e as coordenadas do eixo x e y, como mostra a imagem a baixo.

Figura 24 - Resultado do reconhecimento de um gesto



Fonte: Extraída do protótipo desenvolvido.

4 COMPARATIVO DE RESULTADOS

Neste capítulo é apresentado o comparativo de resultados verificados na execução dos métodos, utilizando um protótipo desenvolvido na linguagem *Python*, utilizando o framework⁷ *OpenCV*.

A biblioteca *OpenCV®*, desenvolvida pela Intel em 2000, permite manipulação de dados de imagens, manipulação de matrizes e vetores, desenvolvimento de rotinas de álgebra linear, estrutura de dados dinâmicas, desenvolvimento de algoritmos de processamento de imagem, análise estrutural, calibração de câmera, análise de movimento (*tracking*), reconhecimento de objetos, GUI básica e rotulagem de imagem. (Barboza, 2009, p. 2)

A opção por escolher uma biblioteca especializada em visão computacional a desenvolver as funções do zero é devido ao fato que este trabalho é focado no estudo e aplicação de técnicas e conceitos de visão computacional. Além disso, a implementação do sistema está alinhada a boas práticas de desenvolvimento de software. “Engenharia de software orientada a reuso tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvido e, assim, reduzir os custos e riscos. Geralmente também proporciona a entrega mais rápida do software.” (SOMMERVILLE, 2011, p. 23).

4.1 Ambiente de *hardware* e sistema operacional

Foi utilizado para testes um computador com 4 Gb de RAM, Intel® Core(TM)2 Quad -QM CPU 2.50 GHz com sistema operacional Windows 7 Home Premium de 64 bits.

Figura 25 – Ambiente de hardware e sistema operacional utilizado



Fonte: Imagem extraída da configuração do computador.

⁷ *Framework* de aplicações é definido por Sommerville (2011) como uma estrutura genérica estendida para se criar uma aplicação ou sub-sistema mais específico.

4.2 Equipamento para aquisição de imagens

Foi utilizado uma câmera que capta a luz visível, no caso, uma câmera ligada ao computador, da marca *LeaderShip*, modelo 3526, *Webcam Night Vision*.

Figura 26 – Objeto de captura da imagem



Fonte: <http://www.leadershipe.com.br/Webcam-Night-Vision.html>.

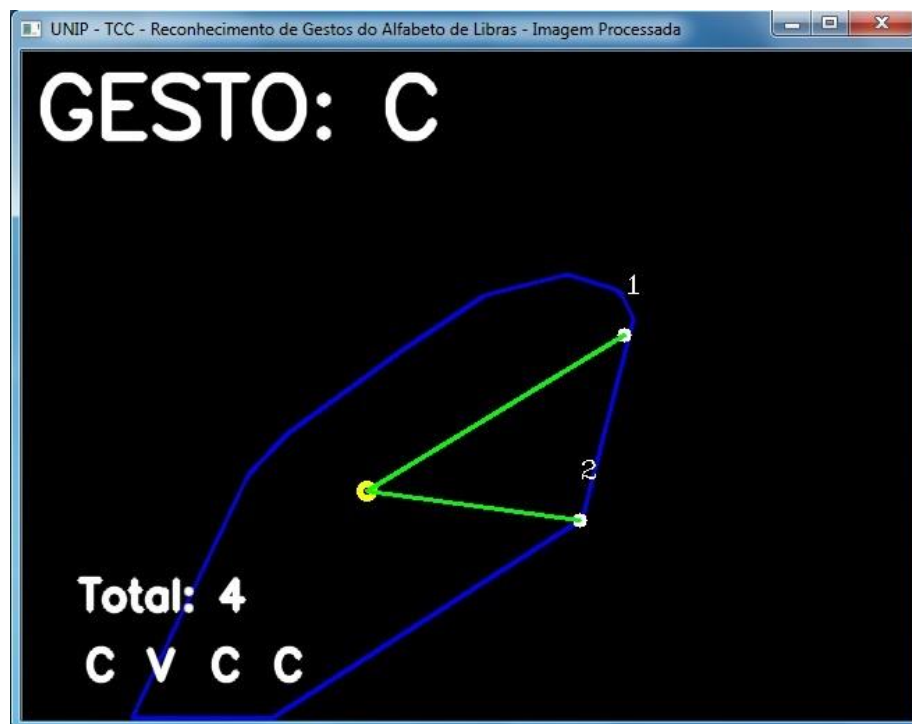
4.3 Cenário controlado

Foram feitos testes em um cenário controlado. O controlado é definido neste trabalho como um ambiente de iluminação artificial constante, sem variações, capturando imagens em um fundo de cor sólida, no caso, cor branca.

4.4 Gestos de teste

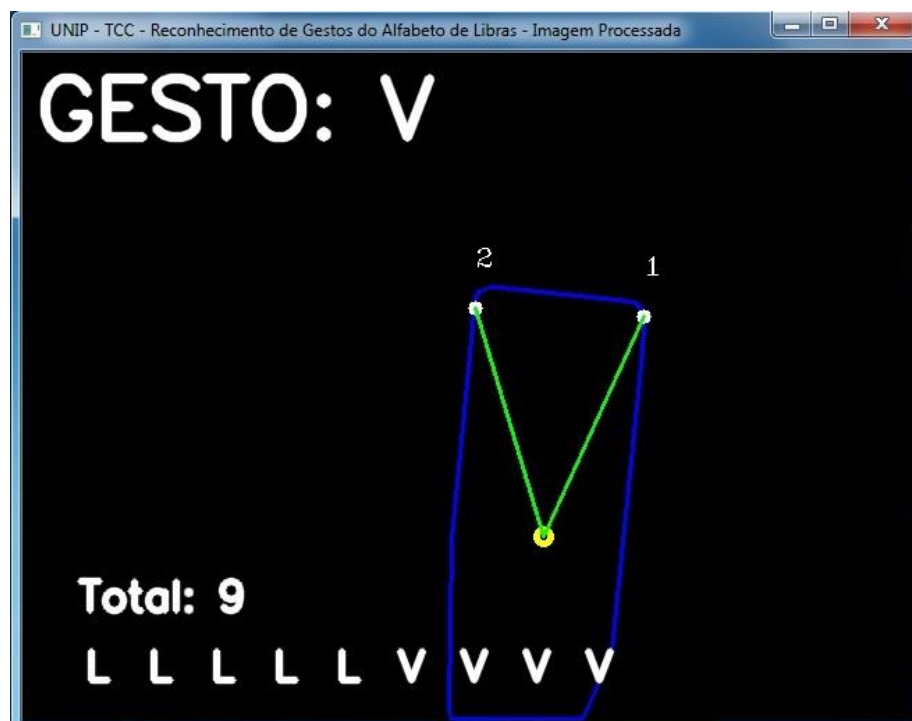
Para testes, foram utilizados as letras L, U, C e V do alfabeto digital de Libras. O algoritmo do programa desenvolvido identifica a letra através de algumas características como, o ângulo formado entre os dedos detectados e o centro da área da mão.

Figura 27 - Gesto reconhecido C



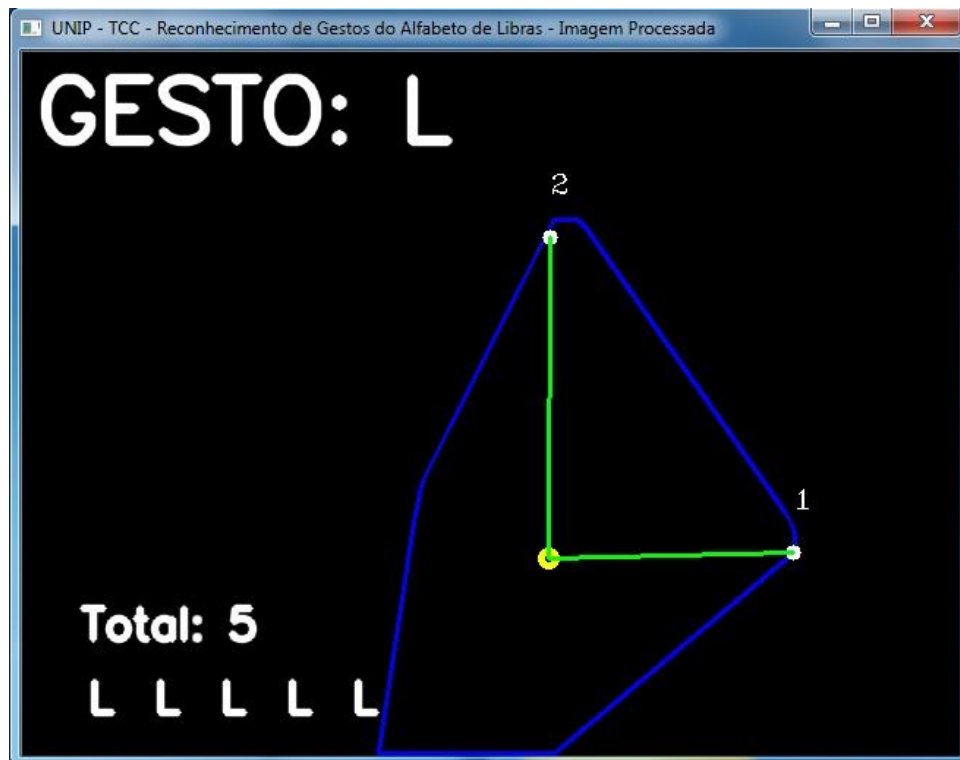
Fonte: Extraída do protótipo desenvolvido.

Figura 28 - Gesto reconhecido V



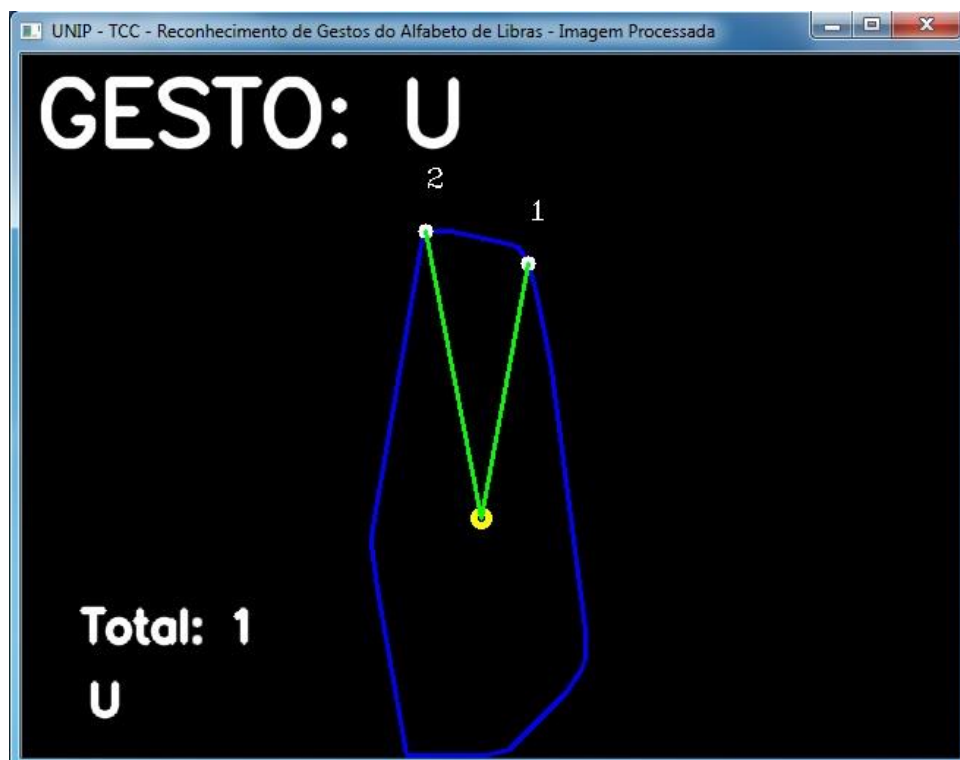
Fonte: Extraída do protótipo desenvolvido.

Figura 29 - Gesto reconhecido L



Fonte: Extraída do protótipo desenvolvido.

Figura 30 - Gesto reconhecido U



Fonte: Extraída do protótipo desenvolvido.

4.5 Resultados finais sem filtro de suavização

Tabela 2 – Resultados sem filtro de suavização

Gesto	Total a reconhecer	Tempo processo por gesto	Erros	Porcentagem Acerto
L	10	45 ms	0	100 %
U	10	59 ms	2	80 %
V	10	48 ms	0	100 %
C	10	103 ms	2	80%

Fonte: Elaborada pelos autores.

4.6 Resultados finais sem filtros morfológicos

Tabela 3 – Resultados sem filtros morfológicos

Gesto	Total a reconhecer	Tempo processo por gesto	Erros	Porcentagem Acerto
L	10	70 ms	1	90 %
U	10	101 ms	5	50 %
V	10	52 ms	0	100 %
C	10	125 ms	7	30"%

Fonte: Elaborada pelos autores.

4.7 Resultados finais com aplicação de filtros de suavização e morfológicos

Foram aplicados os filtros de suavização de Gauss, bem como os filtros morfológicos de abertura e fechamento. Todos apresentados no capítulo anterior.

Tabela 4 – Resultados com filtros de suavização e morfológicos

Gesto	Total a reconhecer	Tempo processo por gesto	Erros	Porcentagem Acerto
L	10	57 ms	0	100 %
U	10	67 ms	0	100 %
V	10	53 ms	0	100 %
C	10	56 ms	0	100"%

Fonte: Elaborada pelos autores.

4.8 Conclusão sobre a análise dos resultados

Os gestos L e V são os que possuem os melhores resultados de acerto e tempo de processamento, independente da aplicação de filtros.

Quando há gestos mais complexos, como o U e o C, a qualidade do resultado depende da aplicação de filtros de suavização e morfológicos.

Assim, conclui-se que para reconhecimento de gestos em Libras, feito sob as condições acima mencionadas, é imprescindível o uso de técnicas de pré-processamento, a fim de realçar a imagem para as etapas subsequentes.

5 CONCLUSÃO

Neste trabalho foi apresentado uma metodologia para reconhecer gestos do alfabeto digital de Libras. Durante os estudos e o desenvolvimento do protótipo, foi possível comprovar a hipótese de que é possível um dispositivo computacional atuar como um intérprete entre o alfabeto digital da linguagem gestual e a Língua Portuguesa.

Estudando diversos conceitos, técnicas e algoritmos, foi possível capturar uma informação armazenada em um vídeo. Sendo este capturado em tempo real. Analogicamente a uma linha de produção de uma indústria, a matéria-prima, o vídeo, foi passado por várias fases de beneficiamento e, no final, trouxe o produto final, que é a informação. Esta que é representada por um gesto do alfabeto digital.

Foi constatado que um sistema desta natureza deve ter algoritmos preparados para tratar padrões de informações e não a informação em si. Como exemplo, citamos a pesquisa neste texto.

Um sinal do alfabeto digital é executado por diferentes pessoas. Sendo que, cada uma possui características biológicas específicas. Além disso, os gestos podem ser executados em diferentes ângulos. Cabe ao sistema baseado em visão a responsabilidade de aplicar corretamente técnicas de processamento digital de imagens com o intuito de obter a informação de qual símbolo representa na Língua Portuguesa.

A grande conclusão deste trabalho foi confirmar que a visão computacional é uma área promissora, fazendo com que sistemas atuem com um periférico de entrada de dados fora do eixo do teclado e mouse, que é alguma câmera sensível a algum espectro de luz. É possível criar soluções específicas para problemas de diversas naturezas, capturando imagens ou vídeos e aplicando um conjunto de passos para conseguir extrair uma informação.

O alfabeto digital de Libras possui gestos que são formatos por gestos estáticos e dinâmicos. Neste trabalho comprovamos a hipótese reconhecendo apenas as letras V, U, L e C. Todas elas de natureza estática, aplicamos apenas as características do ângulo formado entre as pontas dos dedos e o centro da área da mão.

Porém, este trabalho pode ser evoluído como aplicação de novas características a fim de reconhecer outros sinais do alfabeto, principalmente gestos

dinâmicos.

Para reconhecer gestos dinâmicos a componente temporal deve ser levada em consideração. Gestos dinâmicos podem envolver várias posturas, trajetórias e velocidade ao longo do tempo. Considerando que podem ocorrer erros e imperfeições na segmentação, estes sistemas de reconhecimento devem utilizar modelos estatísticos que absorvam estes erros e levem em consideração o desenvolvimento do gesto ao longo do tempo. (ANJO, 2013, p. 77).

5.1 Trabalhos Futuros

Conforme salientado em capítulos anteriores, a Visão Computacional é uma área multidisciplinar devido ao fato de recorrer à inteligência artificial para criação de muitas soluções.

Gonzalez e Woods (2000) diz que a inteligência artificial, através de conceitos de redes neurais pode atuar na criação de algoritmos que auxiliam em limiarizações binárias de regiões de imagens digitais. Através de análise de amostras de padrões é possível um algoritmo em outro momento tomar decisões. “O processo pelo qual um padrão de treinamento é usado na obtenção das funções de decisão é chamado de aprendizado ou treinamento.” (GONZALEZ; WOODS, 2000, p. 424).

Assim sendo, é salientada a sugestão para Trabalhos Futuros à criação de uma tratativa diferenciada na limiarização de objetos utilizando conceitos de redes neurais. E é esperado que este texto sirva como inspiração no desenvolvimento de outras soluções em visão computacional, bem como, no aprimoramento e criação de novos algoritmos de processamento e análise de imagens digitais.

REFERÊNCIAS

ANJO, Mauro Dos Santos. **Avaliação das técnicas de segmentação, modelagem e classificação para o reconhecimento automático de gestos e proposta de uma solução para classificar gesto da libras em tempo real**. São Paulo, 2012. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli_10001999.pdf>. Acesso em: 21 de dez. 2015.

AZEVEDO, Eduardo; CONCI, Aura. **COMPUTAÇÃO GRÁFICA: Geração de imagens**. 11. ed. Rio de Janeiro: Editora Campus, 2003.

BARBOZA, Daniel Ponciano dos Santos. **Estudo da Biblioteca OpenCV**. Rio de Janeiro, 2009. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli_10001999.pdf>. Acesso em: 21 de set. 2015.

BEZERRA, Miguel. **MÉTODOS BASEADOS NA REGRA DO VIZINHO MAIS PRÓXIMO PARA RECONHECIMENTO DE IMAGENS**. Disponível em: <<http://tcc.ecomp.poli.br/20052/MiguelEugenio.pdf>>. Acesso em: 20 de set. 2015.

BRASIL. **Decreto nº 5.626, de 22 de dezembro de 2005**. Regulamenta a Lei nº 10.436, de 24 de abril de 2002, que dispõe sobre a Língua Brasileira de Sinais – Libras e o artigo da Lei nº 10.098, de 19 de dezembro de 2000. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/D5626.htm>. Acesso em: 24 ago. 2015.

BRASIL, **Lei nº 10.436 de 24 de abril de 2002**. Dispõe sobre a Língua Brasileira de Sinais – Libras e dá outras providências. Disponível em: <<http://portal.mec.gov.br/seesp/arquivos/pdf/lei10436.pdf>>. Acesso em: 23 ago. 2015.

BORBA, Victor Ubiracy. **Estudo Comparativo de Algoritmos de Extração de Fundo**. Marília, 2014. Disponível em: < http://aberto.univem.edu.br/bitstream/handle/11077/1_024/Victor%20Ubiracy%20Borba.pdf?sequence=1>. Acesso em: 08 dez. 2015.

BUENO, Marcelo Lemes. **Detecção de bordas através do algoritmo Canny**. Disponível em: <<http://www.inf.ufsc.br/~visao/2000/Bordas/index.html>>. Acesso em: 22 de set. 2015.

CAMPOS, Andréa Gomes. **Detecção e análise de contornos em imagem 2D**. São Carlos, 1998. Disponível em: < http://www.teses.usp.br/teses/disponiveis/76/7_6132/tde-23092008-110948/pt-br.php>. Acesso em: 06 dez. 2015.

CARDIM, Guilherme Pina. **Desenvolvimento do sistema computacional cartomorph para processamento de imagem de sensoriamento remoto**. Disponível em: <http://www2.fct.unesp.br/pos/cartografia/docs/teses/d_cardim_gp.pdf>. Acesso em: 07 de dez. 2015.

CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. **COMPUTAÇÃO GRÁFICA: Teoria e Prática**. 2ª. Edição. Rio de Janeiro: Elsevier, 2008.

COSTA, Pedro Bastos. **Calibração de padrões dimensionais de comprimento utilizando visão computacional**. Niterói, 2012. Disponível em: < <http://www.mec.ufrj.br/pdfteses/PedroBastosCosta2012.pdf>>. Acesso em: 06 de dez. 2015.

DAVIS, Clodoveu. **Simplificação de poligonais**. 2000. Disponível em: <<http://mundogeo.com/blog/2000/06/01/simplificacao-de-poligonais/>>. Acesso em: 06 dez. 2015.

DANTAS, Paulo Felipe dos Santos. **Lousa digital baseada em visão computacional**. Ceará, 2015. Disponível em: <<http://www.repositoriobib.ufc.br/00001d/00001d26.pdf>>. Acesso em: 21 de set. 2015.

GONZALEZ, Rafael, WOODS, Richard. **Processamento de imagens digitais**. 1ª edição. São Paulo: Blucher, 2000.

HIGASHINO, Wilson Akio. Estudo Comparativo de Algoritmos de Subtração de Fundo em Sequências de Imagens. Campinas 2006. Disponível em: < <http://www.bibliotecadigital.unicamp.br/document/?code=vtls000382691>>. Acesso em: 05 dez. 2015.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MASUTANI, Victor Hugo. Et al. **Reconhecimento de mãos usando cores e formas aplicada no projeto de uma interface gestual**. Disponível em: <http://iris.sel.eesc.usp.br/wvc/Anais_WVC2012/pdf/97139.pdf>. Acesso em: 05 dez. 2015.

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

ROGAWSKJ, Jon. **Cálculo**. 2º Ed. Porto Alegre, 2009.

SCURI, Antônio. **Fundamentos da Imagem Digital**. Rio de Janeiro: Tecgraf/PUC-RIO, 2002. Disponível em: <<http://www.inf.ufes.br/~thomas/graphics/www/apostolsCIV2801ScuriImgDigital.pdf>>. Acesso em 13. Ago de 2015.

SILVA, Luiza. **Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais**. Disponível em: <http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=7587@1>. Acesso em: 20 de set. 2015.

SOMMERVILLE, Ian. **Engenharia de Software**. 9.a edição. São Paulo: Pearson Prentice Hall, 2011.

TRUYENQUE, Michel. **Uma aplicação de visão computacional que utiliza gestos da mão para interagir com o computador**. Rio de Janeiro, 2005. Disponível em: <http://www.tecgraf.pucRio.br/publications/diss_2005_alain_aplicacao_visao_computacional.pdf>. Acesso em: 21 set. 2015.