

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

Danilo de Oliveira Dorotheu

Diego da Silva Santana

Marcio Fernandes Cruz

Thiago Guy Mozol Vicente

Prof. Me. Gley Fabiano C. Xavier

## **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.

## **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 na nação como é a língua portuguesa.

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.

No contexto da aplicação serão demonstradas as etapas comuns a qualquer sistema de visão computacional, serão discutidos conceitos e, por fim, feito uma proposta de um conjunto de técnicas necessárias para resolver o problema em questão.

## **2 DESENVOLVIMENTO**

### **2.1 Um interpretador de alfabeto digital em Libras**

O alfabeto digital, único para cada língua de sinais, traz a possibilidade de se entender e soletrar as letras entre pessoas ouvintes e surdas. Porém, como visto acima, a língua de sinais não é só o alfabeto digital e, exige uma tradução muito maior envolvendo a tridimensionalidade e o contexto da situação onde se está praticando o uso da língua.

Com as facilidades trazidas pela Informática, existem empresas que criaram

aplicativos que fazem a tradução da língua oral-auditiva para a visual-gestual. No caso de Libras, existe o aplicativo “Hand Talk”, que além de traduzir o alfabeto digital, traduz outros gestos. Embora uma ferramenta interessante e que facilita o trabalho de tradução, porém não elimina o papel do tradutor/intérprete humano.

Os computadores que realizam traduções automáticas oferecem uma tradução razoável de uma língua para outra, mas não pode fornecer uma tradução perfeita o tempo todo. A Hand Talk é uma potente ferramenta complementar ao trabalho do intérprete de Libras. Na internet, por exemplo, é impossível atualmente os humanos interpretarem em Língua de Sinais os bilhões de novas notícias diárias dentro de milhões de sites espalhados pelo mundo, automaticamente. Já o Hugo consegue está presente em todos esses sites, ao mesmo tempo, levando acessibilidade para milhões de surdos, instantaneamente. (TALK, 2015).

Além da dificuldade apresentada no aplicativo “Hand Talk”, existem outras limitações. Numa comunicação, muitas vezes, envolve o emissor e o receptor. Sendo assim, um receptor de sinais em Libras pode responder com gestos em Libras e, a tecnologia atual do “Hand Talk”, não permite esta tradução. Mesmo diante de todas estas limitações, é demonstrado em seu site o seu valor para a sociedade. “Eleito pela ONU o melhor app social do mundo” (TALK, 2015).

Enfim, é demonstrado a proposta deste trabalho acadêmico, onde foi executado um estudo para fazer com que uma entrada de dados, feita através de sinal em língua de sinais, possa ser traduzida em uma letra no computador. Serão usadas algumas letras do alfabeto digital de Libras.

## **2.2 Conceituando a Visão Computacional**

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.” (CONCI; AZEVEDO; LETA, 2008, p. 5).

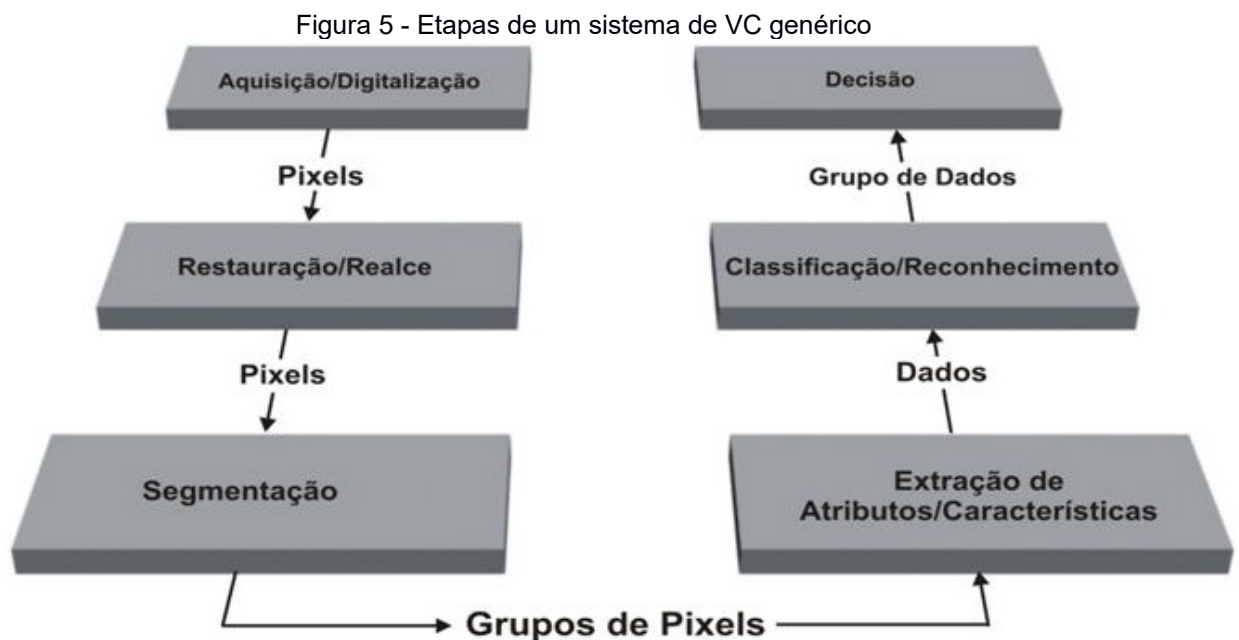
Sobre a abrangência de atuação, o autor cita como exemplo que essa área atua em reconhecimento de pessoas, objetos e peças fora do padrão de uma linha de produção, sensoriamento remoto, etc.

É citado por Conci, Azevedo e Leta (2008, p. 5) que a visão computacional pode atuar em conjunto com a área da 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. Pela figura 5 pode se observar as etapas.



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. A primeira etapa de qualquer sistema de visão computacional é a de aquisição de imagens como será apresentada na próxima sessão.

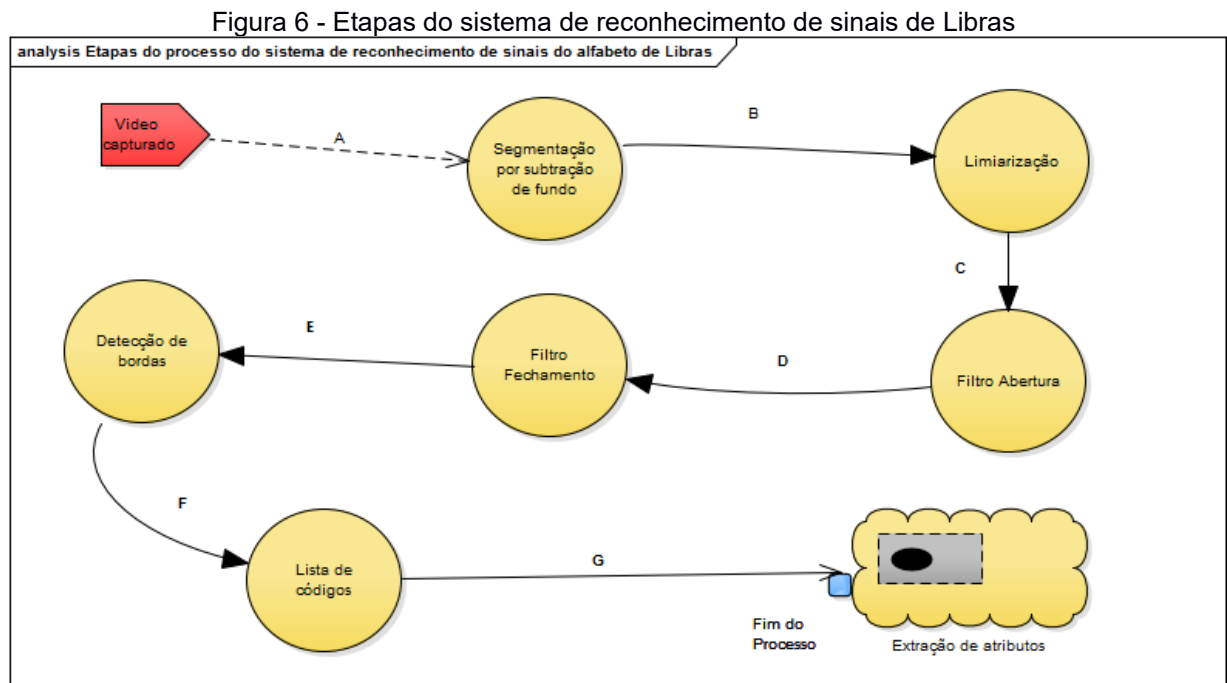
## 2.3 Automatizando a interpretação de um sinal de Libras

Para o desenvolvimento do programa do computador proposto, este trabalho

se utiliza das funções de um *framework* de aplicações<sup>1</sup> especializado em computação gráfica, o *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 computação gráfica 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).



Fonte: Elaborada pelos autores.

(A): O vídeo é enviado ao método de segmentação.

<sup>1</sup> *Framework* de aplicações é definido por Sommerville (2011) como uma estrutura genérica estendida para se criar uma aplicação ou sub-sistema mais específico.

- (B): A imagem colorida é transformada em preto e branco.
- (C): Aplicado filtro morfológico de Abertura.
- (D): Aplicado filtro morfológico de Fechamento.
- (E): É detectado as bordas.
- (F): A borda da imagem é representada em uma lista de códigos.
- (G): O sistema compara a lista de códigos da imagem com a base de dados e encontra a mais próxima para verificar se o gesto da imagem inicial é um sinal de Libras.

### 2.3.1 Equipamento para aquisição da imagem

Quanto ao sensor que captará a imagem será uma câmera de vídeo comum acoplada ao computador. Existem sensores específicos para captar e interpretar movimentos como o *kinect*<sup>2</sup> da empresa Microsoft®. Certamente a utilização deste aparelho traria maior eficiência e acertabilidade nos reconhecimentos de gestos. Entretanto, a utilização deste encapsularia conceitos e técnicas de processamento de imagens ao qual se pretende explorar no desenvolvimento deste trabalho.

### 2.3.2 Um tipo de sistema que não opera em tempo real

Para sistemas de visão computacional, o correto é trocar a expressão “tempo real” por “fortemente acoplado” devido ao tempo de resposta deste tipo de software. Segundo Truyenque (2005), a mudança de expressão se justifica devido ao fato de haver um tempo de resposta da aplicação desde a aquisição, passando pelas fases de processamento de imagem até o reconhecimento desta. Ainda segundo o autor, este tempo é de latência do sistema e, pode variar de aplicação para aplicação.

Neste trabalho, para reduzir o tempo de processamento, é proposto logo na primeira fase aplicar um ambiente controlado para aquisição de imagem.

### 2.3.3 Importância de um ambiente controlado

O ambiente controlado é definido neste trabalho como um local de iluminação artificial e com um fundo (que será usado para contraste de objeto) de cor sólida.

Num ambiente previamente conhecido, o processo inicial de aquisição da

<sup>2</sup>*Kinect* é um periférico que tem o objetivo de reconhecer e analisar movimentos do corpo humano. Ele originalmente foi feito para o console de videogame Xbox, mas, o fabricante, a Microsoft Corporation disponibiliza um kit de desenvolvimento para que desenvolvedores possam criar programas de computador que fazem uso deste aparelho. Disponível em: <>. Acesso em 26 set. de 2015.

imagem traz um produto de saída de ótima qualidade para as etapas subsequentes.

O objetivo principal deste ambiente é reduzir a probabilidade de se encontrar sombras e ruídos. Estes podem ser incorporados erroneamente ao objeto a ser analisado, atrapalhando o reconhecimento de um eventual gesto válido em etapas subsequentes. “Imagens reais frequentemente sofrem degradações durante seu processo de aquisição, transmissão ou processamento. Essa degradação é normalmente chamada de ruído.” (PEDRINI; SCHATZ, 2008, p. 27).

No programa de computador apresentado nesse trabalho não serão aplicados técnicas de realce ou restauração da imagem digital adquirida. Entretanto na etapa de segmentação que será vista na próxima sessão, serão aplicados filtros morfológicos para melhorar a imagem do objeto presente na imagem captada.

#### **2.3.4 Segmentando a imagem de uma mão humana**

O objetivo da segmentação é separar o objeto a ser analisado do fundo da imagem. Esse objeto é separado em nova imagem digital e esta será passada a próxima etapa de processamento.

Neste trabalho é utilizado o algoritmo KNN, acessado através da função *createBackgroundSubtractorKNN* do *OpenCV*.

Nas subseções seguintes será descrito a técnica para subtrair o fundo da imagem, limiarização e aplicação de melhorias nesta para passar a próxima etapa.

#### **2.3.5 Descartando o fundo irrelevante**

O objeto “mão humana” que conterà o gesto a ser interpretado será captado em um movimento dentro de um vídeo que contém um fundo estático. Sendo assim, podemos usar o fator movimento para facilitar o reconhecimento de objetos.

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 a ser observada [...]. (GONZALEZ; WOODS, 2000, p. 331).

A característica principal de algoritmos que utilizam segmentação de imagens por subtração de fundo é conseguir uma imagem de referência que representa o

fundo estático. Para conseguir essa imagem de referência, o algoritmo utiliza o conceito de treinamento de máquina.

Basicamente, a subtração do fundo consiste em subtrair a imagem atual de outra usada como referência, ou seja, uma imagem contendo somente o fundo da cena e construída a partir de uma sequência de imagens com variações de iluminação durante um período de treinamento. A subtração de fundo é utilizada para capturar apenas objetos não estáticos (em movimento) e novos objetos em uma cena, que é um passo fundamental [...]. (BORBA 2014 apud SCHNEIDER 2007, p. 19).

### 2.3.6 Aplicando filtros para refinamento

Para melhoramento dos aspectos da imagem da mão humana, serão aplicados neste trabalho o filtro morfológico de abertura e fechamento. Estes utilizam internamente outros dois filtros morfológicos: erosão e dilatação.

[...] a dilatação expande uma imagem, enquanto a erosão a reduz. [...] A abertura geralmente suaviza o contorno de uma imagem, quebra istmos estreios e elimina protusões finas. O fechamento também tende a suavizar os contornos, mas, em oposição à abertura, geralmente funde as quebras em golfos finos, elimina pequenos buracos e preenche fendas em um contorno. (GONZALEZ; WOODS, 2000, p. 373).

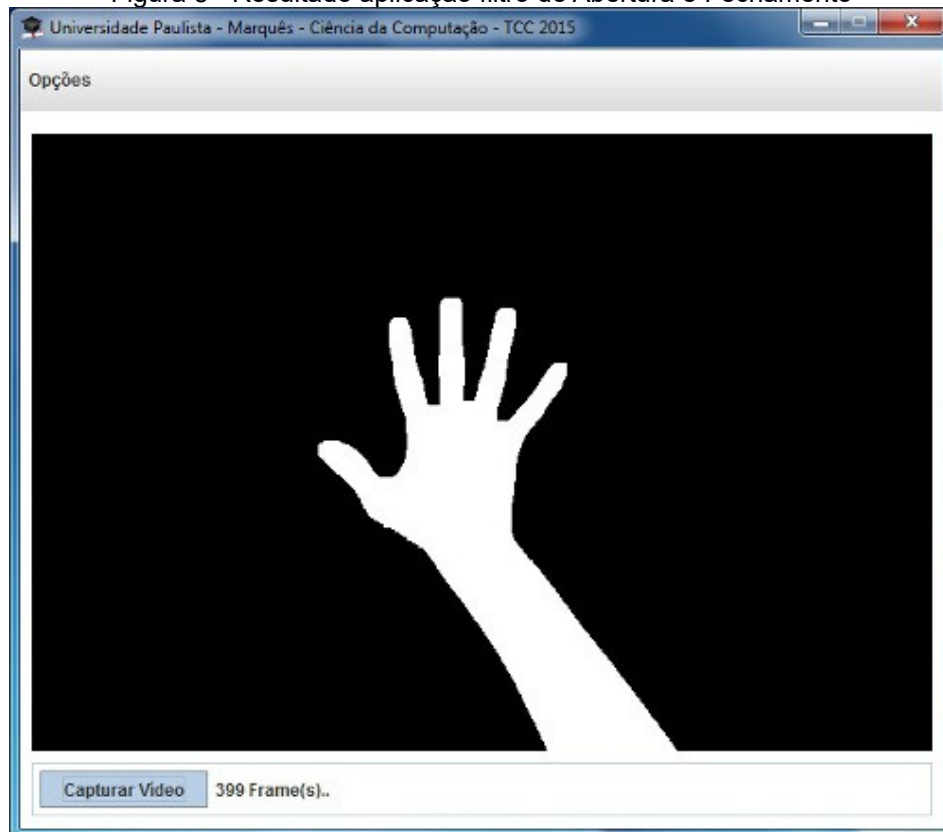
Sobre os filtros morfológicos em si, recorremos a Pedrini e Scharzt (2008) que nos diz como foram fundamentados. Os filtros morfológicos fazem parte da metodologia da morfologia matemática. “A morfologia matemática consiste em uma metodologia para análise de imagens que permite a construção de operadores úteis para a descrição de objetos em imagens.” (PEDRINI; SCHARTZ, 2008, p. 327). Ainda sobre esses filtros, o autor enfatiza que são baseados na teoria dos conjuntos.

Sem se adentrar em explicações matemáticas, o filtro morfológico de abertura é o resultado da aplicação de uma dilatação sobre uma imagem que sofreu erosão. Enquanto que o filtro de fechamento é o resultado da aplicação do método de erosão sobre uma imagem dilatada. (PEDRINI; SCHARTZ, 2008, p. 350).

Neste trabalho, é utilizado a função *Opening* e em seguida *Closing* do *OpenCV*.



Figura 8 - Resultado aplicação filtro de Abertura e Fechamento



Fonte: Extraída do programa desenvolvido.

### 2.3.7 Detectando bordas e contornos da mão

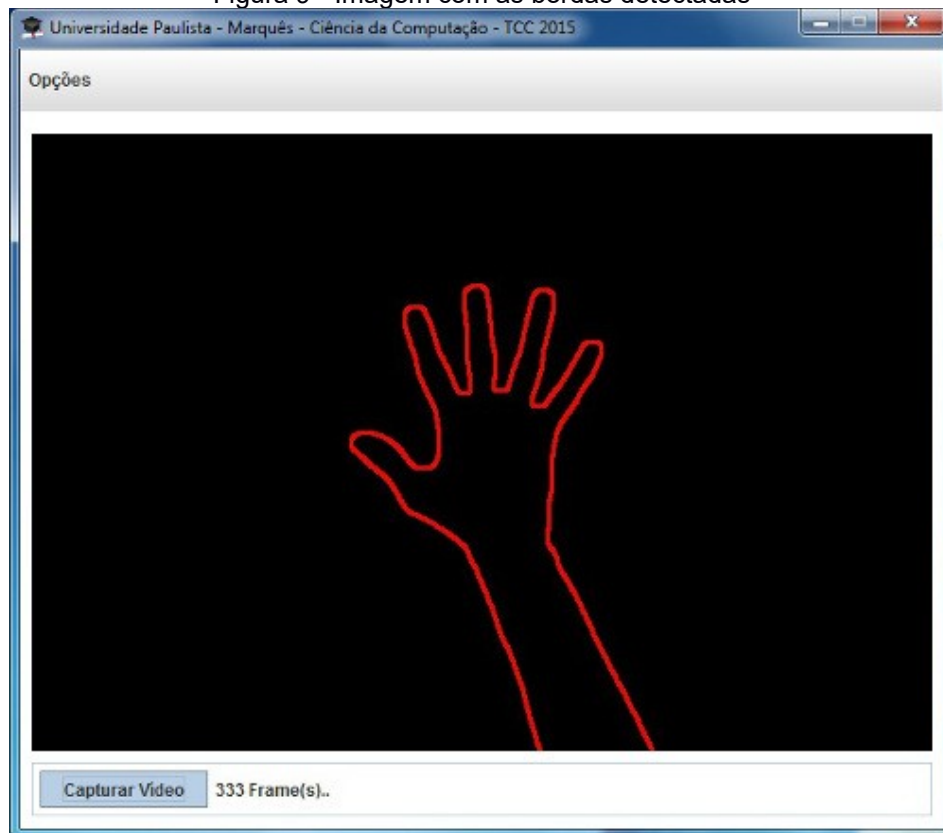
A imagem de entrada nesse processo é uma figura com duas cores. A abordagem desse trabalho é a partir da forma do objeto mão, conseguir localizar um padrão de letra que o represente, interpretando o gesto. Portanto, será aplicado método para detectar as bordas, eliminando o resto da figura. Ou seja, a parte exterior e interior da mesma.

Sem descrever detalhes do funcionamento, será utilizado neste trabalho o método *Canny*, “[...] um método para detecção de bordas que procura otimizar a localização de pontos da borda na presença de ruído[...]” (PEDRINI; SCHARTZ, 2008, p. 166).

Para conseguir a silhueta de um objeto, inicialmente devem-se mapear as bordas internas e externas deste. “Uma borda é o limite ou a fronteira entre duas regiões com propriedades relativamente distintas de nível de cinza.” (PEDRINI;

SCHARTZ, 2008, p. 153).

Figura 9 - Imagem com as bordas detectadas



Fonte: Extraída do programa desenvolvido.

No trabalho prático é utilizada a função *Canny* para detectar as bordas. E a etapa de segmentação termina na descoberta das bordas. Na próxima sessão será discutida a etapa de representação e descrição do objeto “mão”.

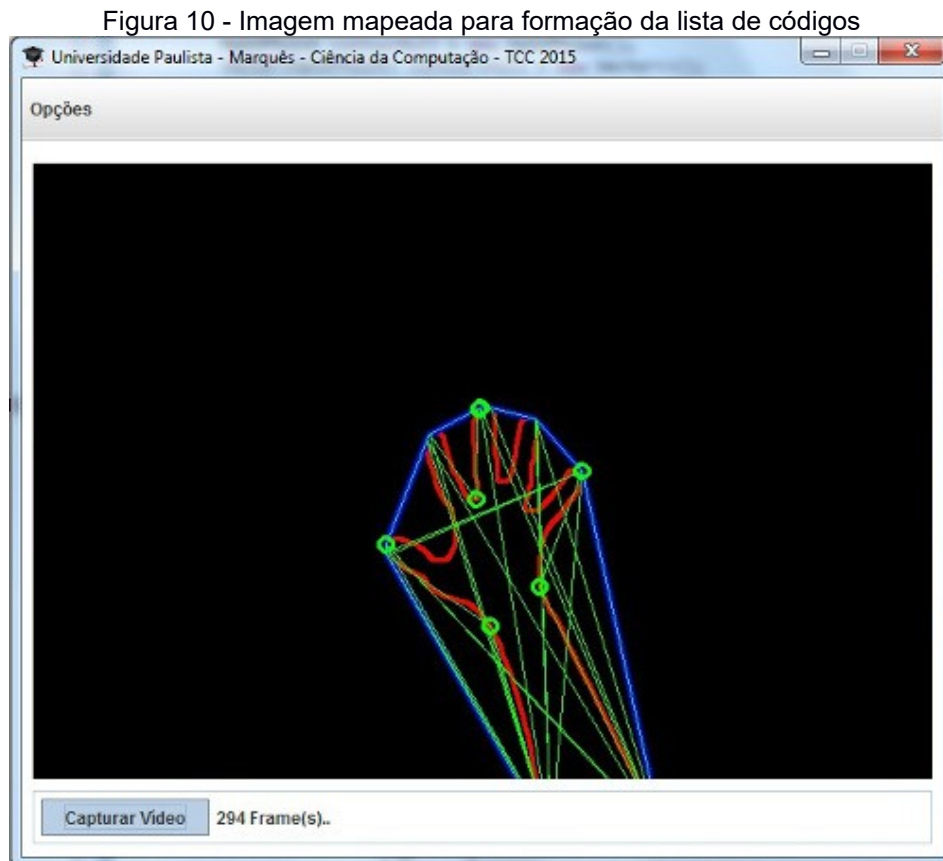
### 2.3.8 Representando os códigos do contorno da mão

Nesta etapa do processamento, usamos o contorno da imagem para representar em uma estrutura de dados no computador.

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

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; SCHARTZ, 2008, p. 250).

Sem maiores detalhes de funcionamento, conseguimos ter 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,



Fonte: Extraída do programa desenvolvido.

A imagem de uma mão, que tem seu contorno detectado em etapa anterior, possui um identificador que se chama lista de códigos. Este é criado pela função *FindContours* da *OpenCV*

A próxima etapa do processamento será o reconhecimento deste objeto como um sinal válido ou não de Libras.

### 2.3.9 Reconhecendo um gesto válido do alfabeto digital

Esta é a última etapa do sistema de visão computacional que tem o objetivo de verificar a lista de códigos de determinada imagem de um objeto identificado em etapas anteriores possui uma representação próxima a um padrão preexistente na

base de dados.

É retomada a execução do algoritmo K-NN da etapa de segmentação, mas, agora com o objetivo de identificar um padrão dentro do conjunto válido do alfabeto digital de Libras. Sendo que cada letra ou número é uma classe de padrões e, possui uma lista de cadeia de códigos que a representam. “Aqui, um padrão é uma descrição quantitativa ou estrutural de um objeto ou alguma outra entidade de interesse de uma imagem. [...] Uma classe de padrões é uma família de padrões que compartilhem algumas propriedades comuns.” (GONZALEZ; WOODS, 2000, p. 409).

O algoritmo para reconhecimento é bem direto. Vai receber como entrada a lista de código conseguida com a sequência de processamento até este momento. E, vai verificar na base de dados o padrão mais próximo. Se encontrar um padrão, identificará a classe e consequentemente o caractere que o representa. Caso contrário, retornará como “gesto não identificado” ao sistema.

As funções do *OpenCV* que usaremos nesta fase do trabalho é a *Flann* (acrônimo do inglês *Fast Approximate Nearest Neighbor*) e *FlannBasedMatcher*. Como dito em sessões anteriores, o parâmetro K é igual a 1. Ou seja, queremos verificar se existe pelo menos um padrão próximo para identificar se uma lista de código é válida ou não.

A invocação da função *Flann* da biblioteca *OpenCV* dentro da aplicação desenvolvida na linguagem Java não demonstraram o resultado esperado. O que inviabilizou a continuidade no desenvolvimento do *OpenCV* junto com o Java, pelo menos no aspecto de funções de busca de padrões em base de dados.

Foi retomado a pesquisa e foi verificado junto a própria documentação do *OpenCV* que a linguagem de programação Python possui uma alta compatibilidade na utilização com este framework.

Assim, está sendo desenvolvido uma outra aplicação em Python que tem o mesmo objetivo da aplicação inicial em Java, que é receber uma sequência de imagens e extrair a informação do gesto que o representa no alfabeto digital de Libras. A sequência de métodos utilizadas na linguagem Python é a mesma aplicada na linguagem Java. Ou seja, o desenvolvimento em Java não foi em vão pois ajudou a trilhar o caminho para a acertabilidade em outra plataforma.

Conclui-se aqui o estudo dos conceitos e técnicas para o desenvolvimento de uma aplicação que utiliza conceitos de visão computacional para atuar no domínio

do problema de reconhecimento de gestos do alfabeto digital de Libras.

### **3 CONCLUSÃO**

A contribuição principal deste trabalho foi demonstrar que a visão computacional é uma área inovadora e abre espaço para o desenvolvimento de excelentes projetos para facilitar a vida na sociedade.

Recorremos à literatura para descrever que o funcionamento do sistema de visão humana é mais complexo, eficiente e assertivo comparado ao de visão artificial. Além de capacidade de processamento e percepção do mundo exterior mais apurado há outro fator, a inteligência.

O fato do sistema de visão humana ser mais eficiente não deve ser encarado como motivo desencorajador para que se não se busque a criação de melhores algoritmos e soluções computacionais para processamento e análise de imagem. Foi descrito vários exemplos onde o uso da visão computacional se faz presente.

No contexto do trabalho, foi proposto a criação de um sistema de interpretação de sinais de um alfabeto de Libras. Na condução deste assunto, foram estudados as etapas, conceitos e técnicas específicas que buscassem a interpretação automatizada de uma letra, feita através de um gesto da mão humana.

Inicialmente foi utilizado a linguagem Java junto com o framework OpenCV. Não foi obtido sucesso no fechamento do processo pois, na última etapa, que buscava informações na base, o método Flann se mostrou problemático. Assim, foi iniciado uma nova aplicação utilizando o mesmo framework OpenCV compilado para o ambiente Python.

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.

Uma sugestão de trabalho futuro é o desenvolvimento de um sistema de visão computacional para interpretar a linguagem Libras em si. É um trabalho que necessitaria no mínimo de estudos de neurolinguística e inteligência artificial. Isso devido ao fato da linguagem Libras ser de comunicação humana como qualquer outra. Automatizar a interpretação de uma linguagem gestual dentro de todo o contexto da situação que se encontra determinada conversação seria um grande

desafio para os cientistas da computação.

#### 4 REFERÊNCIAS BIBLIOGRÁFICAS

ANATOMIA, **O olho Humano**. Disponível em: <<http://www.laboratoriorigor.com.br/anatomia.html>>. Acesso em: 6 set. 2015.

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

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.

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

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.

FELIPE, Tanya A. **Libras em Contexto: Curso Básico: Livro do Estudante**. 8. ed. Rio de Janeiro: WalPrint Gráfica e Editora, 2007.

GERRIG, Richard; ZIMBARDO, Philip. **A psicologia e a vida**. 16ª. Edição. São Paulo: Artmed, 2004.

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

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

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

DOROTHEU, Danilo et al. **Uma aplicação para interpretar sinais do alfabeto digital de Libras**. São Paulo: Unip, 2015.