Identificação de Caracteres Libras por Visão Computacional

Vanessa R. C. Leite

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) – Rio de Janeiro – RJ – Brasil

{vleite}@inf.puc-rio.br

Abstract. Gestures are an essential complement to spoken language and the primary communication tool in the absence of speech. In Brazil, communication through gestures was regulated by creating a new language: the Brazilian Signal Language (LIBRAS, from Portuguese "Língua Brasileira de Sinais"). It is through that deaf people can communicate. However, most of those without disabilities don't know LIBRAS, hindering communication and inclusion of deaf people. Thus, the automatic recognition of LIBRAS provides an opportunity for inclusion, as they open doors for translating gestures into text and even speech, allowing deaf people to communicate with everyone.

Resumo. Os gestos são utilizados de forma completar à fala, e na ausência desta, eles são as principais ferramentas da comunicação. No Brasil, a comunicação através de gestos foi regulamentada, criando-se uma nova língua: a Língua Brasileira de Sinais (LIBRAS). É por meio dela que pessoas com deficiência auditiva se comunicam. Entretanto, a maioria das pessoas sem deficiências não sabem LIBRAS, dificultando a comunicação e a inclusão daquelas pessoas. Assim, o reconhecimento automático de LIBRAS possibilita uma oportunidade de inclusão das pessoas com deficiência, uma vez que abrem portas para a conversão dos gestos em texto, e até mesmo em fala.

1. Introdução

Os gestos são muito importantes na comunicação, sendo utilizados de forma complementar à fala. Na ausência da linguagem falada, os gestos recebem uma conotação especial, passando de coadjuvante a atores principais da comunicação. Dessa forma, os gestos passam a constituir uma nova língua com uma gramática própria. A linguagem de sinais recebe o *status* de língua por que possui todos os níveis linguísticos necessários: fonológico, morfológico, sintático e semântico [INES 2010].

O reconhecimento automático de gestos humanos pode ser utilizado em diversas aplicações úteis à sociedade como, por exemplo, em ambientes virtuais, sistemas de segurança, interação homem-máquina etc [Garg et al. 2009] [Sánchez-Nielsen et al. 2003]. Em especial, o reconhecimento de caracteres da linguagem de sinais abre uma possibilidade de inclusão das pessoas com deficiências auditivas, à medida que fornece um mecanismo para conversão da linguagem de sinais em texto ou fala, o que possibilita a comunicação com aqueles que não sabem essa linguagem.

Esse reconhecimento automático pode ser feito, basicamente, por meio de:

i. **dispositivos rastreadores** como luvas, roupas, cabos etc. A utilização de dispositivos garante uma maior precisão no reconhecimento, além de evitar problemas como o de oclusão. Entretanto, esses dispositivos costumam ser caros e limitam a liberdade de movimento do usuário, e por vezes causam algum desconforto [LaViola 1999]; ou

ii. **visão computacional**, que utiliza, em geral, apenas uma câmera, e permite que o usuário tenha uma maior liberdade de movimento. Além disso, tal abordagem permite a utilização da informação de cor, o que em alguns casos pode ser útil.

1.1. Motivação

No Brasil, segundo o IBGE (Instituto Brasileiro de Geografia e Estatísticas), 166.400 pessoas são surdas e quase 900 mil tem dificuldade permanente em ouvir [IBGE 2000]. Para facilitar a comunicação entre as pessoas com e sem deficiência auditiva foi criada a LIBRAS (Língua Brasileira de Sinais), que é uma mistura da língua francesa de sinais com a língua de sinais brasileira antiga [Menezes 2010]. LIBRAS possui gestos que reproduzem caracteres (letras de A a Z e números de 0 a 9) e também palavras, conforme pode ser visto no [DicionarioLibras 2008]. Na Figura 1 estão os caracteres LIBRAS que reproduzem o alfabeto, alguns caracteres são estáticos, enquanto outros são dinâmicos.

Figura 1 - Caracteres LIBRAS utilizados para soletrar palavras

Apesar de LIBRAS ter o objetivo de facilitar a comunicação das pessoas com deficiência auditiva, a maioria das pessoas que não possuem esse tipo de deficiência não sabem falar em LIBRAS. Sendo assim, a principal motivação deste trabalho é proporcionar um método que permita facilitar a comunicação entre aquelas pessoas com deficiência auditiva e aqueles que não conhecem LIBRAS.

1.2. Objetivo

O objetivo deste trabalho é propor um procedimento para o reconhecimento automático dos caracteres de LIBRAS capturados por uma câmera e convertê-los para texto. Este procedimento é tratado aqui em três etapas: segmentação da imagem da mão, extração

de características desta imagem e, por fim, classificação através de métodos de aprendizagem de máquina.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta alguns trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada, explicando como foram feitas a segmentação, a definição das características e a classificação das imagens. Na Seção 4, apresentaremos uma implementação e os resultados obtidos a partir dela. Finalmente, na Seção 5 apresentamos as conclusões e trabalhos futuros.

2. Trabalhos relacionados

O reconhecimento automático de caracteres da linguagem de sinais tem sido bastante abordado na literatura. De uma maneira geral, o processo para o reconhecimento usando visão computacional, segue as três etapas já enunciadas: segmentação, extração de características e classificação. Em [Carneiro 2009], que serviu como principal referência para este trabalho, é utilizado uma métrica simples para segmentação baseada apenas no padrão de cor da imagem. Esse trabalho utiliza momentos invariantes de HU para a extração de características e redes SOM (Self-Organizing Maps) para fazer a classificação. Já em [Pistori et al. 2007] foi utilizada uma plataforma de código aberto para tratamento da imagem (SIGUS, [Pistori et al. 2006]) e HMM (Hidden Markov Models) para a classificação, objetivando reconhecer, não os caracteres, mas algumas palavras próprias da língua. Em [Pizzolato et al. 2010] é feita a identificação de caracteres através de extração simples de características da imagem e HMM para classificação.

3. Método

Nesta seção, descrevemos a metodologia utilizada para o desenvolvimento deste trabalho, apresentamos o funcionamento da segmentação e dos Momentos Invariantes de HU, bem como o classificador utilizado.

3.1. Segmentação da Imagem

Segmentar uma imagem significa dividir a imagem em regiões e selecionar uma determinada região de interesse. Em geral, isso é feito atribuindo-se valores conhecidos aos *pixels* como, por exemplo, "1" caso o *pixel* pertença à região de interesse e "0" caso contrário. Neste trabalho, dado um ambiente no qual serão reproduzidos os gestos dos caracteres LIBRAS, faz-se necessário segmentar a imagem, removendo o fundo e identificando apenas a mão do usuário (a região de interesse).

Existem várias técnicas de segmentação de imagem, tais como limiar em tons de cinza, segmentação orientada a crescimento de regiões, agrupamento por pixel, agrupamento por histograma etc. A maioria dessas técnicas levam em consideração o valor original do *pixel,* e o classificam baseando-se em padrões conhecidos de cor. Neste trabalho, para identificar a mão do usuário, também usamos uma segmentação que se baseia em imagens coloridas, uma vez que a identificação de pele em imagens coloridas pode ser feita de forma simples, conforme apresentado por [Oliveira 2006] [Ribeiro 2008].

Seguindo a proposta de [Carneiro 2009], a segmentação utilizada neste trabalho baseia-se na conversão do padrão de cor da imagem para YCbCr, que é um padrão de

cor no qual o Y representa a luminância da imagem e os componentes Cb e Cr representam o azul e o vermelho, respectivamente. Uma vez a imagem convertida, são aplicados limiares nos canais Cb e Cr.

3.2. Momentos Invariantes de HU

Para poder determinar a classificação da região de interesse é necessário construir um vetor de características que possuam medidas que sejam similares para objetos de mesma classe (neste trabalho, o mesmo gesto). Em geral, posição e tamanho não são boas características, um avião continua sendo avião em maior ou menor escala, por exemplo. Entretanto, características ligadas a cor, ou textura, tendem a ser invariantes, ou seja, independente de escala, rotação ou translação, elas permanecem as mesmas. Com base nisso, tem-se os momentos invariantes que são medidas puramente estatísticas da distribuição dos pontos.

Os Momentos Invariantes de HU são definidos com base em uma imagem binária MxN e uma função f(x,y) que representa o estado do pixel (x,y): preto ou branco. Um momento de ordem (p+q) é definido pela Equação 1:

$$m_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^p y^q f(x, y)$$

Equação 1 - Definição de momento invariante

O momento de ordem 0 (m_{00}) representa a superfície da imagem, enquanto os momentos de ordem 1 (m_{10}) e (m_{01}) definem o centro de gravidade (x_g e y_g) da imagem, conforme a Equação 2.

$$x_g = \frac{m_{10}}{m_{00}} \quad \text{e} \quad y_g = \frac{m_{01}}{m_{00}}$$

Equação 2 – Centros de gravidade x_g e y_g

Com o intuito de ser invariante à rotação e translação HU [HU 1962] definiu os momentos centrais n_{pq}, mostrados na Equação 3:

$$n_{pq} = \sum_{x=1}^{N} \sum_{y=1}^{M} (x - x_g)^p (y - y_g)^q f(x, y)$$

Equação 3 – Definição do momento central

Os momentos centrais de ordem 2 permitem achar os eixos principais de inércia, os prolongamentos e as orientações da forma. Para que os momentos sejam invariantes à escala, os mesmos devem ser normalizados pelo tamanho da imagem, conforme

Equação 4.

$$\mu_{pq} = n^{p+q} \gamma_{pq} \quad \text{onde} \quad \gamma = \frac{1}{(n_{20} + n_{02})^{2} + 1}$$

Equação 4 – Normalização dos momentos

Assim, para construir um vetor de características que seja invariante à rotação, escala e translação, utiliza-se os 7 momentos invariantes de HU (de ordem 2 e 3), conforme as Equações (5 – 11).

$$\varphi(1) = \mu_{20} + \mu_{02}$$

Equação 5 – Momento 1

$$\varphi(2) = (\mu_{20} - \mu_{02})^{2} + 4\mu_{11}^{2}$$

Equação 6 – Momento 2

$$\varphi(3) = (\mu_{30} - 3\mu_{12})^{2} + (3\mu_{21} - \mu_{03})^{2}$$

Equação 7 – Momento 3

$$\varphi(4) = (\mu_{30} + \mu_{12})^{2} + (\mu_{21} + \mu_{03})^{2}$$

Equação 8 - Momento 4

$$\varphi(5) = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^{2} - 3(\mu_{21} + \mu_{03})^{2}] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^{2} - (\mu_{21} + \mu_{03})^{2}]$$

Equação 9 – Momento 5

$$\varphi(6) = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^{2} - (\mu_{21} + \mu_{03})^{2}] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$

Equação 10 – Momento 6

$$\varphi(\ )=(\ \mu\ )(\mu+\ )[(\mu+\mu\ )\ (\mu+\mu\ )\ ]\ 7\ 3\mu\ 3\mu\ 3$$

$$(\mu\ )(\mu+\mu\ )[\ (\mu+\mu\ )\ (\mu+\mu\ )\ ]$$

$$3\mu\ 3$$

Equação 11 – Momento 7

3.3. Classificação das Imagens

Para a classificação das imagens foi usado o SVM, que é uma técnica de aprendizado de máquina introduzida por Vapnik e sua equipe [Osuna et al. 1997]. O SVM é basicamente uma implementação do método de minimização de risco estrutural para o treinamento de classificadores (aprendizagem de máquinas). Uma das principais características do SVM é que ele possui um bom desempenho no reconhecimento de padrões nos casos de grande volume de dados.

A classificação de dados linearmente separáveis utilizando um SVM consiste na criação de um hiperplano ótimo que separa os dados em duas classes, como mostra a Figura 2. Dado um espaço em que os dados possam assumir apenas dois valores (+1, - 1), um hiperplano separa estes dados de forma que um dos lados contenha apenas dados da classe 1, e do lado oposto apenas dados da classe -1 [Burges 1998]. Esse processo é feito com base em um treinamento prévio num conjunto finito de amostras.

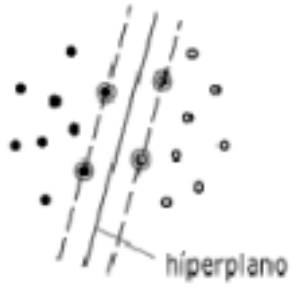

Figura 2 - Hiperplano ótimo de separação de

Atualmente o SVM tem sido utilizados para a resolução de diversos problemas como, reconhecimento de faces de pessoas, reconhecimento de escrita manual, reconhecimento de objetos, entre outros [Burges 1998].

4. Implementação e Resultados

Nesta seção será comentada peculiaridades da implementação, bem como os resultados obtidos.

A implementação deste trabalho foi desenvolvida em C++, em plataforma Linux, com a IDE Eclipse. Utilizou-se a biblioteca OpenCV 2.1 [OpenCV 2010] para facilitar na captura e tratamentos de imagem, e também a biblioteca LIBSVM [LIBSVM 2001] para o treinamento e classificação. Neste trabalho foram tratados apenas as 5 primeiras letras do alfabeto em Libras, como forma de validá-lo.

4.1. Segmentação

Para realizar a segmentação, primeiramente foi feita uma conversão do padrão de cor da imagem de RGB para YcbCr. Essa conversão foi realizada utilizando-se uma função do OpenCV.

Após a imagem convertida, aplicou-se o limiar da Equação 12, que destaca a pele, que é nossa região de interesse.

$$77 \leq Cb \leq 127 \; e \; 133 \leq Cr \leq 193$$

Equação 12 – Limiar dos canais Cb e Cr

Na Figura 3 observa-se o resultado obtido da segmentação. Note que realmente a região de interesse é obtida e que não há nenhuma outra região selecionada.

Figura 3 - Exemplo de mão segmentada. À esquerda imagem original, à direita imagem segmentada e binária.

4.2. Momentos Invariantes de HU e Classificação

Para calcular os momentos invariantes de HU, bastou seguir as equações citadas na subseção 3.2. Apesar dos cálculos serem realizados a cada frame, não houve notável perda de eficiência nos cálculos.

Os movimentos invariantes são calculados para cada frame, e ao fim do processamento são gerados os valores dos 7 principais momentos. Esses valores são repassados para o SVM para realizar a classificação.

Antes de iniciar o processo de classificação é necessário fazer o treinamento, que é realizado de forma simples: para cada imagem de treinamento é calculado os 7 momentos de HU, e esses valores são colocados em um arquivo obedecendo o seguinte formato:

```
<classe> <caracteristica>:<valor> <caracteristica>:<valor> …
```

Onde `<classe>` representa o índice da classe, `<caracteristica>` representa o índice da característica, que em nosso caso, são apenas 7 e `<valor>` representa o valor correspondente da característica. Nesse arquivo, deve ser armazenada uma classe por linha. A base gerada nesse trabalho teve poucas amostras, apenas 8 para cada uma das 5 letras tratadas, gerando um arquivo de 40 linhas. Esse arquivo gerado é passado para a LIBSVM que através de um *script* em Python gera um arquivo modelo, que será utilizado na classificação.

A classificação é feita calculando a probabilidade das características passadas fazerem parte de alguma classe do arquivo modelo. A base utilizada não teve expressividade suficiente para manter alto nível de acerto, uma vez que, por exemplo, o caracter 'A' e o 'E' em LIBRAS possuem alta semelhança. A Figura 4 mostra alguns resultados obtidos.

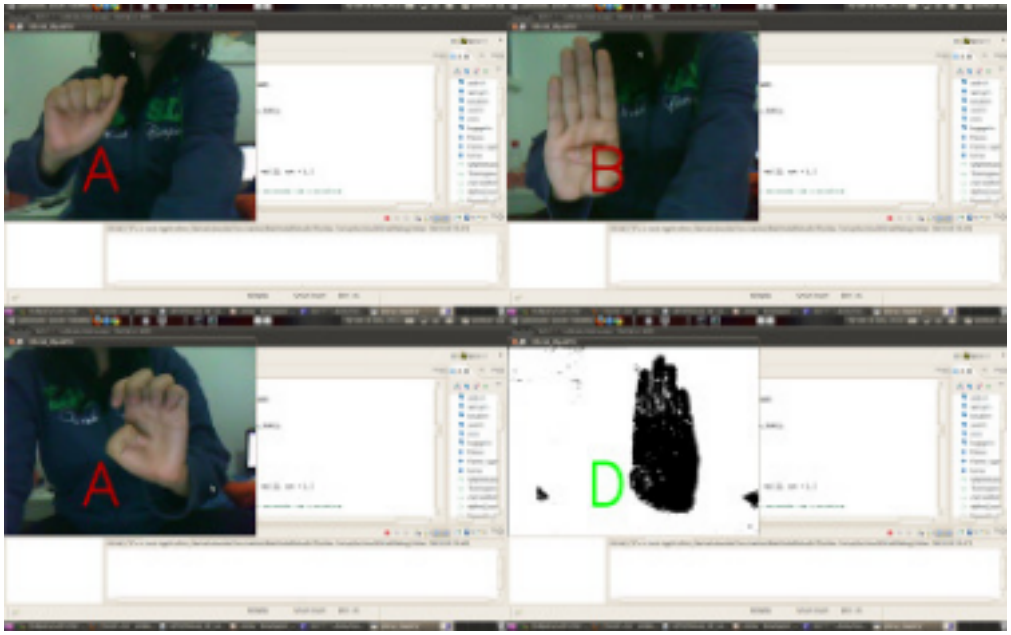

Figura 4 - Imagens superiores: classificação correta das letras A e B; inferiores: reconhecimento da letra E erroneamente como A, e da letra B como D.

5. Conclusão e Trabalhos Futuros

Após o desenvolvimento deste trabalho notou-se que, apesar da métrica para segmentação ser bem simples, ela é bastante eficiente. Entretanto, em ambientes que possuam materiais próximos a tons de pele como objetos amarelados, eles são considerados como parte da região de interesse, além de fazer com que haja a necessidade de um posicionamento da câmera no qual o rosto não fique visível, pois a pele é detectada, incluindo mais elementos indesejados.

As características extraídas através dos momentos invariantes de HU se mostraram características realmente relevantes, entretanto, com o aumento de número de classes notou-se, empiricamente, que 7 momentos podem ser características insuficientes. Já na classificação, concluiu-se que utilizar a LIBSVM é bastante simples e tem um excelente

retorno, sendo a falha neste trabalho apenas a ausência de uma base de treinamento suficientemente completa.

Como trabalho futuro, pode-se incluir alguma outra característica durante a segmentação, para garantir que nenhum objeto errôneo seja segmentado, independente de fundo e posicionamento da câmera. No que se refere à extração de característica, pode-se incluir os momentos de imagem, ou seja, características estatísticas como variância, desvio padrão, e até mesmo excentricidade e orientação da elipse que melhor se ajusta à imagem segmentada. E, é claro, estender o trabalho para as demais letras do alfabeto LIBRAS, tratando ainda as letras que são representadas por gestos dinâmicos, e criando uma base de dados maior, que seja satisfatória.

Referências

[INES 2010] Instituto Nacional de Eduação de Surdos – INES, <http://www.ines.gov.br/ines_livros/11/11_011.HTM>, acessado em Novembro/10.

[IBGE 2000] Instituto Brasileiro de Geografia e Estatísticas – IBGE, Censo 2000: <http://www.ibge.gov.br/home/presidencia/noticias/27062003censo.shtm>, acessado em Outubro/10.

[Sánchez-Nielsen et al. 2003] Sánchez-Nielsen, E., Antón-Canalís, L. and Hernández Tejera, M. (2003) "Hand Gesture Recognition for Human-Machine Interaction" In: *Journal of WSCG*, Vol.12, No.1-3

[Garg et al. 2009] Garg, R., Shriram, N., Gupta, V. and Agrawal, V. (2009) "A Biometric Security Based Electronic Gadget Control Using Hand Gestures" Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09.

[LaViola 1999] LaViola, J. J. (1999) "A survey of hand posture and gesture recognition techniques and technology" Technical report, Brown University, Providence, RI, USA, 1999.

[Menezes 2010] Menezes, E. T. and Santos, T. H. "LIBRAS (Língua Brasileira de Sinais)" (verbete). *Dicionário Interativo da Educação Brasileira* - EducaBrasil. São Paulo: Midiamix Editora, 2002, <http://www.educabrasil.com.br/eb/dic/dicionario.asp?id=40>, acessado em Novembro/10.

[DicionarioLibras 2008] Acessibilidade Brasil "Dicionário da Língua Brasileira de Sinais" <http://www.acessobrasil.org.br/libras/>, acessado em Novembro/10.

[Pistori et al. 2007] Souza, K. P., Dias, J. B. and Pistori, H. (2007) "Reconhecimento Automático de Gestos da Língua Brasileira de Sinais utilizando Visão Computacional" III Workshop de Visão Computacional, 2007, São José do Rio Preto, São Paulo.

[Pistori et al. 2006] Pistori, H., Martins, P. S., Pereira, M. C. and Neto, J. J. (2006) "Sigus - plataforma de apoio ao desenvolvimento de sistemas para inclusão digital de pessoas com necessidades especiais." IV Congresso Iberdiscap: Tecnologias de Apoio a Portadores de Deficiência, Vitória, Fevereiro/2006.

[Ribeiro 2008] Ribeiro, J. M. (2008) “Segmentação de pele humana em imagens coloridas baseada em valores das médias da vizinhança em subimagens” Dissertação de Mestrado – Escola de Engenharia de São Carlos

[Oliveira et al. 2006] Oliveira, T. M., Cortez, P. C., Silva, W. P., Soares, J. M., and Barroso, G. C. (2006) “Segmentação de pele humana em imagens de vídeo utilizando wavelet e redes neurais.” II Workshop de Visão Computacional, 2006

[HU 1962] Hu, M. K. (1962) “Visual Pattern Recognition by Moment Invariants”, IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962

[OpenCV 2010] OpenCV 2.1 <http://opencv.willowgarage.com/wiki/> acessado em Outubro/10.

[LIBSVM 2001] Chang, Chih-Chung and Lin, Chih-Jen (2001) “LIBSVM: a library for support vector machines”, Software disponivel em <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

[SVM 2010] Hsu, Chih-Wei.,

Chang, Chih-Chung and Lin, Chih-Jen (2010) “a Pratical Guide to Support Vector Classification”, disponivel em <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

[Osuna et al. 1997] Osuna, E., Freund, R., and Girosi, F. (1997). “Training

support vector machines: an application to face detection”. CVPR’97, Porto Rico, pages 130–136.

[Burges 1998] Burges, C. J. C. (1998). “A tutorial on support vector machines

for pattern recognition”. Data Mining and Knowledge Discovery, 2(2):121–167.