

# Um sistema de informação extensível para o reconhecimento automático de LIBRAS

Luciano A. Digiampietri<sup>1</sup>, Beatriz Teodoro<sup>1</sup>, Caio R. N. Santiago<sup>1</sup>,  
Guilherme A. Oliveira<sup>1</sup>, Jonatas C. Araujo<sup>1</sup>

<sup>1</sup>Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)  
Av. Arlindo Bértio, Ermelino Matarazzo – 03828-000 – São Paulo – SP – Brasil

digiampietri@usp.br

**Abstract.** *This paper presents an information system for the automatic recognition of LIBRAS, grounded on two pillars: a configurable and extensible environment for the management of sign language processing experiments, based on the use of scientific workflows, and a set of modules for image and video processing, composed of image segmentation and classification methods.*

**Resumo.** *Este artigo apresenta um sistema de informação para o reconhecimento automático de LIBRAS, fundamentado em dois pilares: um ambiente configurável e extensível para o gerenciamento de experimentos de processamento de línguas de sinais, baseado no uso de workflows científicos e um conjunto de módulos desenvolvidos especificamente para o processamento de imagens e vídeos, composto por métodos para a segmentação e classificação de imagens.*

## 1. Introdução e Motivação

Nos últimos anos, diversos esforços têm sido feitos para aumentar a acessibilidade e incluir digitalmente pessoas que até então não tinham condições (financeiras, físicas ou educacionais) de utilizar computadores. Além disso, muitos tradutores automáticos de texto foram desenvolvidos, ampliando o acesso aos mais diferentes textos de variadas culturas, bem como, facilitando a comunicação entre pessoas que falam diferentes idiomas. Esses esforços já produziram muitos frutos, porém há ainda muito trabalho a ser feito.

Enquanto resultados muito satisfatórios já podem ser observados na tradução de textos, o reconhecimento automático e tradução de línguas gestuais ainda está em um estado inicial de desenvolvimento. Este artigo está focado no reconhecimento automático de LIBRAS (Língua Brasileira de Sinais), de forma a tentar facilitar a comunicação entre surdos dialogando em LIBRAS e ouvintes que não conheçam esta língua. A interface entre LIBRAS e português falado precisa de duas funcionalidades principais: (i) o reconhecimento automático de LIBRAS e sua conversão para português (textual ou oral, sintetizado automaticamente) e (ii) o reconhecimento de voz e sua tradução visual para LIBRAS (utilizando, por exemplo, animações).

Este artigo está focado no primeiro objetivo e combina o trabalho desenvolvido em dois projetos de pesquisa diferentes. O primeiro projeto tem por objetivo o desenvolvimento dos diversos módulos básicos para processamento de imagens e vídeos de forma a permitir o reconhecimento automático de LIBRAS e será apresentado na Subseção 3.1.

O segundo trata do desenvolvimento de um ambiente para gerenciamento de experimentos científicos, onde o usuário pode combinar atividades (módulos) já desenvolvidos de forma a montar, graficamente, o experimento desejado. O usuário também pode desenvolver novos módulos e executar os experimentos montados. Este ambiente será apresentado na Subseção 3.2. Neste artigo os resultados destes dois projetos são combinados de forma a se obter um sistema de informação extensível e configurável para o processamento automático de LIBRAS.

O restante deste artigo está organizado da seguinte forma. A Seção 2 sumariza os trabalhos correlatos referentes ao reconhecimento automático de diferentes línguas de sinais. A Seção 3 apresenta o sistema proposto, descrevendo todos os módulos já desenvolvidos. Por fim, a Seção 4 contém as conclusões e trabalhos futuros.

## 2. Conceitos Básicos e Trabalhos Correlatos

Diversos trabalhos relacionados ao reconhecimento de línguas de sinais foram publicados ao longo dos últimos anos. Nesta seção, uma breve apresentação de alguns dos principais e mais recentes trabalhos relacionados a este assunto serão apresentados.

Existem cinco parâmetros relacionados à realização de sinais em qualquer língua sinalizada: (a) configuração da mão (há 63 configurações diferentes); (b) movimento; (c) ponto de articulação (local do espaço de sinalização onde o sinal é realizado) [Stokoe 2005]; (d) orientação da palma da mão [Klima e Bellugi 1979]; e (e) expressões não manuais (por exemplo, faciais). Neste artigo, estamos utilizando apenas os parâmetros (a) e (d) para o reconhecimento de sinais, porém todos os parâmetros serão considerados em versões futuras do ambiente proposto.

É importante destacar que não é objetivo deste artigo apresentar a melhor técnica para o reconhecimento automático de língua de sinais, mas sim apresentar um sistema de informação para processamento automático de língua de sinais que permita facilmente a incorporação de extensões de forma a melhorar os resultados obtidos. De qualquer forma, algumas técnicas foram desenvolvidas neste trabalho e resultados interessantes foram obtidos, conforme será apresentado na Subseção 3.1.

Para vislumbrar o estado da arte sobre reconhecimento de línguas de sinais, foi feita uma pequena revisão sistemática sobre o assunto. Para manter esta revisão concisa, optou por considerar apenas os artigos mais relevantes de duas bases de artigos científicos, a IEEE (IEEEExplore<sup>1</sup>) e a ACM (ACM Digital Library<sup>2</sup>). As consultas foram feitas utilizando a expressão “*sign language recognition*” e os sete primeiros artigos resultantes de cada base foram selecionados e revisados (segundo o critério de relevância de cada uma). Por questões de espaço, este artigo apresenta apenas um resumo desta revisão.

A Tabela 1 contém um sumário das principais características dos trabalhos revisados. Todos os trabalhos revisados utilizam os parâmetros configuração de mão, orientação da palma e movimento, porém quase nenhum utiliza expressões não manuais. A segmentação dos vídeos é um processo extremamente complexo, a maioria dos trabalhos utilizou ambientes controlados para a aquisição dos vídeos, enquanto outros utilizaram luvas especiais (e assim não processaram vídeos e sim o sinais produzidos por

---

<sup>1</sup>ieeexplore.ieee.org

<sup>2</sup>dl.acm.org

estas luvas). Vale destacar que desta lista de correlatos o trabalho de Quan (2010) é uma proposta de sistema e apenas um trabalho tem como objetivo explícito o processamento de língua de sinais em tempo real [Huang et al. 2010].

**Tabela 1. Resumo da revisão bibliográfica**

| Referência                | Método de Aquisição Imagens                               | Parâmetros Utilizados                                                                  | Técnica de Reconhecimento                               |
|---------------------------|-----------------------------------------------------------|----------------------------------------------------------------------------------------|---------------------------------------------------------|
| [Maebatake et al. 2008]   | não consta                                                | configuração de mão, orientação da palma, posição e movimento                          | <i>Multi-Stream HMM</i>                                 |
| [Quan 2010]               | vídeos de mãos com fundos brancos                         | configuração de mão, orientação da palma e movimento                                   | <i>Support Vector Machine (SVM)</i>                     |
| [Huang et al. 2010]       | vídeos convencionais                                      | configuração de mão, posição, orientação da palma, movimento e expressões não manuais. | <i>Gabor Wavelet Transforms</i>                         |
| [Kumarage et al. 2011]    | vídeos obtidos por duas câmeras em um estúdio             | configuração de mão, posição, orientação da palma, movimento e expressões não manuais. | Distância entre Imagens                                 |
| [Bauer e Hienz 2000]      | vídeos gravados em um estúdio e luvas de cores diferentes | configuração de mão, orientação da palma, posição e movimento                          | <i>Hidden Markov Model (HMM)</i>                        |
| [Theodorakis et al. 2009] | vídeos gravados em um estúdio, sem luvas                  | configuração de mão, orientação da palma, e movimento                                  | <i>Product HMM</i>                                      |
| [Jiangqin et al. 1998]    | uso de luvas de dados <i>data gloves</i>                  | configuração de mão, orientação da palma, posição e movimento                          | <i>Hidden Markov Model (HMM)</i>                        |
| [Holt et al. 2011]        | vídeos 3D gravados em um estúdio, sem luvas               | configuração de mão, orientação da palma, posição e movimento                          | <i>Combined Discriminative Feature Detectors (CDFD)</i> |
| [Fang et al. 2003]        | uso de luvas de dados <i>data gloves</i>                  | configuração de mão, orientação da palma, posição e movimento                          | <i>Hierarchical Decision Trees</i>                      |
| [Zhang et al. 2004]       | vídeos gravados com luvas coloridas                       | configuração de mão, orientação da palma, posição e movimento                          | <i>Tied-Mixture Density HMM</i>                         |
| [Michael et al. 2009]     | vídeos gravados em um estúdio, sem luvas                  | configuração de mão, posição orientação da palma, movimento expressões não manuais.    | <i>Stacked Support Vector Machine</i>                   |
| [Li et al. 2010]          | uso de sensores EMG ( <i>electromyography</i> )           | configuração de mão, orientação da palma, e movimento                                  | <i>Multi-Stream HMM</i>                                 |
| [Zhang et al. 2005]       | uso de luvas de dados <i>data gloves</i>                  | configuração de mão, orientação da palma, posição e movimento                          | Bosted HMM                                              |
| [Caridakis et al. 2008]   | vídeos gravados em um estúdio, sem luvas                  | configuração de mão, orientação da palma, posição e movimento                          | <i>Self-Organizing Maps</i>                             |

### 3. Sistema de Informação Desenvolvido

Esta seção apresenta o sistema de informação desenvolvido que tem como objetivo ser um ambiente flexível para o processamento de imagens e vídeos em LIBRAS. Inicialmente, a Subseção 3.1 apresenta os módulos desenvolvidos especificamente para o processamento de imagens e vídeos, em especial aqueles relacionados a LIBRAS. Estes módulos possuem as atividades básicas que poderão ser combinadas e executadas na forma de workflows. A Subseção 3.2 apresenta de maneira sucinta o ambiente para gerenciamento e execução de workflows, incluindo um exemplo de workflow para processamento de uma imagem relacionada à LIBRAS.

### 3.1. Módulos para o Processamento de LIBRAS

Um conjunto de módulos para processamento de imagens e vídeos foram desenvolvidos de forma a transformar o ambiente de gerenciamento de workflows que será apresentado na Subseção 3.2 em um sistema para processamento de LIBRAS. Esta seção apresenta os principais módulos desenvolvidos, além de listar os módulos futuros.

Os módulos desenvolvidos foram classificados nas seguintes categorias: (a) métodos gerais; (b) segmentação de imagens; (c) extração de características das imagens; e (d) reconhecimento de LIBRAS. Cada uma dessas categorias será descrita nas próximas subseções. Cada função desenvolvida nos módulos para o processamento de LIBRAS pode ser executada e combinada com outras funções no ambiente para execução de workflows.

#### 3.1.1. Métodos Gerais

Esta seção apresenta diversos métodos gerais para o processamento de imagens e vídeos, que são utilizadas como ferramentas básicas de outras atividades. Cada uma das funções será brevemente descrita a seguir.

- **Remoção de fundo:** método para a remoção do fundo de uma imagem utilizando um limiar de cor;
- **Conversão de imagem:** conversão de imagem colorida para tons de cinza ou preto e branco;
- **Separação dos objetos de uma imagem:** separa os objetos contínuos de uma imagem em sub-imagens;
- **Concatenação dos pixels de duas imagens:** cria uma imagem com a união dos pixels pretos de duas imagens preto e branco;
- **Intersecção dos pixels de duas imagens:** cria uma imagem com a intersecção dos pixels pretos de duas imagens preto e branco;
- **Visualização de uma imagem:** cria um formulário (interface gráfica) para a visualização de uma dada imagem;
- **Redimensionar imagem:** amplia ou reduz uma imagem, de acordo com os parâmetros de entrada;
- **Reforçar contornos:** reforça os contornos dos objetos de uma dada imagem;
- **Aumentar contraste:** aumenta o contraste de uma dada imagem;
- **Suavizar imagem:** executa um filtro passa-baixa em todos os pixels de uma imagem, considerando a vizinhança de tamanho 8;
- **Processamento em lote de um diretório:** método utilizado para processar todas as imagens de um dado diretório de entrada. O usuário escolhe o método de processamento que será aplicado em cada imagem.

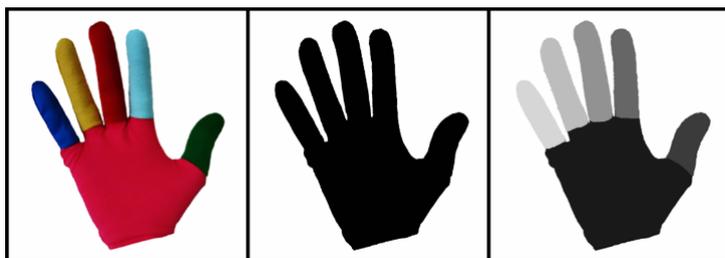
#### 3.1.2. Segmentação de Imagens

Neste trabalho a segmentação de imagens está sendo utilizada para identificar a mão dentro de uma imagem e também para separar a mão em seis regiões: os cinco dedos (mínimo, anelar, médio, indicador e polegar) e a palma. Em alguns trabalhos, a remoção do fundo e

a segmentação propriamente dita são feitas em uma única etapa, em outros são duas etapas diferentes. Os três métodos de segmentação que estão sendo utilizados neste trabalho executam essas duas tarefas de uma vez: remoção do fundo e segmentação.

Com a segmentação da mão de uma imagem é possível descobrir informações de dois (dos cinco) parâmetros relacionados a realização de sinais em línguas gestuais: *configuração da mão* e *orientação da palma*. Ao se analisar a mão em uma sequência de imagens oriunda de um vídeo é possível obter informações de um terceiro parâmetro, o *movimento da mão*. Para se obter informações sobre os demais parâmetros (*ponto de articulação* e *expressões não manuais*) é necessário identificar outros elementos na imagem e não somente a mão. Este artigo só lida com a segmentação da mão e reconhecimento de sinais a partir desta segmentação. A identificação e o processamento dos demais parâmetros serão tratados em trabalhos futuros.

A Figura 1 apresenta um exemplo da segmentação utilizada neste trabalho. A imagem a esquerda da figura apresenta uma foto com a luva colorida utilizada em um de nossos bancos de imagens. A imagem do centro apresenta a imagem segmentada com toda a mão, já a imagem da direita apresenta a imagem com a mão dividida em 6 segmentos (a palma e os cinco dedos).



**Figura 1. Imagem com fundo removido; segmentação de toda a mão e segmentação da mão em 6 partes**

Neste trabalho, foram desenvolvidos três estratégias semiautomáticas de segmentação. A primeira utiliza um mapa de cores que é utilizado para identificar se cada pixel da imagem é parte de um dos cinco dedos, da palma da mão ou do fundo da imagem. Este mapa é fornecido pelo usuário e pode conter várias cores diferentes para indicar cada segmento de interesse da imagem. É utilizada a distância euclidiana e um limiar para se classificar cada pixel da imagem como sendo parte de um dos dedos, palma ou fundo da imagem. Além do segmentador propriamente dito, foi desenvolvida uma ferramenta na qual o usuário recorta um pedaço de uma imagem e o sistema informa a cor média daquela região. Esta cor poderá ser inserida pelo usuário no mapa de cores.

A segunda estratégia utiliza a técnica de inteligência artificial de agrupamento (*clusterização*) para agrupar os pixels das imagens de acordo com suas cores, após esse agrupamento, o usuário poderá clicar em um grupo (*cluster*) e informar ao sistema a qual segmento esse grupo pertence. A Figura 2 apresenta uma imagem original e a imagem resultante da clusterização de seus pixels em 30 grupos, usando o algoritmo *Simple K-Means*.

A terceira estratégia utiliza como entrada a segmentação manual de algumas imagens. Baseada nesta segmentação, um algoritmo de classificação é executado para clas-



**Figura 2. Imagem original e imagem resultante da clusterização**

sificar os pixels das demais imagens como pertencentes ao fundo, a um dos dedos ou a palma da mão. Ao invés de implementar algoritmos de classificação, optamos por testar os diferentes algoritmos de classificação presentes no sistema Weka<sup>3</sup> [Hall et al. 2009] e com o resultado destes testes foi feita a escolha do algoritmo que seria usado.

Para o teste dos classificadores, foram segmentadas manualmente 26 imagens e extraídos 1400 pixels dessas imagens. Os algoritmos de classificação foram executados utilizando-se *10-fold cross-validation* e a média dos acertos é apresentada na Tabela 2.

**Tabela 2. Resultados da classificação de pixels**

| Classificador                    | Resultado |
|----------------------------------|-----------|
| meta.RotationForest              | 93,0714 % |
| functions.MultilayerPerceptron   | 92,7857 % |
| trees.FT                         | 92,4286 % |
| trees.LMT                        | 91,8571 % |
| functions.SimpleLogistic         | 91,3571 % |
| lazy.Ibk                         | 91,1429 % |
| rules.Nnge                       | 91,0714 % |
| functions.Logistic               | 91,0714 % |
| lazy.IB1                         | 91 %      |
| trees.RandomForest               | 90,1429 % |
| meta.Decorate                    | 90,0714 % |
| bayes.NaiveBayesMultinomial      | 89,9286 % |
| meta.END                         | 89,7857 % |
| meta.Bagging                     | 89,5714 % |
| meta.MultiClassClassifier        | 89,4286 % |
| trees.J48                        | 89,3571 % |
| meta.AttributeSelectedClassifier | 89,3571 % |
| meta.ClassificationViaRegression | 89,2857 % |
| trees.J48graft                   | 89,2143 % |
| rules.PART                       | 89,0714 % |

Pelo fato do Weka possuir uma API de fácil utilização, o algoritmo selecionado (o *RotationForest*) não foi reimplementado em nosso sistema, em vez disso, ele é executado pelo Weka. Foi desenvolvida uma ferramenta que converte a floresta resultante da execução do *RotationForest* em um sistema especialista que recebe as cores de um dado pixel e retorna a classe desse pixel. O segmentador chama este sistema especialista para classificar cada pixel de uma imagem.

### 3.1.3. Extração de Características

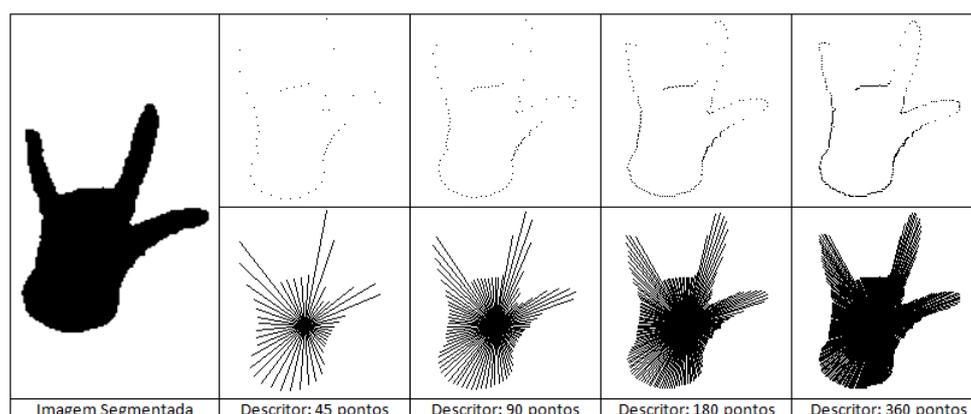
Tentar comparar imagens pixel a pixel é uma tarefa computacionalmente impraticável, pois mesmo fotos tiradas com baixa resolução possuem dezenas de milhares de pixels.

<sup>3</sup><http://www.cs.waikato.ac.nz/ml/weka/>

Desta forma, é interessante reduzir a dimensionalidade deste problema. Uma das maneiras mais comuns de se fazer isso é utilizar descritores de imagens, que nada mais são do que maneiras diferentes de se caracterizar uma dada imagem. Por exemplo, um descritor extremamente simples de uma imagem em preto e branco é a porcentagem de pixels pretos dessa imagem. Com esse descritor podemos comparar duas imagens considerando apenas suas porcentagens de pixels pretos, não tendo assim que comparar as imagens pixel a pixel. Obviamente, o uso de descritores pode gerar perda de informação, mesmo assim, eles são amplamente utilizados na recuperação de imagens por conteúdo.

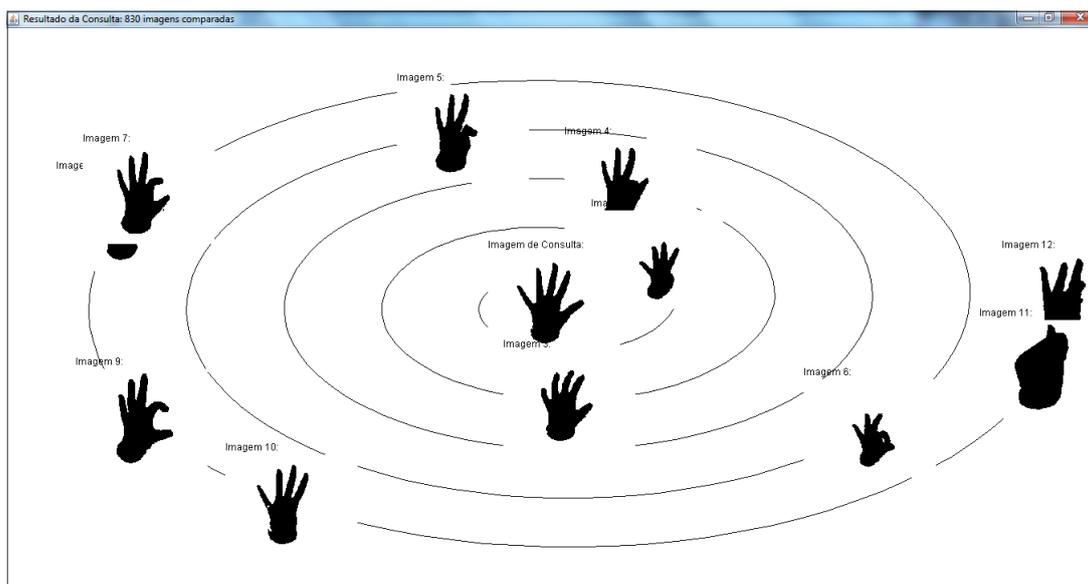
Para se obter as informações de um dado descritor de imagens são desenvolvidos extratores de característica (um para cada descritor). Em um trabalho anterior, foram desenvolvidos onze extratores de características de imagens [Valença 2010]. Estes extratores são de uso geral e estão disponíveis em nosso sistema. Porém, nesta seção será apresentado apenas um destes onze descritores, pois é aquele mais adequado para o processamento de imagens relacionadas a LIBRAS. Além disso, dois outros extratores foram desenvolvidos especificamente para a extração de características de LIBRAS, conforme será visto a seguir.

O *extrator de forma* implementado neste trabalho consiste em, dada uma imagem com apenas um objeto em preto e branco, identificar o centro de gravidade desta imagem e a partir dele calcular o tamanho dos  $n$  raios que saem deste centro e atingem a borda da imagem. Por exemplo, se  $n = 36$  então, a cada 10 graus, será calculado o valor do raio entre o centro de gravidade da imagem e a borda externa da imagem. Por fim, esses raios são normalizados para valerem de 0 a 1. Este extrator é interessante por reduzir as informações de uma imagem de, por exemplo, 307.200 pixels (imagem de 480 x 640 pontos) para  $n$  pontos. Além disso, esse extrator consegue recuperar imagens de maneira invariante ao seu tamanho (já que os raios são normalizados) e invariante a rotação (ao se utilizar uma distância não euclidiana). A Figura 3 apresenta exemplos desse extrator sendo aplicado a uma imagem para diferentes valores de  $n$ .



**Figura 3. Caraterística extraída utilizando o descritor de forma**

As imagens podem ser comparadas utilizando os extratores/descriptores implementados. A Figura 4 apresenta uma ferramenta que foi desenvolvida para comparar as imagens de um diretório com uma imagem de consulta. Imagens mais próximas ao centro da tela indicam imagens mais próximas da imagem de consulta (proximidade calculada de acordo com o extrator de forma apresentado).



**Figura 4. Resultado de busca utilizando o descritor de forma**

Outros dois extratores de características foram desenvolvidos, especificamente para lidar com imagens relacionadas à LIBRAS. Estes extratores recebem como entrada imagens segmentadas em seis regiões: palma da mão e cada um dos dedos (conforme exemplo de segmentação apresentado na Figura 1). O primeiro extrator calcula a *área proporcional de cada segmento*, por exemplo, em uma dada imagem segmentada de uma mão a palma corresponde a 65% da imagem, o indicador a 10% e assim por diante. Assim sendo, este extrator descreve a imagem em 6 valores reais. O segundo extrator calcula a *posição relativa de cada segmento* da mão em relação ao centro de gravidade da mão, ou seja, em um plano bidimensional, o centro de gravidade é colocado na coordenada (0;0) e a coordenada relativa do centro de gravidade de cada um dos segmentos é calculada e normalizada para que todas as coordenadas tenham seus valores entre -1 e +1. Desta forma este extrator caracteriza cada imagem em 12 valores (6 coordenadas bidimensionais).

### 3.1.4. Reconhecimento de Libras

Conforme apresentado, este artigo utiliza apenas dois parâmetros da realização de sinais para o reconhecimento de LIBRAS: a configuração da mão e a orientação da palma. Estes parâmetros não são utilizados diretamente, mas sim através dos extratores apresentados na seção anterior.

O reconhecimento de LIBRAS neste artigo é realizado através da medida da distância entre imagens já classificadas e as novas imagens, e o sinal atribuído a uma nova imagem corresponde ao sinal da imagem classificada mais “parecida” com a nova imagem, de acordo com o descritor utilizado.

Para se identificar a letra do alfabeto que está sendo sinalizada (26 possibilidades), nosso sistema obteve a taxa de classificação correta (verdadeiros positivos) apresentada na Tabela 3. Nesta avaliação dos extratores foi criado um novo extrator que descreve as imagens concatenando as informações produzidas pelos dois extratores específicos para

LIBRAS desenvolvidos neste trabalho. É importante observar nesta tabela que os extratores específicos para LIBRAS, por mais simples que sejam, apresentaram resultados bastante interessantes, sendo que ambos obtiveram resultados melhores do que o extrator de forma para uso geral em processamento de imagens.

**Tabela 3. Resultados da classificação de imagens**

| <b>Extrator</b>                    | <b>Resultado</b> |
|------------------------------------|------------------|
| Extrator de forma                  | 61,64%           |
| Área proporcional de cada segmento | 65,75%           |
| Posição relativa de cada segmento  | 83,56%           |
| Combinação dos extratores 2 e 3    | 87,67%           |

### 3.2. Ambiente de Workflows

O ambiente para gerenciamento de experimentos científicos na forma de workflows utilizado neste trabalho é uma extensão de uma ferramenta cujo desenvolvimento iniciou-se há cerca de dez anos, chamada originalmente de WOODSS [Medeiros et al. 2005]. A versão atual da ferramenta conta com um editor de workflows onde cada atividade pode ser um serviço Web, um método na linguagem Java ou um aplicativo local. A interface gráfica permite a criação, edição e execução de workflows (experimentos científicos), bem como a conversão de um workflow em um código executável, que então poderá ser executado fora do ambiente desenvolvido. Detalhes sobre a versão atual do ambiente podem ser encontradas em [Digiampietri et al. 2011]. Por ser de uso geral, o ambiente está sendo utilizado em diferentes áreas de conhecimento e já foi aplicado no processamento de imagens biológicas [Andrioli et al. 2012].

Um workflow é composto por *atividades* (retângulos na representação gráfica), sendo que cada atividade pode possuir um conjunto de entradas e saídas; *fluxos de dados* (setas pretas) que ligam uma saída de uma atividade à entrada de outra; e *fluxos de controle* (setas cinzas) que indicam que uma atividade só poderá ser iniciada após a execução de outra.

A Figura 5 contém a cópia de tela do ambiente para a execução de workflows. Nesta tela há um workflow para o processamento inicial de uma imagem. O workflow desta figura é composto por seis atividades: a atividade mais a esquerda recebe o nome do arquivo de uma imagem e converte essa imagem para um formato interno utilizado pelo sistema; a atividade no meio e acima do workflow é responsável pela segmentação da imagem de entrada (segmentação utilizando a técnica mapa de cores, apresentada na seção anterior); a atividade acima e a direita é responsável por filtrar a imagem segmentada. As três atividades mais abaixo da figura são responsáveis por exibir o estado atual de processamento da imagem de entrada (a mais a esquerda exibe a imagem original, a do meio exibe a imagem segmentada e a da direita exibe a imagem segmentada após a execução de um filtro). As setas pretas indicam o fluxo de dados entre as atividades. As entradas das atividades que não possuem nenhuma seta chegando nelas devem ser preenchidas pelo usuário. Neste exemplo, o usuário preencheu o nome do arquivo que foi processado pelo workflow, bem como os títulos que serão exibidos nas janelas de resultado de exibição de imagens.

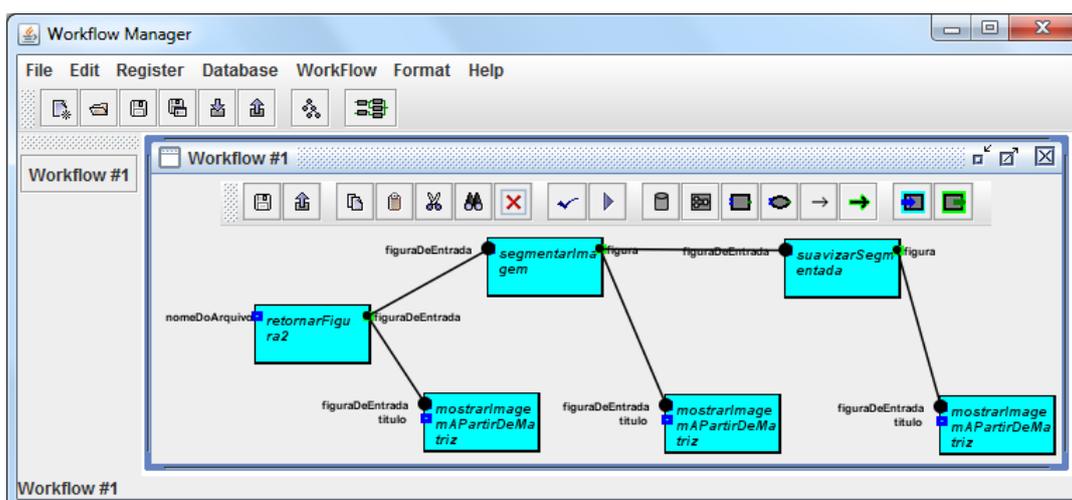


Figura 5. Tela do Gerenciador de Workflows

O resultado da execução do workflow da Figura 5 é composto por três “janelas” contendo a imagem original, imagem segmentada e imagem segmentada e filtrada, essas três janelas são apresentadas na Figura 6.

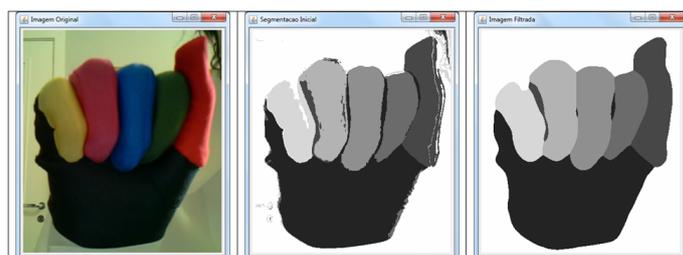


Figura 6. Resultados da Execução do Workflow da Figura 5

#### 4. Conclusões e Trabalhos Futuros

Comunicação e compartilhamento de conhecimento são peças chave para uma sociedade que segue em direção à globalização e tenta prover diferentes formas de inclusão aos seus cidadãos. Entre os desafios ligados à comunicação, destacamos a necessidade de melhores interfaces entre ouvintes e surdos. Este artigo descreveu alguns esforços para se permitir o reconhecimento automático de LIBRAS, tarefa que ainda hoje possui diversos desafios.

Este artigo apresentou um ambiente flexível para o reconhecimento de LIBRAS, composto por um sistema de gerenciamento de experimentos na forma de workflows, no qual o usuário pode adicionar novas atividades ou combinar as atividades existentes de maneira a obter os resultados desejados.

Além disso, diversos módulos para o processamento de imagens e vídeos e, especificamente, para o processamento de LIBRAS foram desenvolvidos e estão disponíveis no sistema.

Como trabalho futuro pretende-se estender o sistema desenvolvido de forma a tratar os cinco parâmetros relacionados à realização de sinais em LIBRAS. Além disso,

pretende-se desenvolver segmentadores de imagens e extratores de características mais precisos.

## Agradecimentos

O trabalho apresentado neste artigo foi parcialmente financiado pela FAPESP (Projeto Jovem Pesquisador processo 2009/10413-5 e Bolsa de Iniciação Científica processo 2011/07968-5), pelo CNPq (Bolsa de Iniciação Científica) e pelo Programa de Educação Tutorial (MEC/SESu).

Agradecemos também à ex-aluna do grupo, Ana Carolina Valença que implementou a versão original dos extratores de características de imagens, à professora Sarajane Marques Peres que forneceu o primeiro conjunto de imagens para testes e à professora de LIBRAS Maria Carolina Casati que gravou os novos vídeos utilizados nesta pesquisa, bem como ajudou na anotação das imagens e validação do sistema.

## Referências

- Andrioli, L. P., Digiampietri, L. A., de Barros, L. P., e Machado-Lima, A. (2012). Hucklebein is part of a combinatorial repression code in the anterior blastoderm. *Developmental Biology*, 361(1):177 – 185.
- Bauer, B. e Hienz, H. (2000). Relevant features for video-based continuous sign language recognition. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 440 –445.
- Caridakis, G., Diamanti, O., Karpouzis, K., e Maragos, P. (2008). Automatic sign language recognition: vision based feature extraction and probabilistic recognition scheme from multiple cues. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, PETRA '08*, pages 89:1–89:8, New York, NY, USA. ACM.
- Digiampietri, L. A., Pérez-Alcázar, J. J., e Freitas, R. S., Araújo, J. C., Éric H. Ostroski, e Santiago, C. R. N. (2011). Uso de planejamento em inteligência artificial para o desenvolvimento automático de software. In *Autonomous Software Systems (AutoSoft 2011)*, page 10.
- Fang, G., Gao, W., e Zhao, D. (2003). Large vocabulary sign language recognition based on hierarchical decision trees. In *Proceedings of the 5th international conference on Multimodal interfaces, ICMI '03*, pages 125–131, New York, NY, USA. ACM.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., e Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.
- Holt, G. A. T., Doorn, A. J. V., Reinders, M. J. T., Hendriks, E. A., e Ridder, H. D. (2011). Human-inspired search for redundancy in automatic sign language recognition. *ACM Trans. Appl. Percept.*, 8:15:1–15:15.
- Huang, Z., Jiang, D., e Zhao, W. (2010). Study of sign language recognition based on gabor wavelet transforms. In *Computer Design and Applications (ICDDA), 2010 International Conference on*, volume 1, pages V1–151 –V1–154.
- Jiangqin, W., wen, G., yibo, S., wei, L., e bo, P. (1998). A simple sign language recognition system based on data glove. In *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on*, volume 2, pages 1257 –1260 vol.2.

- Klima, E. e Bellugi, U. (1979). *The signs of language*. Cambridge University Press.
- Kumarage, D., Fernando, S., Fernando, P., Madushanka, D., e Samarasinghe, R. (2011). Real-time sign language gesture recognition using still-image comparison and motion recognition. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pages 169–174.
- Li, Y., Chen, X., Tian, J., Zhang, X., Wang, K., e Yang, J. (2010). Automatic recognition of sign language subwords based on portable accelerometer and emg sensors. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI '10*, pages 17:1–17:7, New York, NY, USA. ACM.
- Maebatake, M., Suzuki, I., Nishida, M., Horiuchi, Y., e Kuroiwa, S. (2008). Sign language recognition based on position and movement using multi-stream hmm. In *Proceedings of the 2008 Second International Symposium on Universal Communication, ISUC '08*, pages 478–481, Washington, DC, USA. IEEE Computer Society.
- Medeiros, C., Perez-Alcazar, J., Digiampietri, L., Pastorello, G., Santanche, A., Torres, R., Madeira, E., e Bacarin, E. (2005). WOODSS and the Web: Annotating and Reusing Scientific Workflows. *ACM SIGMOD Record*, 34(3):18–23.
- Michael, N., Metaxas, D., e Neidle, C. (2009). Spatial and temporal pyramids for grammatical expression recognition of american sign language. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility, Assets '09*, pages 75–82, New York, NY, USA. ACM.
- Quan, Y. (2010). Chinese sign language recognition based on video sequence appearance modeling. In *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*, pages 1537–1542.
- Stokoe, W. C. (2005). Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of Deaf Studies and Deaf Education*, 10(1):3–37.
- Theodorakis, S., Katsamanis, A., e Maragos, P. (2009). Product-hmms for automatic sign language recognition. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1601–1604.
- Valença, A. C. (2010). Uso de mineração de dados para aperfeiçoar sistemas de recuperação de imagens por conteúdo. Technical report, Universidade de São Paulo.
- Zhang, L.-G., Chen, X., Wang, C., Chen, Y., e Gao, W. (2005). Recognition of sign language subwords based on boosted hidden markov models. In *Proceedings of the 7th international conference on Multimodal interfaces, ICMI '05*, pages 282–287, New York, NY, USA. ACM.
- Zhang, L.-G., Chen, Y., Fang, G., Chen, X., e Gao, W. (2004). A vision-based sign language recognition system using tied-mixture density hmm. In *Proceedings of the 6th international conference on Multimodal interfaces, ICMI '04*, pages 198–204, New York, NY, USA. ACM.