

ANÁLISE DE TÉCNICAS DE EXTRAÇÃO DE CARACTERÍSTICAS PARA VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS

Trabalho de Conclusão de Curso
Engenharia da Computação

Gearlles Viana Ferreira

Orientador: Prof. Dr. Byron Leite Dantas Bezerra

Gearlles Viana Ferreira

ANÁLISE DE TÉCNICAS DE EXTRAÇÃO DE CARACTERÍSTICAS PARA VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco

Universidade de Pernambuco

Escola Politécnica de Pernambuco

Graduação em Engenharia de Computação

Orientador: Prof. Dr. Byron Leite Dantas Bezerra

Recife - PE, Brasil

18 de junho de 2015

Declaro que revisei o Trabalho de Conclusão de Curso sob o título “*ANÁLISE DE TÉCNICAS DE EXTRAÇÃO DE CARACTERÍSTICAS PARA VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS*”, de autoria de *Gearlles Viana Ferreira*, e que estou de acordo com a entrega do mesmo.

Recife, _____ / _____ / _____

Prof. Dr. Byron Leite Dantas Bezerra
Orientador

À minha família.

Agradecimentos

Agradeço primeiramente à Deus por ter me dado forças e oportunidades para alcançar vários de meus objetivos.

Agradeço à minha família, por sempre estarem presentes em todas etapas da minha vida, por terem me dado oportunidades, apoio, conselhos e confiança. Agradeço o amor que vocês me dão todos os dias.

Agradeço aos meus amigos Nestor, Felipe, Arthur e Milla que, de todos esses anos de amizade, sempre estiveram presentes nas alegrias e sufocos na universidade. Sem vocês nada disso seria possível.

Agradeço a POLI-UPE por todas oportunidades concedidas. Especialmente, agradeço meu orientador Prof. Byron e o Prof. Bruno por terem confiado em mim, terem acompanhado minha carreira na universidade, desde os primeiros períodos de monitoria e disciplinas até meu trabalho de conclusão de curso.

Agradeço meus amigos de trabalho Saulo, Antonio, Robson, Max, Everton, Denis e Diego por todo apoio e grandes momentos. Por toda sabedoria compartilhada e pela compreensão nos momentos mais difíceis.

Resumo

A verificação automática de assinaturas *off-line*, também conhecida como estática, tem como desafios fatores como variação entre assinaturas do mesmo autor, a semelhança de escrita entre autores diferentes e também falsificações, principalmente as habilidosas. Na literatura, diversas abordagens utilizam o processo de extração de características baseado no contorno. A abordagem proposta nesse trabalho compara técnicas de extração de características utilizadas para reconhecimento de gestos também baseadas em contorno. No modelo, duas classes são consideradas: autoria ou não-autoria e as seguintes etapas fazem parte da abordagem: aquisição de dados, pré-processamento, remoção de *background*, identificação do contorno, extração de características, minimização do modelo, extração dos pontos mais significativos e finalmente o processo de decisão.

Palavras-chave: Verificação de Assinaturas, Análise de Documentos Questionados, Reconhecimento de padrões.

Abstract

Automatic off-line signature verification, also known as Automatic static signature verification, is still a challenge due mainly to factors such as the variability presented by samples from the same writer, the similarity among samples from different writers and also forgeries types, mainly the skillful forgery. In the literature, various approaches use contour-based features in the features extracion process. The approach presented in this work compares features extraction techniques that were used to gesture recognition, also using contour-based features. In the model, two classes are considered: genuine and forgery. The following steps are part of the approach: data acquisition, pre-processing, background removal, contour identification, features extraction, model minimization, most-significant points extracion and finally, the decision process.

Keywords: Signature Verification, Questioned Document Analysis.

Lista de ilustrações

Figura 1 – (a) Assinatura original, (b) Sobreposição de várias assinaturas do mesmo autor.	13
Figura 2 – Informações como pressão, coordenada em função do tempo e velocidade podem ser obtidas através de uma mesa digitalizadora. Fonte: IBM (2015).	16
Figura 3 – Filtro de mediana aplicado a um pixel. O valor do pixel 10 é substituído pela mediana de todos seus vizinhos (inclusive ele mesmo), cujo valor é 4. Fonte: Kamya (2015).	16
Figura 4 – Função gaussiana em duas dimensões. Fonte: Fisher (2015).	17
Figura 5 – Imagem original com ruído (esquerda), aplicação do filtro gaussiano (centro) e aplicação do filtro de mediana (direita). Fonte: Gouillart (2015).	17
Figura 6 – Hitogramas de imagens. O histograma pode ser utilizado para inferir certas características da imagem. A imagem mais escura (esquerda) possui um histograma mais próximo da origem, enquanto que o histograma da imagem mais clara (direita) possui valores mais altos. Fonte: Usernet (2015).	18
Figura 7 – Modelo de um sistema de Reconhecimento de Padrões. A extração de características pode ser entendida como o mapeamento de um ponto x de um espaço m -dimensional para um q -dimensional. Da mesma forma, a classificação, de um espaço q -dimensional, para um r -dimensional. Fonte: SANTOS (2004).	19
Figura 8 – Diagrama hierárquico de classificação dos tipos de abordagens de verificação de assinaturas. Fonte: Autoria própria.	22
Figura 9 – Amostra de assinatura inclinada em relação à linha base.	23
Figura 10 – Amostra de assinaturas com inclinação axial. Fonte: SANTOS (2004).	24
Figura 11 – Amostra de assinaturas com diferentes calibres.	24
Figura 12 – Tela inicial do DSSV.	25
Figura 13 – Tela de verificação de assinaturas do DSSV.	25
Figura 14 – Vizinhança norte, sul, leste e oeste do ponto P . Fonte: Barros et al. (2013b).	27
Figura 15 – (a) A imagem de entrada, (b) Imagem integral calculada através da imagem (a). Cada pixel representa a soma dos pixels vizinhos acima e esquerda. Fonte: Mathworks (2015).	28
Figura 16 – (a) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{yy}(x, \sigma))$, (b) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{xy}(x, \sigma))$, (c) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{xx}(x, \sigma))$. Fonte: OpenCV (2015a).	29

Figura 17 – Resultado da aplicação do Filtro Laplaciano Gaussiana nas imagens da Figura 16. Fonte: OpenCV (2015a).	29
Figura 18 – Relação entre Octave e escala para aplicação dos filtros. Fonte: Bay et al. (2008).	30
Figura 19 – Amostra de imagens presentes na base italiana da competição SigWI- Comp2015.	33
Figura 20 – Módulo que implementa as técnicas de extração de características LCS, SURF, CLCS e CSURF.	34
Figura 21 – Pontos selecionados pelas técnicas. Imagem original (superior), pontos selecionados pelo LCS (meio), pontos selecionados pelo CLCS (inferior).	35

Lista de tabelas

Tabela 1 – Comparação dos resultados utilizando diferentes técnicas de extração de características.	35
Tabela 2 – Comparação de tempo de processamento das técnicas de extração de características.	35

Lista de abreviaturas e siglas

PDI	Processamento Digital de Imagens
LCS	<i>Local Countour Sequence</i>
CLCS	<i>Convexity Local Countour Sequence</i>
SIFT	<i>Scale-invariant Feature Transform</i>
SURF	<i>Speed Up Robust Features</i>
ICDAR	<i>International Conference on Document Analysis and Recognition</i>

Sumário

1	INTRODUÇÃO	13
1.1	Qualificação do Problema	13
1.2	Objetivos	13
1.2.1	Objetivos Gerais	14
1.2.2	Objetivos Específicos	14
1.3	Estrutura da Monografia	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Introdução	15
2.2	Processamento Digital de Imagens	15
2.2.1	Aquisição da imagem	15
2.2.2	Pré-processamento de Imagens	16
2.2.2.1	Filtros de suavização	16
2.2.3	Segmentação de Imagens	17
2.2.3.1	Limiarização	18
2.3	Reconhecimento de Padrões	19
3	VERIFICAÇÃO DE ASSINATURA	21
3.1	Introdução	21
3.2	Tipos de abordagens na Verificação de Assinaturas	21
3.3	Métodos de Verificação	22
3.4	Características de assinaturas	23
3.4.1	Inclinação em relação à linha base	23
3.4.2	Inclinação axial	24
3.4.3	Calibre	24
3.5	Estudo de caso: DSSV	24
4	EXTRAÇÃO DE CARACTERÍSTICAS	26
4.1	<i>Local Contour Sequence (LCS)</i>	26
4.2	<i>Speeded Up Robust Features (SURF)</i>	27
4.3	Abordagem Convexa	31
4.3.1	<i>Convexity Local Countour Sequence (CLCS)</i>	31
4.3.2	<i>Convexty Speeded Up Robust Features (CSURF)</i>	32
5	EXPERIMENTOS E RESULTADOS	33
5.1	Base de dados	33

5.2	Metodologia e resultados	34
6	CONSIDERAÇÕES FINAIS	36
6.1	Conclusões	36
6.2	Trabalhos Futuros	36
	REFERÊNCIAS	37
	APÊNDICE A – APÊNDICE A: CÓDIGO PRINCIPAL	40

1 Introdução

1.1 Qualificação do Problema

No contexto do Aprendizado de Máquina, Classificação é um processo que procura identificar a qual classe uma nova observação pertence, baseado no conhecimento atual. O procedimento de classificação é um método formal para realizar tais julgamentos repetidamente em novas situações (MICHIE; SPIEGELHALTER; TAYLOR, 1994).

A verificação automática de assinaturas manuscritas é o ato de classificar uma assinatura de uma pessoa em relação a sua autenticidade. As classes podem ser, por exemplo, genuínas e forjadas. Tal processo é necessário em diferentes campos do nosso dia-a-dia, desde pagamento de cheques, transações comerciais e processos judiciais. Esse processo é utilizado para verificar a identidade de uma pessoa através de características presentes em sua assinatura. Dessa forma, a qualidade da classificação depende diretamente da qualidade das características extraídas da assinatura.

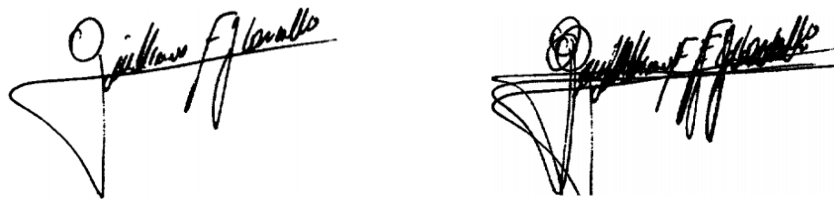


Figura 1 – (a) Assinatura original, (b) Sobreposição de várias assinaturas do mesmo autor.

Entretanto, uma pessoa nunca reproduz a assinatura da mesma forma. É natural não existir homogeneidade nas diversas assinaturas realizadas por um ser humano, no que se diz respeito a tamanho e distribuição dos caracteres (RIBEIRO; VASCONCELOS, 1999), como observado na Figura 1. Há dois tipos de variação na assinatura que são considerados na literatura: variação de curto-prazo, que depende da condição psicológica do assinante; e a variação de longo-prazo, que depende de variações físicas do assinante (membros físicos) (PIRLO; IMPEDOVO, 2013).

Dessa forma, dadas as variações presentes nas assinaturas de uma pessoa, é de grande importância que as características utilizadas sejam robustas o suficiente para preservar as informações intrínsecas de cada autor.

1.2 Objetivos

Nesta sessão são expostos os objetivos gerais e em seguida os objetivos específicos.

1.2.1 Objetivos Gerais

O trabalho tem como objetivo realizar a comparação de técnicas de extração de características (LCS, SURF, CLCS e CSURF) aplicadas ao contexto da verificação de assinaturas manuscritas. Além disso, objetiva-se comparar os resultados obtidos entre as técnicas utilizadas.

1.2.2 Objetivos Específicos

Este trabalho tem os seguintes aspectos como objetivos específicos:

- Implementar técnicas de extração de características;
- Integrar as técnicas de extração de características implementadas em um sistema de verificação de assinaturas;
- Análise e comparação dos resultados obtidos no sistema de verificação de assinatura.

1.3 Estrutura da Monografia

Este documento está estruturado em cinco capítulos. O Capítulo 2 apresenta os fundamentos e técnicas de Processamento Digital de Imagens, como Aquisição de imagens, Pré-processamento e Segmentação. O Capítulo 3 aborda a Verificação de Assinatura, expondo conceitos, tipos e métodos do tema. O Capítulo 4 apresenta as técnicas de extração de características baseadas em Reconhecimento de Gestos que foram modificadas para uso na verificação de assinaturas. O Capítulo 5 expõe os experimentos realizados, a metodologia que o trabalho segue e os resultados obtidos. Por fim, o Capítulo 6 apresenta as conclusões acerca da análise de técnicas de extração de características, assim como extensões que podem ser desenvolvidas desse trabalho no futuro.

2 Fundamentação teórica

2.1 Introdução

Neste Capítulo são apresentados alguns fundamentos e técnicas de processamento digital de imagens, dividido em três etapas principais: a aquisição, o pré-processamento e a segmentação de imagens.

2.2 Processamento Digital de Imagens

O termo Processamento Digital de Imagem (PDI) se refere ao processamento de uma imagem digital onde a entrada e a saída do sistema são imagens. PDI tem como objetivo principal realizar uma melhoria na imagem a fim de facilitar a extração de informações e outros processamentos. Quando utilizado junto com técnicas de Verificação de Assinatura, técnicas de PDI tem como função aguçar o contorno da escrita e encontrar contorno de toda assinatura, por exemplo. Para as técnicas de extração de características discutidas nesse trabalho, o principal objetivo do processamento de imagens é análise do contorno.

2.2.1 Aquisição da imagem

As imagens são geradas através de três elementos principais: uma fonte de luz, o objeto observado e o observador. O objeto observado utiliza a luz proveniente da fonte de iluminação e parte dela é refletida e outra parte absorvida. A parte da luz refletida é captada pelo observador.

Para a criação de uma imagem digital, é necessário realizar a amostragem e quantização da informação obtida pelos sensores, pois estão em formato de onda contínua. A imagem analógica pode ser representada por uma função contínua em x e y . O processo de amostragem capta informação das coordenadas x e y , enquanto que o processo de quantização capta informação relacionada à intensidade da função contínua.

Como resultado, uma matriz bidimensional é obtida. O valor de uma posição da matriz representa a intensidade naquele ponto, termo conhecido como *pixel*. Matematicamente uma imagem pode ser definida por $f(x, y)$ onde x e y são as coordenadas e o valor de f é a intensidade.

No contexto da verificação de assinatura, dispositivos como *scanner*, câmera e mesa digitalizadora são os mais utilizados. Os dois primeiros dispositivos são utilizados para verificação *off-line*, enquanto que o último para verificação *on-line*, já que características

como pressão, coordenada em função do tempo e velocidade podem ser analisadas. A descrição da verificação *on-line* e *off-line* são discutidas no Capítulo 3.



(a) Amostra de mesa digitalizadora.



(b) Assinaturas correspondentes.

Figura 2 – Informações como pressão, coordenada em função do tempo e velocidade podem ser obtidas através de uma mesa digitalizadora. Fonte: IBM (2015).

2.2.2 Pré-processamento de Imagens

O pré-processamento de imagens tem como objetivos principais a correção de distorções geométricas na imagem e a remoção de ruídos.

2.2.2.1 Filtros de suavização

Os filtros de suavização, também conhecidos como filtros de média e filtros passa-baixa, são filtros que suavizam a imagem através do cálculo da média dos *pixels* vizinhos (GONZALEZ; WOODS, 2006). São exemplos de filtros de suavização: o filtro de mediana e filtro Gaussiano.

O filtro de mediana é um filtro espacial não-linear onde o valor do pixel analisado é substituído pela mediana dos pixels vizinhos na imagem que está sob a máscara. Da estatística, a mediana é o valor que separa uma série na metade. Como exemplo, a Figura 3 ilustra filtro de mediana.

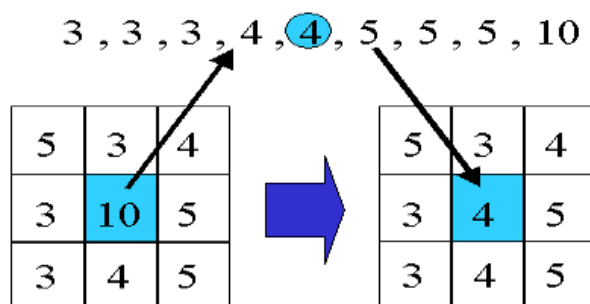


Figura 3 – Filtro de mediana aplicado a um pixel. O valor do pixel 10 é substituído pela mediana de todos seus vizinhos (inclusive ele mesmo), cujo valor é 4. Fonte: Kamya (2015).

O filtro Gaussiano também pode ser usado como filtro passa-baixa, utilizando a função Gaussiana para obtenção dos valores da máscara que será aplicada na imagem.

Utilizando σ como desvio padrão, a função Gaussiana é definida na Equação 2.1.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

Considerando uma imagem é formada por pixels, ou seja, é uma função discreta, a função G da Equação 2.1 pode ser aproximada para sua forma discreta de distribuição. A Figura 4 mostra a função gaussiana e sua aproximação para uma máscara 4×4 .

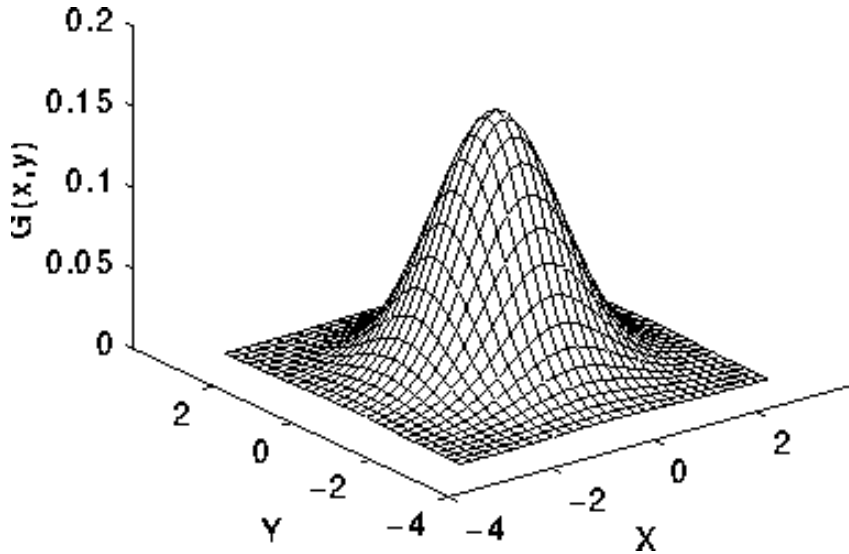


Figura 4 – Função gaussiana em duas dimensões. Fonte: Fisher (2015).

Os filtros de suavização podem ser utilizados para remoção ou redução de ruídos de uma imagem. Convertendo a imagem para o domínio da frequência, é observado que alguns tipos de ruídos ficam distribuídos no espectro de frequências mais baixas (GONZALEZ; WOODS, 2006). Por esse fato, os filtros de suavização tem ótimos resultados para esses tipos de problemas, como observado na Figura 5.



Figura 5 – Imagem original com ruído (esquerda), aplicação do filtro gaussiano (centro) e aplicação do filtro de mediana (direita). Fonte: Gouillart (2015).

2.2.3 Segmentação de Imagens

A segmentação de imagem é o processo de divisão de uma imagem em múltiplas partes. É usualmente utilizada para identificar objetos ou outras informações relevantes em imagens digitais.

2.2.3.1 Limiarização

Utilizando a análise do histograma, a limiarização tem como objetivo escolher os valores limites para os pixels. O histograma é a representação gráfica da distribuição das frequências de uma imagem em determinadas classes. No contexto do processamento digital de imagens, o histograma indica a frequência de cada valor de pixel em uma imagem e pode ser utilizado para inferir certas características da imagem, como observado na Figura 6. O eixo x é a frequência e o eixo y é o valor do pixel.



Figura 6 – Histogramas de imagens. O histograma pode ser utilizado para inferir certas características da imagem. A imagem mais escura (esquerda) possui um histograma mais próximo da origem, enquanto que o histograma da imagem mais clara (direita) possui valores mais altos. Fonte: [Usernet \(2015\)](#).

A limiarização é um método específico que consiste em dividir a imagem em duas classes: o fundo e a região de interesse (objeto à ser analisado). Essa divisão é feita a partir de um determinado limiar L que separa o histograma em duas partes. Para cada *pixel* da imagem é realizada a operação descrita na Equação 2.2.

$$g(x, y) = \begin{cases} 255, & g(x, y) < L \\ 0, & \text{caso contrário} \end{cases} \quad (2.2)$$

Quando um limiar constante L é aplicado em toda a imagem, o processo é chamado de limiarização global. Caso esse limiar varie, o processo é chamado de limiarização variada. Quando este limiar é local, o processo só é realizado dependendo das propriedades dos vizinhos do píxel. Caso o limiar varie de acordo com o valor do pixel, esta limiarização variada é também chamada de adaptativa ou dinâmica. As terminologias destes termos não são universais e é comum encontrá-las trocadas na literatura ([GONZALEZ; WOODS, 2006](#)).

2.3 Reconhecimento de Padrões

Um padrão é uma descrição de um objeto que pode ser um conjunto de medidas ou observações. São normalmente representadas através de um vetor ou notação de matriz (ABBAS; CIESIELSKI, 1995). No processo de verificação de assinaturas, uma assinatura é um exemplo de padrão que pode ser representado por uma matriz de pixels.

O reconhecimento de padrões pode ser definido como a categorização de dados de entrada em classes identificáveis, via extração de características significantes ou atributos de detalhes relevantes. Conseqüentemente o objetivo fundamental do reconhecimento de padrões é a classificação. Um sistema de reconhecimento de padrões básico divide-se em duas fases. A primeira é a extração das características e a segunda a classificação (ABBAS; CIESIELSKI, 1995; SANTOS, 2004).

Características são quaisquer medidas extraíveis de um padrão que podem contribuir para a classificação, sendo que as mesmas podem ser representadas por valores contínuos ou discretos.

Após a extração de características, ocorre o processo de treinamento, durante o qual apresenta-se repetidamente ao classificador um conjunto de padrões de entrada juntamente com a categoria à qual cada padrão apresentado pertence. Posteriormente, na etapa de testes apresenta-se um novo padrão que pertence à mesma população de padrões utilizada no treinamento. O classificador deve ser capaz de identificar a classe daquele padrão particular, pela informação que ele extraiu dos dados de treinamento.

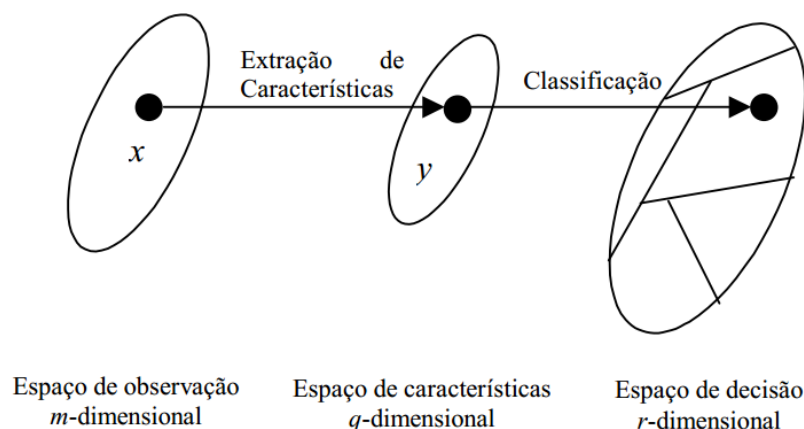


Figura 7 – Modelo de um sistema de Reconhecimento de Padrões. A extração de características pode ser entendida como o mapeamento de um ponto x de um espaço m -dimensional para um q -dimensional. Da mesma forma, a classificação, de um espaço q -dimensional, para um r -dimensional. Fonte: SANTOS (2004).

De acordo com HAYKIN (2000), um padrão é representado por um conjunto de m observações, que pode ser entendido como um ponto x de um espaço de observações m -dimensional. A extração de características é descrita por uma transformação que mapeia

o ponto x para um ponto intermediário y no espaço de características q -dimensional, com $q < m$, como indicado na Figura 7.

Esta transformação pode ser vista como uma redução de dimensionalidade (compressão de dados), cuja utilização é justificada por simplificar a tarefa de classificação. A própria classificação é descrita como uma transformação que mapeia o ponto intermediário y para uma das classes em um espaço de decisão r -dimensional, em que r é o número de classes a ser distinguida.

De acordo com HAYKIN (2000), os seres humanos têm habilidades inerentes ao reconhecimento de padrões, pois os sentidos atuam como receptores que captam dados do ambiente e são capazes de reconhecer a fonte dos dados, quase que imediatamente e sem esforço. Eles podem, por exemplo, reconhecer um rosto familiar de uma pessoa, muito embora esta pessoa tenha envelhecido ou tenham cicatrizes provenientes de acidentes.

3 Verificação de assinatura

3.1 Introdução

A preocupação com a segurança da informação e o surgimento de novas técnicas de falsificação de identidade forçou a sociedade a buscar novas maneiras para reconhecer a identidade de um indivíduo. Métodos convencionais de identificação como carteira de identidade, número de segurança e reconhecimento facial não são confiáveis por serem facilmente burladas. Por essa razão, empresas e governos estão promovendo o reconhecimento biométrico como uma forma mais segura de identificar pessoas com má fé e controlar acesso a áreas privadas (IMPEDOVO; PIRLO; RUSSO, 2014).

Um sistema biométrico é essencialmente um sistema de reconhecimento de padrão que funciona através da aquisição de informação biométrica de um indivíduo, extraindo um conjunto de características das informações obtidas, e as comparando com características extraídas de um conjunto de dados. Nesse contexto, assinaturas manuscritas ocupam lugar especial dentre as técnicas de biometria e isso se deve ao fato de que assinaturas são o meio mais difundido para verificação pessoal (IMPEDOVO; PIRLO, 2008), além de ser legalmente aceito para essa finalidade em muitos países (OHISHI et al., 2001; WIJESOMA et al., 2001).

Majoritariamente há dois tipos de abordagens para verificação de assinaturas. Se o processo de verificação utiliza características obtidas no momento que a assinatura é escrita (pressão, velocidade, coordenada em função do tempo), esse método é dito *on-line* ou dinâmico. Um exemplo desse método *on-line* ou dinâmico é a verificação que ocorre com o auxílio de um *tablet*. Nessa abordagem, é necessário de um dispositivo para o processamento dedicado para o momento da produção da assinatura. Por outro lado, quando o processo de verificação é baseado em características estáticas, como por exemplo assinando em um papel e digitalizando, esse método é dito *off-line* ou estático.

3.2 Tipos de abordagens na Verificação de Assinaturas

Os métodos de verificação de assinatura *off-line* apresentam duas classes de características extraídas da imagem de acordo com Justino (2001). Uma classe é chamada de estática, pois possui a capacidade de representar as características relacionadas à forma, como por exemplo, o comprimento e a altura máxima de uma assinatura. A segunda classe chamada de pseudodinâmica, possui a capacidade de representar as características dinâmicas da escrita como tensão do traçado, inclinação e pressão. Métodos baseados em

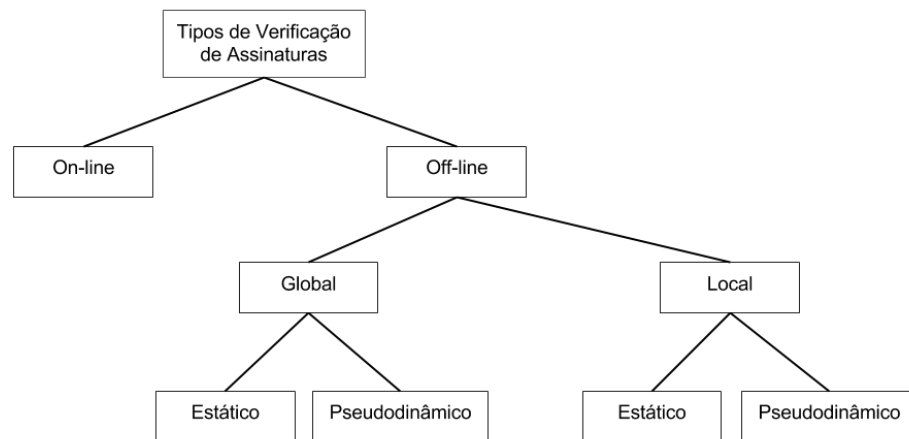


Figura 8 – Diagrama hierárquico de classificação dos tipos de abordagens de verificação de assinaturas. Fonte: Autoria própria.

características estáticas são usados geralmente para identificar falsificações simples. A razão é que estes métodos mostram-se mais aptos para descrever características relacionadas à forma da assinatura. Já as abordagens pseudodinâmicas são capazes de capturar detalhes da movimentação da escrita, sendo portanto mais adequados para identificar falsificações habilidosas (SANTOS, 2004).

A abordagem usada para manipular características em uma imagem determina a eficiência de um método na resolução do problema. De acordo com Justino (2001) tais métodos podem ser:

- Globais, que usam as características gerais da imagem, sendo mais insensíveis às variações intrapessoais, porém com menor capacidade de distinção de falsificações simuladas;
- Locais, que dependem de um processo de segmentação da imagem principal em subimagens a serem analisadas, descrevendo a particularidade do traçado e sendo mais eficientes na identificação de todos os tipos de falsificações.

O diagrama apresentado na Figura 8 demonstra como os métodos de verificação de assinatura são classificados.

3.3 Métodos de Verificação

A escolha do tipo de representação constitui uma etapa essencial na elaboração de um sistema de verificação. As dificuldades surgem principalmente em relação à maneira como são tratadas as entidades naturais usadas para obter a descrição matemática, induzida

por um método teórico formal. Segundo [Justino \(2001\)](#) e [SANTOS \(2004\)](#), os dois tipos de métodos formais são:

- Métodos estruturais: buscam descrever informações geométricas de maneira estrutural, representando formas complexas a partir de componentes elementares, chamadas primitivas.
- Métodos estatísticos: consistem em efetuar as medições do espaço métrico através da estatística. O aprendizado é executado através da separação de um conjunto de amostras em classes obedecendo a um conjunto de características comuns. A modelagem estatística se beneficia dos processos automáticos. Os principais métodos estatísticos são os paramétricos e os não- paramétricos.

3.4 Características de assinaturas

Por mais que alguma semelhança nas assinaturas ou escrita de cada autor possa existir, elementos indicadores de individualidade estarão presentes na sequência de letras de uma palavra. São características discriminantes observadas de perto pelo perito durante uma perícia. A seguir são apresentadas algumas dessas características individuais e quem podem ser utilizadas como parâmetro para identificação de uma fraude.

3.4.1 Inclinação em relação à linha base

Muitos documentos apresentam uma linha onde o autor deve realizar sua assinatura. Tomando essa linha como linha base, a assinatura do autor pode apresentar graus distintos de inclinação, usualmente para cima da linha em documentos pautados, como ilustrado na [Figura 9](#).



Figura 9 – Amostra de assinatura inclinada em relação à linha base.

3.4.2 Inclinação axial

A inclinação axial é o a inclinação da escrita, em relação ao eixo vertical, formando um ângulo com o eixo horizontal. Esse eixo, representado por uma linha de base imaginária. Essa inclinação pode ocorrer à esquerda ou à direita, conforme Figura 10.



Figura 10 – Amostra de assinaturas com inclinação axial. Fonte: SANTOS (2004).

3.4.3 Calibre

O Calibre de uma assinatura é o tamanho da escrita em termos de largura e altura. Assinaturas de alto calibre tendem a terem maior altura e largura que assinaturas de baixo calibre. Esse tipo de característica é intrínseca de cada autor e é melhor visualizada em assinaturas cursivas, como demonstrado na Figura 11.



Figura 11 – Amostra de assinaturas com diferentes calibres.

3.5 Estudo de caso: DSSV

O *Document Solutions Signature Verification* (DSSV) é um software de verificação de assinaturas desenvolvido pela Document Solutions, empresa genuinamente brasileira que desenvolve e comercializa componentes inteligentes e soluções nas áreas de Processamento de Imagens, Reconhecimento de Padrões e Recuperação de Informação. A Figura 12 mostra a tela inicial do DSSV.

O DSSV é desenvolvido na linguagem C++ e permite a verificação da autenticidade de assinaturas manuscritas comparando imagens de assinaturas manuscritas com seus equivalentes genuínos em cheques bancários, contratos, cartões de identificação e acesso. O DSSV permite o recorte da assinatura de interesse para posterior armazenamento ou somente o armazenamento de um arquivo contendo as características da imagem, podendo até ser desprezado a imagem da assinatura após a extração de suas características.

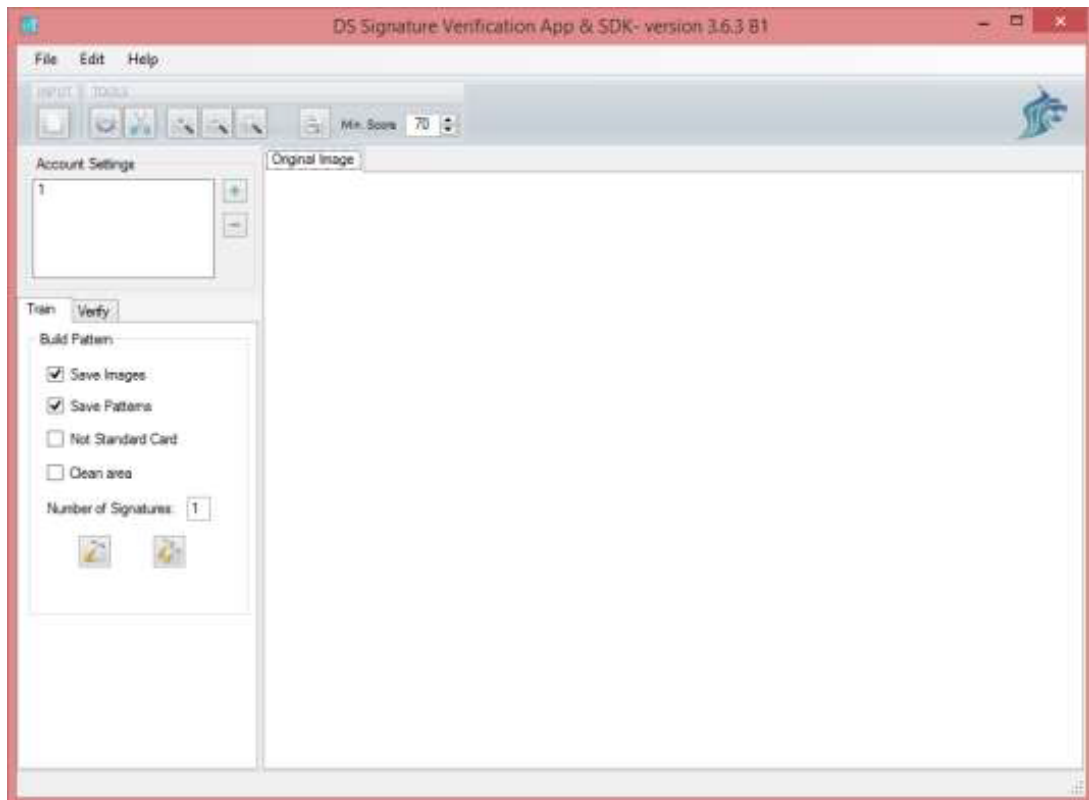


Figura 12 – Tela inicial do DSSV.

O DSSV permite que o usuário salve a imagem processada e os padrões. Além disso, é oferecida uma opção para escolha de qual classificador usar, dentre eles uma Rede Neural Perceptron de Múltiplas Camadas e o Dynamic Time Warping (DTW). Este último, utilizado neste trabalho. Além dessas configurações, também é possível escolher limiares, posição de assinatura e definição de *templates*.

A verificação de assinatura se dá através da opção *Verify*. O usuário primeiramente deve selecionar uma imagem alvo que será verificada. A verificação poderá retornar três tipos de respostas: a resposta será *Positive* se assinatura alvo pertencer à conta selecionada; a resposta será *Negative* se a assinatura alvo não pertencer à conta selecionada; ou a resposta será *Undefined* se o sistema não conseguir inferir se a assinatura pertence ou não a conta de acordo com os limiares de certeza passados como parâmetro. Além disso, como visualizado na Figura 13, o *software* também retorna um campo *Score*, que representa o grau de confiança no resultado obtido.



Figura 13 – Tela de verificação de assinaturas do DSSV.

4 Extração de características

O objetivo da extração de características é transformar uma imagem em uma representação que possa ser usada em um classificador. Na literatura, diversos tipos de parâmetros são utilizados para essa etapa. Pesquisadores consideram parâmetros baseados em orientação (POTTIER; BUREL, 1994), direção (PAL et al., 2011), forma geométrica (HAN; SETHI, 1995) e estrutura (PALANISWAMI et al., 1991).

Além desses, também são utilizados parâmetros baseados no contorno, que não são exclusivos da área de Verificação de Assinaturas (ABUSHARIAH et al., 2012; PAL et al., 2012; GHANDALI; MOGHADDAM, 2008). Técnicas de Reconhecimento de Gestos utilizam extração de características baseadas no contorno da mão para classificar qual o gesto que está sendo executado em determinado momento. Este capítulo explica algumas dessas técnicas que foram utilizadas para o sistema de verificação de assinaturas e como elas foram adaptadas para essa finalidade.

4.1 *Local Contour Sequence (LCS)*

O *Local Countour Sequence* (LCS) é uma técnica para descrição de contorno de uma imagem, inicialmente utilizada para descrição de gestos através de imagens de uma mão, representando a posição que ela está assumindo naquela imagem. O LCS recebe como entrada a imagem binária e encontra o primeiro pixel que faz parte do contorno, percorrendo a imagem de cima para baixo e da esquerda para direita. Tomando esse pixel como referência, o algoritmo ordena todos os demais pixels em ordem horária.

A ordenação no sentido horário dos pixels é feita utilizando um algoritmo de ordenação baseado em vizinhança. Neste algoritmo, dado que o primeiro pixel encontrado é representado pelas coordenadas (x, y) e marcado como p_0 , verifica-se se os pixels dos seus vizinhos norte $(x, y + 1)$, nordeste $(x + 1, y + 1)$, leste $(x + 1, y)$, sudeste $(x + 1, y - 1)$, sul $(x, y - 1)$, sudoeste $(x - 1, y - 1)$, oeste $(x - 1, y)$ ou noroeste $(x - 1, y + 1)$ são 1, conforme ilustrado na Figura 14.

Seguindo tal ordem, a verificação é feita até encontrar o primeiro ponto, chamado de p_1 . O ponto p_1 é utilizado como referência para obtenção do ponto p_2 através da mesma análise de vizinhança. Esses passos são repetidos até que todos os pontos do contorno sejam percorridos, seguindo a restrição que um ponto não pode ser selecionado mais de uma vez.

Após o cálculo dos pontos, toma-se h_i como pixel na posição i , ou seja, $h_i = (x_i, h_i)$. O cálculo das distâncias retornadas pelo LCS é feito entre o pixel h_i e uma reta formada

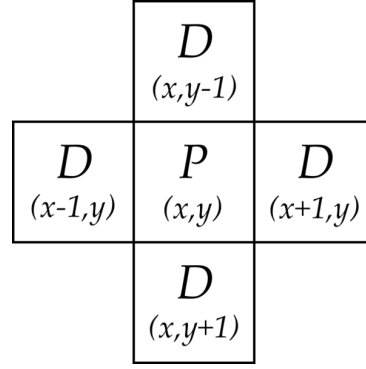


Figura 14 – Vizinhança norte, sul, leste e oeste do ponto P . Fonte: [Barros et al. \(2013b\)](#).

entre os pontos $h_{[i-(w-1)/2]}$ e $h_{[i+(w-1)/2]}$, onde w é um parâmetro do LCS. A distância h do pixel i é definido pela equação

$$h(i) = \left\lfloor \frac{\mu_i}{v_i} \right\rfloor, \quad (4.1)$$

onde μ_i e v_i são definidos pelas equações 4.2 e 4.3.

$$\begin{aligned} \mu_i = & x_i[y_i - (w - 1/2) - y_i + (w - 1/2)] \\ & + y_i[x_i + (w - 1/2) - x_i - (w - 1/2)] \\ & + [y_i + (w - 1/2)][x_i + (w - 1/2)] \\ & - [y_i - (w - 1/2)][x_i - (w - 1/2)] \end{aligned} \quad (4.2)$$

$$v_i = \sqrt{(y_i - (w - 1/2) - y_i + (w - 1/2))^2 + (x_i - (w - 1/2) - x_i + (w - 1/2))^2} \quad (4.3)$$

Algumas características do LCS são importantes ressaltar. A técnica é invariante à escala, ou seja, imagens rotacionadas, obtidas através de um dispositivo móvel, por exemplo, terão resultados semelhantes. O algoritmo também é invariante à translação, pois as distâncias do ponto analisado e a reta formada pelos pontos vizinhos vizinhos sempre será a mesma, independente de qual ponto iniciar a avaliação. O valor do parâmetro w afeta o resultado do vetor de características. Um valor alto de w deixa o algoritmo muito sensível, possivelmente considerando ruídos. Por outro lado, um valor baixo de w pode gerar muitas características, aumentando o custo computacional.

4.2 Speeded Up Robust Features (SURF)

Em 1999 foi publicado um paper que introduzia um novo algoritmo para detecção e descrição de características locais em uma imagem que fosse invariante à escala, o

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

(a) Imagem original

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

(b) Imagem integral

Figura 15 – (a) A imagem de entrada, (b) Imagem integral calculada através da imagem (a). Cada pixel representa a soma dos pixels vizinhos acima e esquerda. Fonte: Mathworks (2015).

Scale-invariant Feature Transform (SIFT) (LOWE, 1999). Um dos problemas do SIFT era o desempenho. Então, em 2008, foi publicado um novo paper “SURF: Speeded Up Robust Features” que introduzia um novo algoritmo chamado SURF. Como o próprio nome sugere, uma versão otimizada do SIFT (BAY et al., 2008).

Para encontrar os pontos de interesse em uma imagem, isto é, os pontos dos quais serão obtidos os descritores, o SURF utiliza uma abordagem baseada em uma aproximação da matriz Hessiana (YITAO et al., 2011). Através do uso de imagens integrais (CROW, 1984), o SURF consegue diminuir o custo computacional do algoritmo, resultando em uma execução rápida de todo o processo de extração dos pontos de interesse.

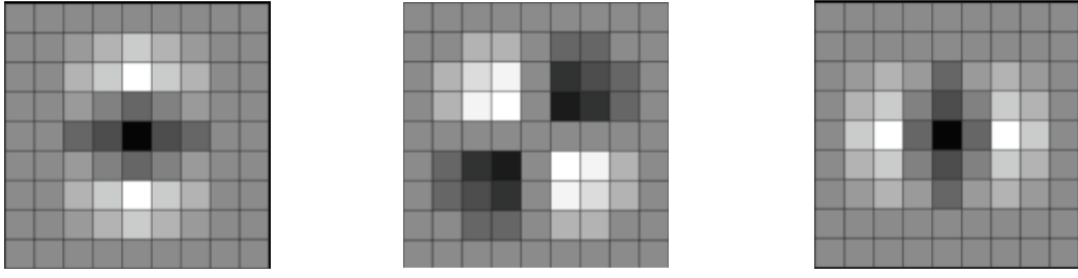
Uma imagem integral, representa a soma dos pixels de uma imagem I com seus vizinhos acima e a esquerda (VIOLA; JONES, 2001). A Equação 4.4 ajuda a definir a imagem integral I .

$$I = \sum I(x-1, y-1) \quad (4.4)$$

A detecção dos pontos de interesse do SURF está baseada na matriz Hessiana, através da busca por estruturas parecidas, ou seja, nas quais o determinante da matriz é máximo. Dado um ponto $\chi = (x, y)$ em uma imagem I , a matriz Hessiana $\chi(x, \sigma)$ em x e com uma escala σ é definida por:

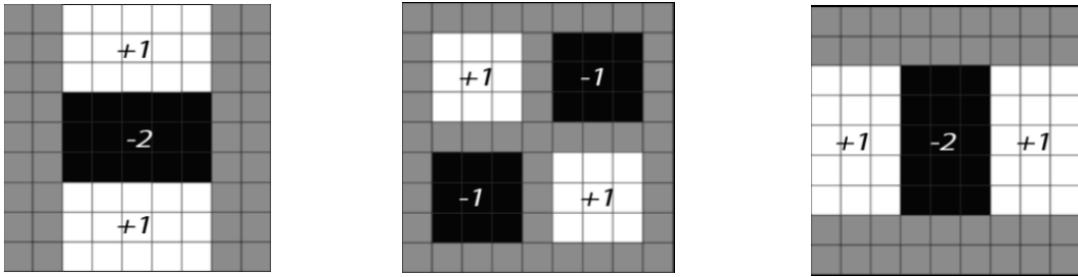
$$\chi(x, y) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (4.5)$$

Onde $L_{xx}(x, \sigma)$ é a convolução da derivada Gaussiana de segunda ordem $\frac{\partial^2}{\partial x^2}g(\sigma)$ com a imagem I em x . Da mesma forma descobre-se $L_{xy}(x, \sigma)$ e $L_{yy}(x, \sigma)$. A Figura 16a exibe a derivada gaussiana de segunda ordem em $L_{yy}(x, \sigma)$. Já a Figura 16b exibe a derivada gaussiana de segunda ordem em $L_{xy}(x, \sigma)$ e a Figura 16c exibe a derivada da gaussiana de segunda ordem em $L_{xx}(x, \sigma)$. Para todas as imagens da Figura 16, uma escala σ de 1.2 foi utilizada.



(a) Derivada Gaussiana de segunda ordem em $L_{yy}(x, \sigma)$. (b) Derivada Gaussiana de segunda ordem em $L_{xy}(x, \sigma)$. (c) Derivada Gaussiana de segunda ordem em $L_{xx}(x, \sigma)$.

Figura 16 – (a) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{yy}(x, \sigma))$, (b) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{xy}(x, \sigma))$, (c) Derivada Gaussiana de segunda ordem aplicadas em $y(L_{xx}(x, \sigma))$. Fonte: [OpenCV \(2015a\)](#).



(a) Aplicação do filtro Laplaciano Gaussiano na Figura 16a. (b) Aplicação do filtro Laplaciano Gaussiano na Figura 16b. (c) Aplicação do filtro Laplaciano Gaussiano na Figura 16c.

Figura 17 – Resultado da aplicação do Filtro Laplaciano Gaussiano nas imagens da Figura 16. Fonte: [OpenCV \(2015a\)](#).

Já que a aplicação do Filtro Hessiano original não é ideal para aplicações computacionais, um Filtro Laplaciano da Gaussiana ([LOWE, 2004](#)) é utilizado, de forma a transformar o resultado obtido pela aplicação da matriz Hessiana em valores computacionais. Um Filtro 9×9 normalmente é utilizado, pois adequa-se bem a maioria dos casos. Denomina-se D_{xx} a aplicação do Filtro Laplaciano da Gaussiana na Derivada Gaussiana de segunda ordem $L_{xx}(x, \sigma)$. Da mesma forma, encontra-se D_{xy} e D_{yy} da imagem. O resultado da aplicação do Filtro Laplaciano da Gaussiana nas imagens da Figura 16 pode ser visto nas imagens da Figura 17.

Tomando-se um ponto qualquer x , para encontrar as regiões de interesse de x , o SURF utiliza o determinante da matriz Hessiana para o ponto x . Os pontos com determinantes próximos são mapeados em uma região similar, delimitando aquela como uma região de interesse. O determinante da matriz Hessiana χ pode ser calculado utilizando a aproximação de D_{xx} , D_{xy} e D_{yy} , dado por

$$\det(\chi) = D_{xx}D_{yy} - (0.9D_{xy})^2. \quad (4.6)$$

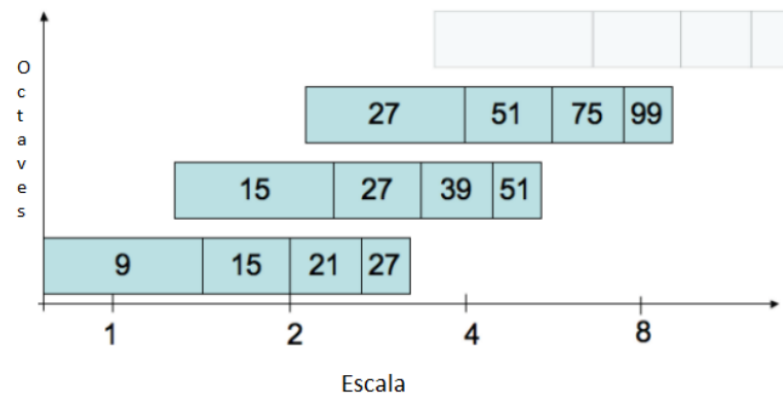


Figura 18 – Relação entre Octave e escala para aplicação dos filtros. Fonte: [Bay et al. \(2008\)](#).

As regiões de interesse encontradas precisam ser invariantes à escala, de forma que imagens com tamanhos diferentes possam ser comparadas e para tanto um espaço de escala é implementado. Espaços de escala são utilizados geralmente como uma pirâmide de imagens. A cada nível da pirâmide, a imagem é borrada com um Filtro Gaussiano e então redimensionada. Como o SURF trabalha com imagens integrais, o custo computacional de aplicar filtros maiores na mesma imagem é menor do que redimensionar toda a imagem, fazendo com que em cada nível da pirâmide seja aplicado um Filtro gaussiano em escala maior. No trabalho de ([LOWE, 2004](#)), ele aplica uma Diferença de Gaussianas em todos os níveis da pirâmide, de forma a extrair contornos e regiões comuns a todas elas.

Para extração dos pontos de interesse, dentro da região de interesse, o espaço de escala é dividido em Octaves. Cada Octave representa um mapeamento das respostas obtidas pela convolução da imagem com um filtro, que varia em escala. No total, uma Octave contém o mapeamento das respostas de um filtro com escala dobrada, ou seja, o filtro vai crescendo até atingir uma escala que possui o dobro do tamanho original. A Figura 18 exibe a relação entre as Octave e as escalas para aplicação dos filtros em cada uma delas.

A imagem original é aplicada dentro do espaço de escalas, através do filtro com escalas diferentes em cada Octave. Isso cria uma pirâmide de imagens, contendo a imagem original na base e todas as suas derivadas nos níveis mais acima. A partir das áreas de regiões encontradas em cada Octave, uma supressão não-máxima de tamanho 3×3 é aplicada em cada escala de cada Octave. Esse processo resulta na extração de intensidades díspares, realçando as regiões de transição da imagem. Para cada escala aplicada, a supressão não-máxima também é aplicada para as regiões acima e abaixo, realçando as áreas de interesse nas diversas escalas. Após essa operação, os pontos de interesse (χ') são interpolados utilizando a matriz Hessiana escrita no formato de uma expansão de uma

série de Taylor centrada no ponto X :

$$\chi' = -\frac{\partial^2 H^{-1}}{\partial X^2} \frac{\partial H}{\partial X} \quad (4.7)$$

Após extrair os pontos de interesse, o SURF obtém descritores para esse ponto. De forma que esses descritores sejam invariantes à rotação, o SURF extrai a orientação de cada ponto. Para tanto a transformada de Haar é calculada para as direções x e y para os vizinhos em um raio de $6s$, onde s delimita a escala em que os pontos foram encontrados. Mais uma vez o uso de imagens integrais auxiliam na aplicação da transformada de Haar, agilizando o seu cálculo.

4.3 Abordagem Convexa

A Abordagem Convexa foi proposta por [Barros et al. \(2013a\)](#) e foi utilizada para Reconhecimento de Gestos através de quadros de imagens extraídos de vídeos de gestos. É composta pelas etapas: remoção de *background*, identificação do contorno, extração de características, minimização do modelo e extração dos pontos mais significativos. O Algoritmo 1 expõe essas etapas. Como entrada, a técnica recebe um conjunto de “imagens brutas” (*i.e.* imagens sem pré-processamento) que devem ser extraídas características.

Para a primeira etapa do algoritmo, a remoção do *background*, é utilizada a técnica de Limiarização de Otsu ([OTSU, 1975](#)), resultando em uma imagem binária contendo apenas a forma do objeto. A fim de diminuir a quantidade de ruído presente na imagem e suavizá-la, aplica-se um Filtro de Mediana ([GONZALEZ; WOODS, 2006](#)).

Para realizar a obtenção do contorno, o Filtro de Detecção de Borda de Canny é utilizado. O resultado nesta etapa contém informações mais representativas da forma do objeto. O contorno da mão é resultado como entrada para a terceira etapa, a Extração de Características.

A extração de características deve ser feita de tal forma que um número mínimo de pontos que representam a forma do objeto sejam selecionados. Isso é realizado através do algoritmo de Douglas-Peucker ([HECKBERT; GARLAND, 1997](#)) que cria uma curva de aproximação dos pontos extraídos na etapa anterior, resultado em um polígono que representa a forma do objeto. Logo após, utiliza-se o algoritmo de Sklansky ([SKLANSKY, 1982](#)) para minimizar o modelo geométrico e excluir toda informação redundante.

4.3.1 Convexity Local Countour Sequence (CLCS)

O *Convexity Local Countour Sequence* (CLCS) é uma técnica que, invés de extrair características a partir dos pontos do contorno do objeto, ela procura minimizar o contorno

(BARROS et al., 2013a). Dessa forma, pontos que representam o objeto mais eficientemente são selecionados.

Finalmente, o cálculo de distâncias é utilizado para extrair as principais características do objetos. O CLCS consiste em uma adaptação do LCS. Enquanto que no LCS o conceito padrão de janela é utilizado para calcular os valores dos pontos, no CLCS todos pontos selecionados são utilizados sequencialmente para o cálculo da distância LCS, gerando um vetor de saída (BARROS et al., 2013a).

Algoritmo 1: Extração de características usando o CLCS.

Input: conjunto de imagens I

Output: características *features* de todas as imagens de I

```

1  $features \leftarrow \emptyset$ ;
2 begin
3   for  $imagem \in I$  do
4      $temp \leftarrow removeBackground(imagem)$ ;
5      $temp \leftarrow EncontrarContorno(temp)$ ;
6      $temp \leftarrow AlgoritmoDouglasPeucker(temp)$ ;
7      $temp \leftarrow AlgoritmoSkanky(temp)$ ;
8      $resultado \leftarrow LCS(temp)$ ;
9     adicionar  $resultado$  em  $features$ 
10  return  $features$ 

```

4.3.2 Convexty Speeded Up Robust Features (CSURF)

O *Convexty Speeded Up Robust Features* nada mais é que a aplicação do SURF na Abordagem Convexa.

5 Experimentos e Resultados

Este capítulo relata os experimentos e resultados obtidos neste trabalho. De forma mais detalhada, são analisados os resultados na classificação de imagens de assinaturas manuscritas utilizando técnicas de extração de características descritas no Capítulo 3, a saber: (i) LCS, (ii) CLCS, (iii) SURF e (iv) CSURF.

Para tal, implementou-se as técnicas de extração de características utilizando a linguagem de programação C#. A escolha por essa linguagem se deu devido à facilidade e eficiência no processamento de matrizes e consequentemente imagens, assim como a vasta gama de bibliotecas. Foram usadas bibliotecas como AForge ([AFORGE.NET](#), 2015) para processamento digital de imagens; Emgu CV ([EMGUCV](#), 2015) para uso do OpenCV ([OPENCV](#), 2015b) na plataforma .Net e, consequentemente, ferramentas de visão computacional; Accord.NET ([ACCORD.NET](#), 2015) para uso de ferramentas de estatística e aprendizado de máquina.

5.1 Base de dados

A base de dados utilizada neste trabalho foi a mesma base disponível para a competição *Signature Verification and Writer Identification Competitions for On- and Offline Skilled Forgeries* (SigWIComp2015), competição realizada em 2015 como parte do da conferência *International Conference on Document Analysis and Recognition* (ICDAR). A base de dados está disponível no *website* da competição ([SIGWICOMP2015](#), 2015).

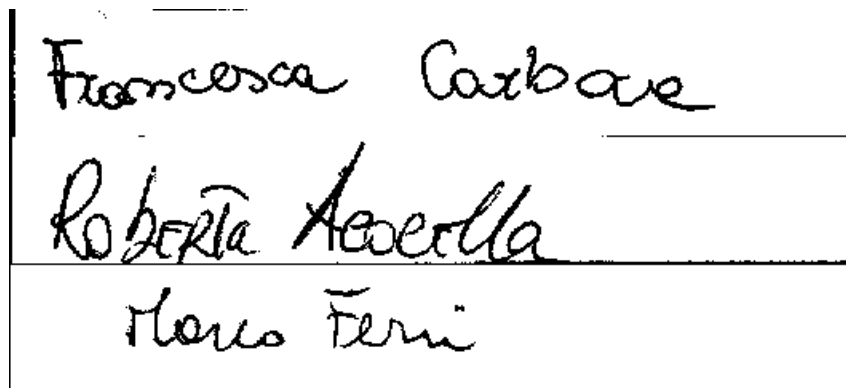


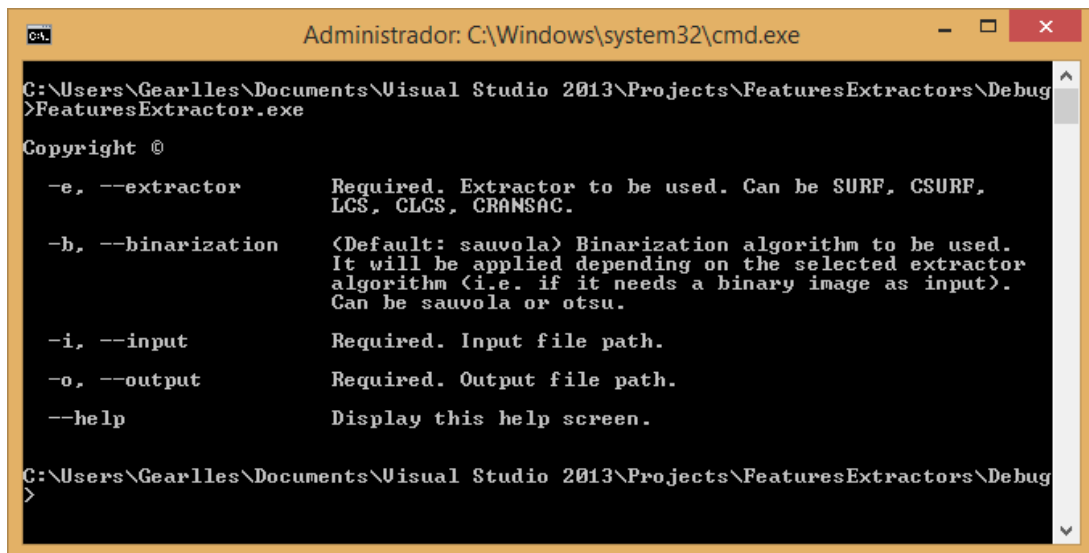
Figura 19 – Amostra de imagens presentes na base italiana da competição SigWIComp2015.

A base do SigWIComp2015 possui assinaturas escritas em três diferentes idiomas: Italiano, Bengali e Alemão. Entretanto, para este trabalho apenas o idioma Italiano foi utilizado. A base Italiana está dividida no conjunto de treino e no conjunto de testes,

entretanto apenas o conjunto de treino está disponível publicamente. Ela possui imagens de 50 autores diferentes, cada um com 5 imagens. A [Figure 19](#) mostra exemplos de imagens da base.

5.2 Metodologia e resultados

O DSSV foi utilizando como ferramenta para classificação utilizando o *Dynamic Time Warping* (DTW). Entretanto, como o DSSV é feito na linguagem C++ e as técnicas de extração de características foram implementadas em C#, foi-se necessária realizar uma integração. Essa integração foi feita através da compilação do código que implementa as extrações de características e a chamada por linha de comando do executável. Dessa forma, enviando como entrada a imagem, têm-se como saída o vetor de características da imagem correspondente.



```
Administrador: C:\Windows\system32\cmd.exe
C:\Users\Gearlles\Documents\Visual Studio 2013\Projects\FeaturesExtractors\Debug
>FeaturesExtractor.exe
Copyright ©

-e, --extractor      Required. Extractor to be used. Can be SURF, CSURF,
                    LCS, CLCS, CRANSAC.
-b, --binarization   (Default: sauvola) Binarization algorithm to be used.
                    It will be applied depending on the selected extractor
                    algorithm (i.e. if it needs a binary image as input).
                    Can be sauvola or otsu.
-i, --input          Required. Input file path.
-o, --output          Required. Output file path.
--help              Display this help screen.

C:\Users\Gearlles\Documents\Visual Studio 2013\Projects\FeaturesExtractors\Debug
>
```

Figura 20 – Módulo que implementa as técnicas de extração de características LCS, SURF, CLCS e CSURF.

Como ilustrado na [Figura 20](#), o *software* que implementa as extrações de características recebe como argumentos a técnica de extração, o caminho da imagem de entrada, o caminho do arquivo de saída e a técnica de binarização, que pode ser Sauvola ou Otsu.

Como citado anteriormente, o software de extração de características é chamado por linha de comando pelo DSSV e sua saída, que é um arquivo com as características, é analisada para de fato ser passada como entrada para o classificador DTW. Utilizou-se limiar positivo $Th1 = 0,5$ e limiar negativo $Th2 = 0,5$, isso quer dizer que, em respostas positivas, valores menores que 0,5 são desconsiderados. Da mesma forma, em respostas negativas, valores menores que 0,5 também são desconsiderados.

A [Tabela 1](#) contém os valores percentuais de verdadeiro positivo e falso negativo de cada algoritmo de extração de características. Como observado, utilizando o DTW

Extração de característica	Verdadeiro Positivo	Falso Positivo
LCS	54,637%	42,017%
CLCS	58,854%	39,324%
SURF	69,363%	34,891%
CSURF	70,824%	29,421%

Tabela 1 – Comparação dos resultados utilizando diferentes técnicas de extração de características.

como classificador, o LCS tem uma taxa de 54,637% de verdadeiro positivos e 42,017% de falso positivos, enquanto que o CLCS tem 58,854% e 39,324%, o SURF tem 69,363% e 34,891%, e o CSURF 70,824% e 29,421%, respectivamente.

A aplicação da Abordagem Convexa nas técnicas LCS e SURF, formando respectivamente o CLCS e o CSURF, gerou taxas de reconhecimento maiores do que quando não aplicada. Para a classificação, todos os resultados obtidos com o CLCS e o CSURF foram maiores que os obtidos pelo LCS e o SURF. Em relação ao tempo de processamento, como abordados na Tabela 2, as técnicas da Abordagem convexa também dão uma resposta mais rápida em relação ao tempo de processamento das técnicas cujo são originárias. Esse fato se deve ao tamanho e aos elementos do vetor de característica extraídos pela Abordagem Convexa (ver Figura 21). Como os pontos selecionados para a extração conseguem representar melhor a forma da assinatura, as taxas e tempo de classificação foram mais elevados.

Extração de característica	Tempo de processamento (milissegundos)
LCS	668
CLCS	76
SURF	163
CSURF	123

Tabela 2 – Comparação de tempo de processamento das técnicas de extração de características.

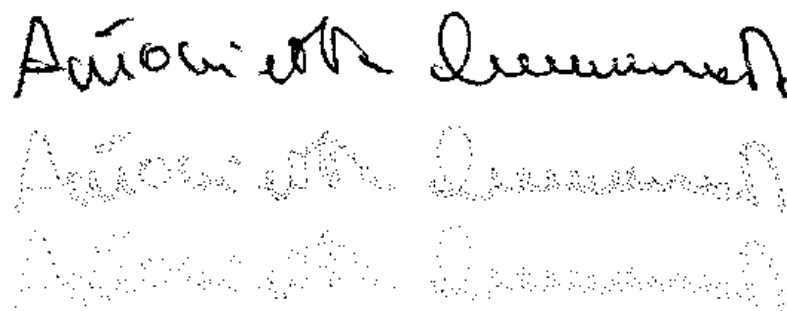


Figura 21 – Pontos selecionados pelas técnicas. Imagem original (superior), pontos selecionados pelo LCS (meio), pontos selecionados pelo CLCS (inferior).

6 Considerações Finais

Neste capítulo são apresentadas as conclusões obtidas com o trabalho, bem como sugestões para trabalhos futuros.

6.1 Conclusões

O objetivo deste trabalho é o desenvolvimento de um sistema de verificação de assinaturas que integra técnicas de extração de características baseadas em contorno, inicialmente utilizadas para Reconhecimento de Gestos. Para alcançar tal objetivo, foram estudadas diversas técnicas de processamento digital de imagens, reconhecimento de padrões e extração de características necessárias para o desenvolvimento do trabalho.

As características dos algoritmos utilizados para Reconhecimento de Gestos não é suficiente para representar uma assinatura, para posterior etapa de classificação. Dessa forma, os resultados mostraram que foi necessária uma adaptação das técnicas citadas para adequação em verificação de assinatura.

Após a implementação e adaptação das técnicas, utilizou-se o *Document Solutions Signature Verification* (DSSV), um *software* de verificação de assinaturas desenvolvido pela empresa Stefanini Document Solutions.

6.2 Trabalhos Futuros

Este trabalho pode ser estendido em vários pontos. Os seguintes pontos podem ser desenvolvidos:

- Integração com técnicas específicas de extração de características da área de Verificação de Assinaturas para futuras comparações;
- Uso de verificação *on-line* para maior taxa de classificação;
- Desenvolvimento de um aplicativo para dispositivos móveis que realiza todo sistema de verificação de assinaturas.

Referências

- ABBAS, R.; CIESIELSKI, V. A prototype system for offline signature verification using multilayered feed forward neural networks. *Internet site: <http://citeseer.nj.nec.com/25661.html>*, Citeseer, 1995.
- ABUSHARIAH, A. A. et al. Automatic person identification system using handwritten signatures. In: IEEE. *Computer and Communication Engineering (ICCCE), 2012 International Conference on*. [S.l.], 2012. p. 560–565.
- ACCORD.NET. *Accord.NET*. 2015. <<http://accord-framework.net/intro.html>>. Acessado em 15/06/2015.
- AFORGE.NET. *AForge.NET*. 2015. <<http://www.aforgenet.com/>>. Acessado em 15/06/2015.
- BARROS, P. V. et al. Convexity local contour sequences for gesture recognition. In: ACM. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. [S.l.], 2013. p. 34–39.
- BARROS, P. V. et al. An effective dynamic gesture recognition system based on the feature vector reduction for surf and lcs. In: *Artificial Neural Networks and Machine Learning–ICANN 2013*. [S.l.]: Springer, 2013. p. 412–419.
- BAY, H. et al. Speeded-up robust features (surf). *Computer vision and image understanding*, Elsevier, v. 110, n. 3, p. 346–359, 2008.
- CROW, F. C. Summed-area tables for texture mapping. *ACM SIGGRAPH computer graphics*, ACM, v. 18, n. 3, p. 207–212, 1984.
- EMGUCV. *EmguCV*. 2015. <http://www.emgu.com/wiki/index.php/Main_Page>. Acessado em 15/06/2015.
- FISHER, R. *Gaussian Smoothing*. 2015. <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>>. Acessado em 10/06/2015.
- GHANDALI, S.; MOGHADDAM, M. E. A method for off-line persian signature identification and verification using dwf and image fusion. In: IEEE. *Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on*. [S.l.], 2008. p. 315–319.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.
- GOUILLART, E. *Image manipulation and processing using Numpy and Scipy*. 2015. <http://scipy-lectures.github.io/advanced/image_processing/>. Acessado em 10/06/2015.
- HAN, K.; SETHI, I. K. Signature identification via local association of features. In: IEEE. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. [S.l.], 1995. v. 1, p. 187–190.

- HAYKIN, S. S. Redes neurais artificiais: princípio e prática. 2ª Edição, Bookman, São Paulo, Brasil, 2000.
- HECKBERT, P. S.; GARLAND, M. *Survey of polygonal surface simplification algorithms*. [S.l.], 1997.
- IBM. *Dynamic Signature Verification (DSV)*. 2015. <http://www.research.ibm.com/haifa/dept/imt/dpm/focus_dsv.shtml>. Acessado em 12/06/2015.
- IMPEDOVO, D.; PIRLO, G. Automatic signature verification: The state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 38, n. 5, p. 609–635, Sept 2008. ISSN 1094-6977.
- IMPEDOVO, D.; PIRLO, G.; RUSSO, M. Recent advances in offline signature identification. In: *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. [S.l.: s.n.], 2014. p. 639–642. ISSN 2167-6445.
- JUSTINO, E. O grafismo e os modelos escondidos de markov na verificação automática de assinaturas. 2001.
- KAMYA, S. *MDBUTMF*. 2015. <<http://www.mathworks.com/matlabcentral/fileexchange/46563-mdbutmf>>. Acessado em 10/06/2015.
- LOWE, D. Object recognition from local scale-invariant features. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, n. 2, p. 91–110, 2004.
- MATHWORKS. *Integral Image MATLAB*. 2015. <<http://www.mathworks.com/help/vision/ref/integralimage.html>>. Acessado em 15/05/2015.
- MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. Machine learning, neural and statistical classification. Citeseer, 1994.
- OHISHI, T. et al. Pen-input on-line signature verification with position, pressure, inclination trajectories. In: *Parallel and Distributed Processing Symposium., Proceedings 15th International*. [S.l.: s.n.], 2001. p. 1757–1763. ISSN 1530-2075.
- OPENCV. *Introduction to SURF (Speeded-Up Robust Features)*. 2015. <http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html>. Acessado em 15/05/2015.
- OPENCV. *OpenCV*. 2015. <<http://opencv.org/>>. Acessado em 15/06/2015.
- OTSU, N. A threshold selection method from gray-level histograms. *Automatica*, v. 11, n. 285-296, p. 23–27, 1975.
- PAL, S. et al. Off-line signature identification using background and foreground information. In: IEEE. *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*. [S.l.], 2011. p. 672–677.
- PAL, S. et al. Multi-script off-line signature identification. In: IEEE. *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*. [S.l.], 2012. p. 236–240.

- PALANISWAMI, M. et al. Signature identification using neural networks. In: IEEE. *TEN-CON'91.1991 IEEE Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems*. [S.l.], 1991. v. 2, p. 149–152.
- PIRLO, G.; IMPEDOVO, D. Cosine similarity for analysis and verification of static signatures. *IET biometrics*, IET, v. 2, n. 4, p. 151–158, 2013.
- POTTIER, I.; BUREL, G. Identification and authentication of handwritten signatures with a connectionist approach. In: IEEE. *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*. [S.l.], 1994. v. 5, p. 2948–2951.
- RIBEIRO, J.; VASCONCELOS, G. Arquiteturas construtivas autoassociativas para a verificação automática de assinaturas. In: *Proceedings of the 4 th Brazilian Conference on Neural Networks, São José dos Campos, São Paulo*. [S.l.: s.n.], 1999. p. 263–268.
- SANTOS, C. R. *Análise de assinaturas manuscritas baseada nos princípios da grafoscopia*. Tese (Doutorado) — Pontifícia Universidade Católica do Paraná, 2004.
- SIGWICOMP2015. *SigWiComp2015*. 2015. <<http://www.dfki.uni-kl.de/afha2015/SigWiComp.html>>. Acessado em 15/06/2015.
- SKLANSKY, J. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, Elsevier, v. 1, n. 2, p. 79–83, 1982.
- USERNET. *Guide to understanding and reading histograms*. 2015. <<http://www.usernetsite.com/photography/guide-to-understanding-and-reading-histograms.php>>. Acessado em 10/06/2015.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.], 2001. v. 1, p. I–511.
- WIJESOMA, W. et al. Online handwritten signature verification for electronic commerce over the internet. In: ZHONG, N. et al. (Ed.). *Web Intelligence: Research and Development*. Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2198). p. 227–236. ISBN 978-3-540-42730-8. Disponível em: <http://dx.doi.org/10.1007/3-540-45490-X_27>.
- YITAO, C. et al. Improving hessian matrix detector for surf. *IEICE transactions on information and systems*, The Institute of Electronics, Information and Communication Engineers, v. 94, n. 4, p. 921–925, 2011.

APÊNDICE A – Apêndice A: Código principal

```

namespace Main
{
    class Program
    {
        static int Main(string[] args)
        {
            var options = new Options();
            if (CommandLine.Parser.Default.ParseArguments(args,
                options))
            {
                //var desktopPath = Environment.GetFolderPath(
                //    Environment.SpecialFolder.Desktop);
                var inputFile = options.inputFile;
                var inputFileImage = Bitmap.FromFile(inputFile);
                var inputBitmap = new Bitmap(inputFileImage);

                FeaturesExtraction fe = new FeaturesExtraction()
                    ;
                List<double> caracteristicas;

                switch(options.extractor)
                {
                    case "SURF":
                        inputBitmap = preProcess(inputBitmap,
                            options);
                        caracteristicas = fe.DOSURF(inputBitmap)
                            ;
                        break;
                    case "CSURF":
                        inputBitmap = preProcess(inputBitmap,
                            options);
                        caracteristicas = fe.DOCSURF(inputBitmap
                            );
                }
            }
        }
    }
}

```

```

        break;
    case "LCS":
        inputBitmap = preProcess(inputBitmap,
            options);
        características = fe.DOLCS(inputBitmap);
        break;
    case "CLCS":
        inputBitmap = preProcess(inputBitmap,
            options);
        características = fe.DOCLCS(inputBitmap,
            null);
        break;
    case "CRANSAC":
        double[] cRANSACOutput = CIPBR.doCIPBR(
            inputBitmap);
        características = new List<double>(
            cRANSACOutput);
        break;
    default:
        throw new ArgumentException(String.
            Format("Unknown parameter \"{0}\".",
                Use"—help to check valid arguments."
                , options.extractor));
}

Console.WriteLine(String.Format("Processing_
    completed. "));
Console.WriteLine(String.Format("Features_vector
    size:{0}", características.Count));

File.WriteAllLines(
    options.outputFile // <== Put the file name
        here
    , características.Select(d => d.ToString(
        CultureInfo.InvariantCulture)).ToArray()
    );
}
return 0;
}

```

	}	
}		