

ALGORITMO PARA ESTIMAÇÃO DA FREQUÊNCIA ESPACIAL DE BIOMETRIAS A PARTIR DE PROCESSAMENTO DE IMAGEM E MÉTODO DE DERIVADAS

Felipe Cafezeiro Plech; Jovelino Torres dos Santos² Valéria Loureiro da Silva³

¹ Graduado em Engenharia Elétrica; fcplech@gmail.com

² Mestre em Engenharia Elétrica; Centro Universitário SENAI CIMATEC; Salvador-BA; jovelino.torres@fieb.org.br

³ Doutora em Física; Centro Universitário SENAI CIMATEC; Salvador-BA; valeria.dasilva@fieb.org.br

RESUMO

A necessidade de otimização da resolução de scanners biométricos pelas fabricantes para atingir melhores qualidades de imagem e para busca de certificação dos mesmos é uma realidade. Entretanto, nem sempre atingir uma resolução extremamente alta é a melhor solução, podendo gerar gastos desnecessários com componentes ópticos e eletrônicos de maior qualidade. Com base nesta problemática, este resumo expandido tem como objetivo apresentar um algoritmo para estimar a frequência espacial de biometrias para identificação da resolução necessária para um scanner biométrico, poupando maiores custos com gastos materiais.

PALAVRAS-CHAVE: Digitais, scanner, biometria, frequência, resolução.

1. INTRODUÇÃO

Para o desenvolvimento de scanners biométricos de média a alta resolução é de suma importância que seja identificado qual a resolução necessária para resolver uma digital em um sistema óptico. Os scanners comerciais que são utilizados em situações cotidianas geralmente possuem um resolução de 500dpi, considerados modelos de baixa resolução. Os modelos de alta resolução, na faixa de 800dpi até 2000dpi, podem ser fabricados sobre dimensionados, ou seja, com uma resolução muito acima do que era necessário para o seu ramo de aplicação. Isso implica com um maior custo de fabricação e desenvolvimento para o produto.

Visto isso, esse resumo expandido tem como objetivo apresentar um algoritmo para identificação da frequência espacial de digitais através de processamento de imagem e derivadas.

2. METODOLOGIA

Para validar o método, o algoritmo de processamento de imagens foi primeiramente desenvolvido no MATLAB®, um software otimizado para operações matriciais¹, para posteriormente ser portado para Python utilizando uma biblioteca mais otimizada, nesse caso a OpenCV.

A OpenCV (*Open Source Computer Vision Library*) é uma biblioteca gratuita para operações complexas de imagens tanto para uso acadêmico como para uso comercial, contendo interface para C++, Java e Python. A mesma foi desenvolvida com foco em eficiência computacional para sistemas de tempo real de processamento de imagens.

Segundo o Instituto Nacional de Pesquisas Espaciais (INPE), o processamento de imagens é um conjunto de técnicas voltadas para a análise de dados multidimensionais, adquiridos por diversos tipos de sensores, e recebem o nome de processamento digital de imagens, ou seja, é a manipulação de uma imagem por computador de modo que a entrada e a saída do processo são imagens.

Pode-se dividir o processamento de imagem realizado neste projeto em três etapas: pré-processamento, realce e análise, que serão discutidos no próximo tópico.

3. RESULTADOS E DISCUSSÃO

As imagens capturadas por câmeras digitais nem sempre representam fielmente a realidade. Os sinais analógicos possuem ruídos, que são sinais interferentes de natureza aleatória que provocam a degradação do sinal de interesse durante seu processamento.

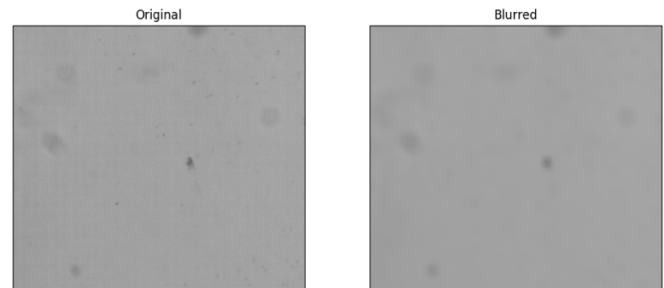
Antes de entrar na etapa de realce, é necessário atenuar esses ruídos da imagem, caso contrário, os mesmos também serão amplificados nos procedimentos posteriores. O fundo das imagens de digitais é uniforme, logo qualquer componente de alta frequência deve ser removida ou atenuada³. Para isso, foi utilizado um filtro passa-baixa comumente chamado de *blur*. Na figura 1, pode-se observar a imagem original.

Basicamente, o filtro realiza uma convolução entre um operador matricial e a imagem de interesse tirando uma média dos pixels vizinhos². O filtro pode ser comparado a uma média móvel em duas dimensões. O efeito do filtro pode ser observado na Figura 2.

Figura 1: Imagem original utilizada como exemplo.



Figura 2: Comparação entre o plano de fundo original (esquerda) e após o filtro (direita). Imagens com zoom.

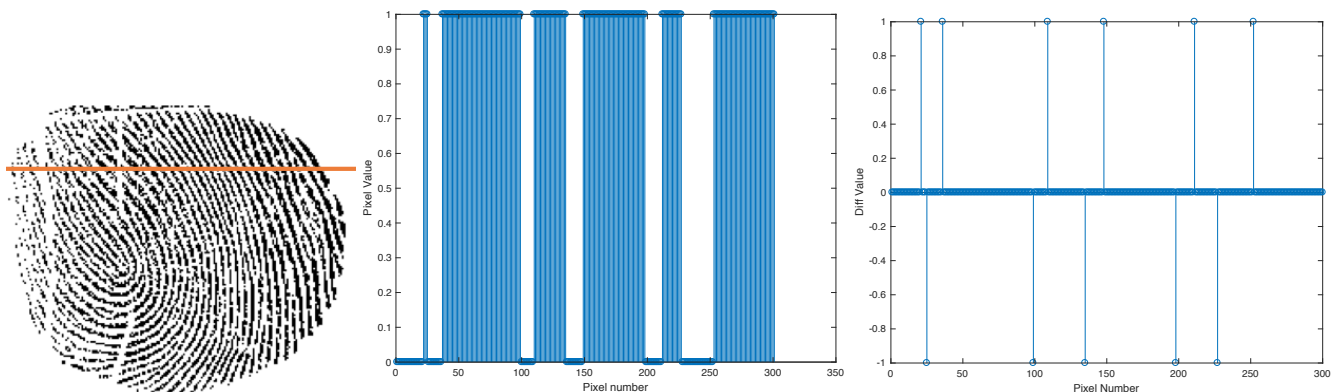


Após o pré-processamento, foi possível dar início ao procedimento de realce dos dados relevantes da imagem. Para realçar a digital do plano de fundo, foi utilizado um método de segmentação chamado de *thresholding*, com objetivo de gerar imagens binárias. Os métodos mais simples de *thresholding* substituem cada pixel de uma determinada imagem por preto, caso sua intensidade for menor que uma constante fixa, ou branco caso seja maior que essa mesma constante. O resultado é uma imagem segmentada com apenas valores de 0 e 255, caso a imagem tenha uma profundidade de cor de 8-bits⁴.

A partir da imagem segmentada são realizadas algumas operações para determinar qual a frequência espacial da digital capturada. Para isso, são necessários dois parâmetros iniciais: a magnificação do sistema óptico e o tamanho do pixel do sensor utilizado.

Para dar início ao procedimento, é retirado um *array* na direção perpendicular das digitais, como é mostrado na Figura 3a e 3b. Para saber a maior frequência nas digitais, é necessário obter a espessura dos vales das papilas (representados por branco), por terem menores dimensões. Logo, é necessário contar o número de pixels em cada vale da digital. Aplica-se então, uma derivada no *array* para se obter cada instante em que houve uma mudança de preto para branco e vice-versa, como mostrado na Figura 3c.

Figura 3: a) Exemplo de como se é obtido os dados do array. b) Gráfico demonstrando o padrão do array c) Vetor derivado.



Com base no vetor derivado, conta-se quantos pixels existem entre um valor de derivada negativa e uma derivada positiva, que representam o número de pixels de um vale. A partir desse número de pixels encontrado, é possível calcular qual é a dimensão do vale no plano da imagem, a partir da equação 1⁵:

$$T_{vale_sensor} = N_{pixels} * Pixel_{size} \text{ um} \quad [1]$$

Para saber qual a dimensão do vale no plano do objeto, divide-se o tamanho no plano da imagem pela magnificação do sistema óptico, como mostrado na equação 2.

$$T_{v\grave{a}le_object} = \frac{T_{v\grave{a}le_sensor}}{magnification} \text{ um} \quad [2]$$

Sabendo qual o tamanho do vale no plano do objeto, calcula-se o a frequência espacial. Como a frequência espacial é dada em pares de linha por milímetro (lp/mm), divide-se mil micrometros pelo dobro da espessura de um vale, como pode ser visto na equação 3⁵.

$$Frequ\^{\text{e}}ncia \text{ Espacial} = \frac{1000 \text{ um}}{2 * T_{v\grave{a}le_object}} \text{ lp/mm} \quad [3]$$

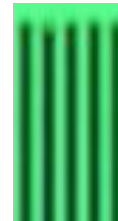
Calculando a frequência espacial de um banco de dados de digitais, pode-se fazer uma média para obter o valor de resolução necessário para o sistema óptico de um *scanner* biométrico.

Foram realizados testes com *targets*, componentes ópticos de alta precisão para testes de resolução contendo linhas a uma determinada frequência. Os resultados foram bastantes satisfatórios para uma primeira versão do algoritmo, apresentando valores com um erro de aproximadamente ± 1 lp/mm. Pode ser visto abaixo dois cortes da área de um *target* onde a frequência espacial é de 10 lp/mm e a outra de 20 lp/mm. Utilizando o algoritmo nesta imagem foi obtido um resultado de 10.56 lp/mm e de 20.51 lp/mm respectivamente.

Figura 5: Padrões de linhas de um *target* com frequência de 10 lp/mm.



Figura 6: Padrões de linhas de um *target* com frequência de 20 lp/mm.



4. CONSIDERAÇÕES FINAIS

A partir da frequência estimada das imagens das digitais, é possível ter um melhor direcionamento no momento do design óptico do sistema do *scanner* poupando custos de desenvolvimento e de fabricação, pois quanto maior a resolução do sistema será necessário utilizar componentes ópticos (prismas, espelhos e lentes) e eletrônicos (câmeras) de melhor qualidade.

Agradecimentos

Agradecemos a EMBRAPPII, SEBRAE, SENAI-CIMATEC e a empresa Akiyama pelo apoio financeiro e tecnológico.

5. REFERÊNCIAS

¹ The MathWorks, **MATLAB 2015a, Inc.**, Natick, Massachusetts, Estados Unidos. Disponível em: <https://www.mathworks.com/>

² OpenCV, **Smoothing Images**. Disponível em: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html

³ Fabris, E., **Ruídos em Sistemas Eletrônicos**, ». Disponível em: http://www.ece.ufrgs.br/~fabris/E04055/E04055_Ruido.pdf

⁴ OpenCV, **Image Thresholding**. Disponível em: <
https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html>

⁵ Edmund Optics, **Pixel Sizes and Optics**. Disponível em: <
<https://www.edmundoptics.com/resources/application-notes/imaging/pixel-sizes-and-optics/>>