

DESENVOLVIMENTO DE UMA TOOLBOX PARA O AUXÍLIO DO ENSINO-APRENDIZAGEM DO PROCESSAMENTO DIGITAL DE IMAGENS

Emmanuel Diego Gonçalves de Freitas – emannueldiego@gmail.com,
Francisco Alves de Oliveira Júnior – chico.alves.jr@gmail.com,
Evandson Claude Seabra Dantas – evandson@live.com,
Jones Clécio Otaviano Dias Júnior – jonesjr.prof@gmail.com,
Suzete Élide Nóbrega Correia – suzete.correia@gmail.com
Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB.
Av. João da Mata, 256, Jaguaribe.
CEP: 58.015-020 - João Pessoa - Paraíba

***Resumo:** Um quadro comum e recorrente no ensino das engenharias é a utilização de softwares pré-programados para a realização prática das mais diversas atividades e projetos. Um dos exemplos mais clássicos é a utilização do MatLab[®] que ganhou bastante espaço no Brasil pela sua praticidade e grande acervo de algoritmos e métodos. Toda essa praticidade envolve um alto custo computacional o que inviabiliza a implantação direta do código em sistemas embarcados que ganharam uma fatia muito grande do mercado nos últimos anos. O Processamento Digital de Imagens (PDI) é uma área da pesquisa científica bastante produtiva e desafiadora, com uma gama variada de aplicações em diversos ramos da iniciativa científica, comercial e técnica. Uma nova forma de acelerar o ensino-aprendizagem bem como projetar o conhecimento para a aplicabilidade prática e mais eficiente desta área é utilizar bibliotecas pré-programadas específicas para tal fim. O OpenCV é uma das mais conceituadas bibliotecas para realização de atividades relacionadas ao PDI. O desenvolvimento de uma toolbox com novas funcionalidades e métodos mais simplificados da biblioteca OpenCV é objeto de estudo e desenvolvimento neste artigo.*

***Palavras-chave:** OpenCV, Processamento Digital de Imagens, Ensino nas Engenharias.*

1. INTRODUÇÃO

A aplicação prática da ciência na confecção de novas ferramentas cada vez mais eficientes e duradouras é um dos principais focos e desafios da engenharia. Para que isto seja possível, é vital acompanhar o desenvolvimento tecnológico mundial, aprimorar e implementar novos métodos em sistemas mais eficientes. Estamos vivendo uma era do rápido desenvolvimento dos sistemas embarcados, ao qual são *hardwares* de baixo-custo, com pequenas dimensões e uma gama de ferramentas. Dentre as ferramentas mais comuns, estão câmeras com capacidade que vão desde 2 até 41 Megapixels (ALAKARHU, 2013) e processadores que podem executar vídeos na resolução de 4K (SHAH, 2014). Recursos como

estes permite o desenvolvimento de aplicações diversas, principalmente aqueles que envolvem o Processamento Digital de Imagens (PDI).

O PDI é uma área de estudo de cursos superiores e de pós-graduação. Os estudos relacionados a imagens auxiliam a identificar e quantificar diversos itens inerentes a esta área, tendo aplicabilidade em diversos campos de pesquisa.

Muitas vezes, para que seja possível realizar a identificação de padrões em imagens é necessário aplicar diferentes métodos que envolvem alta complexidade ao qual custa tempo de desenvolvimento e inviabiliza muitas pesquisas. Uma prática comum é utilizar *softwares* como MatLab® que incluem diversas ferramentas que agilizam o trabalho. A utilização das funções do MatLab também abstrai o funcionamento de diversos algoritmos vitais para a aprendizagem do PDI.

Desta maneira, procura-se encontrar uma ferramenta ideal que otimize o desempenho, permita uma rápida implementação, demonstre bons resultados e permitam um maior aprendizado. Então, optou-se pela utilização do OpenCV, que, além de ser gratuita de código-fonte aberto, demonstra ter ótimo desempenho, grande número de funções, apresenta boa portabilidade e ajuda a definir melhor os conceitos do PDI.

Com o objetivo de melhorar ainda mais a relação ensino-aprendizagem e facilitar a implementação dos conceitos firmados em sala de aula, esse trabalho apresenta trabalho uma *toolbox* didática para o ensino de PDI, com novos algoritmos e simplificações de funções pré-existentes da biblioteca OpenCV.

2. PROCESSAMENTO DIGITAL DE IMAGENS

2.1. Processamento digital de imagens

O processamento digital de imagens é a área do conhecimento que trata da modelagem matemática, análise, projeto e implementação em *software* e em *hardware* de sistemas digitais voltados ao tratamento de informação pictórica com o propósito de torná-la mais adequada à interpretação por seres humanos ou máquinas ou para obter maior eficiência de armazenamento e transmissão (BATISTA, 2005).

As etapas fundamentais do processamento de uma imagem estão ilustrados na “Figura 1” (GONZALEZ & WOODS, 2007).

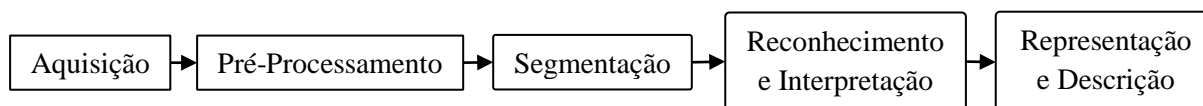


Figura 1: Etapas do Processamento Digital de Imagens

Na etapa de aquisição a imagem é convertida em uma representação numérica adequada para a próxima etapa do processamento digital. Essa etapa é dividida em dois elementos. O primeiro elemento é um transdutor que tem como função converter em um sinal elétrico a faixa de frequência do espectro da luz captada por um sensor. O segundo elemento é um conversor analógico-digital que tem como função gerar uma representação digital do sinal elétrico convertido pelo transdutor. Para um sinal ser digitalizado ele precisa ser quantizado e amostrado. Na amostragem a imagem será convertida em uma matriz de $M \times N$ pontos, em que cada ponto da matriz é denominado *pixel* (menor elemento da imagem). A quantização será responsável por atribuir um valor inteiro a cada *pixel*. Após a obtenção da imagem digital, o próximo passo é realizar o pré-processamento da imagem. O objetivo é melhorar a



imagem de forma a aumentar as chances de sucesso nas etapas seguintes. Nesta etapa, são aplicadas técnicas para suavização, correção de brilho e contraste e remoção de ruído da imagem analisada.

A segmentação divide a imagem em partes ou objetos constituintes. Em geral, a segmentação automática é uma das tarefas mais complexas no processamento digital de imagens. A saída do estágio de segmentação consiste de dados que indicam fronteiras de uma região. Dentre as técnicas usadas para a segmentação, podem ser citadas a limiarização e detecção de contornos. A segmentação pode ser feita por diferentes métodos e pode selecionar regiões, bordas e/ou preenchimentos da imagem.

Um método para descrever os dados pictóricos é fornecido através da representação e descrição, que busca informação quantitativa de interesse ou que sejam básicas para a discriminação entre classes de objetos. Técnicas de afinamento, código da cadeia, podem ser usados na descrição de objetos da imagem. Em se tratando de objetos com forma poligonal, descritores tais como área e perímetro, podem ser usadas na diferenciação das formas.

O último estágio envolve o reconhecimento e interpretação. Técnicas de inteligência computacional, a exemplo de redes neurais, podem ser usadas para dar significado a informação fornecida pelos descritores empregados.

2.2. OpenCV: Uma biblioteca de desenvolvimento de visão computacional

Desenvolvida inicialmente pela Intel® no ano 1999, o OpenCV (acrônimo de: *Open Source Computer Vision Library*) é uma biblioteca multiplataforma totalmente gratuita de código-fonte aberto para utilização acadêmica e comercial do desenvolvimento de aplicativos na área de Visão computacional (BRADSKI, 2008). Esta biblioteca pode ser executada em diversas plataformas como Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux e Mac OS. O OpenCV tem como principal foco a eficiência computacional e o desenvolvimento de aplicações em tempo real. Para isto, a biblioteca foi desenvolvida em C e C++ podendo tirar proveito de diversos núcleos de processadores. Um dos principais objetivos do OpenCV é prover uma infraestrutura de Processamento de Imagens simples de utilizar para ajudar na confecção de aplicações sofisticadas de tempo real. Esta biblioteca contém mais de 500 funções que abrangem diversas área do PDI como: Inspeção industrial de produtos, análise médica, segurança, calibração de câmera, visão robótica, dentre outras (BRADSKI, 2008).

O OpenCV tem uma estrutura modular, o que significa que cada módulo inclui rotinas e funções compartilhadas ou estáticas. Alguns dos módulos mais utilizados são:

- **core** – Módulo que define a estrutura e funcionalidades básicas das matrizes multidimensionais;
- **imgproc** – Módulo de processamento digital de imagens que contém filtragem linear e não linear, transformações geométricas, conversões de cores, histogramas e diversas outras ferramentas;
- **video** – Módulo de vídeo que permite a análise de quadros, subtração de fundo e rastreamento de objetos;
- **calib3d** – Módulo de calibração 3D. Permite a calibração da câmera e fornece diversos algoritmos que permitem a múltipla vista geométrica;
- **features2d** – Módulo que fornece diversos algoritmos para realizar métricas na imagem;

- **objdetect** – Módulo de reconhecimento que permite a detecção de objetos e instância de classes pré-definidas (i.e., rostos, olhos, canecas, carros, ...);
- **highgui** – Módulo gráfico. Permite criar uma interface gráfica para facilitar a iteração com o usuário;
- **gpu** – Módulo GPU. Fornece um conjunto de algoritmos que otimizam o funcionamento dos diferentes módulos do OpenCV;

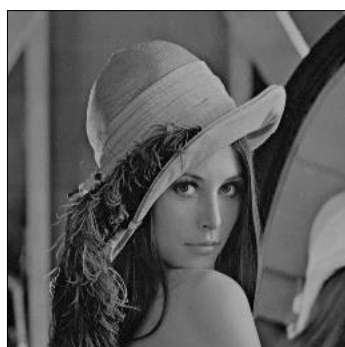
3. APRESENTAÇÃO DA TOOLBOX

Criada como uma camada de *software* superior para algumas funções já existentes do OpenCV, a *toolbox* proposta objetiva simplificar a entrada de dados e tornar mais intuitivo o uso de funções para processamento digital de imagens, haja vista que, muitas vezes, a aplicação de funções OpenCV não é feita com uma sintaxe tão natural quanto a conhecida por pessoas de pouca experiência com programação de computadores.

Trata-se de um conjunto de funções que oferecem interface simplificada para funções OpenCV ou oferecem novas rotinas ainda não implementadas. São exemplos das rotinas desenvolvidas: 5 rotinas para realce e filtragem de imagens (Brilho e Contraste, Passa-Baixas, Passa-Faixas, Passa-Alta e Mediana), 9 rotinas de limiarização (Média Padrão, *Mean Grey Level*, Dez Porcento de Preto (PARKER, 2010), Dois Picos, Seleção Iterativa (RIDLER, 1978), Otsu (OTSU, 1979), Borda, Pun (PUN, 1980) e Kapur (KAPUR, 1981)), 6 rotinas para detecção de bordas (Vetor Gradiente, Roberts, Sobel, Prewitt, Canny (CANNY, 1986) e Marr-Hildreth), 4 rotinas para morfologia matemática (Erosão, Dilatação, Erosão com Dilatação e Dilatação com Erosão) e 3 de afinamento (Stentiford, Zhang-Suen (ZHANG & SUEN, 1984) e Holt).

4. TESTES E RESULTADOS

Nessa seção, serão exibidos alguns resultados encontrados utilizando a *toolbox* desenvolvida. Para tanto serão utilizadas as seguintes imagens como entrada:



(a)



(b)

Figura 2: (a) Imagem de entrada “lena.pgm”. (b) Imagem de entrada “peppers.pgm”

4.1. Controle de contraste e de brilho

As imagens anteriores são representadas em tons de cinza. Seus *pixels* são compostos por valores de intensidade que variam entre 0 e 255 (FILHO, 1999). A Equação 1 descreve uma transformação linear de intensidade que é utilizada para realizar o realce de uma imagem.

$$g(x, y) = f(x, y) \cdot c + b, \quad (1)$$

em que: $g(\cdot, \cdot)$ representa a intensidade do *pixel* resultante, $f(\cdot, \cdot)$ a intensidade do *pixel* original, c é o fator de contraste e b a constante de brilho. x e y são coordenadas do *pixel* analisado.

Disposto dos recursos criados na *toolbox*, aplicou-se o realce para a “Figura 2a” com o parâmetro de contraste fixo em 1, mostrando o efeito da variável de brilho (Figura 3) e com o brilho fixo em 0, mostrando o efeito do contraste na imagem (Figura 4).



Figura 3: (a) Resultado para brilho $b = -60$. (b) Resultado para brilho $b = 60$. (c) Resultado para brilho $b = 120$.



Figura 4: (a) Resultado para contraste $c = 0,5$. (b) Resultado para contraste $c = 1,5$.
 (c) Resultado para contraste $c = 2$.

4.2. Limiarização

A segmentação simples, com apenas um nível, é utilizada como alternativa para separar a imagem em duas regiões: uma composta pelo objeto (*foreground*) e outra composta pelo do fundo da imagem (*background*) (Gonzales, 2007).

Neste tópico, é testado o recurso de limiarização da *toolbox* desenvolvida para diferentes métodos de limiarização que são aplicados à “Figura 2a”. Dentre os principais métodos testados estão o da média padrão (cujo limiar situa-se no meio do histograma), Nível

Médio de Cinza (*Mean Grey Level*), 10% de Preto, Seleção Iterativa (Thussel), Otsu e Dois Picos.



Figura 5: (a) Média Padrão (limiar 127); (b) Nível Médio de Cinza (limiar 94); (c) 10% de Preto (limiar 28); (d) Seleção Iterativa (limiar 82); (e) Otsu (limiar 83); (f) Dois Picos (limiar 109); (g) Borda (limiar 127); (h) Pun (limiar 51); (i) Kapur (limiar 162).

4.3. Aplicação de máscaras

A *toolbox* didática proposta também permite realizar a convolução da imagem com uma máscara pré-existente ou determinada pelo usuário, desde que a mesma seja declarada em uma matriz 3x3. Para verificar seu funcionamento, utiliza-se nos próximos testes os filtros definidos na “Tabela 1”:

Tabela 1 - Máscaras a serem testadas.

Tipo de Filtro	Máscaras Aplicadas
Filtro passa-alta	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
Filtro Prewitt vertical	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
Filtro para detecção de linhas diagonais	$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$

A convolução é feita via chamada de função e os resultados são exibidos na tela, tal como os mostrados na “Figura 6”.



Figura 6: Efeito da aplicação de máscaras. (a) Filtro passa-alta; (b) Filtro Prewitt vertical; (c) Filtro para detecção de linhas diagonais.

A seguir, são mostradas imagens resultantes obtidas após o uso dos filtros da Média e da Mediana aplicados à “Figura 2b”, com recursos da *toolbox* desenvolvida. Para fins didáticos, gerou-se um ruído “sal e pimenta”¹ para melhor demonstrar a eficiência dos respectivos filtros.

¹ Um ruído “sal e pimenta” é um ruído impulsivo cuja duração, frequência e intensidade são indeterminadas.

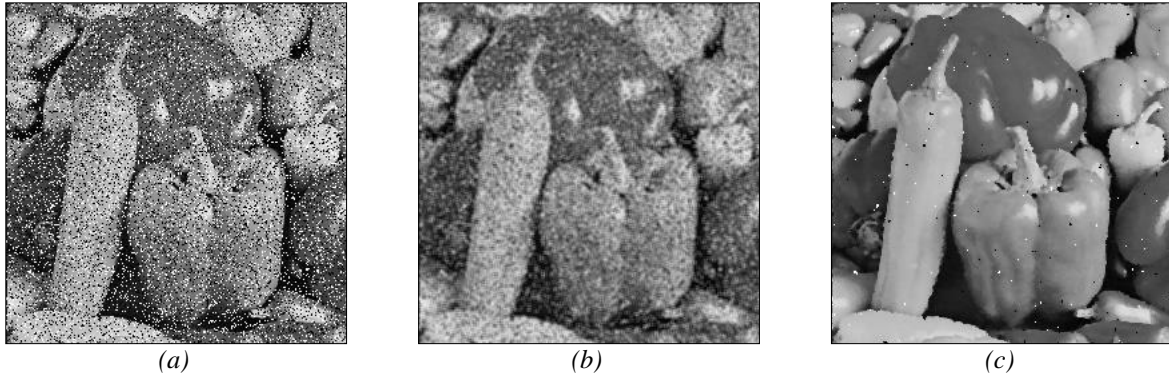


Figura 7: Efeito da aplicabilidade dos filtros. (a) Figura 2b com ruído impulsivo; (b) Filtro da Média; (c) Filtro da Mediana.

4.4. Detecção de bordas

A *toolbox* criada também detêm o recurso para a detecção de borda. Para isto, foram implementados como função os métodos comumente utilizados, sendo eles: Laplaciano, Kirsch, Canny, Sobel, Roberts e Prewitt. A “Figura 8” demonstra o efeito dos métodos citados aplicados à “Figura 2b”.

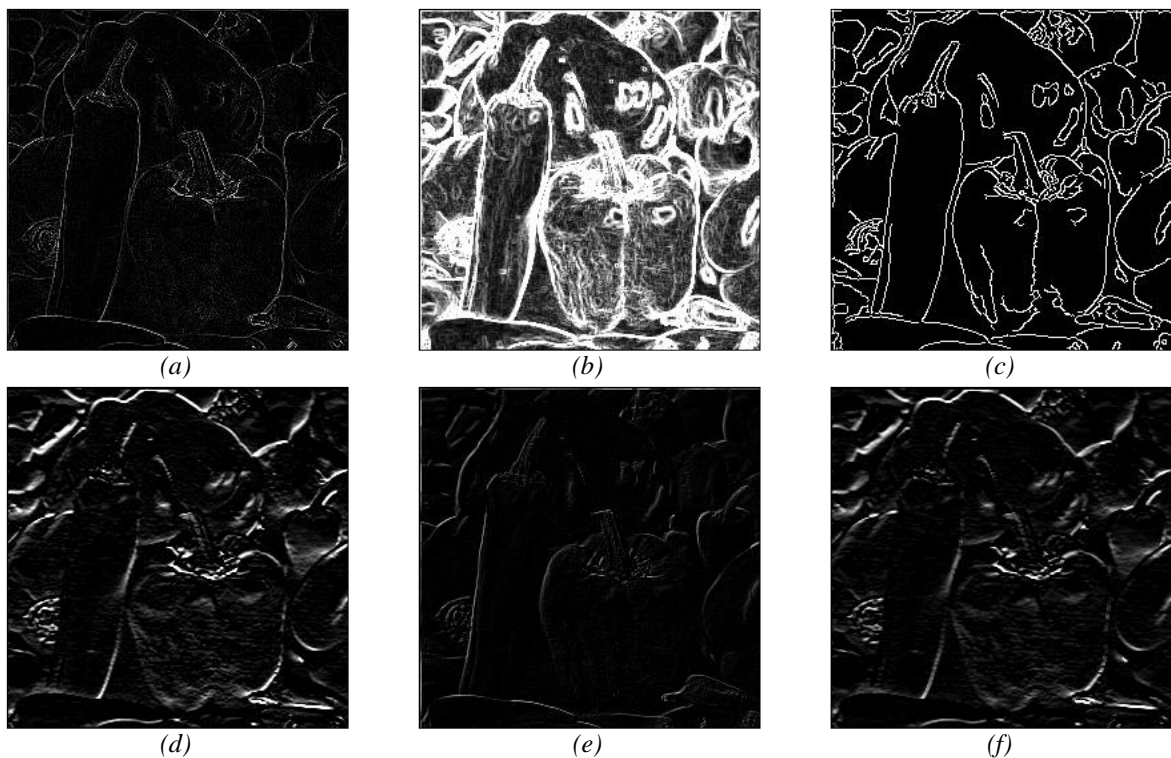


Figura 8: Diferentes métodos de detecção de borda aplicados à Figura 2b. (a) Método do Laplaciano; (b) Método de Kirsch; (c) Método de Canny; (d) Método do Sobel; (e) Método de Roberts; (f) Método de Prewitt.

4.5. Morfologia Matemática

Uma das técnicas bastante utilizadas no tratamento de imagens é a morfologia matemática. A morfologia foi desenvolvida inicialmente para imagens binários e foi posteriormente estendida para tons de cinza. Essa técnica procura preencher espaços vazios ou

remover *pixels* vazios a fim de ajudar a diferenciar os objetos contidos em uma imagem. As operações básicas da morfologia são a erosão ao qual *pixels* que não atendem um dado padrão são apagados da imagem, e a dilatação, em que pequenos espaços vazios são preenchidos com outros *pixels* de mesmo padrão. Na *toolbox* desenvolvida, foram melhoradas a funcionalidade ao qual foram simplificadas para melhor utilização. Na “Figura 9” são demonstrados os processos de Erosão, Dilatação, Erosão seguida de Dilatação e Dilatação seguida de Erosão.



Figura 9: Morfologia matemática aplicado à Figura 1a. (a) Erosão; (b) Dilatação; (c) Erosão seguida de Dilatação; (d) Dilatação seguida de Erosão;

4.6. Afinamento

A operação denominada afinamento procura remover todos os *pixels* redundantes em uma imagem criando uma nova imagem simplificada com largura de um único *pixel*. O problema para os algoritmos de afinamento é o de determinar quais *pixels* são redundantes em uma imagem. Este processo costuma envolver várias interações até que não todos os critérios tenham sido alcançados. Os principais critérios para afinar uma imagem são:

- As regiões precisam ter largura de um *pixel*
- Os *pixels* da imagem resultante precisam ficar próximos ao centro da região de cruzamento de regiões
- O número de esqueletos sejam o mesmo que o número de regiões inicial da imagem

Existem alguns algoritmos clássicos na literatura, os mais famosos: Stanford, Zhang-Suen e Holt foram implementados na *toolbox* didática. Na “Figura 10” são apresentados os resultados dos algoritmos de afinamento para uma impressão digital.

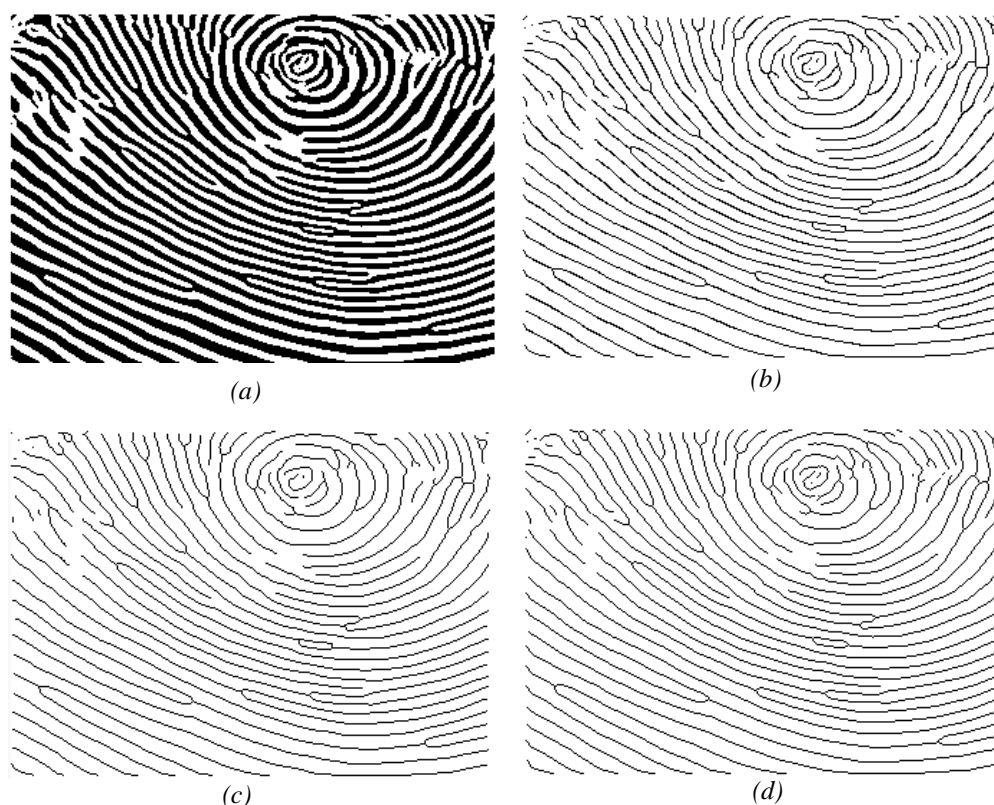


Figura 10: Técnicas de afinamento. (a) Imagem Original; (b) Método de Zhang-Suen; (c) Método de Holt; (d) Método de Stentiford.

5. CONSIDERAÇÕES FINAIS

O OpenCV fornece diversos métodos de processamento digital de imagens, aliado ao desempenho e a portabilidade. A *toolbox* desenvolvida facilitou ainda mais a relação ensino-aprendizado da disciplina de Processamento Digital de Sinais ao mesmo tempo que auxilia o desenvolvimento de novas aplicabilidades.

Em uma perspectiva didática, a *toolbox* representa uma nova metodologia para o ensino de Processamento Digital de Imagens, haja vista que professor e alunos poderão ter meios para avaliar em uma situação prática as teorias abordadas na disciplina acadêmica. Sua vantagem está na facilidade de implementar as rotinas de PDI dentro de projetos de *software* desenvolvidos em linguagem C++, isto é importante quando se tem necessidade de realizar apresentações experimentais durante uma aula didática, situação esta muito comum no processo de ensino-aprendizagem de assuntos abordados nos cursos da área de tecnologia.

Para trabalhos futuros, pretende-se ampliar ainda mais o acervo da *toolbox* com novos métodos, otimizando cada vez mais as funções para que continuem inseridas no contexto da eficiência e aplicabilidade em tempo real.



6. REFERÊNCIAS / CITAÇÕES

ALAKARHU, Juha; DINNING, Damian e SALMELIN, Eero. NOKIA PureView imaging technology. Disponível em: <<http://press.nokia.com/wp-content/uploads/mediaplugin/doc/nokia-808-pureview-whitepaper.pdf>> Acesso em: 8 de junho de 2014.

BATISTA, L. V.; Introdução ao Processamento Digital de Imagens, Apostila, UFPB, 2005.

BRADSKI, G. e KAEHLER, A. Learning OpenCV, Computer Vision with OpenCV Library. Ed. O'REILLY Media Inc., United States, September of 2008.

CANNY, J.; "A Computational Approach to Edge Detection, IEEE Tran. Pattern Analysis and Machine Intelligence", 8(6), pp. 679—698, 1986.

FILHO, O. M. e NETO, H. V. "Processamento Digital de Imagens", Brasport, Rio de Janeiro, 1999.

GONZALES, R. C. e WOODS, R. E. "Digital Image Processing", Trird Edition, 976 pages, Prentice Hall, August of 2007.

MELLO, Carlos A; "Processamento Digital de Sinais", Centro de Informática, UFPE, 2013. Disponível em: <<http://www.cin.ufpe.br/~cabm/pds/PDS.pdf>>. Acesso em 9 de junho de 2014.

OTSU, N.; "A threshold selection method from gray-level histograms". *IEEE Trans. Sys., Man., Cyber.* 9 (1): 62–66, 1979

PARKER, J. R.; "Algorithms for Image Processing and Computer Vision", 2ª edição, Ed. Wiley, 504 Páginas, Dezembro de 2010.

RIDLER, T. W. e CALVARD, S.; "Picture Theresholding Using an Iterative Selection Method", IEEE Transaction on Systems, Man, and Cybernetics, v. SMC 8, no.8, August 1978.

SHAH, Agam. Qualcomm reveals 64-bit Snapdragon 810 and 808. Disponível em: <<http://www.greenbot.com/article/2140500/qualcomm-reveals-64-bit-snapdragon-810-its-fastest-ever-mobile-chip.html>> Acesso em: 8 de junho de 2014.

ZHANG, T. Y. e SUEN, C. Y.; "A Fast Parallel Algorithm for Thinning Digital Patters", Communications of the ACM, V. 23, N. 3, March 1984.



DEVELOPMENT OF A TOOLBOX FOR TEACHING-LEARNING AID OF DIGITAL IMAGE PROCESSING

Abstract: *A common and recurrent framework of engineering education is the use pre-programmed software for the practical realization of various projects and activities. One of the classic examples is the use of MatLab® who have earned enough space in Brazil by practicality, large collection of algorithms and methods. All this practicality involves a high computational cost, which prevents direct code implementation in embedded systems that have gained a very large market share in recent years. Digital Image Processing (DIP) is an area of very productive and challenging scientific research, with a diverse range of applications in various branches of science, business and technical initiative. A new way to accelerate teaching and learning as well as design knowledge to practical and more efficient applicability of this area is to use specific preprogrammed libraries for this purpose. OpenCV is one of the most prestigious libraries to perform activities related to DIPs. The spread and development of a toolbox are proposed in this article.*

Key-words: *OpenCV, Digital Image Processing, Teaching in Engineering.*