

Prova 1:

Filtragem Digital, Transformação Linear e Histograma

Rafael Luiz Franco de Campos



1 INTRODUÇÃO

Quando não se tem uma imagem com uma boa qualidade ou algum tipo de distorção e esta é utilizada em algum tipo de aplicação, como em imagens médicas (raio-X, ressonância magnética, etc.), deve ser feita a correção necessária para facilitar o diagnóstico médico por exemplo. A utilização de filtros em processamento digital de imagens têm como objetivo principal extrair informações e transformar a imagem, eliminando ruídos, realçando bordas e segmentando [1]. Em geral, a aquisição de uma imagem gera erros, que são chamados "ruídos", os quais causam uma deteriorização da imagem, como por exemplo, em fotografias de satélites em que a turbulência do ar altera a imagem original. Torna-se preciso então, utilizar algoritmos para restaurar essas imagens. O uso de filtros e a técnica de realce de imagens são poderosas ferramentas usadas na restauração de imagens.

Como o objetivo geral é processar uma imagem para obter da mesma, resultados que sejam mais apropriados que a imagem inicial, para uma aplicação específica, neste relatório serão abordados os principais filtros de suavização lineares e não lineares utilizados para a eliminação de ruídos e detecção de bordas em imagens, bem como a aplicação da transformação linear e o uso do histograma.

2 FILTRAGEM DIGITAL

A filtragem digital normalmente tem como objetivo realçar ou suavizar uma imagem, corrigindo ruídos ou o contraste, melhorando muitas vezes, a qualidade da imagem. Os métodos de filtragem de imagens são normalmente classificados em duas categorias: as técnicas de filtragem espacial e as técnicas de filtragem no domínio da frequência [2].

2.1 Filtragem no domínio espacial

As técnicas de filtragem no domínio espacial são aquelas que atuam diretamente sobre a matriz de pixels que é a imagem digitalizada, normalmente utilizando operações de convolução com máscaras, ou seja, trabalha com uma máscara, uma janela, que percorre toda a imagem realizando as operações de filtragem desejadas. Em geral, esta técnica é computacionalmente mais eficiente e requer menos recursos de processamento para sua realização.

A manipulação da imagem utilizando as técnicas desse domínio, removendo determinada característica ou ainda destacando/suavizando determinado detalhe, somente é possível se conhecer a posição de cada pixel de uma

imagem. Existem dois tipos de filtros: lineares e os não lineares [3].

2.1.1 Filtros lineares

Usam máscaras que realizam somas ponderadas das intensidades de pixels ao longo da imagem (convolução). Este filtro destaca/suaviza a imagem sem alterar o nível de cinza da imagem de origem. Ele cria uma imagem nova excluindo as linhas e colunas sem diminuir a resolução. Nesta classificação de filtros, encontram-se os filtros lineares de suavização (filtros passa-baixa), sendo o mais conhecido o filtro de média.

2.1.2 Filtros não-lineares

Diz respeito a qualquer filtro que não é uma soma ponderada de pixels. Os filtros não-lineares alteram a imagem, sem diminuir sua resolução e tem como principais aplicações: a segmentação, a eliminação de ruídos, o reconhecimento de texturas e formas, a detecção de bordas, linhas e manchas, aplicando modificações sem se preocupar com níveis de cinza da imagem de origem. Para detectar determinadas características, como a borda, utiliza-se de operadores, sendo Roberts e Sobel filtros exemplos de detecção de borda. Outro filtro bastante utilizado não linear é o da mediana, que busca preservar as bordas e detalhes da imagem, ou então diminuir a aparência borrada da imagem.

2.2 Filtragem no domínio da frequência

Consiste em transformar a representação da imagem no espaço para o domínio da frequência, aplicando, por exemplo, a transformada de Fourier na imagem, realizando-se em seguida a filtragem desejada e depois calculando-se a transformada inversa de Fourier para obter o resultado da imagem filtrada [4]. Em suma, a ideia é transformar cada pixel da imagem do domínio espacial para o domínio de frequência usando a Transformada de Fourier. Após a transformação, aplicar filtro(s) na imagem digital através das técnicas de processamento de imagens. Por fim, é realizado o processo inverso, onde a imagem obtida da aplicação do(s) filtro(s) é transformada novamente no domínio espacial, utilizando a Transformada Inversa de Fourier, gerando assim uma nova imagem processada.

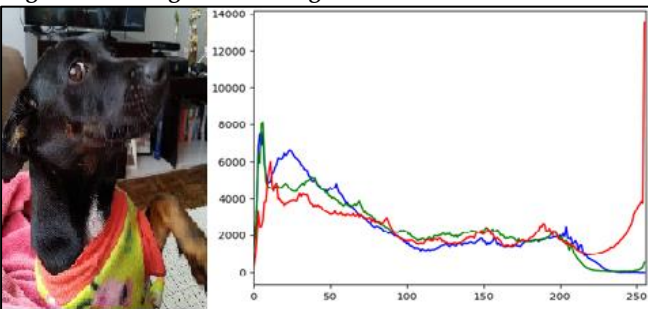
3 TRANSFORMAÇÃO LINEAR

Em Matemática, uma transformação linear é um tipo particular de função entre dois espaços vetoriais que preserva as operações de adição vetorial e multiplicação por escalar. Uma transformação linear também pode ser chamada de aplicação linear ou mapa linear [5]. Ela é usada para girar, rotacionar, diminuir, aumentar coisas no plano e no espaço, e podem ser aplicadas em imagens, já que formam uma matriz “gigante”, transformando assim, cada valor na matriz [6].

4 HISTOGRAMA

Um histograma é uma ferramenta de análise e representação de dados quantitativos, agrupados em classes de frequência que permite distinguir forma, o ponto central e a variação da distribuição, além de outros dados como amplitude e simetria na distribuição dos dados [7]. Na imagem, o histograma pode ser usado para demonstrar a quantidade de pixels com um determinado valor, no caso da Figura 1, apresenta a frequência dos valores dos pixels nas camadas RGB da imagem.

Figura 1 – Histograma da imagem RGB



Fonte: Autoria Própria.

5 DESENVOLVIMENTO

Como solução das questões da prova, foi desenvolvido um programa na linguagem de programação Python, com os filtros propostos, aplicação da transformação linear e representação do histograma. O código pode ser encontrado no repositório do GitHub [8].

5.1 Questão 1

Essa questão propõe a aplicação dos operadores: Canny, Prewitt, Sobel, Roberts, Média, Gaussiano e Laplaciano. Os filtros foram aplicados em uma imagem em tons de cinza.

5.1.1 Canny

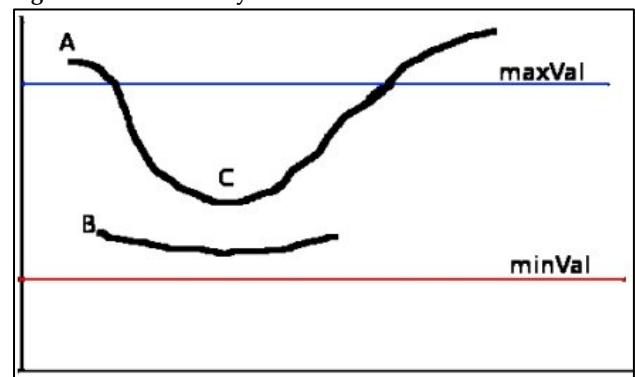
O Canny Edge Detection é um algoritmo popular de detecção de bordas. Foi desenvolvido por John F. Canny em 1986 [9]. Basicamente o Canny envolve:

1. Aplicar um filtro gaussiano para suavizar a imagem e remover o ruído.
2. Encontrar os gradientes de intensidade da imagem.
3. Aplicar Sobel duplo para determinar bordas potenciais.
4. Aplicar o processo de “hysteresis” para verificar se o

pixel faz parte de uma borda “forte” suprimindo todas as outras bordas que são fracas e não conectadas a bordas fortes.

Para a aplicação do operador Canny, foi utilizada a função fornecida pelo próprio opencv, `cv2.Canny()`. Tem como primeiro parâmetro a imagem que se deseja aplicar, o segundo e terceiro argumentos são os valores `minVal` e `maxVal`, respectivamente. Esses valores são usados no passo 4 (hysteresis), onde as arestas acima do `maxVal` são consideradas arestas seguras (bordas), arestas abaixo de `minVal` não são consideradas bordas. Arestas entre `minVal` e `maxVal`, podem ser consideradas bordas se tiverem conectadas a arestas seguras, caso não estejam conectadas serão desconsideradas, como mostra a figura abaixo:

Figura 2 – Limiar de hysteresis



Fonte: Canny Edge Detection - OpenCV.

5.1.2 Sobel

Esse operador se baseia na convolução da imagem com um filtro pequeno, separável e com valor inteiro na direção horizontal e vertical e, portanto, é relativamente barato em termos de cálculos [10].

Figura 3 – Máscara Sobel

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fonte: Comparing Edge Detection Methods.

Para a aplicação do operador sobel, foi utilizada a recomendação de um tutorial do próprio opencv, onde dizia para converter a imagem para outro tipo (`cv2.CV_64F`) aplicar o filtro e converter novamente para o tipo original, já que aplicando o operador no outro tipo surtiria mais efeito.

5.1.3 Prewitt

O operador Prewitt é semelhante ao operador Sobel e é usado para detectar bordas verticais e horizontais nas imagens. No entanto, diferentemente do Sobel, esse operador não enfatiza os pixels mais próximos do centro da máscara.

Figura 4 – Máscara Prewitt

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Fonte: Algoritmos de Detecção de Bordas Implementados em FPGA [11].

Para a implementação do operador Prewitt, foram usados kernels (kernelx para detector as bordas horizontais e kernely para detector as bordas verticais) com os valores da máscara de Prewitt. Depois foram somadas as duas imagens resultantes da aplicação dessas máscaras e então exibida a imagem final.

5.1.4 Roberts

No operador de Roberts calcula as diferenças de níveis de cinza por diferenças cruzadas. Este operador é o menos indicado, pois o resultado contém muitos ruídos, e como as máscaras têm dimensão 2X2, ele não destaca muito a visualização da borda quando comparado com às das outras máscaras estudadas.

Figura 5 – Máscara Roberts

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Fonte: Algoritmos de Detecção de Bordas Implementados em FPGA.

Na aplicação do operador Roberts, foram usados kernels (kernelx para detector as bordas horizontais e kernely para detector as bordas verticais) com os valores da máscara de Roberts. Depois foram somadas as duas imagens resultantes da aplicação dessas máscaras e então exibida a imagem final.

5.1.5 Média

É um filtro simples, intuitivo e fácil de ser desenvolvido, porém, eficiente para o propósito de filtrar imagens, pois ele reduz a quantidade de variação de intensidade entre um pixel e seus vizinhos, eliminando ruídos. A ideia é simplesmente substituir cada valor de pixel em uma imagem com o valor médio de seus vizinhos, incluindo ele mesmo, o que produz o efeito de eliminar os valores de pixels que são representativos de seus arredores.

Figura 6 – Filtro da Média

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Fonte: Processamento Digital de Imagens.

Para a aplicação do filtro da media, foi implementada uma função simulando a aplicação de uma máscara 3x3.

5.1.6 Gaussiano

O filtro Gaussiano geralmente é usado como um filtro de passa-baixa por deixar passar as baixas frequências, mas elimina os valores relacionados às altas frequências. O filtro Gaussiano tem esse nome por usar a função Gaussiana para obter os valores da mascara Figura.

Figura 7 – Função Gaussiana

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

Fonte: Técnicas de Suavização de Imagens e Eliminação de Ruídos [12].

A figura 8 mostra a máscara típica para implementar o filtro Gaussiano.

Figura 8 – Máscara 5x5 com desvio padrão 1

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Fonte: Técnicas de Suavização de Imagens e Eliminação de Ruídos.

Para a aplicação do filtro gaussiano foi utilizada a função fornecida pelo opencv cv2.GaussianBlur(), sendo o primeiro parâmetro a imagem e o Segundo a o tamanho da mascara que deseja aplicar.

5.1.7 Laplaciano

Laplaciano é um pouco diferente dos métodos que discutimos até agora. Diferentemente dos detectores de borda de Sobel e Prewitt, esse detector de bordas usa apenas um núcleo. Ele calcula derivadas de segunda ordem em uma única passagem. Dois núcleos pequenos comumente usados são:

Figura 9 – Núcleos Laplacianos

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Fonte: Comparing Edge Detection Methods.

Para a aplicação do filtro gaussiano foi utilizada a função fornecida pelo opencv cv2.Laplacian(), sendo o primeiro parâmetro a imagem e o Segundo o tipo.

5.2 Questão 2

Essa questão propõe o uso da transformação linear para diminuir os níveis de energia de uma imagem colorida. Foi então desenvolvida uma função que percorre toda a imagem colorida passando nos 3 canais de cor RGB. Através do intervalo que foi calibrado visualmente e fornecido a transformação linear, é verificado se o pixel possui valor abaixo do limite mínimo, se está acima do limite máximo ou se está entre esse intervalo da transformação linear. Para cada caso aplica um cálculo. Como a ideia é diminuir a energia da imagem, nos pixels com maior intensidade são aplicadas subtrações do valor para então tornar a imagem escura, mas sem distorcer a mesma.

5.3 Questão 3

Essa questão propõe o uso de histograma e transformação linear para separar a parte mais escura da imagem. Para a solução da questão, a imagem é convertida para tons de cinza e então foi plotado o histograma da imagem, verificado o valor mais indicado para ser o limite mínimo e o limite máximo. A partir desse limites, é utilizada a transformação linear para separar a imagem em tons escuros, médios e claros. Então são exibidas as imagens resultantes com seus respectivos histogramas.

6 UTILIZAÇÃO

Para as questões 1 e 2, é fornecido um menu com as opções de escolha entre essas questões. Para verificar os resultados de qualquer uma das duas, basta selecionar uma opção no menu. Para visualizar o resultado da questão 3, basta executar o código.

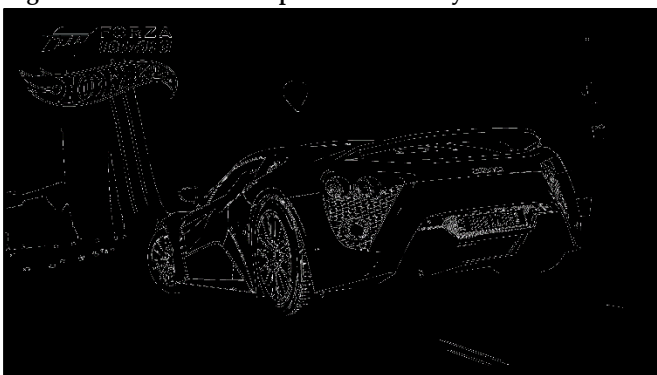
7 RESULTADOS

Os resultados de cada questão podem ser visualizados abaixo.

7.1 Questão 1

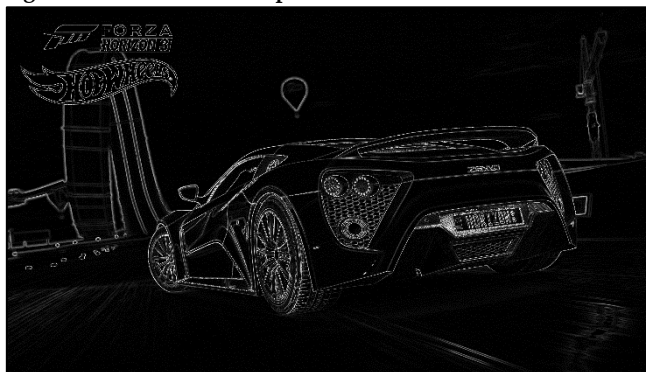
As imagens obtidas após a aplicação de cada operador podem ser visualizadas abaixo.

Figura 10 – Resultado do operador de Canny



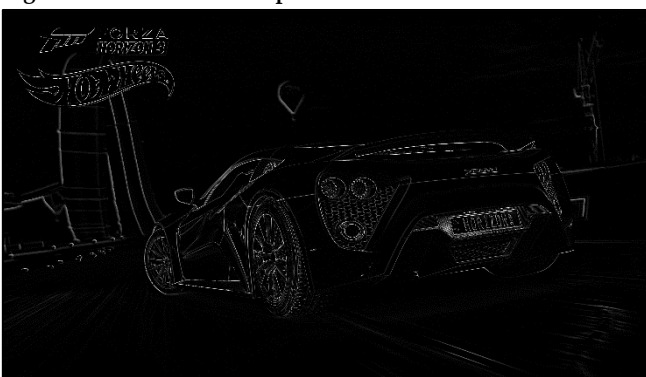
Fonte: Autoria Própria.

Figura 11 – Resultado do operador Sobel



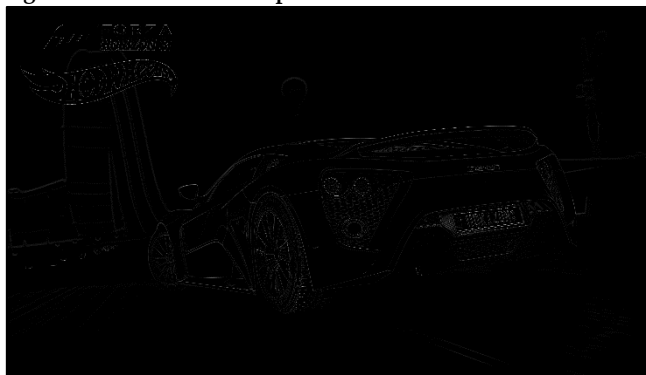
Fonte: Autoria Própria.

Figura 12 – Resultado do operador Prewitt



Fonte: Autoria Própria.

Figura 13 – Resultado do operador Roberts



Fonte: Autoria Própria.

Figura 14 – Resultado do filtro de Média



Fonte: Autoria Própria.

Figura 15 – Resultado do filtro Gaussiano



Fonte: Autoria Própria.

Figura 16 – Resultado do operador Laplaciano



Fonte: Autoria Própria.

7.2 Questão 2

A imagem obtida após a aplicação da transformação linear para a redução dos níveis de energia é demonstrada pela Figura 17.

Figura 17 – Resultado da Transformação linear

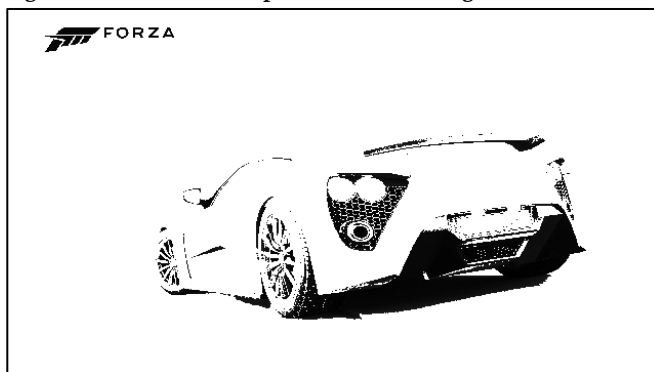


Fonte: Autoria Própria.

7.3 Questão 3

As imagens obtidas da separação após a aplicação da transformação linear das partes escuras da imagem podem ser visualizadas abaixo. Uma observação que pode ser feita, é o fato de que poderia ter sido utilizado o mesmo intervalo da limiarização, separando assim a imagem em dois tons, escuros e claros.

Figura 18 – Resultado da parte escura da imagem



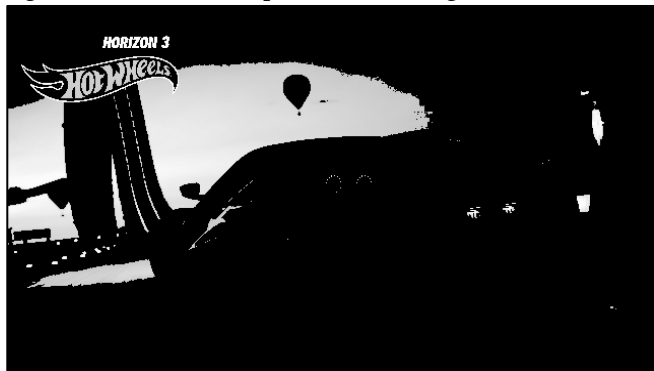
Fonte: Autoria Própria.

Figura 19 – Resultado da parte média da imagem



Fonte: Autoria Própria.

Figura 20 – Resultado da parte clara da imagem



Fonte: Autoria Própria.

REFERÊNCIAS

- [1] PIRES, V. B; BARCELOS, C. A. Z. **Filtros de Suavização para a Eliminação de Ruídos**. Universidade Federal de Goiás. Disponível em: <<https://projetos.extras.ufg.br/conpeex/2004/pibic/exatas/Vinicius.html>>. Acesso: 21 set. 2019.
- [2] MARQUES, O. Filho; VIERA, H. Neto. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. Disponível em: <<http://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>>. Acesso: 20 set. 2019.
- [3] **Filtragem no domínio espacial**. Disponível em: <https://pt.wikipedia.org/wiki/Filtragem_no_dom%C3%ADnio_especial>. Acesso: 21 set. 2019.
- [4] **Filtragem de Imagens**. Capítulo 5. Disponível em:

- <<http://computacaografica.ic.uff.br/transparenciasvol2cap5.pdf>>. Acesso: 20 set. 2019.
- [5] SCHWARTZ, E. **Transformações Lineares No Processamento De Imagens E Na Criptografia**. 2016. Disponível em: <http://dspace.nead.ufsj.edu.br/trabalhospublicos/bitstream/handle/123456789/68/EDUARDO%20SCHWARTZ_12293_assignment_file_TCC-Transforma%C3%A7%C3%B5es%20Lineares.pdf?sequence=1&isAllowed=y>. Acesso: 22 set. 2019.
- [6] OLIVEIRA, G. **O que são transformações lineares?**. Disponível em: <<https://medium.com/@guilhermeoliveira/o-que-s%C3%A3o-transforma%C3%A7%C3%B5es-lineares-26b0eb359825>>. Acesso: 22 set. 2019.
- [7] InfoEscola. **Histograma**. Disponível em: <<https://www.infoescola.com/estatistica/histograma/>>. Acesso 22 set. 2019.
- [8] Repositório GitHub. Disponível em: <<https://github.com/RafaelLFCampos/Prova1VC>>. Acesso: 22 set. 2019.
- [9] **Canny Edge Detection**. Disponível em: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html>. Acesso: 22 set. 2019.
- [10] TSANKASHVILI, N. **Comparing Edge Detection Methods**. Disponível em: <<https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>>. Acesso: 21 set. 2019.
- [11] SALLES, P. Maturana. **Algoritmos de Detecção de Bordas Implementados em FPGA**. Universidade Estadual Paulista. Disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/273-dissertacao_patricia.pdf>. Acesso: 22 set. 2019.
- [12] SANCHES, C. H.; FONTOURA, P. J.; VIERA, P. F.; BATISTA, M. A. **Técnicas de Suavização de Imagens e Eliminação de Ruídos**. Disponível em: <<http://eati.info/eati/2015/assets/anais/Longos/L2.pdf>>. Acesso: 22 set. 2019.