

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA

PSE - Image

Trabalho Prático de Processamento de Imagens Digitais

Bernardo Olímpio

Caio Piantino

Débora Deslandes

Guilherme José

Leonardo Machado

Leandro

Rafael Oliveira

BELO HORIZONTE, 2017

DESENVOLVIMENTO

O trabalho prático teve como proposta a implementação de um programa, implementado em Java, com o objetivo de demonstrar, em uma interface amigável, a aplicação de diversos processos utilizados no processamento de imagens digitais.

Processamento de imagem é qualquer forma de processamento de dados no qual a entrada e saída são imagens. A maioria das técnicas envolve o tratamento da imagem como um sinal bidimensional, no qual são aplicados padrões de processamento de sinal.

Para esse trabalho foram escolhidos os filtros de passa alta e de passa baixa com mascara de convolução, transformações de imagens coloridas em imagens em tons de cinza, transformação da imagem em seu negativo. Também foram implementados métodos de segmentação, interpolação, limiarização, brilho, contraste e plotagem do histograma da imagem.

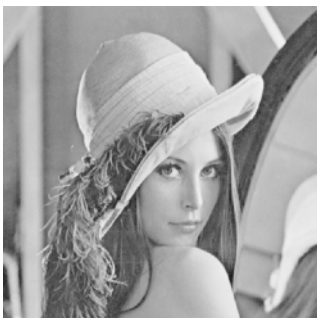
DEFINIÇÕES

FILTRO DE MÉDIA

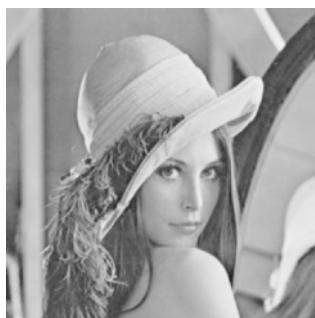
O filtro de média é um filtro linear de suavização que trabalha calculando diretamente o valor da média de uma área quadrada de pixels ao entorno do pixel desejado.

Também conhecidos por ser um dos vários tipos de filtros passa baixa, esse processamento é usado para que se tenha a perda de nitidez da imagem em que está sendo aplicado.

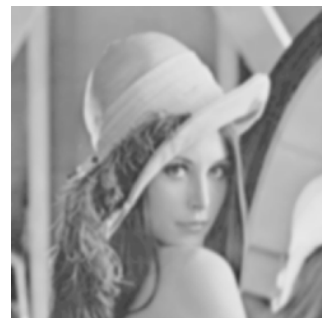
Ao aplicar o filtro de média, o valor de cada pixel analisado, terá seu valor alterado, pela média dos pixels que são vizinhos a estes.



$\alpha = 0$



$\alpha = 1$



$\alpha = 30$

FILTRO GAUSSIANO

O filtro gaussiano é um filtro passa baixa usado para borrar ou desfocar imagens na qual ele é aplicado. Utilizasse a curva de Gauss que determina qualquer probabilidade de uma distribuição normal presente na imagem.

Tem como objetivo reduzir os ruídos presentes na imagem, tendo como resultado a suavização da mesma, fazendo parecer que está sendo vista através de uma lente fora de foco. Seu processo é largamente utilizado no estágio de pré-processamento de imagens para enaltecer a estrutura da imagem em diferentes escalas.

Assim como o filtro de média simples, se utiliza da convolução da imagem, onde o pixel analisado recebe o somatório dos valores de cada um dos seus vizinhos multiplicados pelo valor da mascara usada, e dividido pela soma dos valores da mesma. Sua função genérica é dada pela seguinte fórmula:

$$GB[I]_p = \sum_{q \in S} G_{\sigma}(\|p - q\|) I_q,$$

onde

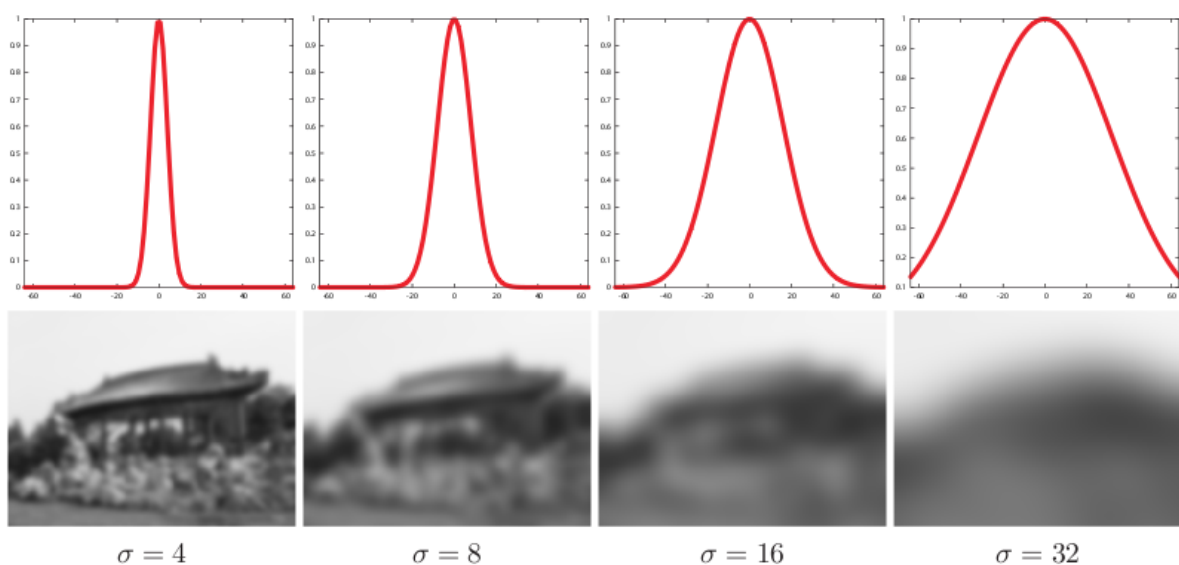
$$G_{\sigma}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

O filtro é obtido através da geração de uma matriz Gaussiana que será utilizada como mascara na convolução da imagem. Esta matriz utilizará a função acima, atribuindo valores a X,Y e S.

0.00078631	0.00655952	0.01330347	0.00655952	0.00078631
0.00655952	0.05472049	0.11097944	0.05472049	0.00655952
0.01330347	0.11097944	0.22507904	0.11097944	0.01330347
0.00655952	0.05472049	0.11097944	0.05472049	0.00655952
0.00078631	0.00655952	0.01330347	0.00655952	0.00078631

Tabela 1 – exemplo de um filtro gaussiano

O resultado da aplicação pode ser vista a seguir:

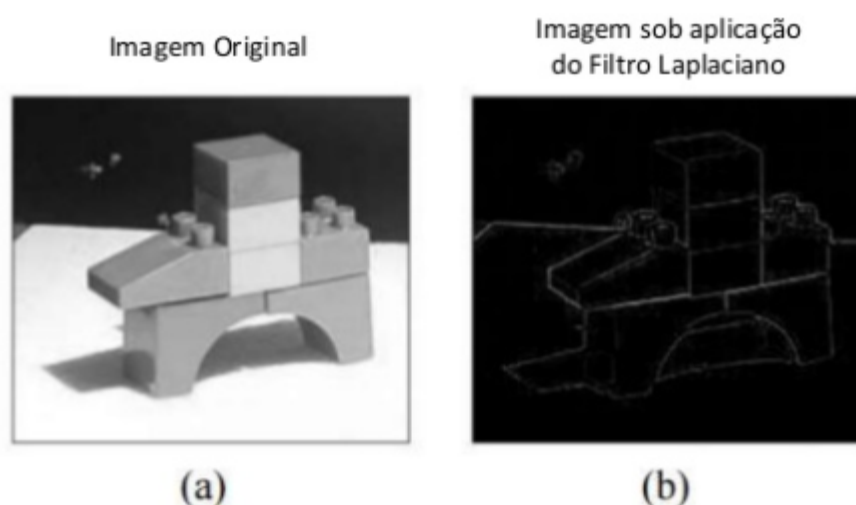


FILTRO LAPLACIANO

É um filtro passa alta de detecção de bordas, que serve para detectar e gerar uma imagem de borda fina após a aplicação da máscara.

O Laplaciano de uma imagem é a derivada segunda do valor da função que descreve a intensidade dos seus pixels. Sua função é representada por:

$$\nabla^2 f(x, y) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$$



FILTRO SOBEL

Se trata de um filtro usado para detectar bordas, realçar linhas horizontais e verticais mais escuras do que o fundo, sem realçar pontos isolados. Destaca as bordas das imagens em que sua máscara é aplicada.

Para sua aplicação, é necessária a utilização de duas matrizes (máscaras), normalmente 3x3, que são convoluídas com a imagem original, sendo 1 delas para as variações horizontais e outra para as verticais. Fazendo a convolução destas máscaras com a imagem gera-se os gradientes verticais e horizontais da imagem que são posteriormente combinadas.

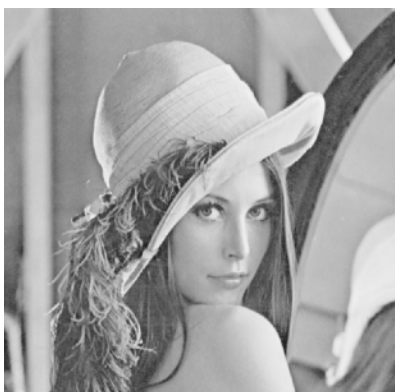
(matrizes para horizontal e vertical)

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{e} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

$$\Theta = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$



ALTERAR IMAGEM PARA TONS DE CINZA

Geralmente é um método usado para converter uma imagem colorida em tons de cinza. Geralmente se aplica para calcular a intensidade da luz em cada pixel em cada faixa do espectro magnético de uma imagem.

Para se converter uma imagem em tons de cinza, devesse utilizar uma escala RGB, adicionando 30% do vermelho, 59 % do verde e 11% do azul independente da escala. O resultado é o nível desejado de cinza.



PEGAR O NEGATIVO DA IMAGEM

É uma função linear inversa, em que o contraste em áreas escuras torne-se claro, e vice-versa. Serve para preservar a imagem, além de poder recupera-la caso ocorra a perda da mesma.

Aplicasse o contraste em áreas de baixos valores para que se tornem altos valores e vice-versa. Sua função pode ser representada por :

$$Y = - (AX + B)$$

Onde:

Y = novo valor do pixel

X = valor original do pixel

A = inclinação da reta (tangente do ângulo)

B = fator de incremento, definido pelos limites mínimo e máximo fornecidos pelo usuário.



LIMIARIZAÇÃO

É um processo de segmentação de imagens que se baseia na diferença dos níveis de cinza que compõem diferentes objetos de uma imagem. Serve para converter imagens em tons de cinza para imagens binárias, por isso essa técnica é conhecida também como binarização.

Para que ocorra a limiarização, precisa-se saber primeiro o que deseja isolar na imagem. Então a imagem pode ser segmentada em dois grupos. Um grupo de pixels com níveis de cinza abaixo do limiar e um grupo com níveis de cinza acima do limiar. Atribui-se então um valor fixo para todos os pixels de mesmo grupo com a seguinte função:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

1 = RGB 255 (branco) e 0 = RGB 0 (preto)

sendo assim a imagem binária.



HISTOGRAMA

Histogramas são conhecidos como distribuição de intensidades e Função de Densidade de probabilidade. A distribuição é representada por colunas discretas que não podem ser divididas ou “quebradas”, correspondentes a números inteiros (em contraposição a números fracionários). Esse conceito assume importância ao se tratar de realce de contraste em imagens. O histograma normalizado é dado por :

$$P_r(r_k) = \frac{n_k}{n}$$

onde $0 \leq r_k \leq L-1$

$k = 0, 1, \dots, L-1$, onde L é o número de níveis de cinza da imagem;

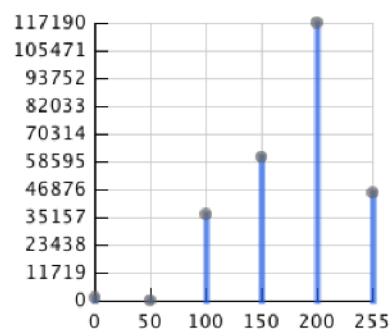
n = número total de pixels da imagem;

n_k = número de pixels cujo nível de cinza corresponde à k ;

$Pr(r_k)$ = Probabilidade do K -ésimo nível de cinza;

Para que serve?

Ele é usado para representar dados digitais de uma imagem.



INTERPOLAÇÃO BILINEAR

A interpolação bilinear é uma extensão da interpolação linear, sendo esta a interpolação de duas funções com duas variáveis. Basicamente é a interpolação linear aplicada duas vezes, contudo em direções diferentes.

É uma técnica de revezamento, usada para se ter uma imagem um pouco mais realista. A interpolação bilinear pode ser usada onde a transformação de imagem perfeita com a correspondência de pixels é impossível, de modo que se pode calcular e atribuir valores de intensidade apropriados aos pixels.

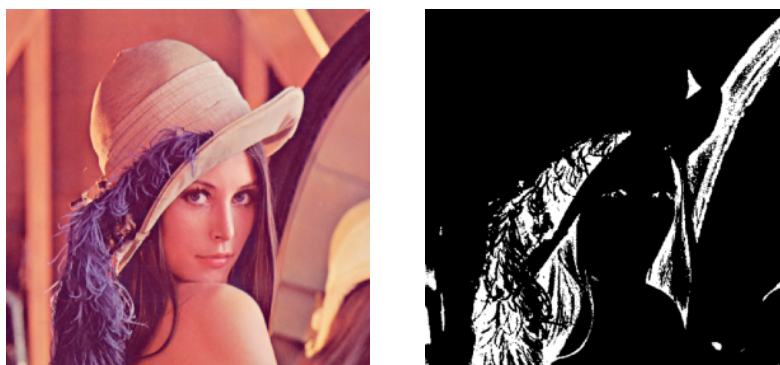
Basicamente ele irá mapear uma localização de pixel de tela para um ponto correspondente no mapa de textura. Uma média ponderada dos atributos como cor e transparência, dos quatro pixels adjacentes é calculada e aplicada ao pixel da tela. Este processo é repetido para cada pixel que forma o objeto sendo texturado.

SEGMENTAÇÃO

É o processo de partição de uma imagem digital em vários segmentos (conjunto de pixels). Seu objetivo é simplificar e/ou alterar a representação de uma imagem para que seja mais significativo e mais fácil de analisar. A segmentação localiza na imagem objetos e limites como linhas e curvas nas imagens.

Ela rotula cada pixel em uma imagem, de modo que os pixels com o mesmo rótulo compartilhem certas características como cor, intensidade ou textura. O resultado disso é um conjunto de segmentos que coletivamente cobrem toda a imagem ou um conjunto de contornos extraídos da imagem.

Filtrando cores azuis principalmente



TRANSFORMADA DE HOUGH

Inicialmente um método elaborado para identificar características em imagens binárias. A Transformada de Hough é um método padrão para detecção de formas que são facilmente parametrizadas (linhas, círculos, elipses, etc.) em imagens digitalizadas.

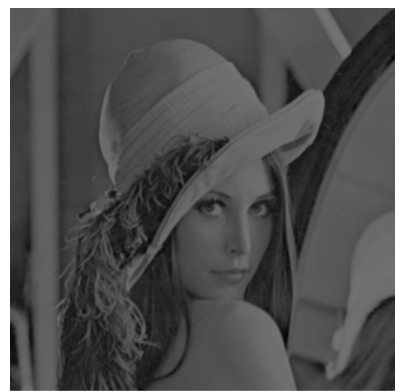
Deve ser aplicada na imagem uma transformação de forma que todos os pontos pertencentes a uma mesma curva sejam mapeados num único ponto de um novo espaço de parametrização da curva procurada.

TRANSFORMADA DE FOURRIER

É uma ferramenta matemática usada na solução de problemas de processamento digital. A transformada possui duas formas, a discreta e a rápida, pois é com elas que ocorre a mudança de domínio de espaço para domínio de frequência. Não há perda de informação durante a mudança de domínios, tendo somente mudado a representação.

BRILHO

É uma variação de intensidade da luz que irradia ou reflete a partir de uma imagem.



CONTRASTE

É a variação nas tonalidades de luz e sombra; claro e escuro; zonas opacas e transparentes na imagem.



RECURSOS USADOS NO TRABALHO

Este trabalho foi desenvolvido em Java, onde foram utilizadas os pacotes **java.awt**, **javax.swing**, **java.io** e **java.util**. O pacote **java.awt** foi utilizado para criar a interface de usuário e ajuda a pintar gráficos e imagens. O pacote **javax.swing** foi usado por fornecer componentes leves que funcionam a partir de uma ação do usuário. O **java.io** para fornecer entradas e saídas do sistema através de fluxos de dados, serialização e sistema de arquivos, além de ter sido usado no back-end para a programação das funções de processamento de imagem implementadas. E o **java.util** que usa estruturas de coleções e no back-end.

MANUAL DO USUÁRIO



Ao abrir o programa você terá uma tela com um Menu Superior e um Menu Lateral. O Menu superior contém 6 botões de ações divididos por uma barra de ações, com as seguintes funções:

- Abrir
- Processar
- Original
- Linha do tempo, representada por uma barra
- Resultado
- Salvar
- Ajuda

ABRIR :

O botão de abrir permite selecionar a origem da foto que você pretende processar. Não existe restrição quanto ao formato da imagem.

PROCESSAR:

O botão de processar deve ser acionado quando já se tem uma imagem aberta e processamentos (encontrados no menu lateral) já selecionados na linha do tempo. Basta clicar que ele processará a imagem seguindo a ordem de processamentos previamente selecionados por você na linha do tempo.

ORIGINAL:

Permite a visualização da imagem original antes, facilitando a comparação com os processamentos.

LINHA DO TEMPO:

É uma linha com os processamentos selecionados a serem realizados.

É possível navegar por cada um dos processamentos na linha do tempo apenas clicando neles, o que permite a visualização dos efeitos gerados na imagem de uma forma mais perceptível.

Também é possível remover um processamento da linha do tempo clicando sobre ele com o botão da direita.

RESULTADO:

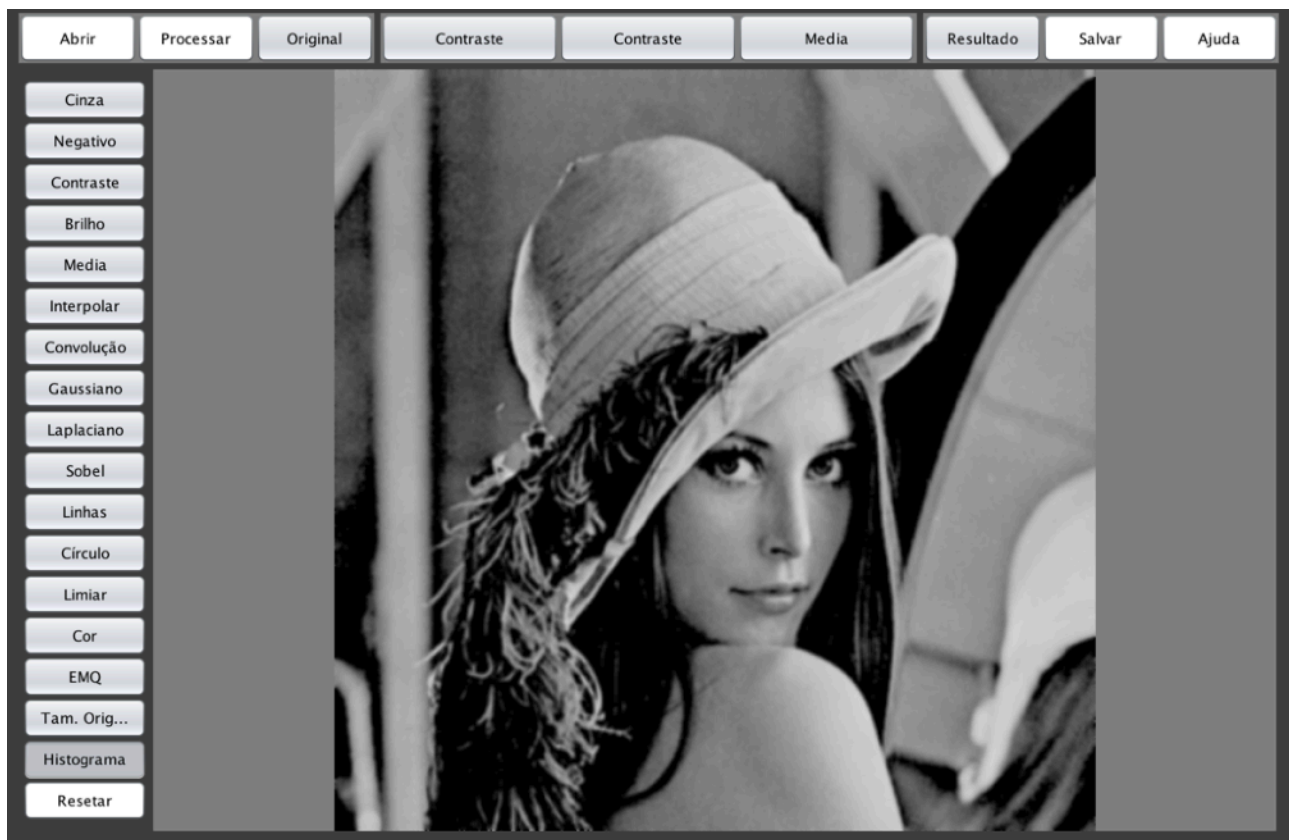
É a imagem final após todos os processamentos. Usado para facilitar a visualização, como é possível caminhar pelos processamentos, basta clicar em resultado para voltar ao resultado final dos processamentos.

SALVAR:

Salva a imagem processada como uma nova imagem.

AJUDA:

Acessa a documentação do projeto e o manual do usuário. Precisa de conexão à internet.



Menu Lateral:

Escala de cinza

O método de escala de cinza recebe a imagem do usuário, itera sobre os pixels da imagem, calculando a média dos canais RGB e seta o pixel da nova imagem com essa média.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Escala de Cinza”, adicionando-o à linha do tempo e selecionando “Processar”.

Negativo

O método negativo recebe a imagem do usuário, itera sobre os pixels da imagem, calcula a cor inversa pegando o valor de RGB e subtraindo o valor do pixel em questão. O novo valor é atribuído a uma nova imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Negativo”, adicionando-o à linha do tempo e selecionando “Processar”.

Média

O método de filtro média recebe a imagem do usuário, itera sobre os pixels da

imagem, cria a máscara de média, realiza uma convolução, divide os valores do RGB com os da máscara para se ter o novo valor do pixel e seta o valor da nova imagem com os valores encontrados.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Mídia”, adicionando-o à linha do tempo e selecionando “Processar”.

Gaussiano

O método Gaussiano recebe a imagem do usuário, itera sobre os pixels da imagem, cria a máscara de Gauss, realiza uma convolução, pega os valores das cores primárias de cada pixel após a convolução com a máscara, seta os novos valores na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Gaussiano”, adicionando-o à linha do tempo e selecionando “Processar”.

Laplaciano

O método Laplaciano recebe a imagem do usuário, itera sobre os pixels da imagem, cria duas máscaras, realiza uma convolução, pega os valores das cores primárias de cada pixel após a convolução com a máscara. Após se aplicar a primeira máscara repete o processo com a segunda máscara. Ocorre o arredondamento dos valores e então os novos valores são setados na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Laplaciano”, adicionando-o à linha do tempo e selecionando “Processar”.

Sobel

O método Sobel recebe a imagem do usuário, itera sobre os pixels da imagem, cria duas máscaras de Sobel, realiza uma convolução, pega os valores das cores primárias de cada pixel após a convolução com a máscara. Após se aplicar a primeira máscara repete o processo com a segunda máscara. Ocorre o arredondamento dos valores e então os novos valores são setados na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Sobel”, adicionando-o à linha do tempo e selecionando “Processar”.

Convolução

O método de convolução recebe a imagem, o número de linhas da máscara, o número de colunas da máscara e os pesos desejados pelo usuário. A partir destes dados cria a máscara. Percorre a imagem pegando os valores das cores primárias de cada pixel após a convolução com a máscara. Divide as cores pelo valor da máscara e aplica os novos valores na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, adicionar as linhas, colunas e pesos,

selecionar “**Convolução**”, adicionando-o à linha do tempo e selecionando “Processar”.

Brilho

O método de Brilho recebe a imagem e a intensidade em valor inteiro do brilho. Após receber a imagem e os valores, verifica o valor dado e então aplicado na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, clicar em “**Brilho**” com o botão da direita definir a quantidade do brilho desejado, selecionar “**Brilho**”, adicionando-o à linha do tempo e selecionando “Processar”.

Os valores do brilho são positivos para aumentar o brilho, negativos para remover o brilho e 0 para não alterar. Dessa forma temos:

- 0 = sem alteração
- 100 = aumentar o brilho em 100% (podendo ter um aumento maior a 100% se esse for o efeito desejado)
- -100 = diminuir o brilho em 100%(podendo diminuir mais que 100% do brilho se esse for o efeito desejado)

Contraste

O método de Contraste recebe a imagem e a intensidade em valor inteiro do contraste. Após receber a imagem e os valores, verifica o valor dado e então aplicado na imagem resultante.

Para utilizá-lo basta abrir a imagem desejada, clicar em “**Contraste**” com o botão da direita definir a quantidade do brilho desejado, selecionar “**Contraste**”, adicionando-o à linha do tempo e selecionando “Processar”.

Os valores do brilho são positivos para aumentar o brilho, negativos para remover o brilho e 0 para não alterar. Dessa forma temos:

- 0 = sem alteração
- 100 = aumentar o brilho em 100% (podendo ter um aumento maior a 100% se esse for o efeito desejado)
- -100 = diminuir o brilho em 100%(podendo diminuir mais que 100% do brilho se esse for o efeito desejado)

Limiar (Limiar Global Padrão (LGP))

O método de limiar recebe a imagem do usuário, itera sobre os pixels da imagem, calculando a média dos canais RGB e seta os pixels da nova imagem com 0, caso esteja abaixo da média, ou 255, caso esteja acima da média. É indicado transformar a imagem em escalas de cinza previamente, pois a média de pixels coloridos quando limiarizados tendem a deixar um leve ruído.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Limiar”, adicionando-o à linha do tempo e selecionando “Processar”.

Histograma

O método de histograma recebe a imagem do usuário, itera sobre os pixels da imagem, calculando a quantidade de pixels

Para utilizá-lo basta abrir a imagem desejada, selecionar “Histograma”, adicionando-o à linha do tempo e selecionando “Processar”.

Erro Médio Quadrático

O método do Erro Médio Quadrático recebe a imagem original e calculo o erro médio quadrático entre essa imagem e a imagem resultante. Para obtermos o valor do erro médio quadrático, precisamos calcular o valor das variáveis sinal e ruído de cada componente RGB o Erro médio quadrático e o PSNR serão calculados pelas seguintes fórmulas: $MSE = (\text{ruído}^2) / (\text{linhas} * \text{colunas})$ e $PSNR = 10 * \log_{10}((\text{max} * \text{max}) / MSE)$ onde $\text{max} = 256$.

Transformada de Hough Cicles:

O método de Hough recebe uma imagem do usuário, o método varre a imagem buscando por círculos e os armazena os círculos encontrados. O mesmo compara os círculos encontrados e remove os similares, após o método gera uma nova imagem com os círculos encontrados.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Círculo”, adicionando-o à linha do tempo e selecionando “Processar”.

Transformada de Hough Lines:

O método de Hough recebe uma imagem do usuário, então executa a transformação do Hough funciona observando um número dessas coordenadas x, y, que geralmente são encontrados por algum tipo de detecção de borda. Cada uma dessas coordenadas é transformada em r, curva theta. Esta curva é discreta, então, na verdade, só olhamos para um certo discreto número de valores theta. As células "Acumulador" em uma matriz de Hough ao longo desta curva são incrementadas para coordenadas X e Y. algoritmo então procura por picos locais na matriz. Quanto maior o pico, mais valores de x e y cruzaram ao longo dessa curva, picos tão altos dão boas indicações de uma linha.

Para utilizá-lo basta abrir a imagem desejada, selecionar “Linhas”, adicionando-o à linha do tempo e selecionando “Processar”.

Filtro de Cor:

O filtro de cor recebe dois parâmetros, rgbMin e rgbMax, que representam os valores mínimo e máximo de uma faixa de cores (em RGB) que serão permitidos. A partir desses valores o filtro gera uma imagem nova, binária, onde somente estão presentes os pixels que estavam dentro da faixa configurada.

Para usar o filtro de cor basta clicar no botão “Cor” com o botão direito do mouse para configurar a faixa RGB permitida e, depois, clicar com o botão esquerdo para agregar “Cor” ao fluxo de processamento.

Tamanho Original:

Botão que liga ou desliga o 'scaling' da imagem sendo visualizada para permitir ver o tamanho real da imagem.

Ligado: Permite a visualização do tamanho real da imagem

Desligado: Faz o scaling da imagem sendo visualizada para ela ficar do tamanho do canvas.

Para usar basta clicar no botão “Tamanho Original”.

Resetar

Volta a imagem para seu original e limpa a linha do tempo.

Referências Bibliográficas

- [1] EUSTÁQUIO, José; QUEIROZ, Rangel De; GOMES, Herman Martins. **Introdução ao processamento digital de imagens**. Rita, Brasil, v. 8, n. 1, 2001.
- [2] JESUS, Edison O.; COSTA JUNIOR, Roberto. **A Utilização de Filtros Gaussianos na Análise de Imagens Digitais**. 2014. Disponível em: <<https://proceedings.sbmec.org.br/sbmec/article/viewFile/477/483>>. Acesso em: 12 nov. 2017.
- [3] OLIVEIRA JUNIOR, Luciano Lucas de. **Filtros Compostos e Adaptativos : o filtro de Gaussiano , Laplaciano do Gaussiano e Gabor (Harmônico e Gaussiano)**. Disponível em: <<http://www2.ic.uff.br/~aconci/gabor.pdf>>. Acesso em: 11 nov. 2017.
- [4] SABADINI, Lucas Martins et al. **Processamento de Imagens -Filtro Laplaciano**. 2016. Disponível em: <<https://pt.slideshare.net/lucassabadini/processamento-de-imagens-filtro-laplaciano>>. Acesso em: 12 nov. 2017.
- [5] PIVETTA, Cleber; MANTOVANI, Gustavo; ZOTTIS, Felipe. **Teoria: Processamento de Imagens**. Disponível em: <<http://www.dpi.inpe.br/spring/teoria/realce/realce.htm>>. Acesso em: 12 nov. 2017.
- [6] PIVETTA, Cleber; MANTOVANI, Gustavo; ZOTTIS, Felipe. **Transformada de Hough**. Disponível em: <[http://www.inf.unioeste.br/~adair/PID/Notas_Aula/Transformada de Hough.pdf](http://www.inf.unioeste.br/~adair/PID/Notas_Aula/Transformada_de_Hough.pdf)>. Acesso em: 12 nov. 2017.
- [7] JAVAC Platform, Standart Edition 7 API Specification<<https://docs.oracle.com/javase/7/docs/api/>>. Acesso em: 16 nov. 2017.