

Atividade: Experimentos com ruídos e filtros no ImageJ (Valor 0,6)

Nesta atividade vocês devem simular no ImageJ pelo menos dois tipos de ruídos artificiais em uma imagem do próprio acadêmico. Em seguida, vocês devem aplicar no mínimo 4 filtros de suavização em alguma imagem com ruído; e 2 algoritmos de detecção de bordas na imagem original/convolucional, utilizando diferentes parâmetros. Na demonstração abaixo foram utilizados os ruídos: *gaussian noise*, *salt and pepper*; os filtros: *smoth*, *gauss blur*, *convolve*, *anisotropic diffusion 2d*; e os algoritmos para detecção de bordas: *find edges*, *canny edge detector*.

Imagem original



8 bits

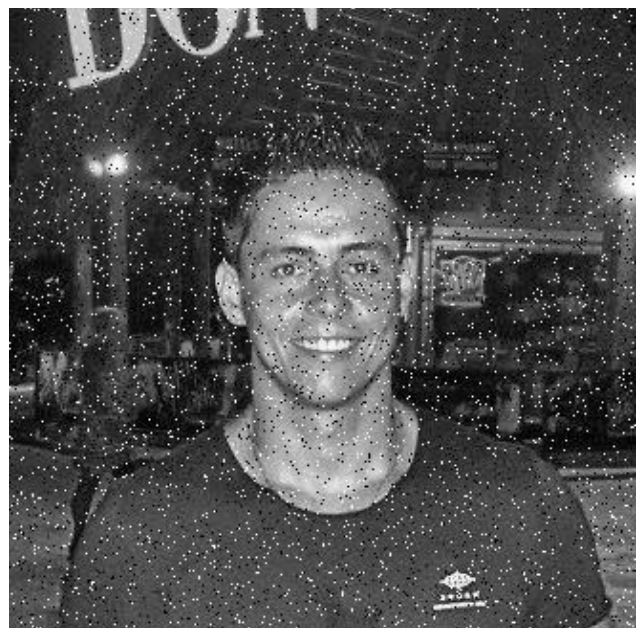


Algoritmos simuladores de ruídos

Gaussian Noise (Desvio Padrão=30)



Salt and Pepper



Gaussian Noise: Adiciona o ruído *Gaussian* atribuindo valor para a média igual a zero e desvio padrão escolhido pelo usuário.

Salt and Pepper: Adiciona o ruído sal (pixel branco) e pimenta (pixel preto) à imagem, substituindo aleatoriamente 2,5% dos pixels da imagem com pixels pretos e 2,5% com pixels brancos. Este comando só funciona com imagens de 8 bits.

Algoritmos de suavização

Simulação dos algoritmos de suavização sob o ruído *Gaussian Noise* e desvio padrão = 30

smoth (média da vizinhança 3×3)



gaussian blur (raio=1.5)



Convolve (máscara=4)



anisotropic diffusion 2D (20 interações com 1 suavização por interação)



Smoth: Este filtro borra a imagem ativa substituindo cada pixel com a média da sua vizinhança 3×3 .

Gauss Blur: Este filtro usa convolução com uma *função de Gauss* para suavizar a imagem.

Median: Reduz o ruído na imagem substituindo cada pixel com a mediana dos valores de pixels vizinhos. Para encontrar a mediana que substituirá cada pixel, o algoritmo exige que os valores de sua vizinhança estejam ordenados, o que consome considerável tempo de operação.

Anisotropic Diffusion 2D: Executa a filtragem por difusão anisotrópica em 8 bits, 16 bits e imagens RGB.

Algoritmos para detecção de bordas

find edges



canny edge detector:



convolve/find edges



convolve/canny edge detector:



find edges: algoritmo que usa um detector de borda chamado Sobel para destacar mudanças bruscas de intensidade na imagem ativa. Duas matrizes de convolução são usadas para calcular as derivadas verticais e horizontais (x e y). A imagem final é produzida pela combinação das duas derivadas usando a raiz quadrada da soma dos quadrados.

canny edge detector: desenvolvido por John F. Canny em 1986, *canny edge detector* é um *plugin* para detecção de bordas que utiliza um algoritmo de múltiplos estágios para detectar uma grande variedade de arestas em imagens.

convolve: é um algoritmo que utiliza uma máscara para transformação de imagens. A aplicação dessa máscara pode determinar as características da imagem, como formas, extração de ruídos ou detecção de bordas. A convolução é uma operação local que leva em consideração a vizinhança de um pixel e a máscara, modificando o valor do pixel correspondente na imagem resultante.

Conclusão

Nesta atividade foram apresentados diferentes tipos de ruídos artificiais, filtros de suavização e algoritmos para detecção de bordas, utilizando diferentes parâmetros.

Para obter uma imagem útil para uma determinada aplicação, uma série de parâmetros precisam ser testados, tais como, brilho, intensidade, filtragem de ruído, detecção de bordas, etc. Após o pré-processamento, a imagem pode, então, ser submetida a um filtro de banda (*threshold*) com o objetivo de transformar as escalas de cinza em pontos binarizados (preto ou branco), facilitando a solução de um problema em particular, como o cálculo de uma área ou a contagem de objetos.

Vale a pena destacar que alguns *plugins* adicionais estão disponíveis e podem ser acrescentados no *ImageJ*. Para instalá-los, basta realizar o *download* do *plugin* (.class) e inseri-lo na pasta “*plugins*” do *ImageJ*. Opcionalmente, o código-fonte (.java) também pode ser baixado e colocado na mesma pasta, permitindo que o usuário contribua com a implementação de novo código para o *plugin*.

Referências

IMAGEJ USER GUIDE – IJ 1.46r Revised edition, 2012.

Disponível em: <https://imagej.nih.gov/ij/docs/guide/146-29.html>. Acesso em: 31 mai. 2016

Obs: Salve suas imagens após as transformações numa pasta compactada (.zip ou .rar) e envie pelo Moodle. Prazo para entrega: 21/03/2021.