

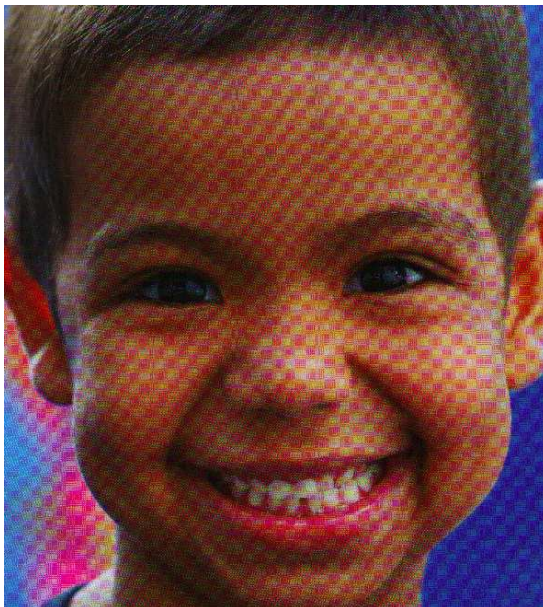
MAC5768 (MAC0417) Visão e Processamento de Imagens
Atividade no. 2

Prof. Dr. Paulo Miranda

¹Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)

Problema 01: Filtragem no domínio da frequência

Dadas duas imagens coloridas com ruído periódico (arquivos “menino.png” e “princesa.png” da Figura 1), faça um programa em Python, utilizando os seus conhecimentos sobre filtragem no domínio da frequência, para remover o ruído presente na imagem.



(a)



(b)

Figura 1: Imagens com ruído periódico. (a) Arquivo “menino.png” com padrão xadrez de interferência. (b) Folha de jornal escaneada apresentando um padrão listrado indesejado do arquivo “princesa.png”.

Instruções:

- Use o OpenCV-Python para a leitura das imagens.
- Utilize a função `cv2.split` do OpenCV-Python para separar os canais do vermelho, verde e azul.
- Faça o processamento de cada canal separadamente usando as implementações da transformada de Fourier e de sua transformada inversa presentes na biblioteca do NumPy ou OpenCV.
- Para cada canal, use a função `cv2.adaptiveThreshold` para identificar as regiões do espectro de elevada amplitude vinculadas ao ruído periódico e ignore a região central do espectro de baixa frequência.

- Faça a dilatação da máscara binária das regiões encontradas no item anterior, antes de apagar suas informações correspondentes no domínio da frequência, de modo a garantir a remoção do ruído periódico e calcule a transformada inversa de Fourier.
- Utilize a função `cv2.merge` para juntar os canais filtrados, gerando assim a imagem colorida filtrada.

Dicas:

- Estude o capítulo 4 do livro [1].
- Estude o tutorial do NumPy em:
<https://docs.scipy.org/doc/numpy/user/quickstart.html>

Problema 02: Árvore de componentes e filtros conexos

Dada uma imagem corrompida por um ruído impulsivo, resultando em múltiplas pequenas regiões pretas (arquivo “fruit.png” da Figura 2), faça um programa em Python para eliminar os pequenos buracos da imagem, utilizando a árvore de componentes.

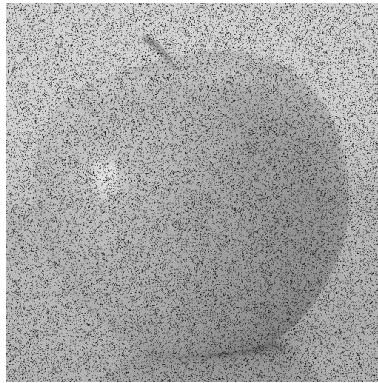


Figura 2: Imagem corrompida do arquivo “fruit.png”.

Dicas:

- Use o OpenCV-Python para a leitura da imagem.
- Utilize o toolbox do `siamxt` [2] para preencher as bacias da imagem de acordo com o atributo de área das bacias¹.

Problema 03: Árvore de componentes e filtragem no domínio da frequência

Dada uma imagem com ruído periódico (arquivo “palhaco.jpg” da Figura 3), faça um programa em Python que:

1. Calcule a transformada de Fourier da imagem.
2. Aplique um filtro de extinção no espectro (magnitude), utilizando o toolbox do `siamxt` [2] com vizinhança-8, de modo a conservar apenas os cinco domos mais relevantes de acordo com o atributo de altura.
3. Calcule os máximos regionais da imagem filtrada do item anterior a fim de identificar as regiões associadas ao ruído periódico, lembrando de ignorar a região central do espectro de baixa frequência.

¹<https://github.com/rmsouza01/siamxt>

4. Faça a dilatação da máscara binária das regiões encontradas no item anterior, antes de apagar suas informações correspondentes no domínio da frequência, de modo a garantir a remoção do ruído periódico e calcule a transformada inversa de Fourier.



Figura 3: Imagem com ruído periódico do arquivo “palhaco.jpg”.

Problema 04: Árvore de componentes e filtros conexos

Para a imagem fornecida abaixo:

3	3	0	2
1	0	2	0
2	1	2	3
4	2	1	3

- (a) Mostre a sua árvore de componentes, considerando vizinhança-4.
- (b) Mostre a sua max-tree, considerando vizinhança-4.
- (c) Faça a remoção de todos os domos com área menor ou igual a 3 (area opening), explicando o passo a passo dessa operação realizada sobre a max-tree, e mostre a imagem filtrada resultante.

Problema 05: Árvore de componentes e filtros conexos

Para a imagem fornecida abaixo:

4	2	1	3
2	1	0	3
1	2	2	3
3	3	1	0

- (a) Mostre a sua árvore de componentes, considerando vizinhança-4.
- (b) Mostre a sua max-tree, considerando vizinhança-4.
- (c) Faça a remoção de todos os domos com volume menor ou igual a 8 (volume opening), explicando o passo a passo dessa operação realizada sobre a max-tree, e mostre a imagem filtrada resultante.

Informações de entrega:

Os arquivos devem ser entregues na forma de um arquivo .zip no PACA.

Para as questões de 1 a 3, vocês devem entregar, além dos códigos dos programas em Python, um pequeno relatório descrevendo o trabalho realizado, com ilustrações de cada passo. Para as questões 4 e 5 deve ser entregue um documento pdf descrevendo as soluções dos problemas.

Referências

- [1] R.C. Gonzalez and R.E. Woods. *Processamento Digital de Imagens, terceira edição*. Pearson, Prentice Hall, São Paulo, SP, 2010.
- [2] Roberto Souza, Letícia Rittner, Rubens Machado, and Roberto Lotufo. iamxt: Max-tree toolbox for image processing and analysis. *SoftwareX*, 6:81 – 84, 2017.