Trabalho 2 – Processamento Digital de Imagens e Sinais – DCC066 – 2018/1
Departamento de Ciência da Computação – UFJF
18 de março de 2018

Prof. Marcelo Bernardes Vieira (marcelo.bernardes@ufjf.edu.br)

Filtros Lineares Espacialmente Invariantes, Teorema da Amostragem, Transformada de Fourier

---

Instruções:

- O valor deste trabalho é de **25 pontos**. O resultado deverá ser apresentado por todos os elementos do grupo até o dia **31 de maio de 2018**, impreterivelmente. Após esta data o grupo receberá nota nula.
- Na apresentação, os grupos deverão demonstrar que suas soluções para cada um dos itens desta especificação funcionam com diversas imagens.
- As especificações abaixo **não têm informações completas para sua implementação**. Os grupos **devem** procurar o professor para tirar dúvidas e entender detalhes do que foi pedido.
- Este trabalho tem a carga de 40 homem·hora: em um grupo de 2 pessoas, cada um deve dedicar 20 horas para finalizar o trabalho.
- Todos os grupos não têm permissão de possuir ou manter consigo o código ou os resultados de outro grupo, incluindo trabalhos antigos. É aconselhável que os grupos mantenham sigilo sobre suas soluções.
- Qualquer fraude, ou mera tentativa de fraude, será punida com **anulação DE TODAS AS NOTAS** de todos os trabalhos de todos os grupos envolvidos. O fato também será relatado à direção do ICE juntamente com uma solicitação de suspensão dos alunos fraudadores.

Projete e implemente um sistema para processamento básico de imagens usando Python:

- Use a biblioteca *Python Imaging Library*-PIL ou PILLOW (na versão 3 do Python), a biblioteca NumPy, a biblioteca SciPy para facilitar o desenvolvimento. Outras bibliotecas podem ser usadas livremente mas deverão ser relatadas ao professor.
- Transformadas com complexidade $O(n^2)$, onde n é o número de pixels, executam muito lentamente em Python mesmo com imagens pequenas. Os alunos podem implementar uma funcionalidades de alto custo computacional em C++. Porém, a funcionalidade deverá ser exportada e utilizada dentro do ambiente Python.
- Implemente **todas** as funcionalidades listadas nesta especificação.
- Na apresentação, o professor pode solicitar que uma imagem específica seja processada. Portanto, os alunos devem construir um sistema flexível, sem qualquer restrição sobre tamanho da imagem de entrada.
- **Atenda a todas as especificações a seguir**. Este trabalho faz parte de um projeto maior.

---

1. **Transformada de Fourier**:

   - Implemente a Transformada Discreta de Fourier **direta** e **inversa** para imagens bidimensionais.
   - Provejam um visualizador de imagens tanto da magnitude quanto da fase dos coeficientes.
   - Implemente os filtros bidimensionais passa-baixa ideal, passa-alta ideal e passa-banda ideal no domínio da frequência.
   - Use a implementação de alguma biblioteca Python (SciPy por exemplo) para verificar se o seu resultado está correto. Use a razão sinal ruído para comparar a diferença entre as reconstruções do seu trabalho e do Python.

2. **Produto de convolução e filtros lineares espacialmente invariantes**:

   - Encontre em alguma biblioteca uma implementação do produto de convolução unidimensional (em x e em y) e bidimensional. Certifiquem-se de que é possivel usar múltiplas opções de extensão de borda: simétrica, periódica e por valor constante. Se não houver suporte a isso, implemente o produto de convolução que tenha essas opções.

- Implemente uma função para geração de núcleos de filtros passa-baixa polinomiais bidimensionais. O programador pode escolher tanto as dimensões quanto o grau do filtro (número de convoluções com filtro box). Use a versão separável do filtro para aumentar o desempenho da funcionalidade.
- Implemente uma função para geração aproximada de núcleos do filtro gaussiano de qualquer dimensão. Use a versão separável do filtro para aumentar o desempenho da funcionalidade.
- Provenha os núcleos filtros passa-alta derivativos clássicos: Prewitt, Sobel, Roberts e Laplaciano.

3. **Amostragem e reconstrução**:

- Em todas as funções de amostragem e reconstrução, assume-se que a razão de aspecto da imagem será sempre o mesmo, ou seja, a proporção entre largura e altura é a mesma na imagem original e na resultante. Uma forma de parametrizar é fornecer uma porcentagem que indicará se a imagem deve ser amostrada (menor que 100%) ou reconstruída (maior que 100%).
- Implemente a função de redução de dimensão através de amostragem pontual da imagem.
- Implemente a função de redução de dimensão através de amostragem por área da imagem.
- Implemente a função de reconstrução por vizinho mais próximo.
- Implemente a função de reconstrução utilizando interpoladores polinomiais.

4. **Testes e verificações mínimas**:

- Mostre um exemplo de cada funcionalidade de transformação de espaço de cor.
- Mostre a validade do teorema da amostragem de Shannon-Whitaker, com exemplos claros de um caso no domínio do espaço e no domínio da frequência.
- Implemente um filtro de borramento e realce de imagens da seguinte forma $I' = (1+w) \cdot I - w \cdot I * G_\sigma$, onde I é a imagem original, G_σ é um núcleo de filtro gaussiano com desvio-padrão σ, e $w \in [-1, 1]$ é um peso para combinar as imagens. Mostre como os resultados variam em função de w e σ. Mostre quais são os melhores parâmetros para borramento e para realce.
- Mostre a eficácia da sua implementação da transformada de Fourier calculando o MSE e o SNR entre a imagem original e sua versão obtida via transformada inversa de Fourier. Ou seja, transforme a imagem original para o domínio da frequência e em seguida aplique a inversa para obter a imagem no domínio do espaço. Calcule o erro entre essa imagem com a original. Compare seu erro com o objetido com a implentação FFT (Fast Fourier Transform) da biblioteca SciPy.
- Mostre os efeitos de cada tipo de amostragem implementada no domínio da frequência. Proveja exemplos de imagens naturais que sofreram *aliasing*.
- Mostre os efeitos de cada tipo de reconstrução implementada no domínio da frequência.
- Mostre exemplos de filtragens com todos os filtros implementados.

5. **Desenvolvendo o trabalho**:

- Antes de começar a codificar, aprenda os conceitos básicos do Python. Há muitos tutoriais na internet. Você pode começar lendo imagens .jpg e .png e criando uma interface simples para visualizá-las. Lembre-se que o grupo deverá fazer uma boa apresentação mostrando a eficácia de seu sistema.
- Não escreva nenhum código antes de entender completamente cada funcionalidade. Tente fazer cada tópico desta especificação na ordem proposta porque grande parte das funcionalidades depende diretamente ou pode ser implementada a partir das antecessoras.
- Não deixe de consultar o livro a respeito de como implementar o que foi pedido. Consulte os livros do curso para desenvolver cada etapa, sobretudo o do Gonzalez. Nele vocês encontrarão vários atalhos e exemplos.
- Aprenda funcionalidades das bibliotecas que podem ser úteis no desenvolvimento do trabalho como: recorte de subimagem, duplicação de imagens, combinação de imagens, redimensionamento de imagens, rotação e espelhamento de imagens, desenho de texto sobre a imagem, etc.