

Trabalho 3 - Processamento Digital de Imagens e Sinais – DCC066 – 2018/1  
Departamento de Ciência da Computação – UFJF  
7 de junho de 2018

Prof. Marcelo Bernardes Vieira ([marcelo.bernardes@ufjf.edu.br](mailto:marcelo.bernardes@ufjf.edu.br))

Compressão de Imagens

---

Instruções:

- O valor deste trabalho é de **25 pontos**. O resultado deverá ser apresentado por todos os elementos do grupo até o dia **3 de julho 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.
  - Opcionalmente, os alunos podem implementar uma funcionalidade em C++. Porém, a funcionalidade deverá ser exportada e utilizando 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. Compressão de imagens:**

- Escolha uma entre as transformadas de Fourier, Discreta do Cosseno, ou Wavelets, e a utilize para fazer compressão com perda de imagens RGB com 24 bits. Implemente a transformada direta e inversa da base que você escolheu.
- Implemente ambos o compressor e o descompressor utilizando como base o trabalho feito até o momento em Python. A eficiência de tempo de compressão não é importante.
- Não é permitido o uso de compressores, existentes com ou sem perda, em nenhuma fase do projeto. Todos os algoritmos devem ser completamente desenvolvidos pelo grupo.
- Vocês podem escolher o espaço de cor que será utilizado. Transformações de espaço de cor podem ser obtidas no pacote PIL ou PILLOW do Python. Lembrem-se que os compressores comerciais usam decomposição luminância-crominância, com redução de bits nos canais de crominância. Mas o espaço RGB também permite bons níveis de compressão.
- Crie um formato específico para o seu compressor. A imagem de entrada deverá ser comprimida e salva neste formato. Quanto menor o número de bytes no final, melhor será a avaliação do grupo.
- Opcionalmente, para aumentar a eficácia da compressão, utilize algum método de codificação entrópica como código de Huffman, codificação aritmética, entre outros.

## 2. Testes e verificações mínimas:

- Faça um diagrama que explica claramente as fases de compressão e de descompressão.
- Mostre o funcionamento do seu compressor em várias imagens reais e sintéticas. Imprima a taxa de compressão (razão entre o número de bytes de saída e o número de bytes de entrada) e o tempo de compressão. O número de bytes de entrada é a quantidade de pixels vezes o somatório do número de bytes usados pelos canais (fixado neste trabalho em 3 bytes ou 24 bits).
- Mostre o funcionamento do seu descompressor. Imprima o tempo de descompressão e a razão sinal-ruído em relação à imagem original no **espaço RGB** com 24 bits de cor.
- As notas serão dadas de acordo com a eficácia do compressor. Será fornecida uma imagem descomprimida para ser comprimida por todos os grupos. Os grupos deverão comprimi-la até a taxa mínima de 27:1. Ou seja, a imagem comprimida deverá ter número de bytes no máximo 27 vezes menor que a imagem descomprimida. O grupo que obtiver maior razão sinal-ruído após a descompressão terá a maior nota.

## 3. Desenvolvendo o trabalho:

- Use como base o código feito nas fases anteriores do trabalho.
- As estratégias usadas em compressores comerciais podem ser úteis para vocês. Leia a respeito dos formatos de compressão mais populares.
- 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, transformação de espaço de cor, etc.