

Tecnologias Multimédia

Aula 10

Manuela. Pereira

mpereira@di.ubi.pt

November 30, 2022

1. Leia a imagem **baboon**, converta-a para níveis de cinza e reduza-a para metade.
2. Calcule a transformada de Fourier da imagem **baboon** (`np.fft.fftshift(np.fft.fft2(baboon))`). O `fftshift` serve para centrar as frequências.
3. Represente a imagem **baboon** no espaço e também em função das frequências.
4. Use as máscaras `c1ec2` definidas no ficheiro **Aula_10.py** para remover as frequências mais baixas. Analise as imagens resultantes, após proceder à inversa da transformada de Fourier (`np.fft.ifft2(hp)`).
5. Use as máscaras `c3ec4` definidas no ficheiro **Aula_10.py** para remover as frequências mais altas. Analise as imagens resultantes, após proceder à inversa da transformada de Fourier.
6. Repita as alíneas anteriores para a imagem **saturn**.
7. Explique as diferenças do efeito da remoção das diferentes frequências em cada uma das imagens.
8. Nas perguntas seguintes use as funções `dct2` e `idct2` definidas no ficheiro **Aula_10.py** para efetuar a transformada DCT e a respetiva inversa.
9. Calcule a transformada DCT das imagens **baboon** e **saturn**. Proceda à representação das imagens no tempo e na frequência. Compare a distribuição das frequências entre as duas imagens. Verifique que neste caso as baixas frequências se encontram no canto superior esquerdo.
10. Proceda à remoção de altas e de baixas frequências nas duas imagens. Após o cálculo da inversa da DCT represente a imagem resultante e relacione o aspeto dela com as frequências que removeu.
11. Use a função `dct2_blocos` definida no ficheiro **Aula_10.py** para efetuar a `dct2` por blocos de 8×8 na imagem **saturn**. Guarde o resultado na variável `saturn_dct_blocos`.
12. Filtre os valores inferiores a um limite (ex. 0.1, 0.01, ...). Para isso multiplica a sua variável resultante da alínea anterior por uma máscara (exemplo de máscara: `np.absolute(saturn_dct_blocos) > thresh * np.max(np.absolute(saturn_dct_blocos))`). Após a inversão da transformada com a função `idct2_blocos` represente a imagem recuperada.
13. Repita as duas alíneas anteriores para a imagem **baboon**. Neste caso remova ou junte algumas colunas para que as dimensões da sua imagem sejam múltiplos de 8 (ex. `baboon8 = baboon[:, 2 : 498 : 1]`).
14. Crie uma função `quantiza` para efetuar a quantização usando a tabela de quantização do jpeg. Pode usar uma variável `fator` que multiplica pela tabela de quantização de forma a alterar estes valores.
15. Crie uma função `dequantiza` que efetua o inverso da anterior.
16. Use as funções anteriores para perceber como a quantização do jpeg está relacionada com o débito/qualidade das imagens comprimidas usando esta norma.