

EXERCÍCIO PROGRAMA 3

Entrega: 29/06/2018

Professor Marcelo Queiroz
Monitor Thilo Koch

Neste EP vamos experimentar com wavelets usando a biblioteca PyWavelets (`pywt`). Crie um Jupyter notebook para Python e resolva as tarefas nele numa forma apresentável deixando todos os cálculos explícitos. Lembre-se que o notebook pode servir como ótima fonte de informação ou de código no futuro para vocês mesmos. Uma apresentação sistêmica e clara ajuda.

1 Básico

- (a) Explore a biblioteca `pywt` e os wavelets que define. Veja `pywt.families`, `pywt.wavelist` e a descrição do wavelet com `print(wavelet)`.
- (b) Crie um sinal artificial de 256 amostras com os seguintes valores: `[0:100]=0.0`; `[100:161]=1.0`; `[161:257]=4.0`.
- (c) Execute uma DWT desse sinal com o wavelet `bior2.2` e plote os coeficientes de aproximação e os coeficientes de detalhes.
- (d) Descreva o resultado em relação ao sinal original.
- (e) Execute a reconstrução do sinal com `upcoef()` e plote-a.
- (f) Compare os resultados quando variar o modo de extensão do sinal para a DWT (parâmetro `mode` da função `dwt`).
- (g) Experimente da mesma maneira com pelo menos mais 2 outros wavelets; descreva e compare os resultados.

2 Sinais de áudio

- (a) Carregue um som de sua escolha, transforme-o com *bior2.2* e resintetize o som original apenas com os coeficientes de aproximação. Descreva o resultado.
- (b) Explore as funções `wavedec` e `waverec` usando o sinal com diferentes níveis de decomposição e aplique *thresholding*. Compare os resultados obtidos por limiares diferentes (semelhante ao EP2).
- (c) Experimente com pelo menos mais 2 outros wavelets. Descreva e compare.

3 Imagens

- (a) Carregue uma imagem de sua escolha e transforme-a em uma imagem cinza.
- (b) Use a função `dwt2` para decompor a imagem e reconstrua-a usando a função `idwt2` somente com os coeficientes de aproximação.
- (c) Repita os itens (b) e (c) do ponto 2 para o caso bidimensional: a imagem.

Bons trabalhos.