

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia
Licenciatura em Engenharia Informática e de Computadores

Codificação de Sinais Multimédia

2º Semestre de 2024/2025

O objetivo deste trabalho é duplo: 1 - implementar e analisar o desempenho dum codificador/descodificador JPEG sequencial para compressão de imagens a tons de cinzento. 2 - explorar os princípios básicos da codificação de vídeo.

1. Pretende-se implementar e analisar o desempenho dum codificador/descodificador JPEG sequencial para compressão de imagens a tons de cinzento.
 - (a) Construa uma função (codificador) que para cada bloco de 8×8 da imagem original efetue a DCT bidimensional e construa outra função (descodificador) que faça a DCT inversa. Ver funções: `cv2.dct()` e `cv2.idct()`. Visualize a imagem depois da DCT e verifique que a DCT inversa resulta na imagem original.
 - (b) Construa uma função (codificador) que para cada bloco de 8×8 de coeficientes da transformação efectuada faça a divisão pela matriz de quantificação (tabela K1 da norma) multiplicada por um factor de qualidade q (ver função `quality_factor` do ficheiro `Tabelas_jpeg.ipynb`). Veja a imagem com o conjunto dos blocos após a quantificação. Construa uma função (descodificador) que realize a operação inversa da quantificação. Junte estas funções às já realizadas e verifique para diferentes fatores de qualidade qual a SNR e veja a imagem descodificada.
 - (c) Construa uma função (codificador) que faça a codificação diferencial dos coeficientes DC após a quantificação. Construa a função inversa para o descodificador.
 - (d) Construa uma função (codificador) que crie um array com a indexação em zig-zag dos coeficientes AC após a quantificação e crie um array com os pares (zero run length, nonzero value). Construa a função inversa para o descodificador.
 - (e) Junte estas funções às já realizadas e veja a imagem descodificada.
 - (f) Construa uma função que dados os arrays das alíneas anteriores use as tabelas do código de Huffman (tabela K3 e K5) e grave num ficheiro a sequência de bits correspondente. (não é necessário usar o formato JFIF)
 - (g) Construa uma função que leia o ficheiro gravado e retorne os arrays com os coeficientes AC e DC.
2. Pretende-se implementar três formas de codificação de vídeo. Cada um destes codificadores deve ser testado com as sequências de imagens disponibilizadas (`bola_seq.zip` e `carro_seq.zip`). Para cada codificador deve medir (frame a frame):
 - (a) a taxa de compressão;
 - (b) a relação sinal-ruído (relação entre a frame original no emissor e a frame final no recetor);
 - (c) a entropia da frame a transmitir;
 - (d) a energia média por pixel da frame a transmitir;
 - (e) o tempo de compressão e descompressão.

No final deve construir gráficos ou tabelas que ilustrem estas medidas em função da frame.

Para os três codificadores use o algoritmo JPEG do OpenCV com uma qualidade de 50%.

As três formas de codificação são:

- (a) Considerar que cada frame é uma intra-frame (I).
- (b) Considerar que todas as frames à exceção da primeira (a I-frame) são inter-frames (P). Neste codificador deve criar as P-frames, que são a diferença entre a frame a codificar e a I-frame, sem compensação de movimento. Visualize a P-frame (ou seja a imagem a transmitir).
- (c) Considerar que todas as frames à exceção da primeira são inter-frames (P). Neste codificador deve implementar a predição da frame a codificar com base na I-frame fazendo a compensação de movimento. A frame a transmitir é a diferença entre a frame a codificar e a sua predição. Sugere-se a construção de três funções:
 - i. uma função que realiza a medição do erro absoluto médio entre dois blocos (tamanho 16×16);
 - ii. uma função que realiza uma pesquisa (pode escolher a full-search ou outra) do macro-bloco da frame a codificar numa janela de pesquisa (-15 a +15) da I-frame;
 - iii. uma função que percorra os blocos da frame a codificar e construa a frame predita;

Visualizar a frame predita, e a frame diferença, bem como os vetores de movimento (use a função `pylab.quiver` para o efeito).