

Especificação do Projeto de Multimédia II

O projeto é um trabalho de grupo (3/4 pessoas) constituído por três componentes:

1. Um programa (aplicação de software) que implementa um codec multimédia sem perdas, isto é, um algoritmo ou técnica de compressão sem perdas –coder / decoder.
 - Os requisitos deste software constituem o 1º elemento de avaliação prática e incluem o cálculo de métricas sobre o desempenho do codec.
2. Um relatório da implementação do programa incluindo que inclui uma revisão bibliográfica (*survey* - um apanhado de todas as utilizações que existem para um determinado codec), explica o funcionamento do codec com um exemplo trabalhado passo a passo (*traçagem*), e discute as métricas obtidas na compressão dos diferentes elementos do Silesia Corpus.
 - A descrição da estrutura e das secções que devem existir no relatório encontram-se no documento modelo / *template word*.
3. Uma apresentação de 12 - 15 minutos sobre o projeto desenvolvido que inclua todos os elementos do grupo de trabalho, e descreve as principais utilizações do codec vistas no relatório, explica o funcionamento do codec (com o exemplo descrito no relatório), e apresenta uma análise crítica dos resultados obtidos para cada uma das 5 métricas.
 - A apresentação pode ser feita em powerpoint ou prezi e é o elemento principal da defesa do projeto em apresentação pública na sala de aula no final do semestre.

Requisitos de software

1. O programa deve implementar um algoritmo de compressão / descompressão sem perdas (codificador de entropia) em Python ou javascript selecionado a partir da listagem de codecs propostos, e pode ser:
 - baseado em códigos de comprimentos variável (codificador estatístico);
 - baseado em códigos de comprimento fixo (codificação baseada em dicionários da família LZ);
 - baseado em códigos numéricos (tal como Golomb - Rice);
 - (*) baseado em algoritmos de inteligência artificial para aprendizagem profunda (*deep learning*), uma tendência recente de *machine learning* que utiliza redes neurais de modo que o programa “aprenda sozinho”.
2. No 1º modo de funcionamento, utilizando o Silesia Corpus, o programa deve abrir cada um dos respetivos ficheiros e comprimi-los, calculando, a seguir, um conjunto de métricas para caracterizar a eficiência do processo de compressão / codificação dos conteúdos.
 - As métricas devem ser calculadas pelo programa a seguir à implementação do algoritmo de compressão / descompressão e incluem:
 1. o rácio de compressão resultante (CR – *compression ratio*),
 2. o comprimento médio do código (ACL – *average code length*) em bits por símbolo, correspondente ao débito binário da sequência comprimida (*compressed bit rate*)
 3. a eficiência do processo de codificação em termos de:
 - o tempo de duração da compressão (ET – *encoding time*)
 - o tempo de duração da descompressão (DT – *decoding time*).
3. Características do [*Silesia Corpus*](#):
 - Este Corpus contém um conjunto de 12 ficheiros de vários tipos multimédia, desenvolvido por Sebastian Deorowicz em 2003 na Universidade *Silesian University of Technology*, Polónia, utilizados inicialmente para testar algoritmos de compressão sem perdas no âmbito

de um [projeto de uma tese de doutoramento](#), e que se tornou num dos Corpus mais utilizados pela comunidade científica de Multimédia para testar o desempenho de algoritmos de compressão sem perdas (*lossless*).

- É possível consultar um resumo do *Silesia Corpus* no [site original do autor](#) e no [Data Compression Resource](#). A versão comprimida do Silesia Corpus está disponível na plataforma Canvas, mas também podem ser obtidas [aqui](#).
 - O *Silesia Corpus* inclui os seguintes conteúdos multimédia:
 - Docs. de texto ASCII: 2 livros, 6 artigos, 1 resumo bibliográfico e 1 conjunto de notícias sem edição (bib, news, book<n>, paper<n>)) representam linguagem humana;
 - Código editável: 3 ficheiros de código de 3 linguagens de programação representam linguagem artificial;
 - Código binário: 2 ficheiros executáveis obj<n> representam tipos de *media* binários tais como áudio ou vídeo;
 - Transcrição de 1 sessão num terminal: trans, representa uma comunicação;
 - 1 ficheiro de dados georreferenciados: geo
 - 1 imagem bitmap bitonal (P/B): pic
4. No 2º modo de funcionamento, o programa deve fazer a traçagem com a sequência de teste, isto é, apresentar no ecrã os passos intermédios dos algoritmos de compressão e de descompressão, mostrando claramente o processo de codificação, isto é, o modo como os códigos comprimidos vão sendo obtidos mediante a aplicação do algoritmo de codificação, e a forma como os símbolos originais descomprimidos são recuperados a partir dos códigos comprimidos no caso do algoritmo de decodificação.
- Para ilustrar estes passos intermédios, o programa deve ser carregado com a seguinte sequência/*String*, cuja codificação deve ser mostrada pelo programa e descrita passo-a-passo na secção 2.3 do artigo científico.

Sequência de teste:

“Não sei quantas almas tenho. Cada momento mudei. Continuamente me estranho. Nunca me vi nem achei. De tanto ser, só tenho alma.

Quem tem alma não tem calma. Quem vê é só o que vê, Quem sente não é quem é, Atento ao que sou e vejo, Torno-me eles e não eu. Cada meu sonho ou desejo, É do que nasce e não meu.”

(12 versos do poema "Não sei quantas almas tenho" de Fernando Pessoa)

Temas propostos para o projeto

Apresenta-se a seguir um conjunto de temas para o desenvolvimento do projeto.

Nº	Tema do projeto de Multimédia II <i>Técnica de compressão</i> sem perdas de texto e imagem	Referência Khalid Sayood (2017)	Referência David Salomon (2010)
1	Codificação de Huffman Adaptativa (codificação estatística de comprimento variável)	caps. 3.1 e 3.4, pp. 41 e 67	Cap. 5.3, pp. 234
2	Codificação LZ77 (baseada em dicionários)	cap. 5.4.1, pp.134 e 135	Cap.6.3, pp. 334
3	Codificação LZ78 (baseada em dicionários)	cap. 5.4.2, pp.139	Cap.6.9, pp. 354
4	Método de codificação estatística Golomb-Rice (utilizado no formato de compressão de imagens JPEG-LS e formato FLAC para compressão sem perdas de áudio), com desenvolvimento dos códigos de Golomb seguidos dos códigos de Rice	caps. 3.5 e 3.6, pp. 75 e 76	Caps. 3.24 e 3.25, pp. 160 e 166
5	Método BZIP2, constituído por: Transformada de Burrows-Wheeler seguida do método Move-to-Front, terminando com um codificador estatístico	caps. 6.4 e 6.4.1, pp. 175	Cap. 11.1, pp. 1089 Cap. 1.5,

	de Huffman (codificação estatística baseada em contexto)		pp. 45
6	Método <i>deflate</i> (LZSS + Huffman), sendo o LZSS uma variante do LZ77 (utilizado nos formatos zip, gzip e PNG)		Cap. 6.25, pp. 399
7 *	Método PPM - <i>Prediction by Partial Matching</i> (codificação estatística preditiva baseada em contexto)	Caps. 6.2 e 6.3, pp. 165 e 167	Cap. 5.14, pp. 292
8	Técnica RLE - Run-Length Encoding para compressão de texto e imagens		Caps. 1.2, 1.3 e 1.4, pp. 31 e 36
9 *	Algoritmo DMC - <i>Dynamic Markov Compression</i> (baseado em contexto)	cap. 6.6, pp. 181	Cap. 11.8, pp. 1134
10 *	Lossless Data Compression using Deep Learning Techniques: <i>Convolutional Neural Networks</i> (CNN)	Seleção de artigos	Seleção de artigos

Recursos para a realização do projeto:

- Documento modelo (*template*) para o artigo científico com explicação dos conteúdos a escrever.
- Ficheiros do corpus Silesia
- Implementações de projetos de anos anteriores.