

# CODECs não destrutivos para imagens monocromáticas

## Guia de utilização de código

Trabalho prático nº2 da cadeira Teoria da Informação, 2º ano, 1º semestre

João Afonso Vieira de Sousa, 2019224599, uc2019224599@student.uc.pt

José Domingos da Silva, 2018296125, uc2018296125@student.uc.pt

Sancho Amaral Simões, 2019217590, uc2019217590@student.uc.pt

Tiago Filipe Santa Ventura, 2019243695, uc2019243695@student.uc.pt

20 de novembro de 2020

# CMP

O *CMP* é um *CODEC* testável e manipulável desenvolvido por alunos do 2º ano da Licenciatura de Engenharia Informática no âmbito da cadeira de Teoria da Informação, que permite comprimir imagens *.bmp* de 8 *bits* em tons de cinzento. Recomenda-se que seja efetuada a análise da documentação do código, em especial da do ficheiro *BPMCodec.py*, para além da leitura deste documento. As instruções abaixo enumeradas assumem que o utilizador possui uma cópia do repositório criado no âmbito da realização deste trabalho prático.

## 1. Instalação / Configuração

a) Abra uma *IDE* (*Integrated Development Environment*) que suporte a linguagem *Python* (ex.: *PyCharm*, *WingIDE*, *Spyder*, etc.).

b) Abra o diretório *TI-TP2* na *IDE* escolhida no passo acima.

O diretório *TI-TP2* aparecerá na aba de navegação do seu *IDE*.

c) Aceda ao *script Python Main.py* localizado em *TI-TP2/source\_code/cmp*.

d) Certifique-se que o caminho absoluto do *script* a executar especificado pela *IDE* coincide com o caminho absoluto do *script Main.py*.

e) Certifique-se que o caminho absoluto do diretório atual (*working directory*) especificado pela *IDE* coincide com o caminho absoluto do diretório onde o *script Main.py* se encontra.

f) Procure pela região (*region*) *Constants* no *script Main.py*.

As constantes definidas nessa região são *TO\_COMPRESS\_PATH*, *COMPRESSED\_PATH*, *TO\_DECOMPRESS\_PATH*, *DECOMPRESSED\_PATH* que indicam, respetivamente, o diretório de ficheiros alvo a comprimir, o diretório onde os ficheiros comprimidos serão colocados, o diretório de ficheiros alvo a descomprimir e o diretório dos ficheiros descomprimidos.

g) Altere os caminhos relativos especificados pelas constantes por outros, caso não deseje manter os valores predefinidos.

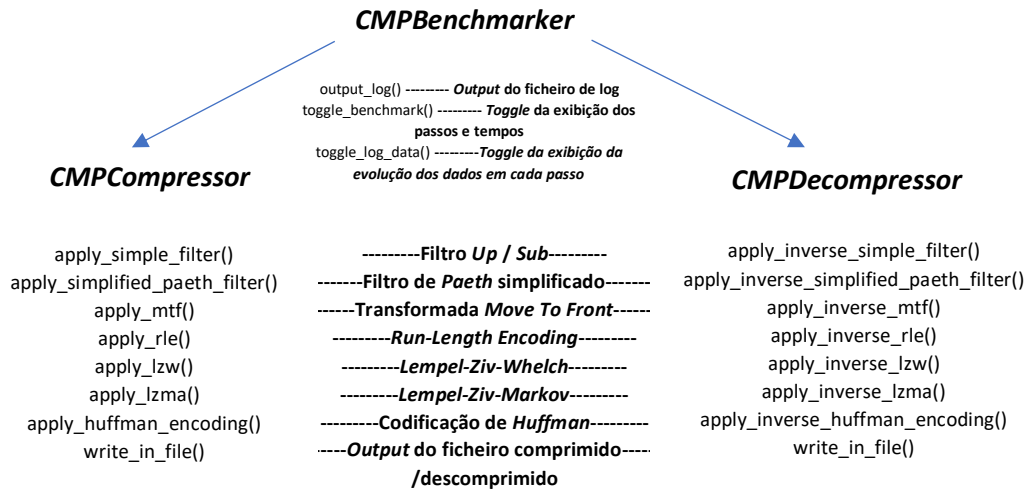
## 2. Modo de utilização

No *script Main.py* encontram-se as funções *compress\_files()* e *decompress\_files()*. A primeira efetua a compressão dos ficheiros de extensão *.bmp* dentro do diretório definido pela constante *TO\_COMPRESS\_PATH* e a segunda descomprime os ficheiros de extensão *.cmp* localizados no diretório especificado pela constante *TO\_DECOMPRESS\_PATH*. Em cada uma destas funções são instanciadas as classes *BMPCompressor* e *CMPDecompressor* que representam o compressor e descompressor, respetivamente, de modo a que o utilizador tenha acesso às diretivas de compressão/descompressão fornecidas pelo *CODEC CMP*.

## 2.1 – Diretivas e funcionalidades fornecidas pelas classes *CMPCompressor* e *CMPDecompressor*

Ambas as classes referidas estendem a classe *CMPBenchmarker* que, por sua vez, fornece funções e variáveis para a finalidade de *logging/benchmarking*.

Abaixo encontra-se um diagrama que explica sucintamente as funcionalidades disponíveis ao utilizador nas classes acima referidas.



**Nota:** Recomenda-se a consulta da documentação das funções acima referidas para melhor entender o seu funcionamento e o propósito dos seus parâmetros.

O utilizador pode escolher várias combinações de transformações aplicadas aos dados alvo, a fim de conseguir apurar qual o algoritmo mais eficiente para a compressão de uma determinada imagem.

## 2.2 – Compression e decompression stack

Convém salientar que o utilizador deve certificar-se que a *compression stack* na função *compress\_files()* é reversa e inversa da *decompression stack* da função *decompress\_files()*, como abaixo ilustrado em pseudocódigo.

<i>compress_files()</i>	<i>decompress_files()</i>	
$t_0$	$(t_n)^{-1}$	$t_i, \forall i, 0 \leq i \leq n$
$t_1$	.	é uma transformação qualquer
.	.	invertível fornecida pelo <i>CMP</i>
.	.	aplicada aos dados em questão.
.	$(t_1)^{-1}$	
$t_n$	$(t_0)^{-1}$	

## 2.3 – Logging e Benchmarking

Para uma recolha mais fácil de dados acerca da *performance* da *compression/uncompression stack* utilizada, como tempo de compressão/descompressão e taxa de compressão, o *CMP* dispõe de mecanismos de *logging* e *benchmarking*.

Um ficheiro *.txt* é criado no diretório especificado pelas constantes *COMPRESSED\_PATH*, aquando da compressão de uma imagem, e *DECOMPRESSED\_PATH*, aquando da descompressão de uma imagem, somente se a função *output\_log()* for chamada.

Exemplos do formato dos ficheiros de *log*:

-----crop.bmp CMP COMPRESSION LOG-----	-----crop.bmp CMP DECOMPRESSION LOG-----
 COMPRESSION STACK:	 DECOMPRESSION STACK:
-> MTF	-> INVERSE HUFFMAN ENCODING
-> RLE	-> INVERSE RLE
-> HUFFMAN ENCODING	-> INVERSE MTF
 TOTAL ELLAPSED COMPRESSION TIME: 0.57 sec.	 TOTAL ELLAPSED UNCOMPRESSION TIME: 0.34 sec.
INITIAL IMAGE SIZE: 117586 bytes	
COMPRESSED IMAGE SIZE: 21117 bytes	
COMPRESSION RATIO: 82.04%	

- Durante a compressão e descompressão das imagens, os eventos que estão a ocorrer podem ser apresentados na consola, as respetivas durações totais e os dados transformados. O *toggling* desta ação pode ser efetuado recorrendo à definição dos parâmetros opcionais *benchmark* e *log\_data* dos construtores das classes *CMPCompressor* e *CMPDecompressor* ou através das funções *toggle\_benchmark()* e *toggle\_log\_data()*.

Exemplos do formato do *output* de *logging/benchmarking* na consola:

```
-----
crop.bmp Compression
-----
Applying up filter...
Elapsed up filtering time: 0.00 sec
Applying RLE encoding...
Elapsed RLE encoding time: 0.05 sec
Applying Huffman encoding...
Elapsed huffman encoding time: 0.02 sec
Total elapsed compression time: 0.07 sec
Writing in file crop.cmp...
-----
```

# BZip2

O *bzip2* é um algoritmo de compressão de ficheiros individuais desenvolvido em 1996 por Julian Seward. No trabalho prático em questão foi utilizada um ficheiro executável para aplicação deste algoritmo (fonte: <https://github.com/phillr/bzip2-windows/releases>).

## 1. Modo de utilização

- a) Abra a linha de comandos do *Windows*.
- b) Através da linha de comandos, navegue para o diretório onde o executável *bzip2.exe* se encontra.
- c) Para comprimir, digite ***bzip2 (/caminho do ficheiro a comprimir|.extensão)*** na linha de comandos. Para descomprimir, digite ***bzip2 -d (/caminho do ficheiro a descomprimir|.bzip2)*** na linha de comandos.

## 2. Flags de execução

<i>-h --help</i>	o texto abaixo é imprimido
<i>-d --decompress</i>	forçar descompressão
<i>-z --compress</i>	forçar compressão
<i>-k --keep</i>	manter (não apagar) ficheiros de <i>input</i>
<i>-f --force</i>	sobrescrever ficheiros de <i>input</i> existentes
<i>-t --test</i>	testar integridade do ficheiro comprimido
<i>-c --stdout</i>	imprimir ficheiro comprimido na consola
<i>-q --quiet</i>	suprimir mensagens de erro não críticas
<i>-v --verbose</i>	<i>output</i> detalhado (um segundo <i>-v</i> torna o <i>output</i> mais detalhado)
<i>-L --license</i>	exibir versão e licença do <i>software</i>
<i>-V --version</i>	exibir versão e licença do <i>software</i>
<i>-s --small</i>	usar menos memória (no máximo 2500KB)
<i>-1 ... -9</i>	definir tamanho dos blocos para 100KB...900KB
<i>--fast</i>	<i>alias</i> para <i>-1</i>
<i>--best</i>	<i>alias</i> para <i>-9</i>

*Bzip2* a ação predefinida é a de comprimir

# Jasper (JPEG2000 - Lossless)

O *Jasper*, também designado por *JPEG2000* é uma variante *lossless* do formato de imagens *JPEG*. No trabalho prático em questão foi utilizada uma versão do *JPEG2000* implementada em C (fonte: <https://www.ece.uvic.ca/~frodo/jasper/>).

## 1. Modo de utilização

a) Abra a linha de comandos do *Windows*.

b) Através da linha de comandos, navegue para o diretório de caminho *jasper-2.0.14\out\install\x64-Debug (default)\bin*.

c) Para comprimir digite

**jasper --input (*|nome do ficheiro a comprimir|.extensão*) --output (*|nome do ficheiro comprimido|.jp2*) --output-format *jp2* ou *jasper -f (*|nome do ficheiro a comprimir|.extensão*) -F (*|nome do ficheiro comprimido|.jp2*) -T *jp2** na linha de comandos.**

Para descomprimir digite

***jasper (*|nome do ficheiro a descomprimir|.jp2*) --output (*|nome do ficheiro descomprimido|.extensão*) --output-format (*formato de saída*) ou *jasper -f (*|nome do ficheiro a descomprimir|.jp2*) -F (*|nome do ficheiro descomprimido|.extensão*) -T (*formato de saída*)* na linha de comandos.***

## 2. Flags de execução

--help            o texto abaixo é imprimido  
--version        imprimir versão e sair  
--verbose        ativar modo detalhado  
--debug-level \$lev    definir o nível de *debug* para *\$lev*  
--input \$file       ler a imagem a partir do ficheiro *\$file* em detrimento do *input standard*  
--input-format \$fmt   especificar o formato da imagem de entrada como *\$fmt* (ver abaixo lista de formatos suportados)  
--input-option \$opt   fornecer a opção *\$opt* ao decodificador  
--output \$file       escrever a imagem no ficheiro *\$file* em detrimento do *output standard*  
--output-format \$fmt   especificar o formato do ficheiro de saída como *\$fmt* (ver abaixo lista de formatos suportados)  
--output-option \$opt   fornecer a opção *\$opt* ao codificador  
--force-srgb        forçar conversão para o espaço de cores *sRGB*

### Abreviaturas:

--input = -f, --input-format = -t, --input-option = -o,  
--output = -F, --output-format = -T, --output-option = -O

### Formatos suportados:

-mif    My Image Format (MIF)  
-pnm    Portable Graymap/Pixmap (PNM)  
-bmp    Microsoft Bitmap (BMP)  
-ras    Sun Rasterfile (RAS)  
-jp2    JPEG-2000 JP2 File Format Syntax (ISO/IEC 15444-1)  
-jpc    JPEG-2000 Code Stream Syntax (ISO/IEC 15444-1)  
-jpg    JPEG (ISO/IEC 10918-1)-pgx    JPEG-2000 VM Format (PGX)

# JPEG (LOSSY)

O *JPEG*, além de um formato de imagens, é um método de compressão destrutivo. No trabalho prático em questão foi utilizada uma versão do *JPEG* implementada em C (fonte: <https://github.com/kornelski/jpeg-compressor/releases>).

## 1. Modo de utilização

### **Windows:**

a) Abra a linha de comandos do *Windows*.

b) Através da linha de comandos, navegue para o diretório de caminho ***jpeg-compressor\_v104\_r1\bin***.

### **Linux:**

a) Abra a linha de comandos do *Linux*.

b) Através da linha de comandos, navegue para o diretório de **caminho *jpeg-compressor\_v104\_r1\bin\_linux***.

c) Para comprimir digite ***jpgc* (*|nome do ficheiro a comprimir|.extensão*) (*|nome do ficheiro comprimido|.jpg*) (*fator de qualidade*)** na linha de comandos ( $1 \leq \text{fator}_{\text{qualidade}} \leq 100$ ).

Para descomprimir digite ***jpgc -d* (*|nome do ficheiro a descomprimir|.jpg*) (*|nome do ficheiro comprimido|.tga*)** na linha de comandos.

## 2. Flags de execução

### **Opções suportadas em ambos os modos (compressão e descompressão):**

***-glog***      adicionar *output* ao ficheiro de *log*

### **Opções suportadas no modo de compressão:**

***-o***          ativar tabelas de *Huffman* Otimizadas (mais lento, mas ficheiros mais pequenos)  
***-luma***      imprimir a imagem com o modo *Y-only*  
***-h1v1, -h2v1, -h2v2***      *chroma subsampling* (predefinido é *Y-only* ou *H2V2*)  
***-m***          teste de compressão memória para memória (em vez de memória para o ficheiro)  
***-w \$file***    escrever imagem comprimida para o ficheiro *\$file.tga*  
***-s***          Usa o ficheiro *stb\_image.c* para descomprimir em vez do ficheiro *jpgd.cpp*