

## Aula Laboratorial Nº 5

### Experiências com codificador JPEG

#### Introdução

Pretende-se neste trabalho executar várias experiências envolvendo o codificador JPEG, usando os seus vários modos de codificação.

#### Preparação

1. Estude com cuidado as notas teóricas e material nela referenciado incidindo sobre codificação de imagem e também [1-3].
2. Vai necessitar do código fonte do codificador JPEG incluído no pacote deste trabalho ou alternativamente o código descarregado de <https://github.com/thorfdbg/libjpeg>. Descompacte o ficheiro libjpeg-master.zip e compile todo o projecto, para obter o executável jpeg.
3. Também vai necessitar da imagem CrowdRun.ppm que é uma imagem não comprimida em formato RGB com 2160 linhas por 3840 colunas e 3x8 bits por pixel. Descompacte o ficheiro CrowdRun.ppm.zip para obter a imagem pronta para as experiências seguintes.

**Nota 1:** Antes de iniciar as experiências, leia com atenção o README que encontra na pasta do software.

**Nota 2:** Após compilar o software corra numa janela powershell (Windows) ou num Shell (Linux ou MacOS) o comando `jpeg -h` para ver a lista de opções suportadas.

**Nota 3:** O comando `jpeg` serve para codificar (entrada é uma imagem, saída é um ficheiro .jpg) e para decodificar (entrada é um ficheiro .jpg e saída é uma imagem).

#### Experimentação

##### Experiência 1 - Comparação entre codificação com perda e sem perda

1. Comprima a imagem CrowdRun em modo JPEG lossless (sem perda) usando o comando

```
jpeg -l -q 100 -c CrowdRun.ppm CrowdRunLS.jpg
```

em que as opções **-l -q 100** especificam codificação com qualidade máxima (lossless) com uso de uma DCT 1D reversível.



Com base nos tamanhos (número de bytes) do ficheiro da imagem original TO e no do ficheiro contendo a imagem comprimida TC, calcule o valor aproximado da taxa de compressão TO/TC.

2. Comprima de novo a imagem original, mas agora em modo lossy (com perda), com factor de qualidade 20 (opção -q 20), usando o comando

```
jpeg -q 20 CrowdRun.ppm CrowdRunQ20.jpg
```

Compare os tamanhos do ficheiro original e comprimido e calcule o valor aproximado da taxa de compressão TO/TC. Abra a imagem original e a comprimida com um visualizador de imagens e observe-as lado a lado. Onde observa mais diferenças ? Porquê ?

3. Agora faça várias compressões com qualidades  $q=25, 50, 75, 100$ , calculando todas as taxas de compressão para cada. Observe cada um dos ficheiros comprimidos (para os vários valores de  $q$ , e determine o menor valor de  $q$ , tal que a imagem comprimida é indistinguível da original. Compare a taxa de compressão conseguida para esse  $q$  e com compressão lossless. Esses valores são muito diferentes ? Quão melhor (percentualmente) é essa taxa de compressão com perda relativamente à compressão sem perda ?

### Experiência 2 – Efeito das tabelas de quantização

4. O codificador JPEG fornecido suporta várias tabelas de quantização dos coeficientes da DCT. Entre elas está uma tabela otimizada tendo em conta as características do sistema visual humano (opção -qt 4). Codifique de novo a imagem usando o comando

```
jpeg -q 20 -qt 4 CrowdRun.ppm CrowdRunQ20QT4.jpg
```

em que “-qt 4” especifica o uso dessa tabela de quantização otimizada. Compare a imagem CrowdRunQ20QT4.jpg com a imagem CrowdRunQ20.jpg Qual lhe parece ter mais qualidade ? Os tamanhos dos ficheiros são muito diferentes ? O que conclui ?

### Experiência 3 – Codificação progressiva

5. O codificador JPEG fornecido suporta codificação progressiva. Codifique a mesma imagem original usando o comando

```
jpeg -v -q 50 CrowdRun.ppm CrowdRunProg.jpg
```

De seguida, usando um editor binário ou um comando Unix como

```
head -c n CrowdRunProg.jpg > CrowdRunTruncn.jpg
```

trunque o ficheiro conservando só os  $n$  primeiros bytes do mesmo.

No caso de Windows deve descarregar e usar o freeware **partcopy** para truncar o ficheiro.



Crie dez ficheiros truncados derivados de CrowdRunProg.jpg, para **n** igual a 20.000, 70.000, 90.000, 120.000, 160.000, 300.000, 350.000, 400.000, 450.000 e 500.000

Com um utilitário de visualização de imagens observe cuidadosamente as várias imagens, verificando que diferenças são observáveis e onde. Deve observar as imagens à escala real. O que observa ? O que conclui acerca da codificação progressiva ?

#### Experiência 4 – Codificação Entrópica

6. O codificador JPEG fornecido suporta codificação Huffman, codificação aritmética e ainda Huffman com optimização dos códigos de Huffman.

a. Codifique a mesma imagem original usando o comando

```
jpeg -q 20 CrowdRun.ppm CrowdRunQ20.jpg
```

e tome nota do tamanho do ficheiro gerado. Neste caso foram usadas as tabelas de Huffman pré-definidas.

b. Codifique a mesma imagem original usando o comando

```
jpeg -q 20 -ar CrowdRun.ppm CrowdRunQ20AR.jpg
```

e tome nota do tamanho do ficheiro gerado. Neste caso foi usada **codificação aritmética** (opção -ar).

c. Por último codifique a mesma imagem original usando o comando

```
jpeg -q 20 -h CrowdRun.ppm CrowdRunQ20HOPT.jpg
```

e tome nota do tamanho do ficheiro gerado. Neste caso foram usadas **tabelas de Huffman optimizadas** (opção -h).

Compare os tamanhos dos três ficheiros e tire conclusões acerca da eficiência dos vários métodos de codificação entrópica. Use o comando **time** existente em sistemas **UNIX** ou o **Measure-command** do **PowerShell de Windows** para contabilizar o tempo gasto em cada codificação. Qual das codificações foi mais rápida ?

## Relatório

Prepare um relatório descrevendo as experiências que fez, explicando o que observou. Se necessário inclua capturas do ecrã que ajudem a perceber as suas explicações. Submeta o relatório no inforestudante.



## Bibliografia

- [1] [Secção 17.3 de “Multimedia Communication Technology”, J.R. Ohm
- [2] Artigo “The JPEG Still Picture Compression Standard”, Gregory K. Wallace
- [3] Capítulo 8 de “Applied Signal Processing A MATLAB-Based Proof of Concept”,  
Thierry Dutoit , Ferran Marqués