

Sistemas Multimédia

2018/2019

Aula Prática 10

I. Compressão de Ficheiros de Imagem

1. Carregue para o *workspace* do MATLAB/Octave o conteúdo da imagem guardada no ficheiro 'Parede_8bit.bmp' que se encontra no Moodle (nesta imagem, a cor de cada pixel é armazenada por 8 bits). Para tal, use o comando:

$\gg [Image, ColorMap] = imread('Parede_8bit.bmp');$

ficando a informação guardada na matriz ($N \times M$) *Image*. Visualize a imagem no MATLAB/Octave.

2. Desenvolva, agora, a função:

Save8bitImage (*Filename*, *Image*)

que guarda num ficheiro (cujo nome é especificado pela *string* *Filename*) a informação contida na matriz *Image*. Esse ficheiro deve seguir a seguinte especificação:

- os primeiros 2 bytes guardam o valor de N (no formato uint16);
- os 2 bytes seguinte guardam o valor de M (no formato uint16);
- os restantes bytes guardam todos os valores de *Image* (no formato uint8).

Teste a função com a matriz gerada na alínea anterior e registe o tamanho (em bytes) desse ficheiro. Tenha o cuidado em atribuir o nome ao ficheiro para que este não coincida com o de outro ficheiro que exista na pasta de trabalho.

3. Para verificar a função anterior, desenvolva agora a função:

$[N, M, Im] = \mathbf{Load8bitImage}(\textit{Filename})$

que carrega a informação guardada no ficheiro criado na pergunta anterior, ficando a matriz da imagem na variável *Im*. Visualize a imagem carregada deste ficheiro, e verifique que é igual à imagem original (da pergunta 1).

4. Implemente, agora, uma função que proceda à compressão (sem perdas) da informação contida na matriz da imagem com base no algoritmo RLE (*Run-Length Encoding*):

$Stream = \mathbf{EncodeImage_RLE}(Image)$

O resultado desta função é uma sequência de bytes (do tipo uint8), guardados no vetor $K \times 1$ *Stream*, correspondente à compressão efetuada na matriz *Image*, onde o varrimento deverá ser efetuado preferencialmente ao longo dos pixels de cada linha. Note que, como o número de ocorrências consecutivas de cada cor é armazenado num uint8, esta função deverá atender a esta limitação.

5. Pretendendo guardar num ficheiro de menor tamanho a informação resultante da aplicação do método RLE à imagem original, desenvolva a função:

Save8bitStream (*Filename*, $N, M, Stream$)

que guarda no ficheiro indicado por *Filename* a sequência de dados produzida pela função da pergunta anterior. De novo, esse ficheiro começa por conter os valores de N

e M (no formato uint16), seguindo-se o conteúdo de $Stream$ (no formato uint8). Teste com a imagem em análise e verifique a taxa de compressão que foi possível obter, quando comparado com o ficheiro criado na pergunta 2.

6. De novo para verificar o funcionamento da função anterior, desenvolva a função que leia a informação contida no ficheiro comprimido anterior:

$[N, M, Stream] = \text{Load8bitStream}(Filename)$

e teste-a (i.e., verifique se a sequência guardada no ficheiro (pergunta 5) é igual à sequência lida a partir do ficheiro).

7. Finalmente, falta decodificar a informação comprimida, para se obter novamente a imagem original. Desenvolva a função:

$Image = \text{DecodeImage_RLE}(N, M, Stream)$

que volta a gerar a matriz dos pixels da imagem que é obtida por interpretação da informação contida em $Stream$ (sequência de dados produzida pelo RLE). Teste a função visualizando a imagem resultante (que terá que ser igual à imagem original).

8. Repita o processo, agora, considerando a imagem 'Arca_8bit.bmp'. Como explica o valor da taxa de compressão obtida com esta nova imagem? O que conclui relativamente ao algoritmo de compressão RLE?