

Sistemas Multimédia

2018/2019

Aula Prática 11

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:

`>> [Image, ColorMap] = imread('Parede_8bit.bmp');`

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

2. Desenvolva a função:

`[Symb, Freq] = ImageSymbols(Image)`

que retorna, em *Symb*, a lista ($K \times 1$) de símbolos (i.e., códigos de cores) existentes na imagem, assim como a frequência (em percentagem) com que ocorre cada símbolo (vetor *Freq*, ($K \times 1$)).

3. Note que o número de símbolos é bastante menor que $2^8 = 256$, pelo que, logo à partida, poder-se-ia comprimi substancialmente a imagem reduzindo o número de bits. Representando cada símbolo pelo índice correspondente do vetor *Symb*, e usando o menor número de bits possível para representar esses índices, qual seria o tamanho do ficheiro que conteria toda a informação da imagem?

4. Usando representação de 8 bits (apenas para simplificar a execução desta alínea), implemente a função:

`Stream = EncodeImage_LZ77(Image, Nw, Mw)`

que codifica a informação contida na matriz *Image* segundo o método LZ77, onde *Nw* é a dimensão (em bytes) da janela deslizante, e *Mw* a dimensão (em bytes) da janela de observação. A matriz deverá ser analisada ao longo de cada linha, não havendo qualquer alteração ao algoritmo aquando da transição para a linha seguinte (i.e., o algoritmo analisa a sequência de dados como se não houvesse mudança de linha). Sugere-se os vários passos sejam devidamente organizados na implementação (por exemplo, fazendo recurso a subfunções para a implementação de tarefas específicas, como pesquisar uma sequência de símbolos na janela deslizante atual). Como é necessário distinguir os códigos dos símbolos com os pares posição-comprimento, os códigos dos símbolos devem ser representados pelos índices em *Symb*, e o primeiro byte do par posição-comprimento deverá ser sinalizado com o seu bit mais significativo a 1 (esta estratégia limita o valor máximo de *Nw* – determine qual esse valor).

Teste esta função para diferentes dimensões das janelas, e observe a dimensão obtida para *Stream* (sendo este um valor indicativo do nível de compressão conseguido).

5. Desenvolva, agora, a função:

SaveLZ77Image (*Filename, N, M, Nw, Mw, Symb, Stream*)

que guarda num ficheiro (cujo nome é especificado pela *string Filename*) a informação contida no vetor *Stream*. 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);
- o byte seguinte guarda a dimensão da janela deslizante *Nw* (no formato uint8);
- o byte seguinte guarda a dimensão da janela de observação *Mw* (no formato uint8);
- o byte seguinte guarda o número de símbolos existentes na imagem original;
- seguidamente, é armazenado o conteúdo da lista de símbolos, *Symb* (um elemento no formato uint8 para cada símbolo);
- os restantes bytes guardam todos os valores de *Stream* (no formato uint8).

Teste a função com os resultados obtidos 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.

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

$[N, M, Nw, Mw, Symb, Stream] = \mathbf{LoadLZ77Image}(Filename)$

que carrega a informação guardada no ficheiro criado na pergunta anterior. Teste a função, verificando que os dados carregados coincidem com os dados gravados anteriormente.

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

$Image = \mathbf{DecodeImage_LZ77}(N, M, Nw, Mw, Symb, Stream)$

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 LZ77). 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'. Compare os resultados obtidos na compressão de ambas as imagens, e compare também com o desempenho do método RLE (visto na aula anterior).