

UNIVERSIDADE DE EVORA
Escola de Ciências e Tecnologias

Engenharia Informática
Teoria da Informação
2017/2018

Compressão e Descompressão de Imagens Binárias

Docente: Miguel Barão
Discentes: Cláudia Dias - 35308
João Queimado - 38176

1 Introdução

Pretende-se desenvolver um sistema de compressão e descompressão de imagens binárias. Uma imagem binária é uma imagem em que cada pixel apenas pode tomar um de dois valores: preto ou branco. Este tipo de imagens podem ser representadas como uma matriz zeros e uns, em que a coluna representa a posição horizontal na imagem e a linha a posição vertical.

O programa deverá receber uma imagem binária com extensão .pbm retornando um ficheiro através do algoritmo LZW a sua compressão com extensão .best

Input: Imagem com extensão .pbm

Output: Ficheiro com extensão .best

2 Algoritmo Utilizado

Utilizou-se o algoritmo LZW, porque se entendeu como o mais adequado para o fim da realização deste trabalho. O algoritmo LZW (Lempel-Ziv-Welch) é um algoritmo de compressão de dados, derivado do algoritmo LZ78, baseado na localização e no registro das padronagens de uma estrutura. É geralmente utilizado em imagens em que não se pode perder a definição original. Nas imagens, o algoritmo lê os valores de pixels de uma imagem bitmap e elabora uma tabela de códigos onde se representam as padronagens repetidas dos pixels encontrados. O codificador LZW reduz, pela compressão, os arquivos de imagens gráficas a 1/3 ou 1/4 de seu tamanho original.

3 Compressão - Funções

3.1 *compress()*

NOME

compress - Comprime a imagem

DESCRIÇÃO

Esta função tem como objetivo comprimir a imagem com extensão .pbm para um array.

3.2 *write file(fo, s, fn)*

NOME

write file - Escreve para o ficheiro .best

DESCRIÇÃO

Esta função é a responsável por escrever para o ficheiro com extensão .best

fo - file output

s - tamanho

fn - file name

3.3 *read file(fn)*

NOME

read file - Lê para a string

DESCRIÇÃO

Esta função possibilita a leitura para a string.

fn - file name

3.4 *main()*

NOME

main - Avalia o input introduzido comprime para o ficheiro com extensão .best

DESCRIÇÃO

A função main utiliza todas as outras funções acima referidas.

4 Descompressão - Funções

4.1 *desompress()*

NOME

descompress - descomprime a imagem

DESCRIÇÃO

Esta função tem como objetivo descomprimir o array com extensão .best

4.2 *write file(fo, s, fn)*

NOME

write file - Escreve do array para o ficheiro .pbm

DESCRIÇÃO

Esta função é a responsável por escrever do array para o ficheiro com extensão .pbm

fo - file output

s - tamanho

fn - file name

4.3 *read_{file}(fn)*

NOME

read file - Lê para a string

DESCRIÇÃO

Esta função possibilita a leitura para a string. fn - file name

4.4 *main()*

NOME

main - Avalia o input introduzido comprime para o ficheiro com extensão .best

DESCRIÇÃO

A função main utiliza todas as outras funções acima referidas.

5 Entropia - Funções

5.1 *read file(fn)*

NOME

read file - lê o ficheiro

DESCRIÇÃO

Esta função tem como objetivo ler o ficheiro original

5.2 *read file comp(fn)*

NOME

read file comp - lê o ficheiro

DESCRIÇÃO

Esta função tem como objetivo ler o ficheiro comprimido

5.3 *probabilidades(ff)*

NOME

probabilidades - calcular as probabilidades

DESCRIÇÃO

Esta função tem como fim calcular quantos uns, quantos zeros e quantos números há na imagem com extensão .pbm. Vai calcular a probabilidade de cada um e cada zero aparecer.

5.4 *entropia(p0,p1)*

NOME

entropia - calcular a entropia

DESCRIÇÃO

Esta função calcula o valor da entropia.

5.5 *condicionada(ff, p0, p1)*

NOME

condicionada - calcular a entropia

DESCRIÇÃO

Esta função calcula o valor da entropia condicional.

5.6 *comprimido(ff)*

NOME

comprimido - calcular a entropia

DESCRIÇÃO

Esta função calcula o valor da entropia do ficheiro comprimido.

5.7 *main()*

NOME

main - Avalia o input introduzido comprime para o ficheiro com extensão .best

DESCRIÇÃO

A função main utiliza todas as outras funções acima referidas.

6 Cálculo da Entropia

6.1 *Entropia $H(X)$*

Há no total 2073600 números de pixéis, sendo 1420586 uns e 653014 zeros.

A probabilidade de uns e zeros calcula-se através do numero total de pixéis, ou seja:

$$p(1) = 1420586/2073600 = 0.685$$

$$p(0) = 653014 / 2073600 = 0.315$$

A entropia vai ser:

$$H(X) = -p(0) * \log_2(p(0)) - p(1) * \log_2(p(1))$$

$$H(X) = 0.899bits$$

6.2 *Entropia Condicional $H(X)$*

$$H(Y|X) = -\text{somatório}(p(x) * \text{somatório}(p(y|x) \log_2 p(y|x)))$$

$$H(Y|X)] = 0.827bits$$

6.3 *Entropia do ficheiro comprimido $H(X)$*

$$H(X)) = 15.577bits$$

7 Conclusão

Este trabalho foi bastante útil para a nossa aprendizagem e para a consolidação dos nossos conhecimentos. Está a funcionar corretamente.