

UNIVERSIDADE DE ÉVORA

DISCIPLINA DE TEORIA DE INFORMAÇÃO

---

# Sistema de Compressão e Descompressão de Imagens

---



UNIVERSIDADE  
DE ÉVORA

*Autores:*

José Ferreira nº 34145

Daniel Soares nº 34222

*Docente:*

Miguel Barão

January 10, 2019

# Índice

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Compressão</b>	<b>3</b>
<b>3</b>	<b>Descompressão</b>	<b>4</b>
<b>4</b>	<b>Cálculos</b>	<b>5</b>
4.0.1	Performance . . . . .	5
4.0.2	Entropia . . . . .	5
4.0.3	Entropia Condicional . . . . .	5
<b>5</b>	<b>Conclusão</b>	<b>6</b>
5.0.1	Performance . . . . .	6
5.0.2	Entropia . . . . .	7
5.0.3	Entropia Condicional . . . . .	8

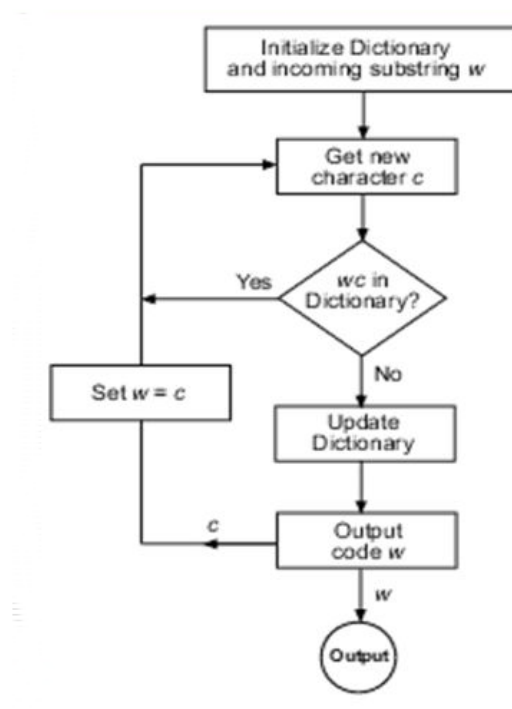
# 1 Introdução

No contexto da disciplina de Teoria de Informação é pretendido que com o uso de um algoritmo de compressão, seja possível comprimir e descomprimir uma imagem guardada com formato PBM (Portable BitMap). Para além destas funções existe a necessidade de determinar a sua entropia, entropia condicional e o desempenho do sistema de compressão escolhido. Sobre estes dados será necessário realizar uma análise e retirar as devidas conclusões. Para tal objectivo foi escolhido o algoritmo de compressão LZW (Lempel-Ziv-Welch). O LZW é um algoritmo de compressão que deriva do algoritmo LZ78, baseado na localização e no registo das padronagens de uma estrutura. Este algoritmo é geralmente usado em imagens em que não se pode perder a definição original.

## 2 Compressão

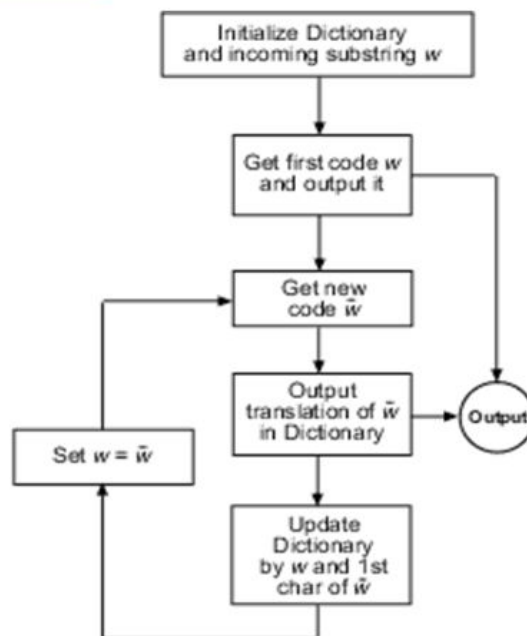
Tal como referido anteriormente o algoritmo de compressão utilizado foi o LZW. Este utiliza repetição de padrões de forma a poupar espaço. São inseridos numa fase inicial os símbolos iniciais como as primeiras duas entradas do dicionário. O algoritmo vai então tentar extender o dicionário criado inicialmente com novos e únicos símbolos, há medida que estes vão aparecendo na informação a comprimir, criados com combinações de símbolos já existentes no dicionário. Este método pode ser utilizado tanto para compressões como para descompressões.

Escolheu-se este algoritmo de compressão pois para além de ser competente para imagens possui uma implementação simples e de fácil compreensão.



### 3 Descompressão

O algoritmo escolhido também é responsável pelo o trabalho de descompressão. O dicionário na fase inicial, do processo de descompressão, encontra-se cheio com todos os símbolos do alfabeto (0 e 1). Ele vai decodificando o ficheiro recebido, lendo o actual ( $a$ ) e o proximo ( $p$ ). Se  $p$  se encontrar no dicionario, é criada uma nova entrada composta por  $a +$  o primeiro símbolo de  $p$ , e esta é inserida na última posição do dicionário. Se  $p$  corresponde a uma entrada não existente no dicionario, cria-se uma entrada composta com o primeiro simbolo de  $a + a$ , e esta é inserida na última posição do dicionário. Desta forma o ficheiro vai sendo descomprimido acompanhando o crescimento do dicionário até que este possua todos os símbolos (combinações dos símbolos iniciais) do ficheiro.



## 4 Cálculos

### 4.0.1 Performance

A performance é obtida fazendo a diferença entre o tamanho do array original e o número de dígitos do ficheiro comprimido e em seguida é feita uma divisão pelo tamanho do array original e uma multiplicação por 100. Isto permite saber qual a percentagem do ficheiro que foi comprimida.

$$p = \frac{t_a - n_d}{t_a} * 100$$

### 4.0.2 Entropia

A entropia é uma medida de incerteza, ou seja trata-se da medida da surpresa gerada por uma fonte. Neste caso permite saber a surpresa gerada por os bits da imagem.

$$H = \sum p(x) * \log(p(x))$$

### 4.0.3 Entropia Condicional

A entropia condicional permite medir a incerteza de uma variável sabendo a ocorrência anterior.

$$H_{condicional} = \sum p(x) \sum p(y|x) * \log(p(y|x))$$

## 5 Conclusão

Depois de analisados os resultados obtidos através da performance, entropia e entropia condicional foi possível chegar às seguintes conclusões:

### 5.0.1 Performance

Neste caso podemos verificar que ela será tanto maior quanto mais repetições de pixels sequenciais que ocorrerem e outro factor influente será as dimensões das imagens. Ou seja, frequentemente quanto maior a imagem original maior será a performance do programa.

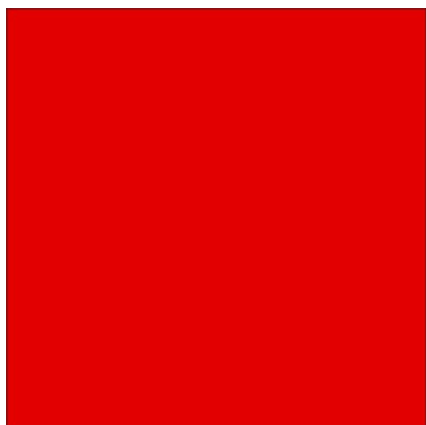


Figure 1: Imagem maior



Figure 2: Imagem pequena

Fig.2 trata-se de um corte de Fig.1. Fig.1 tem  $p = 99.4\%$  e Fig.2 tem  $p = 98.5\%$ .

### 5.0.2 Entropia

Depois de testar diversos casos chegou-se á conclusão que a entropia varia com a informação contida nas imagens. Imagens com vários detalhes possuem níveis de entropia maiores e imagens mais simples entropias menores. Por isso conclui-se que entropias maiores correspondem a imagens com mais elementos presentes (com maior nível de detalhe).



Figure 3: Imagem mais simples

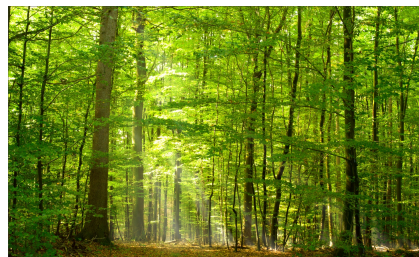


Figure 4: Imagem mais complexa

Neste caso a Fig.3 possui  $H=0.20$  e a Fig.4 possui  $H=0.95$ .



### 5.0.3 Entropia Condicional

Ao calcular a entropia condicional estaremos a assumir que se conhece um evento logo a entropia condicional vai ser a medida de nova informação adquirida depois de um evento ser observado. Esta é menor que a entropia, devido há existência de menos incerteza pois já se conhece o bit anterior. Depois da observação de vários exemplos chegou-se á conclusão que mesmo para imagens com entropias muito semelhantes, ocorria uma maior variação entre a entropia e a entropia condicional na presença de padrões bem definidos. Quanto mais bem definido é o padrão menor é a sua entropia condicional.



Figure 5: Imagem com padrão bem definido



Figure 6: Imagem sem nenhum padrão

Neste caso a Fig.5 possui  $H=0.98$  e uma  $H_{\text{condicional}}=0.55$  e a Fig.6 possui  $H=0.99$  e uma  $H_{\text{condicional}}=0.82$ .