

# RELATÓRIO – PROJECTO 1

## Entropia, redundância e Informação Mútua



FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

### Realizado por:

João Alexandre Santos Cruz 2018288464

André Cristóvão Ferreira da Silva 2018277921

Tomás da Fonseca Pereira Ventura 2018279147

Eurico José Pereira de Sousa 2016225648

### Exercício 3 - introdução:

Neste exercício era pedido a distribuição estatística, a entropia de cada uma das fontes fornecidas e o histograma gerado a partir de cada fonte e o respetivo alfabeto. Para obter os dados pedidos, criámos a rotina `entropia.m` que recebe uma fonte e o seu alfabeto, devolvendo o respetivo valor da entropia, calculada fazendo o produto entre a matriz de probabilidades e o logaritmo de base 2 dessa mesma matriz. A matriz de probabilidades foi por sua vez calculada usando a rotina `ocorrencias.m`, que conta as frequências dos símbolos do alfabeto na fonte. Os símbolos do alfabeto que não estão na fonte não são contabilizados.

Os histogramas são produzidos com recurso à função do matlab `histogram` que apenas recebe a fonte, e faz o histograma ponde os símbolos do alfabeto no eixo do x e as respetivas frequências no eixo do y.

### Resultados:

#### Fonte: `landscape.bmp`

- Entropia da informação: 7.606914

#### Fonte: `MRI.bmp`

- Entropia da informação: 6.860542

#### Fonte: `MRIbin.bmp`

- Entropia da informação: 0.661080

#### Fonte: `soundMono.wav`

- Entropia da informação: 4.065729

#### Fonte: `lyrics.txt`

- Entropia da informação: 4.410705

## Análise dos resultados:

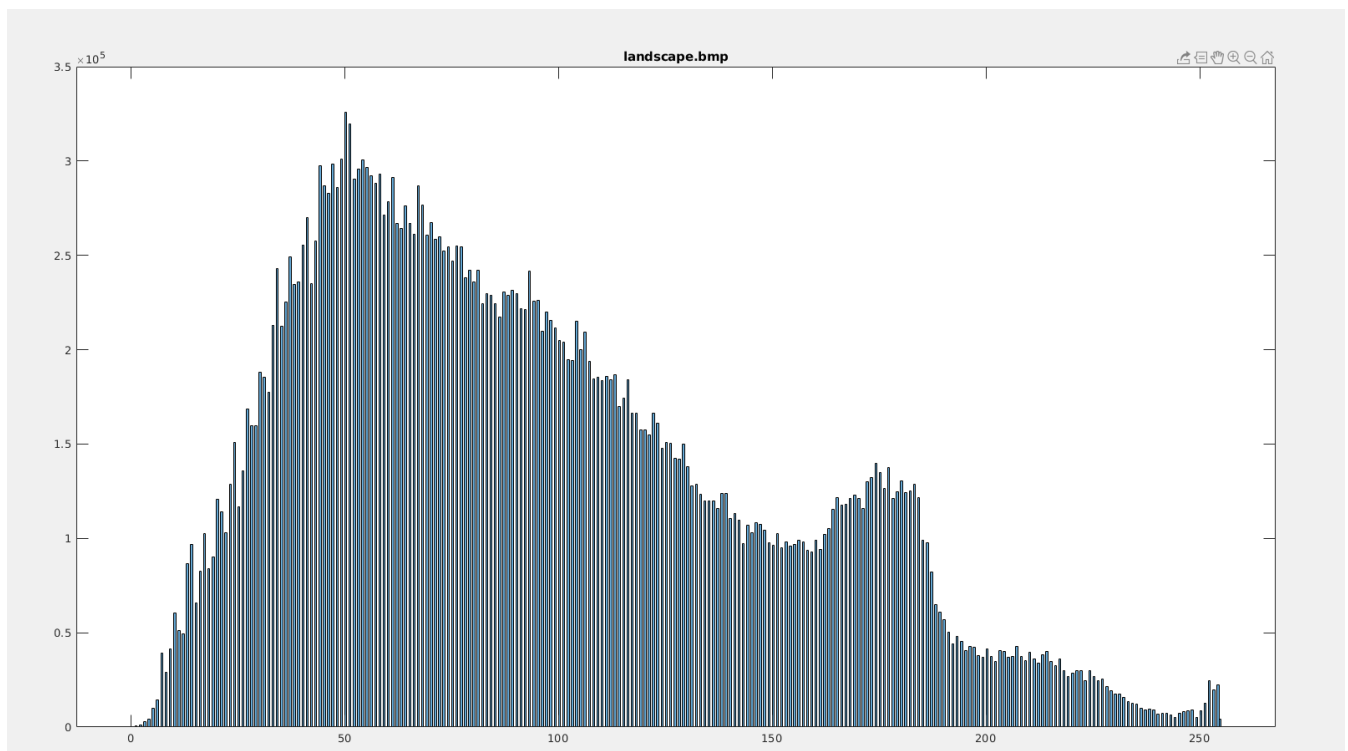
### Imagem – landscape.bmp

Como se pode ver a entropia da primeira imagem tem o valor mais elevado de todos. Isto deve-se ao facto desta fonte ter uma maior dispersão de dados, e por conseguinte, uma maior incerteza associada a cada símbolo, o que significa que num dado pixel da imagem a probabilidade do próximo veicular a mesma informação é baixa.

Observando o histograma produzido a partir desta fonte, verifica-se em primeiro lugar, uma elevada densidade de barras, o que significa que, há uma quantidade de pixéis com valores diferentes, na escala de cinzentos, muito elevada. Em segundo lugar é possível observar, uma maior concentração de pixéis, a variar entre os valores 40 e 60 da escala. Contrastando com uma baixa concentração de pixéis com valores correspondentes a zonas mais claras.

Tudo isto é possível conferir olhando para a imagem fonte, onde se vê que há uma prevalência de tonalidades mais escuras que claras. No entanto, há no diagrama um pico de valores entre o valor 165 e 185, que corresponderá principalmente ao sol que se vê na imagem.

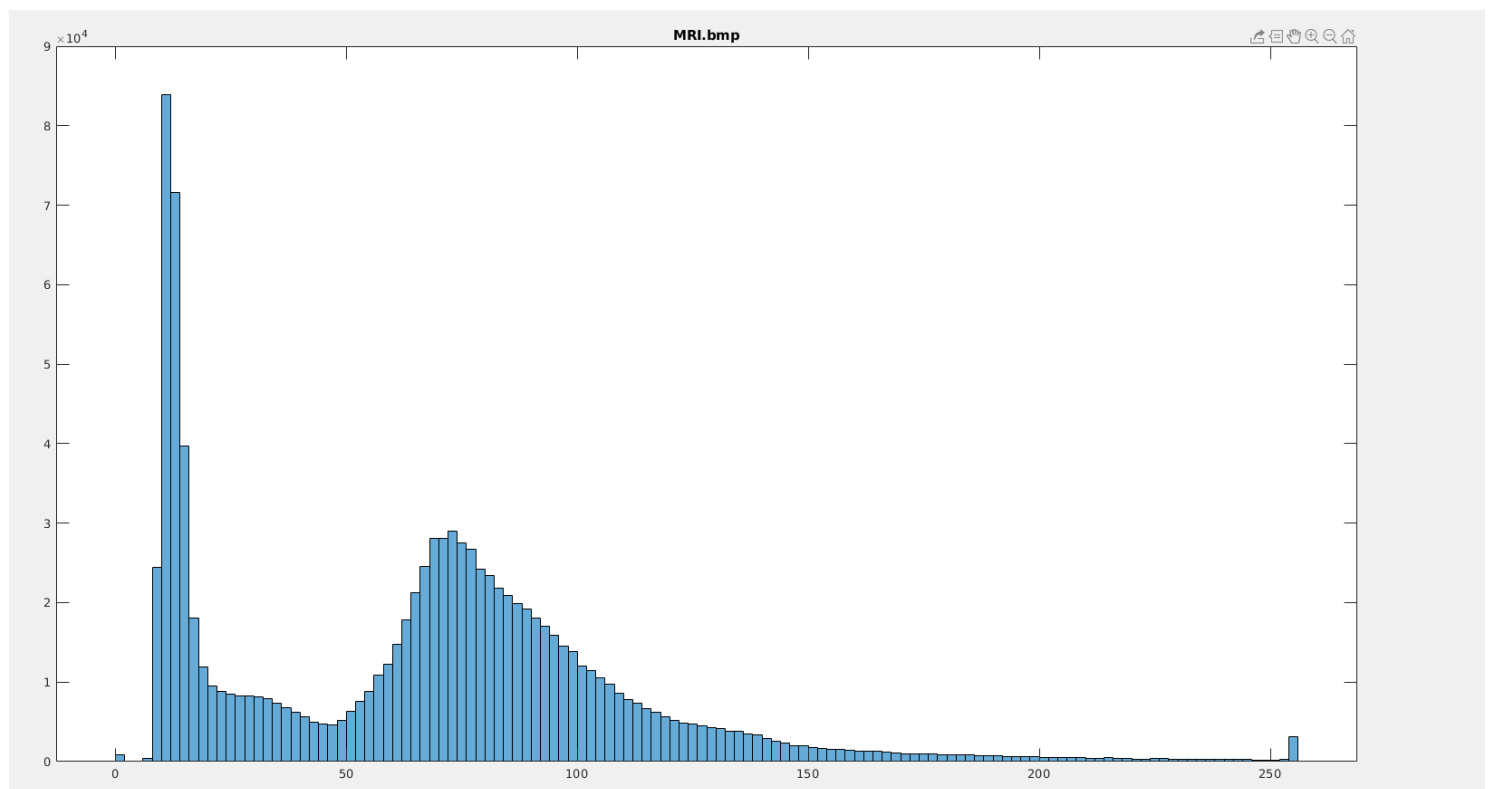
Percebe-se assim com mais clareza, através desta análise do diagrama de barras, o elevado valor da entropia.



### Imagem - MRI.bmp

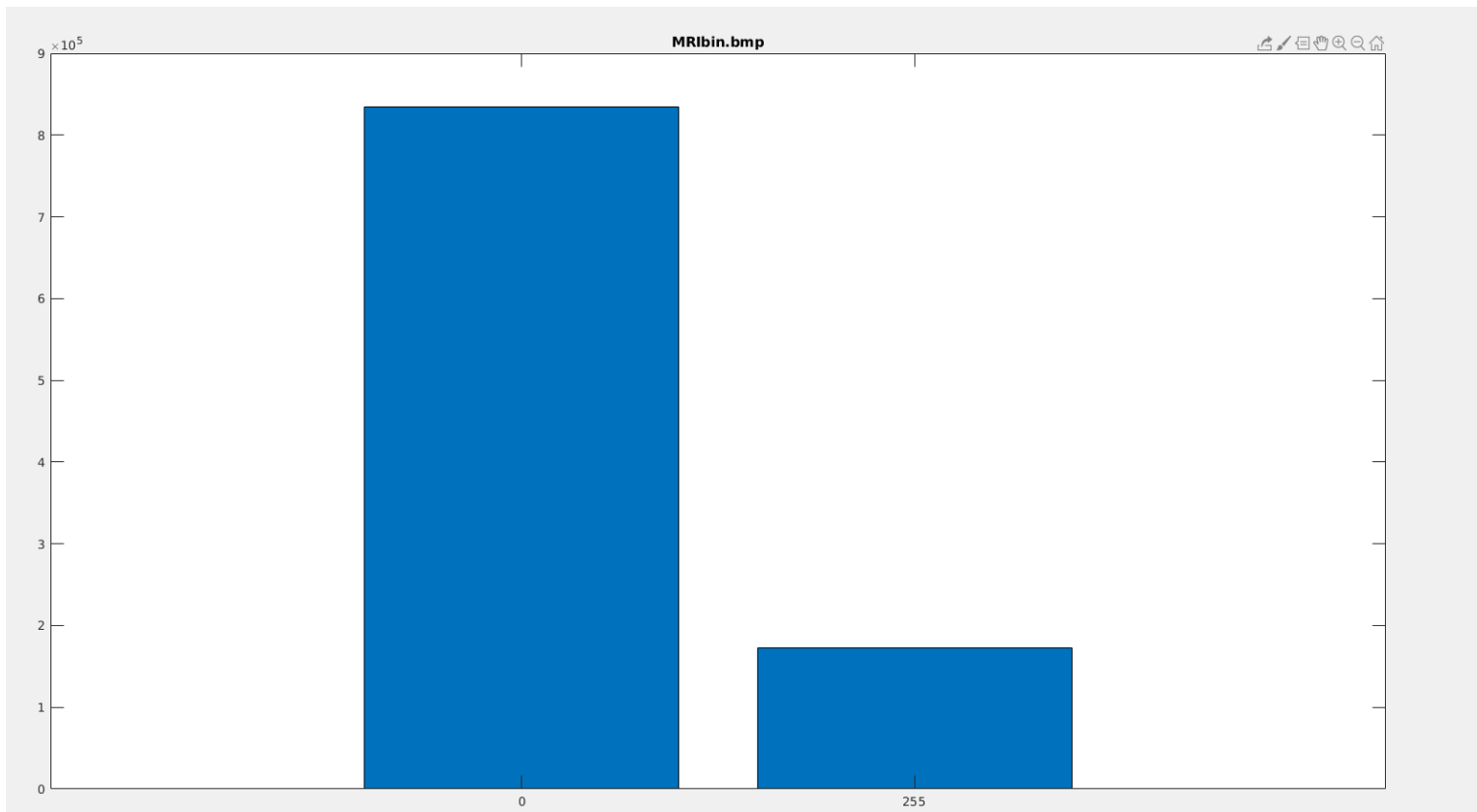
No segundo caso, a imagem MRI também tem uma entropia elevada. Mas esse valor não se deve uma dispersão geral da informação de cada pixel, mas sim há ocorrência de dois picos de valores, bem visíveis, no histograma da imagem.

Fazendo a comparação desta fonte com o seu diagrama, conclui-se que estes picos se devem ao facto de na imagem, haver uma grande área de pixéis onde a informação veiculada por cada um, e o seu adjacente, têm uma grande variância. Esta situação corresponde ao contraste entre o fundo preto, e a elipse cinzenta mais externa. Sendo que o segundo pico corresponde à diferença de tonalidades entre essa elipse externa e uma segunda de pixéis com tonalidade mais escura. Desta forma, compreende-se também o valor da entropia para esta imagem.



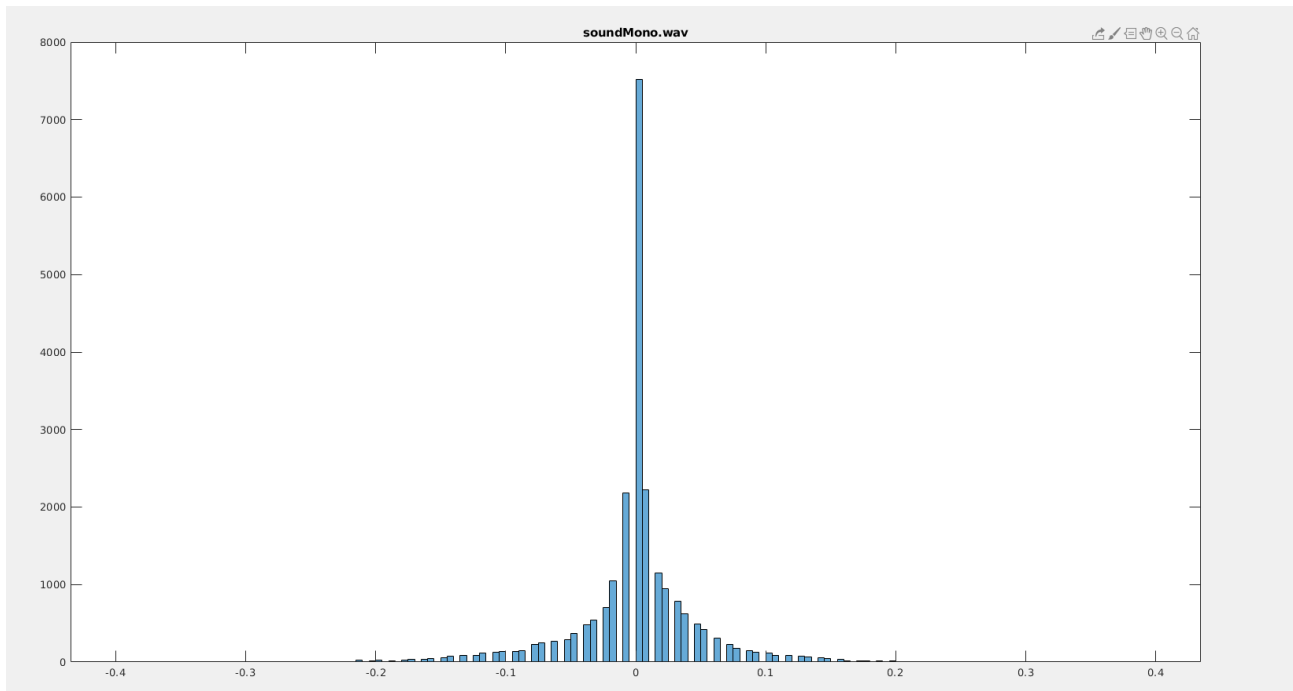
### Imagem – MRIBin.bmp

Seguindo para a terceira imagem, depara-se com uma situação oposta às anteriores, uma vez que a entropia tem um valor muito próximo de zero. Isto explica-se pelo facto da imagem MRIBin ser binarizada, ou seja, o valor dos seus pixéis é 0 ou 255. Logo a probabilidade de um pixel ter a mesma informação do adjacente, é muito elevada, sendo a incerteza baixa. O respetivo diagrama demonstra isso com clareza, porque só há dois conjuntos de valores, 0 e 255.



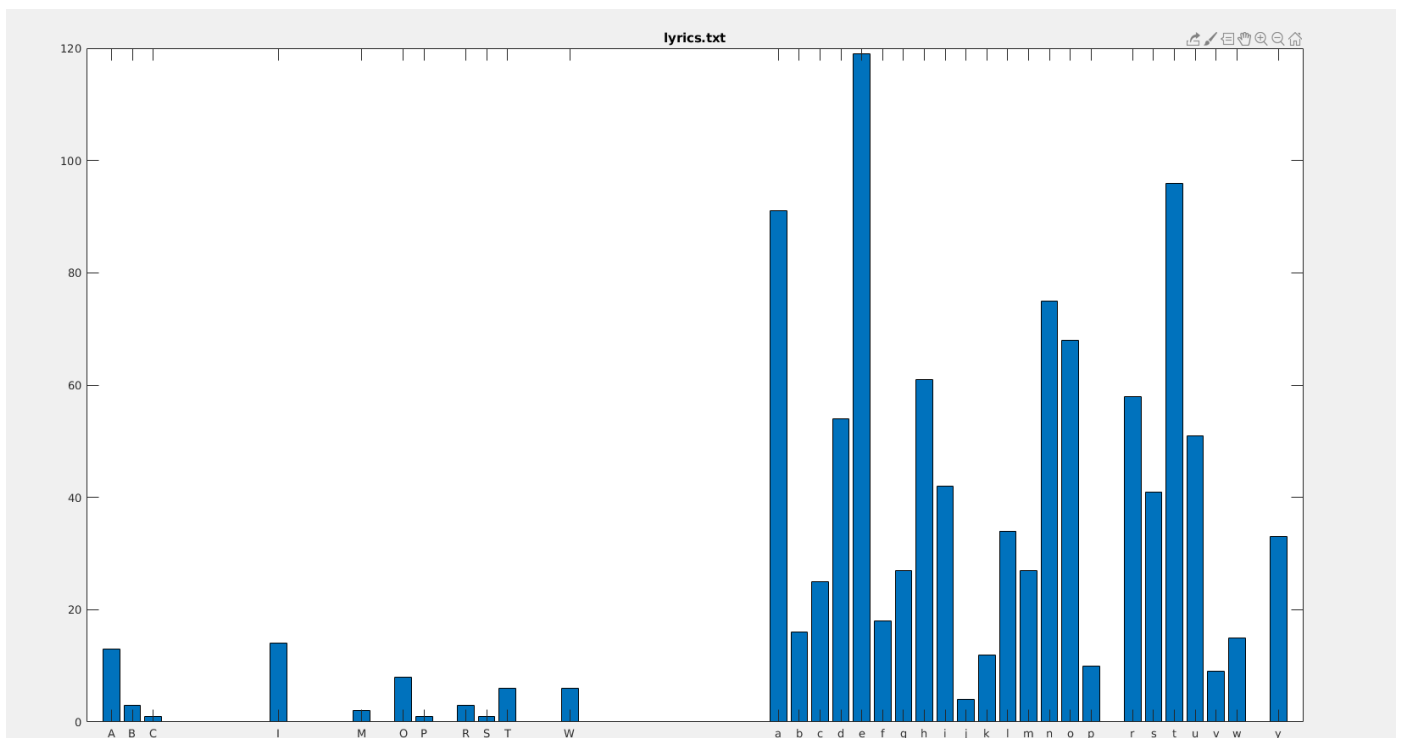
### Som - soundMono.wav

O histograma desta fonte também é caracterizado por uma dispersão elevada das amostras, que como foi mencionado anteriormente promove um aumento da entropia.



### Texto - lyrics.txt

Por último tem-se a fonte texto, que apresenta uma entropia também elevada. Analisando a fonte percebe-se isso pois há uma grande variação entre caracteres existentes no texto. Havendo, no entanto, um conjunto de caracteres que se repetem mais vezes. Por exemplo o carater (e), o (a) e o (t).



## Compressão de dados:

Para que se possa fazer a compressão de dados, é preciso ter em conta o conceito de redundância da informação, no sentido em que se numa dada fonte houver muita informação idêntica entre os símbolos, então há uma elevada redundância nessa fonte.

Isto permite fazer uma compressão não destrutiva dos dados, uma vez que, se pode eliminar essa informação redundante, não afetando a reconstrução da fonte no processo de descompressão, onde ocorre uma reconstrução exata da fonte original.

Neste trabalho todas as fontes que foram fornecidas, usam 8 bits por símbolo para guardar a informação veiculada, (dado obtido a partir dos comandos do MathLab que imprimem a informação de um ficheiro), e em todas as fontes a entropia é menor que 8 bits, conclui-se que é possível comprimir as fontes de forma não destrutiva.

## Compressão máxima:

Para calcular a compressão máxima usa-se a fórmula da entropia, quando é máxima:

$$H_{\text{máx}}(\text{fonte}) = \log_2(\text{alfa}) \quad , \text{ onde alfa é o total de elementos do alfabeto da fonte}$$

$$\text{Compressão máx}(\text{fonte}) = [(H_{\text{máx}}(\text{fonte}) - H(\text{fonte})) / H_{\text{máx}}(\text{fonte})] \times 100$$

### Fonte: [landscape.bmp](#)

- Compressão máxima: 4.913575%

### Fonte: [MRI.bmp](#)

- Compressão máxima: 14.243225%

### Fonte: [MRIbin.bmp](#)

- Compressão máxima: 33.892%

### Fonte: [soundMono.wav](#)

- Compressão máxima: 49.1783875%

### Fonte: [lyrics.txt](#)

- Compressão máxima: 13.302514%

## Exercício 4 - introdução:

Para obter os resultados da entropia pedidos neste exercício, utilizámos a rotina fornecida `hufflen.m`, a rotina `ocorrências.m` já mencionada anteriormente e a rotina `momentoCentradoNumeroDeBits.m`, onde esta última rotina recebe como parâmetros uma fonte, o seu alfabeto e a ordem do momento centrado. Com estes dados, passa a fonte e o alfabeto à rotina `ocorrências.m` que devolve uma matriz com as frequências dos símbolos do alfabeto na fonte. Matriz essa que de seguida é passada à rotina `hufflen.m`, que devolve outra matriz com os códigos de huffman para codificar cada símbolo do alfabeto, que ocorre na fonte. De seguida eleva-se a matriz devolvida por `hufflen.m` ao parâmetro ordem passado à função atual, fazendo a adaptação à ordem do momento.

Por último é calculado o produto desta última matriz com a matriz probabilidade, que também é obtida com recurso à rotina `ocorrências.m`. Sendo o resultado o momento centrado de ordem 1, que corresponde à média do número de bits por símbolo necessário para codificar a informação da fonte.

É também pedida a variância associada à entropia dos códigos de huffman, que é calculada a partir da rotina `varianciaNumeroDeBits.m`. Esta rotina também chama `momentoCentradoNumeroDeBits.m` duas vezes, em primeiro através da rotina `numeroMedioBits.m`, para calcular a média, e em segundo chama-a diretamente, mas com a ordem a 2.

Finalmente a variância é calculada a partir da fórmula:  $V(X) = E(X^2) - (E(X))^2$   
Onde  $E(X^2)$  é o momento centrado de ordem 2 e  $E(X)$  a média.

## Resultados:

### Fonte: `landscape.bmp`

- Entropia da informação: 7.606914
- Número médio de bits: 7.629301
- Variância do número médio de bits: 0.751616

### Fonte: `MRI.bmp`

- Entropia da informação: 6.860542
- Número médio de bits: 6.890996
- Variância do número médio de bits: 2.193081

### Fonte: `MRIbin.bmp`

- Entropia da informação: 0.661080
- Número médio de bits: 1.000000
- Variância do número médio de bits: 0.000000

### Fonte: `soundMono.wav`

- Entropia da informação: 4.065729
- Número médio de bits: 4.110714
- Variância do número médio de bits: 4.333505

### Fonte: `lyrics.txt`

- Entropia da informação: 4.410705
- Número médio de bits: 4.443487
- Variância do número médio de bits: 1.082055



## **Análise dos resultados:**

A partir dos resultados, vê-se que os valores da entropia original e da entropia calculada com os códigos de huffman, não variam muito, sendo que a entropia com códigos de Huffman é maior. Isto deve-se ao facto da fórmula usada no exercício 3 para calcular a entropia, ser ótima. Não sendo possível com qualquer outro algoritmo obter o mesmo valor para o número médio de bits por símbolo.

## **Variância:**

Os códigos de huffman funcionam de forma diferente para fontes com alfabetos binários, e fontes com alfabetos de N elementos.

No caso binário o algoritmo agrupa os dois símbolos que ocorrem na fonte, enquanto que no outro, agrupa os que ocorrem com menos frequência na fonte.

Este processo de agrupamento aumenta a eficiência dos códigos, porque baixa o seu comprimento médio, havendo menos símbolos codificados com códigos mais maiores.

## **Reduzir a variância:**

É possível reduzir a variância sem alterar o comprimento médio do código. Para isso pretende-se obter uma árvore com pouca profundidade, onde os agrupamentos de símbolos são sempre colocados o mais próximo da raiz possível, ou seja, ficam com a maior ordem possível.

Desta forma o comprimento dos símbolos é mais uniforme, diminuindo a possibilidade de ocorrência de erros na decodificação, ficando o código mais eficiente.

No entanto é preciso ter em conta que para fazer o agrupamento dos símbolos na árvore N-ária, é necessário que se tenham sempre N elementos para agrupar até se chegar à raiz. Caso contrário haverá perda de códigos mais curtos.

## Exercício 5 - introdução:

Neste exercício é novamente pedido para calcular a entropia e os gráficos, das fontes, mas agora pretende-se que a informação de cada fonte seja agrupada. Para tal usámos a rotina `EntropiaDeAgrupamentos.m`, que determina o número de bits que estão a ser usados para codificar os símbolos da fonte, e faz o seu cast para 16 bits. Ficando desta forma dois símbolos agrupados, quer isto dizer, que passam a partilhar a mesma célula de memória de 16 bits, que é dividida ao meio. Mantendo assim a integridade de informação original.

Os valores resultantes desta operação são colocados na matriz `bloco` que é depois passada à rotina `entropia.m` que calcula a entropia, sendo preciso depois dividir cada elemento da matriz devolvida por dois, para se obter a entropia do agrupamento.

Para apresentar os gráficos em 3D usámos a rotina `GraphBar.m` que recebe a matriz `bloco`, o alfabeto desta matriz, o título do gráfico e o índice do subplot.

## Resultados:

### Fonte: `landscape.bmp`

- Entropia da informação: 7.606914
- Entropia dos agrupamentos: 6.277266

### Fonte: `MRI.bmp`

- Entropia da informação: 6.860542
- Entropia dos agrupamentos: 5.226929

### Fonte: `MRIbin.bmp`

- Entropia da informação: 0.661080
- Entropia dos agrupamentos: 0.400694

### Fonte: `soundMono.wav`

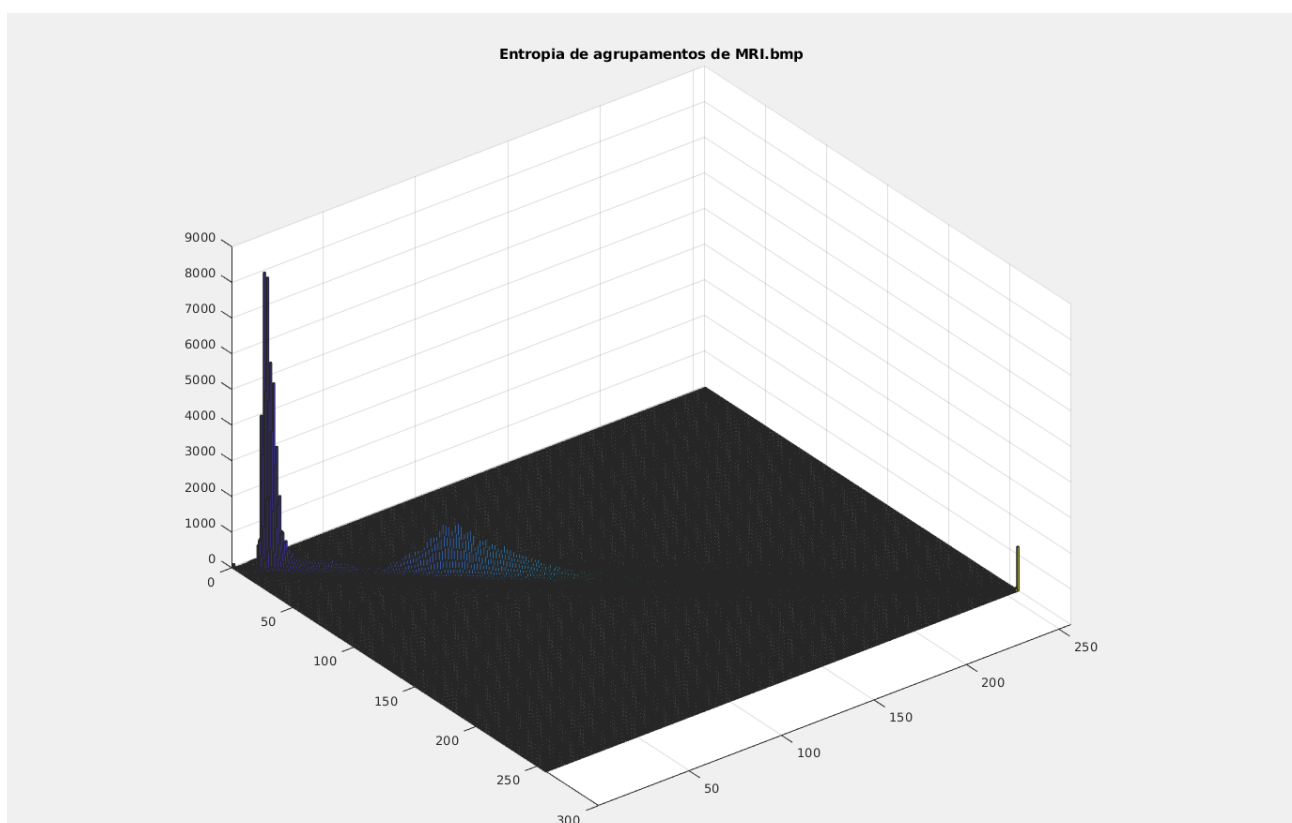
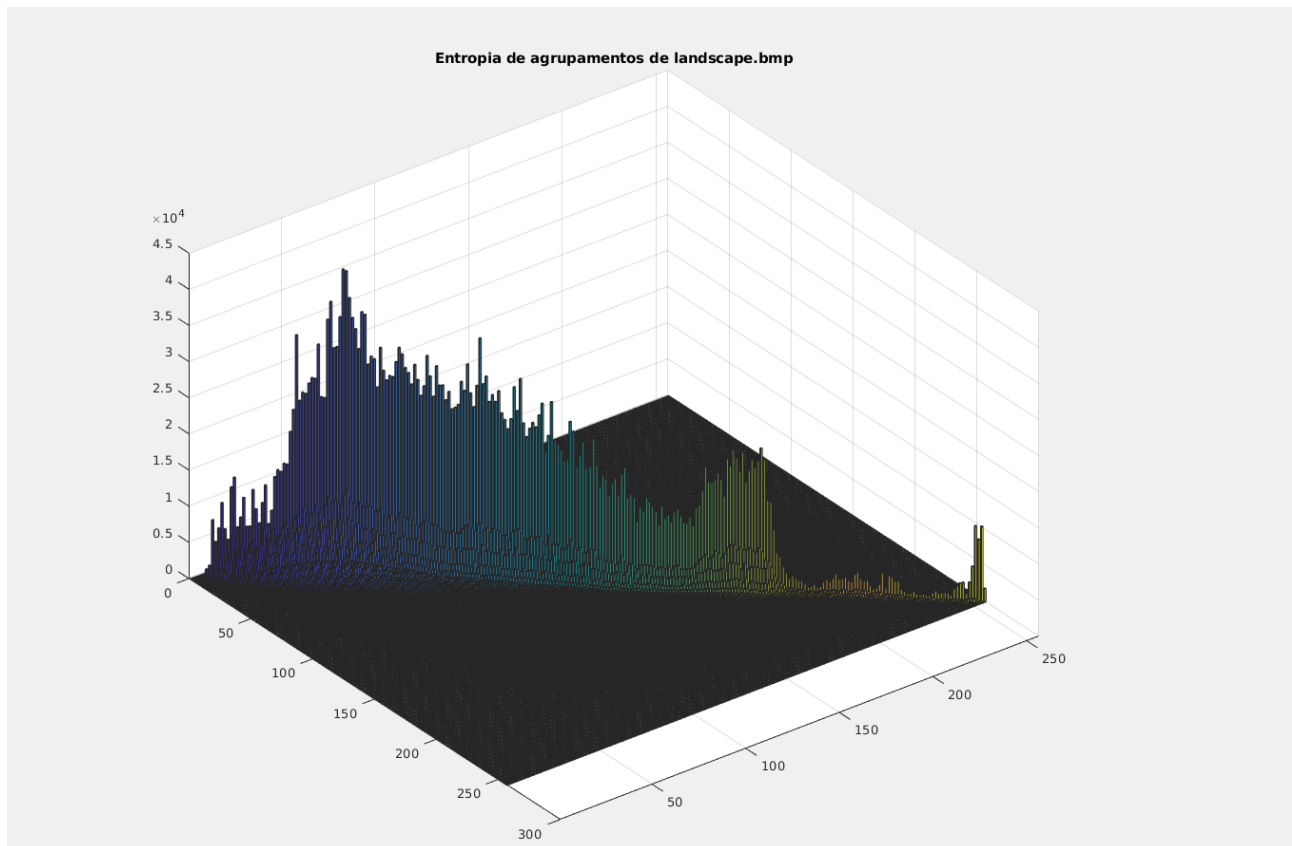
- Entropia da informação: 4.065729
- Entropia dos agrupamentos: 3.310996

### Fonte: `lyrics.txt`

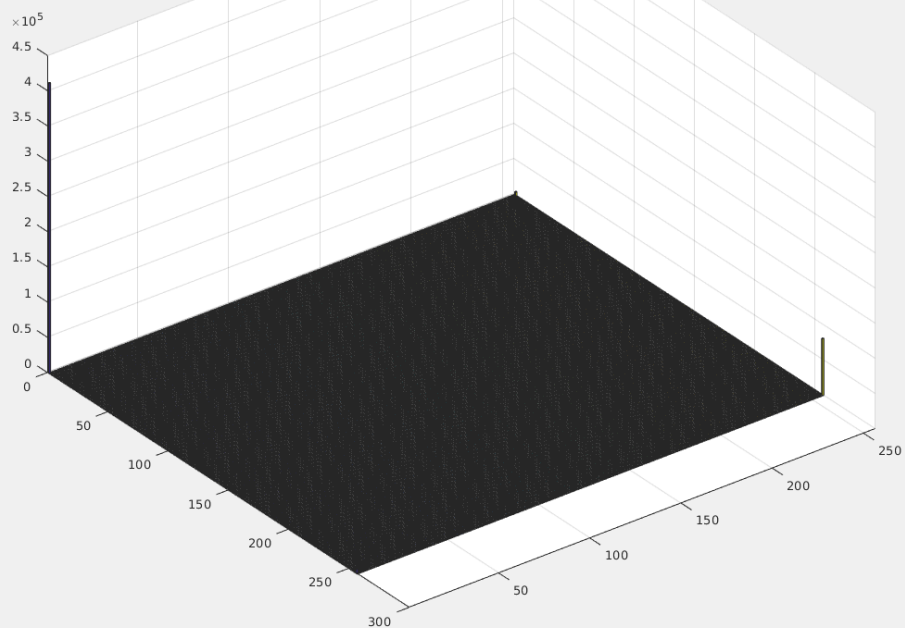
- Entropia da informação: 4.410705
- Entropia dos agrupamentos: 3.652180

## Análise dos resultados:

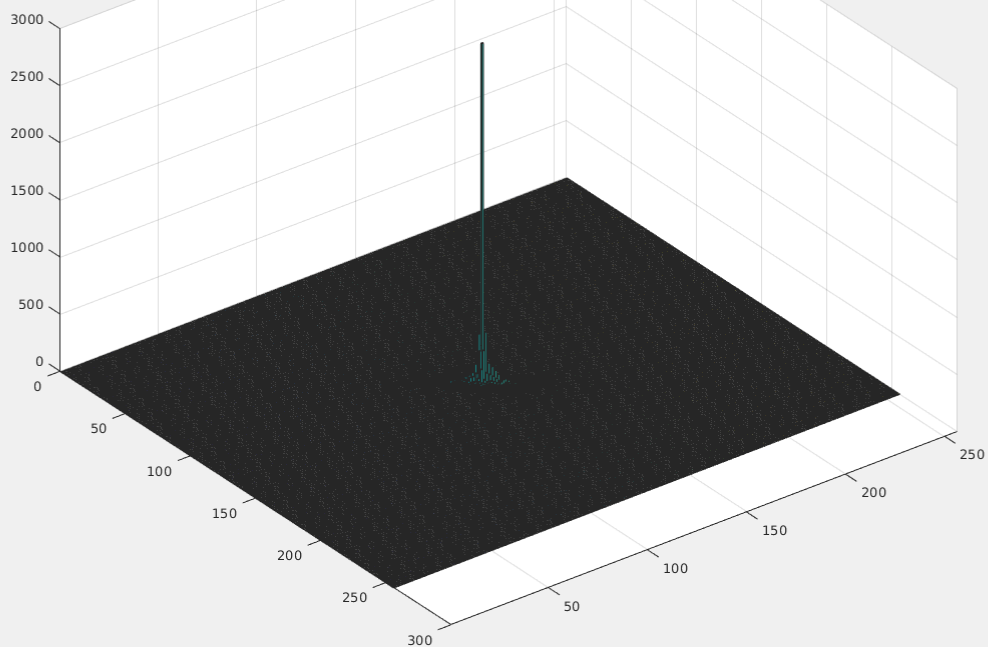
Através dos resultados obtidos, conclui-se que com o agrupamento dos símbolos a entropia diminui sempre, uma vez que são precisos menos bits para codificar cada símbolo.



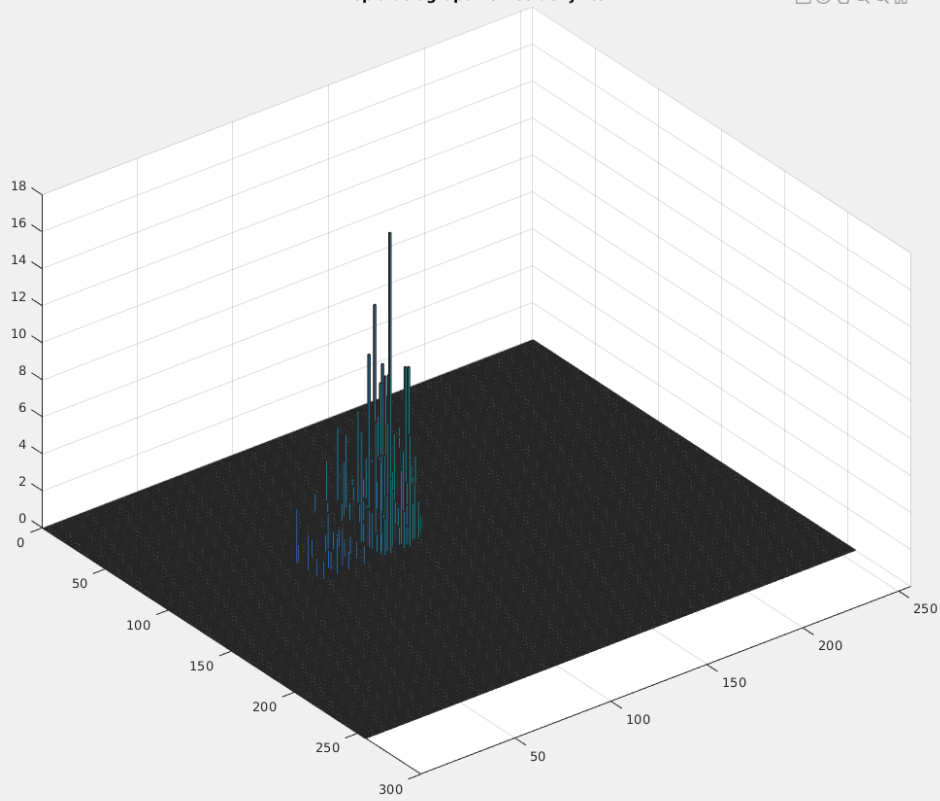
Entropia de agrupamentos de MRlibin.bmp



Entropia de agrupamentos de soundMono.wav



Entropia de agrupamentos de lyrics.txt



## Exercício 6 a) - introdução:

Aqui é pedida a matriz de valores de informação mútua em cada janela. Esta matriz é obtida a partir da rotina InformaçãoMutua.m, que recebe como parâmetros a matriz query, obtida a partir a imagem query, e a matriz target que será a imagem fonte, chamando depois a rotina entropia.m para calcular as respectivas entropias, faz o cast, mencionado no exercício 5, destas duas matrizes, para obter a matriz da informação conjunta. Aplicando por último a fórmula da informação mútua:

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

Devolvendo a matriz resultante.

## Exercício 6 b), bi), bii) - introdução:

Para obter os resultados aqui pedidos usámos as rotinas MatrizInfor.m, que tem como parâmetros a imagem query, a imagem fonte a ser percorrida em saltos definidos pelo step, terceiro parâmetro. Neste processo chama, em cada step, a rotina InformaçãoMutua.m, já explicada, e devolve uma matriz com os valores da informação mútua determinados.

De seguida essa matriz é passada como parâmetro à rotina maiorInformaçãoMutua.m que, tal como nome indica, determina o maior valor da matriz recebida, devolvendo-o. Mas devolve também as coordenadas do maior valor da matriz das informações mútuas.

Por último a rotina catRecognition.m recebe como parâmetros essas coordenadas, a imagem query, e as várias imagens que serviram de fonte. Não devolvendo nada, mas o que faz é desenhar um quadrado com dimensões iguais à da matriz query, usando as coordenadas recebidas para identificar na matriz target a zona onde o valor da informação mútua é maior, pois é aí que está a igualdade entre a query e o target.

## Resultados bi):

### Valor máximo da informação mútua para cada imagem fornecida:

- target\_original.bmp: 1.350032 Posição: (421,316)
- target\_noise.bmp: 1.196346 Posição: (421,316)
- target\_inverted.bmp: 1.350032 Posição: (421,316)
- target\_lightning\_contrast.bmp: 1.224019 Posição: (421,316)

## **Análise dos resultados:**

Analisando os resultados pode concluir-se que em todas as distorções foi possível encontrar a imagem query no target. Onde no target com ruído, o valor da informação mútua é menor, pois esse é seu efeito, altera com mais impacto a informação original. O target invertido tem exatamente o mesmo valor de informação mútua que o original, porque não há modificação degradante da informação. O target com contraste também tem um valor reduzido, uma vez que, ao se alterar o contraste está-se a mudar os valores dos símbolos, pois deixam de estar na mesma zona da escala de cinzentos.

Conclui-se então, que o efeito da informação mútua é permitir a deteção do maior número de símbolos com valores iguais ou muito próximos, de modo a encontrar as zonas mais semelhantes entre duas fontes de informação.

## **Exercício 6 c) - introdução:**

Nesta meta foram usadas as mesmas funções da alínea anterior.

## **Resultados c):**

### **Valor máximo da informação mútua**

- target1.bmp: 1.700786 Posição: (121,196)
- target2.bmp: 1.200681 Posição: (76,151)
- target3.bmp: 1.170559 Posição: (166,91)
- target4.bmp: 1.112329 Posição: (256,196)

## **Análise dos resultados:**

Como se pode ver as rotinas detetaram sempre a face do gato, sendo que a informação mútua tem um valor mais elevado para o target 1, pois este corresponde com a query. Nos outros casos as faces foram encontradas, mas têm valores de informação mútua menores, que isto dizer que as semelhanças detetadas são menos.