





# CODECS LOSSLESS DE IMAGEM

Feito por:

- Pedro Martins
- João Silva

- 
- 
- Existem diversos codecs lossless para a compressão de uma imagem. O mais conhecido é o PNG, contudo são vários os métodos desenvolvidos que atingem taxas de compressão semelhantes. Entre eles existe uma subdivisão entre os mais antigos, como o RLE (Run Length Encoding), Huffman Coding, o LZ\_77, LZ\_78, LZW; e existem os mais recentes, como por exemplo o CALIC ou o Deflate.
  - Uma vez que estes métodos são do tipo lossless, a imagem não irá perder qualidade aquando da compressão.
  - De seguida iremos explicar como funcionam alguns dos algoritmos referenciados acima.

# Run Length Encoding

- Compressão sequencial de uma imagem através da procura de repetição de símbolos.
- Abaixo está presente um exemplo de como este algoritmo funciona:

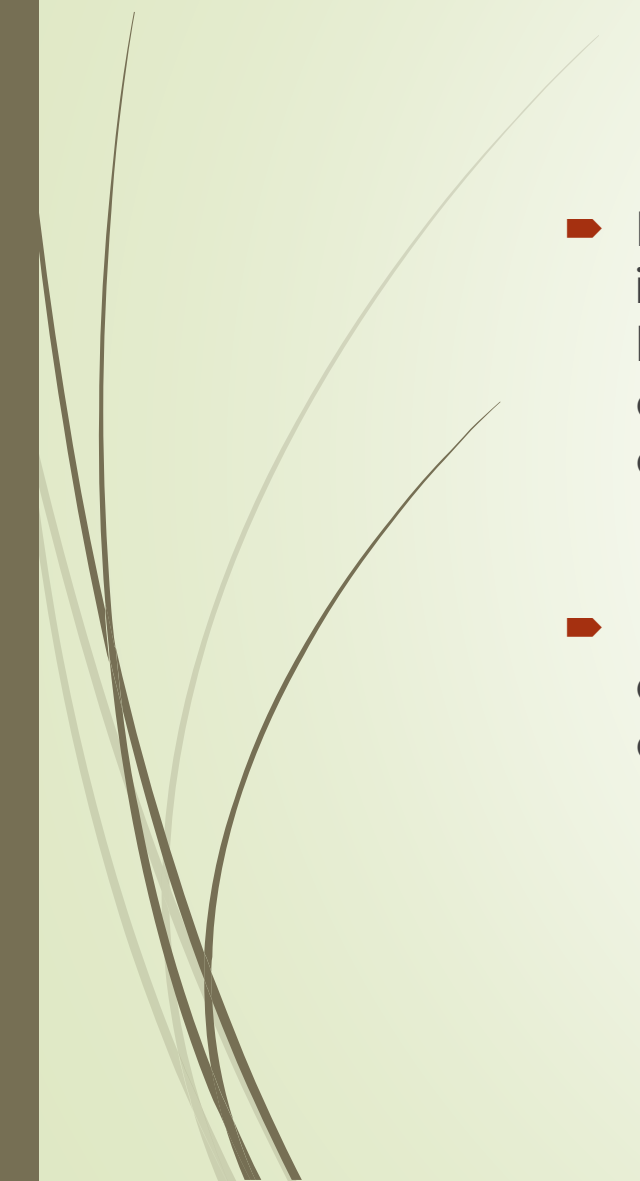
5	5	5	3	3	3	3	8	8	8
---	---	---	---	---	---	---	---	---	---



(5,3)	(3,4)	(8,3)
-------	-------	-------




# Run Length Encoding

- Este algoritmo é bastante favorável na compressão de imagens cujas intensidades dos pixéis são semelhantes. Por exemplo, uma imagem binária consegue uma alta taxa de compressão devido ao facto de variar apenas entre preto e branco. Também é um algoritmo relativamente fácil de executar.
  - Contudo pode apresentar resultados pouco satisfatórios quando utilizado em imagens com bastante variação de cores, uma vez que a repetição de símbolos é mais escassa.
- 




# LZ\_78

- Algoritmo para compressão de imagem que recorre ao uso de um dicionário para armazenar as diversas sequências de símbolos.
  - Este método de compressão é semelhante ao LZ\_77 e ao LZW, sendo mais utilizado e mais eficaz na compressão de texto, face à compressão de imagem.
- 




# Deflate

- Método de compressão lossless que recorre a uma combinação dos algoritmos LZ\_77 (semelhante ao LZ\_78) e à codificação de Huffman.
  - É um algoritmo rápido em velocidade de compressão contudo apresenta limitações. Imagens com poucas repetições irão atingir um baixo nível de compressão, uma vez que vai ser utilizado um número de bits para codificar a imagem semelhante à imagem original, atingindo uma taxa reduzida de compressão.
- 



# Exploração de métodos

- Numa tentativa de comparar qual a taxa de compressão que é possível, analisámos e implementámos diversos códigos na linguagem Python e, usando um dataset que contém 4 imagens no formato .bmp, analisámos a taxa de compressão obtida assim como a rapidez do processo.
- 

# Comparação dos métodos

	Original	RLE	Huffmam Encoding	Deflate
Egg.bmp	17 329 KB	25 622 KB	12 437 KB	6 455 KB
Landscape.bmp	10 749 KB	24 058 KB	10 011 KB	4 341 KB
Zebra.bmp	16 346 KB	27 735 KB	11 982 KB	7 435 KB
Pattern.bmp	46 881 KB	7 911 KB	11 294 KB	2 464 KB



# Taxa de compressão

- Taxa de compressão assenta num valor percentual que representa o quão comprimida uma imagem fica face ao tamanho original.
- Para isso vamos utilizar a seguinte fórmula:


$$TaxaDeCompressão(\%) = 100 - \frac{TamanhoComprimido}{TamanhoOriginal} \times 100$$

# Comparação das taxas de compressão

Original	RLE	Huffmam Encoding	Deflate
Egg.bmp	-47,9%	28,2%	62,8%
Landscape.bmp	-123,8%	6,9%	59,6%
Zebra.bmp	-69,7%	26,7%	54,5%
Pattern.bmp	83,1%	75,9%	94,7%

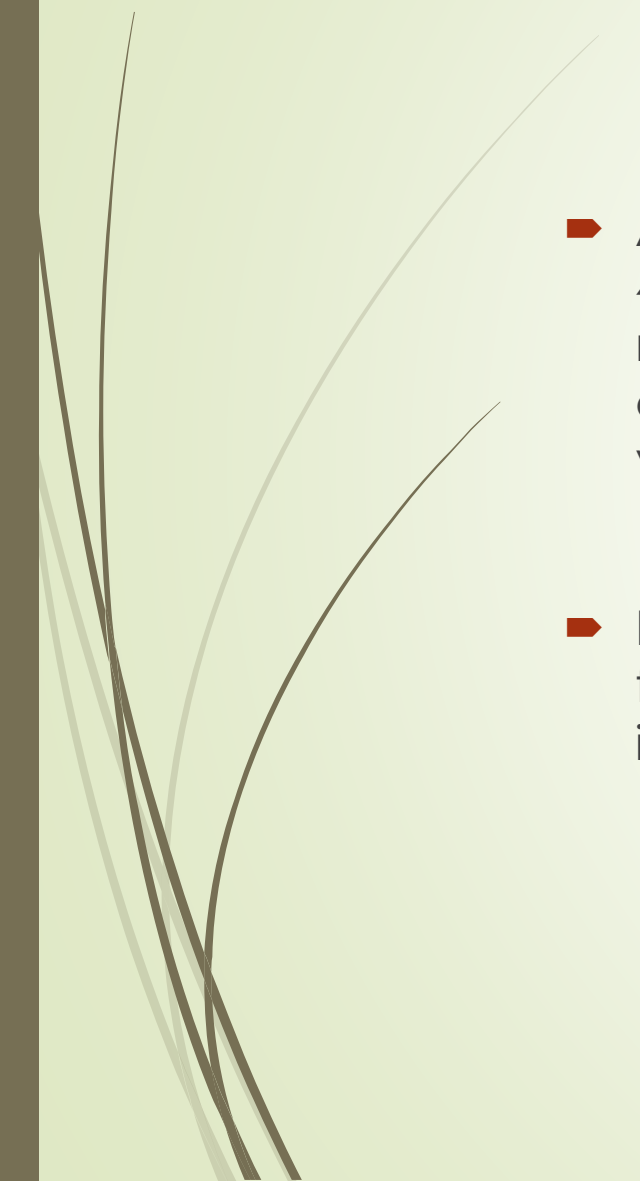


# Melhor taxa de compressão

- Com os resultados acima concluímos que o nosso algoritmo do Deflate, regra geral, é o que atinge melhores taxas de compressão. A média da taxa de compressão para as 4 imagens analisadas é de 67,9%, contra os 34,4% do Huffman Encoding e os -39,6% do RLE (Run Length Encoding).
  - Um dos principais motivos para o Deflate se distinguir dos outros métodos é o facto de utilizar os métodos do Huffman Encoding e do LZ\_77, o que lhe permite aumentar a taxa de compressão.
- 



# RLE e a baixa taxa de compressão

- Após analisar as tabelas anteriores, podemos concluir que o RLE em 3 das 4 imagens aumentou o tamanho do ficheiro final. Isto deve-se a dois motivos. O facto de que este método de compressão é mais eficaz na compressão de texto e o facto dessas 3 imagens terem uma considerável variação de intensidade de cor.
  - Na imagem pattern o resultado foi muito mais satisfatório. Isto deve-se ao facto de essa imagem ser binária, ou seja, apenas constam duas intensidades de cor, branco e preto.
- 



# Rapidez de compressão


- Consiste no tempo que demora a efetuar a compressão de um ficheiro.
- Este valor irá apresentar diferenças de método para método logo, quanto menor for o valor da rapidez de compressão, mais eficiente é a sua aplicação na compressão de imagens.

# Comparação da rapidez de compressão

	RLE	Huffman Encoding	Deflate
Egg.bmp	27,34s	9,34s	1,10s
Landscape.bmp	23,61s	7,09s	0,89s
Zebra.bmp	32,60s	7,55s	0,68s
Pattern.bmp	7,30s	15,62s	0,55s




# Comparação da rapidez de compressão

- O Deflate é o método com maior rapidez de compressão, sendo sempre consistentemente baixo.
  - O RLE mais uma vez é o algoritmo menos capaz, apresentando valores temporais até 50x maiores quando comparado ao Deflate.
- 





# Método de compressão PNG

- O PNG é um método de compressão altamente eficaz e reconhecido que reduz drasticamente o tamanho de uma imagem sem qualquer perda de qualidade, sendo assim um método lossless.
  - Este método utiliza um processo inicial chamado filtração, que troca os pixéis por comparações com os pixéis vizinhos, e que depois recorre ao Deflate para finalizar a compressão.
- 



# Comparação PNG com outros métodos


	Original	RLE	Huffmam Encoding	Deflate	PNG
Egg.bmp	17 329 KB	25 622 KB	12 437 KB	6 455 KB	4 516 KB
Landscape.b mp	10 749 KB	24 058 KB	10 011 KB	4 341 KB	3 246 KB
Zebra.bmp	16 346 KB	27 735 KB	11 982 KB	7 435 KB	5 335 KB
Pattern.bmp	46 881 KB	7 911 KB	11 294 KB	2 464 KB	2 222 KB

# Taxa de compressão PNG

Original	RLE	Huffmam Encoding	Deflate	PNG
Egg.bmp	-47,9%	28,2%	62,8%	73,9%
Landscape.bmp	-123,8%	6,9%	59,6%	69,8%
Zebra.bmp	-69,7%	26,7%	54,5%	67,4%
Pattern.bmp	83,1%	75,9%	94,7%	95,3%




# Taxa de compressão PNG

- O PNG atinge uma taxa média de compressão de 76,6% demonstrando ser altamente eficaz e o porquê de ser um dos métodos de compressão lossless mais utilizados. Face ao nosso Deflate, o que favorece o PNG é o facto do mesmo aplicar um processo de filtração que reduz a variação de intensidade dos pixéis e como tal aumenta a redundância estatística.
  - A segunda etapa deste método é semelhante ao nosso Deflate, uma vez que resulta de uma combinação de Huffman Encoding e LZ\_77.
- 



# Trabalho futuro

- Sugere-se alterações ao método PNG. Primeiramente ao nível da primeira etapa do processo, a filtração. Uma otimização deste passo poderia resultar em ganhos de compressão.
  - Também se equaciona uma possível alteração à segunda etapa do processo, o Deflate. Será possível alterar o LZ\_77 por o mais recente LZ\_78 e se sim, quais seriam os ganhos associados a esta alteração.
- 



# Conclusão

- Este relatório teve como objetivo o estudo aprofundado de vários codecs na compressão de imagens no formato .bmp.
  - Para isso, criámos um estado de arte com os vários métodos existentes e a partir daí estabelecemos comparações entre si e com o método PNG de forma a estudar qual o melhor tanto a nível de taxa de compressão como rapidez.
- 