Compressão de Entropia

Na última aula prática vimos duas versões de um padrão simples de representação de imagens, o chamado padrão `CUIF` Versões 1 (CUIF.1) e 2 (CUIF.2). Para realizar esta aula prática é necessário ter concluído a Prática III. Caso o seu grupo não concluíu, é necessário fazê-lo antes de realizar esta prática. Esta aula prática exigirá o cálculo do PSNR implementado na Prática III.

Nesta aula prática, veremos ao todo 4 versões do padrão `CUIF`:

- CUIF.1. Representa a imagem em um formato RGB;
- CUIF.2. Representa a imagem em um formato RGB com perdas;
- CUIF.3. Representa a imagem em um formato YCbCr.
- CUIF.4. Representa a imagem em um formato YCbCr e compactada usando usando codificação RLE.

O projeto Python da Prática IV contêm 3 arquivos:

- praticaIV.py: contendo métodos para cálculo do PSNR, e uso da classe Cuif para conversão de imagens BMP em CUIF e vice-versa.
- Cuif.py: classe implementando métodos para converter imagens BMP em CUIF e vice-versa
- ColorSpace.py: fornece algumas funções para conversão de espaço de cores
- RLE.py: contém funções para compactar e descompactar dados usando a codificação RLE.

Analise estes arquivos, em especial o arquivo praticaIV.py.

1 CUIF.1

Como visto na aula prática anterior, o padrão `CUI.1` é uma representação RGB separada em canais, de maneira similar ao BMP. Porém, diferente do BMP onde cada *pixel* aparece com seus canais BGR, o `CUI.1` apresenta cada canal R, G e B completos em sequência. Ou seja, ao invés de codificar *pixel*-a-*pixel*, codifica-se canal-a-canal. Cada *pixel* utilizará 1 byte em cada canal.

Exemplo de `CUI.1`, representado em um arquivo CUIF

Vamos supor uma imagem com 2×2 *pixels*:

red = (FF 00 00)$_{16}$ green = (00 FF 00)$_{16}$
blue = (00 00 FF)$_{16}$ gray = (B7 B7 B7)$_{16}$

Para este exemplo, há apenas um estudante no grupo, cuja matrícula é 99132042. Vejamos como fica a organização de um arquivo CUIF para armazenar essa imagem seguindo o padrão `CUI.1`:

byte	valor				significado
	3	2	1	0	
0	–	CUIF			assinatura CUIF
4	–	–	–	1	versão do padrão CUI (CUI.1)
5	–	–	–	1	número de estudantes no grupo
6	2_{10}				largura
10	2_{10}				altura
14	99132042_{10}				matrícula do aluno no grupo
18	–	–	–	FF_{16}	R pixel 0,0
19	–	–	–	00_{16}	R pixel 0,1
20	–	–	–	00_{16}	R pixel 1,0
21	–	–	–	$B7_{16}$	R pixel 1,1
22	–	–	–	00_{16}	G pixel 0,0
23	–	–	–	FF_{16}	G pixel 0,1
24	–	–	–	00_{16}	G pixel 1,0
25	–	–	–	$B7_{16}$	G pixel 1,1
26	–	–	–	00_{16}	B pixel 0,0
27	–	–	–	00_{16}	B pixel 0,1
28	–	–	–	FF_{16}	B pixel 1,0
29	–	–	–	$B7_{16}$	B pixel 1,1

1.1 Roteiro com a codificação CUIF.1

Faça os seguintes procedimentos inciais:

1. Abra o arquivo praticaIV.py, e insira seu código de cálculo do PSNR.
2. Também no arquivo praticaIV.py modifique a lista `matricula` para conter o número de matrículas dos alunos de seu grupo.
3. Execute o praticaIV.py para geração dos arquivos CUIF na versão 1 e a conversão do arquivo CUIF em BMP. Também será impresso o PSNR entre a imagem BMP original e a imagem BMP obtida pela decodificação do arquivo CUIF.1. O valor do PSNR deve ser infinito. Se não for, verifique sua implementação.
4. Alterando o código entregue, codifique a imagem bandeira.bmp usando CUIF.1 e converta este arquivo CUIF em BMP (gerando o bandeira1.bmp).
5. Também gere o PSNR deste arquivo bmp decodificado a partir do CUIF.1.

2 CUIF.2

O padrão `CUIF.2` foi objeto da Prática III. Trata-se de uma codificação RGB com perdas, e os alunos devem indicar como é feita a compressão (atividade da Prática III).

2.1 Roteiro com a codificação CUIF.2

Faça os seguintes procedimentos com o CUIF.2:

1. Usando o código entregue, codificar as imagens lenna.bmp e bandeira.bmp no formato CUIF.2, nomeando os arquivos para lenna2.cuif e bandeira2.cuif, respectivamente.
2. Em seguida, converta estes arquivos Cuif em bmp, nomeando estes arquivos em lenna2.bmp e bandeira2.bmp.
3. Também faça o código necessário para cálculo do PSNR do CUIF.2 para estas duas imagens.

3 CUIF.3

Nosso padrão `CUIF.3` é bastante similar ao `CUIF.1`. Porém, ao invés de representar os *pixels* em RGB, a representação será em YCbCr, de acordo com o formato JPEG File Interchange Format. O documento descrevendo as conversões de RGB para YCbCr e vice-versa está disponível em https://www.w3.org/Graphics/JPEG/jfif3.pdf.

Segue na sequência o formato do arquivo CUIF.3

Exemplo de `CUI.1`, representado em um arquivo CUIF

Vamos supor uma imagem com 2×2 *pixels*. Para este exemplo, há apenas um estudante no grupo, cuja matrícula é 99132042. Vejamos como fica a organização de um arquivo CUIF para armazenar essa imagem seguindo o padrão `CUI.3`:

byte	**valor**				**significado**
	3	2	1	0	
0	–	CUIF			assinatura CUIF
4	–	–	–	3	versão do padrão CUI (CUI.1)
5	–	–	–	1	número de estudantes no grupo
6	2_{10}				largura
10	2_{10}				altura
14	99132042_{10}				matrícula do aluno no grupo
16	–	–	–	FF_{16}	Y pixel 0,0
17	–	–	–	00_{16}	Y pixel 0,1
18	–	–	–	00_{16}	Y pixel 1,0
19	–	–	–	$B7_{16}$	Y pixel 1,1
20	–	–	–	00_{16}	Cb pixel 0,0
21	–	–	–	FF_{16}	Cb pixel 0,1
22	–	–	–	00_{16}	Cb pixel 1,0
23	–	–	–	$B7_{16}$	Cb pixel 1,1
24	–	–	–	00_{16}	Cr pixel 0,0
25	–	–	–	00_{16}	Cr pixel 0,1
26	–	–	–	FF_{16}	Cr pixel 1,0
27	–	–	–	$B7_{16}$	Cr pixel 1,1

3.1 Roteiro com a codificação CUIF.3

Faça os seguintes procedimentos com o CUIF.3:

1. Baixar https://www.w3.org/Graphics/JPEG/jfif3.pdf, e leia a seção *Conversion to and from RGB*, em seguida analise a implementação desta conversão no arquivo ColorSpace.py.
2. Usando o código entregue, codificar as imagens lenna.bmp e bandeira.bmp no formato CUIF.3, nomeando os arquivos para lenna3.cuif bandeira3.cuif.
3. Em seguida, converta estes arquivos Cuif em bmp, nomeando estes arquivos em lenna3.bmp e bandeira3.bmp.
4. Também faça o código necessário para cálculo do PSNR do CUIF.3 para estas duas imagens.

4 Cuif.4

Na implementação de RLE disponibilizada (no arquivo RLE.py), a compressão se baseia na supressão de repetição de símbolos de tamanho de 1 byte (8bits). Nesta implementação, o bit mais significativo de cada byte foi utilizado como flag para indicar se o byte representa um símbolo único (sem repetição) ou contém o número de repetições do próximo byte. Assim, caso o bit mais significativo for 0, o byte representa um símbolo único. Caso este bit seja 1, o restante dos bits indica o número de repetição, e o próximo byte é o símbolo que se repete.

Esta implementação suporta apenas entradas de até 127 símbolos, por utilizar 7 bits para codificar os símbolos (primeiro bit é usado como flag de repetição). Como as imagens consideradas representam valores de Y, Cb e CR com 1 byte (valores de 0 a 255), o valor do byte é dividido por 2 na codificação e multiplicado por 2 na decodificação. Na implementação, foi utilizado o deslocamento de um bit a direita na codificação e deslocamento de um bit a esquerda na decodificação. Assim, sempre que o byte representando um símbolo único for ímpar, haverá erro no bit menos significativo, que será zerado.

Para melhor compreender como funciona o RLE em nossa implementação, vamos analisar a mesma mensagem que abordamos anteriormente: $m = \{3, 3, 194, 5, 1, 1, 1, 2, 3, 5, 1, 4, 3, 3, 3, 3\}$. Esta, seria codificada em binário (1 byte por símbolo) como:

Tabela 1: Codificação binária de m considerando 8 bits/símbolo.

símbolo	binário
3	00000011
3	00000011
194	11000010
5	00000101
1	00000001
1	00000001
1	00000001
2	00000010
3	00000011
5	00000101
1	00000001
4	00000100
3	00000011
3	00000011
3	00000011
3	00000011

Já na codificação RLE proposta na implementação, a representação seria da forma a seguir:

Tabela 2: Codificação RLE conforme implementada para `CUI.4`

binário	significado
10000010	o próximo byte é repetido 2× (3, 3)
00000011	byte que é repetido: 3
01100001	este byte é único, representa 194 dividido por 2
00000010	este byte é único, representa 5 dividido por 2
10000011	o próximo byte é repetido 3× (1, 1, 1)
00000001	este byte é repetido: 1
00000001	este byte é único, representa 2 dividido por 2
00000001	este byte é único, representa 3 dividido por 2
00000010	este byte é único, representa 5 dividido por 2
00000000	este byte é único, representa 1 dividido por 2
00000010	este byte é único, representa 4 dividido por 2
10000100	o próximo byte é repetido 4× (3, 3, 3, 3)
00000011	byte que é repetido: 3

A mensagem decodificada seria: $m' = \{3, 3, 194, 4, 1, 1, 1, 2, 2, 4, 0, 4, 3, 3, 3, 3\}$. Note que há erros! Usando a Equação do MSE, apresentado na Prática III, podemos estimar a MSE entre m e m':

$$MSE(m, m') = \frac{1}{16} \times (1^2 + 1^2 + 1^2 + 1^2) = \frac{1}{16} \times 4 = \frac{1}{4}$$

Através da MSE, podemos obter a PSNR:

$$PSNR(m, m') = 10 \times log_{10}\left(\frac{(2^8 - 1)^2}{MSE(m, m')}\right) = 10 \times log_{10}\left(\frac{65025}{1/4}\right) = 10 \times log_{10}(260100) = 54,1514dB$$

Vimos então que, mesmo a RLE sendo um método de codificação de entropia que por definição não gera erros, por conta de decisões de implementação foram inseridos erros. Durante a execução do roteiro a seguir, busquem analisar as imagens codificadas e verifiquem se é possível identificar esses erros.

4.1 Roteiro com a codificação CUIF.4

Faça os seguintes procedimentos com o CUIF.4:

1. Usando o código entregue, codificar as imagens lenna.bmp e bandeira.bmp no formato CUIF.4, nomeando os arquivos para lenna4.cuif bandeira4.cuif.
2. Em seguida, converta estes arquivos Cuif em bmp, nomeando estes arquivos em lenna4.bmp e bandeira4.bmp.
3. Também faça o código necessário para cálculo do PSNR do CUIF.3 para estas duas imagens.

5 Relatório a ser entregue

Questão 1. Indique as taxas de compressão obtida pelas codificações CUIF.1, CUIF.2, CUIF.3 e CUIF.4 para as imagens lenna e bandeira (razão entre o arquivo original bmp e os arquivos cuif).

Questão 2. Qual codificação resultou na maior taxa de compressão? Justifique porque.

Questão 3. Explique porque a taxa de compressão para o CUIF.4 da imagem bandeira é bem maior para a imagem lenna.

Questão 4. Indique o PSNR medido nas imagens lenna1.bmp, lenna2.bmp, lenna3.bmp e lenna4.bmp quando comparadas com a imagem original lenna.bmp. Justifique os valores obtidos, explicando a fonte dos ruídos gerados em cada codificação.

Questão 5. Indique o PSNR medido na imagem bandeira2.bmp quando comparada com a imagem original bandeira.bmp. Justifique os valor obtido.