

Compressão de Imagens

Luiz Eduardo da Silva

Processamento de Imagens

Ciência da Computação

UNIFAL-MG

1 de agosto de 2022

Agenda

- 1 Princípios da Compressão de imagens
 - Introdução
 - Redundância
- 2 Modelos de Compressão
- 3 Características da Compressão
- 4 Métodos de compressão
 - Árvore de Huffman
 - Codificação LZW
- 5 Codificação JPEG
 - Codificador JPEG
 - Decodificador JPEG
 - Transformada DCT
 - Quantização dos coeficientes
 - Codificação R.L.E.
 - Tabelas
 - Exemplo
- 6 Bibliografia

Agenda

- 1 Princípios da Compressão de imagens
 - Introdução
 - Redundância
- 2 Modelos de Compressão
- 3 Características da Compressão
- 4 Métodos de compressão
- 5 Codificação JPEG
- 6 Bibliografia

Introdução

- Imagem digital gera uma enorme quantidade de dados.
- Compressão de imagem trata o problema de reduzir a quantidade de dados necessários para representar uma imagem digital.
- Teve início, antes do computador digital, com a transmissão de vídeo. Vem crescendo desde o surgimento do computador.

Exemplo - Filme de 2h em Standard Definition

$$30 \frac{\text{frames}}{s} \times (720 \times 480) \frac{\text{pixel}}{\text{frames}} \times 3 \frac{\text{bytes}}{\text{pixels}} = 31.104.000 \text{ bytes}/s$$

$$31.104.000 \frac{\text{bytes}}{s} \times (60^2) \frac{s}{h} \times 2h \cong 2,24 \times 10^{11} \text{ bytes} (224 \text{ GB})$$

Redundância

Redundância de dados é o tema central da compressão de imagens.

- Sejam n_1 e n_2 o espaço necessário para armazenamento do mesmo dado usando codificações diferentes. Seja C_R , a taxa de compressão dada por: $C_R = N_1/N_2$. A redundância (R_D) é dada por:

$$R_D = 1 - \frac{1}{C_R}$$

- Para compressão, três redundâncias básicas podem ser exploradas:
 - Redundância de codificação
 - Redundância de interpixel
 - Redundância psicovisual

Redundância de Codificação

Pixel	Qtd.	Cód. 1	Bits	Cód. 2	Bits
1	19	000	57	11	38
2	25	001	75	01	50
3	21	010	63	10	42
4	16	011	48	001	48
5	8	100	24	0001	32
6	6	101	18	00001	30
7	3	110	9	000001	18
8	2	111	6	000000	12
			300		270

$$C_R = 300/270 = 1.11 \text{ e então } R_D = 1 - \frac{1}{1.11} = 0.099$$

Redundância Interframe



Redundância Psico-visual

Explora a limitação da visão humana que é incapaz de processar toda informação visual que recebe (*Exemplo: imagem original que usa 8 bits por pixel, a mesma imagem representada com 4 bits por pixel, usando técnicas diferentes*)



Critério de Fidelidade

Pode existir perda na compressão da imagem. Sejam $f(x, y)$ e $f'(x, y)$, a imagem original e a imagem comprimida, respectivamente. O erro total é dado por:

$$e_T = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]$$

ou ainda, usando o *erro raiz da média quadrática*:

$$e_{MQ} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]^2}$$

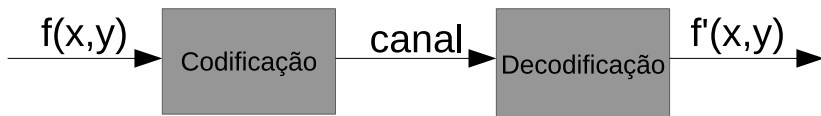
Apesar dessas medidas matemáticas, a melhor avaliação será sempre da percepção humana.

Agenda

- 1 Princípios da Compressão de imagens
- 2 Modelos de Compressão**
- 3 Características da Compressão
- 4 Métodos de compressão
- 5 Codificação JPEG
- 6 Bibliografia

Modelo genérico

- Processo usado para reduzir o tamanho físico da imagem, para otimizar transmissão e armazenamento.
- O processo de compressão está associado a duas fases: a codificação e a decodificação.
- A ideia básica é representar os elementos da imagem com maior frequência usando uma quantidade menor de bits.



Agenda

- 1 Princípios da Compressão de imagens
- 2 Modelos de Compressão
- 3 Características da Compressão**
- 4 Métodos de compressão
- 5 Codificação JPEG
- 6 Bibliografia

Com perda ou sem perda

- Na decodificação, a imagem obtida pode ser exatamente igual ou apresentar alguma alteração (perda).
- A codificação e decodificação podem ser realizadas com perda ou sem perda.
- A perda de informação pode comprometer a qualidade da imagem dificultando a interpretação do seu conteúdo.
- Há aplicações em que uma compressão com perda é intolerável. A compressão de dados de um cliente de banco, por exemplo.

Características

- Compressão sem perdas (lossless)
 - Reversível
 - Imagem reconstruída = imagem original.
 - Apresenta baixas taxas de compressão ($< 3 : 1$)
 - Aplicações: imagens médicas e de satélite.
- Compressão com perdas (lossy)
 - Irreversível
 - Imagem reconstruída = imagem original + ruído.
 - Taxas de compressão elevadas
 - Diversas aplicações: www, ...

Características

Adaptabilidade: Refere-se à característica do processo/ algoritmo de se adaptar aos dados de entrada

- Não adaptativos, em geral, baseiam-se em dicionários estáticos definidos a priori em função de estatísticas de frequência de dados.
 - Usam um dicionário de codificação/compressão que em essência associa os símbolos originais a códigos de menor tamanho.
- Adaptativos: não se baseiam em dicionários pré-definidos, construindo o dicionário em função dos dados de entrada.
- Semi-adaptativos: em geral, é efetuado em dois passos: no primeiro é efetuada uma estatística de entrada, gerando um dicionário; na segunda fase é efetuada a codificação/compressão propriamente dita.

Agenda

- 1 Princípios da Compressão de imagens
- 2 Modelos de Compressão
- 3 Características da Compressão
- 4 Métodos de compressão**
 - Árvore de Huffman
 - Codificação LZW
- 5 Codificação JPEG
- 6 Bibliografia

- O algoritmo de Huffman é um método para compactar um texto/imagem de forma ótima, sobre certos critérios.
- O texto/imagem em questão é composto de símbolos/pixels $S = s_1, s_2, \dots, s_n$, $n > 1$, dos quais se conhece a frequência f_i de ocorrência de cada símbolo/pixel no texto/imagem, ou seja, s_i , acontece f_i vezes dentro do texto/imagem para $1 \leq i \leq n$.

Algoritmo para construção da árvore de Huffman.

```
for (i = 0; i < n-1; i++) {  
    minimo (T', f, F);  
    minimo (T'', f, F);  
    T = T' @ T'';  
    f(T) = f(T') + f(T'');  
    inserir (T, f, F);  
}
```

Exemplo de construção da árvore de Huffman

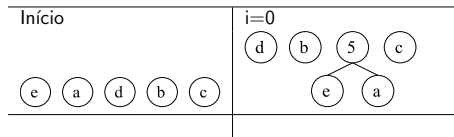
Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2

Início

(e) (a) (d) (b) (c)

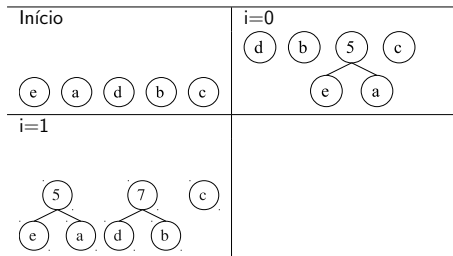
Exemplo de construção da árvore de Huffman

Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2



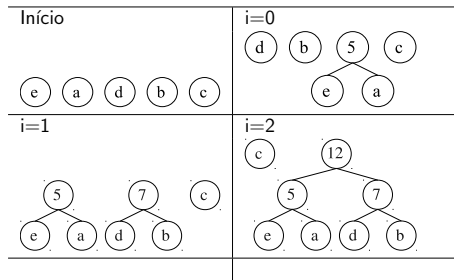
Exemplo de construção da árvore de Huffman

Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2



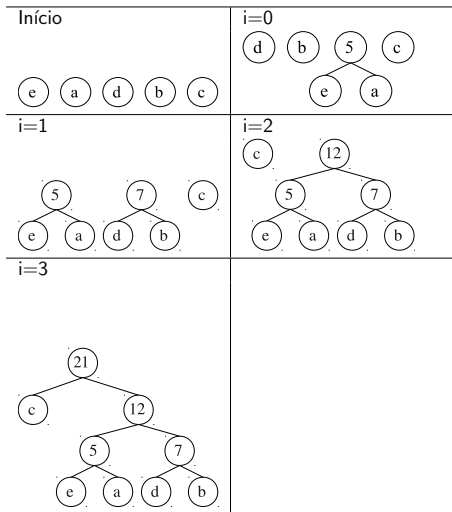
Exemplo de construção da árvore de Huffman

Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2



Exemplo de construção da árvore de Huffman

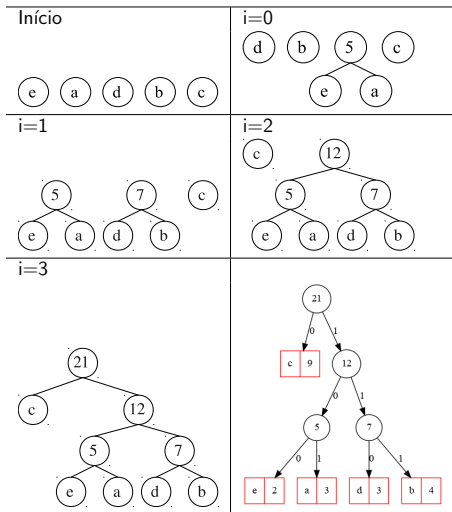
Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2



Exemplo de construção da árvore de Huffman

Símbolo/ Pixel	Frequência
a	3
b	4
c	9
d	3
e	2

Símbolo/ Pixel	Código Huffman
a	101
b	111
c	0
d	110
e	100



Características

- A técnica, chamada codificação de Lempel-Ziv-Welch (LZW), atribui palavras código de tamanho fixo a sequências de símbolos fonte de tamanho variável.
- Uma importante característica da codificação LZW é que ela não requer conhecimento antecipado da probabilidade de ocorrência dos símbolos que serão codificados
- LZW não requer nenhuma informação a priori dos dados de entrada
- LZW comprime e descomprime numa única passada
- Permite implementação simples e rápida execução

Compressão

```
1  inicia dicionario
2  corrente = primeiro símbolo da entrada
3  while (!fimDaEntrada) {
4      simbolo = próximo símbolo da entrada
5      if (corrente + simbolo está no dicionário)
6          corrente = corrente + simbolo
7      else {
8          escreve código corrente
9          adiciona corrente + simbolo no dicionário
10         corrente = simbolo
11     }
12 }
13 escreve código corrente (do último)
```

Sequência atualmente reconhecida	Pixel sendo processado	Saída codificada	Posição no dicionário (palavra código)	Entrada no dicionário
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Descompressão

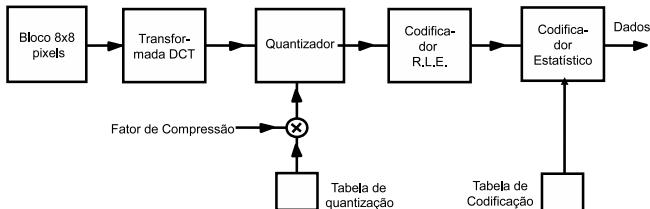
```
1  inicia dicionario
2  anterior = primeiro simbolo da entrada
3  escreve anterior
4  while (!fimDaEntrada) {
5      simbolo = proximo símbolo da entrada
6      cadeia = sequencia do anterior no dicionario
7      adiciona simbolo + cadeia[0] no dicionário
8      escreve sequencia do simbolo no dicionario
9      anterior = simbolo
10 }
```

Agenda

- 1 Princípios da Compressão de imagens
- 2 Modelos de Compressão
- 3 Características da Compressão
- 4 Métodos de compressão
- 5 **Codificação JPEG**
 - Codificador JPEG
 - Decodificador JPEG
 - Transformada DCT
 - Quantização dos coeficientes
 - Codificação R.L.E.
 - Tabelas
 - Exemplo

Características

- Padrão de Compressão de Imagens Estáticas
- Compressão Com Perdas (baseline)
- Taxa de Compressão Indeterminada A Priori
- Baseada em:
 - Transformada DCT
 - Quantização
 - Codificação R.L.E.
 - Codificação Entrópica (Huffman)

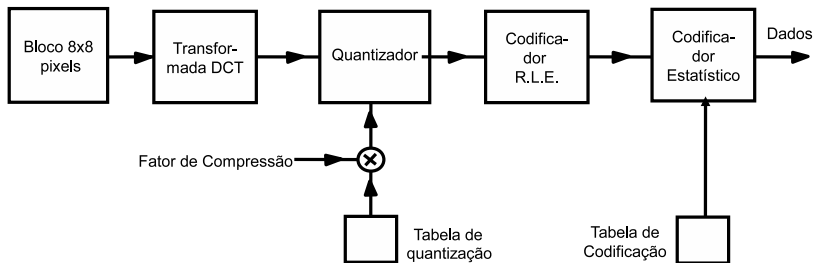


Desempenho

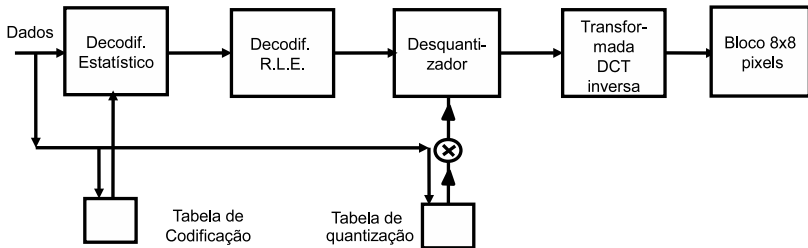
Desempenho típico da codificação JPEG

bits/pixel	Qualidade da Imagem Reconstruída
0,083	imagem reconhecível
0,25	imagem usável
0,75	imagem excelente
2,25	indistinguível da imagem original

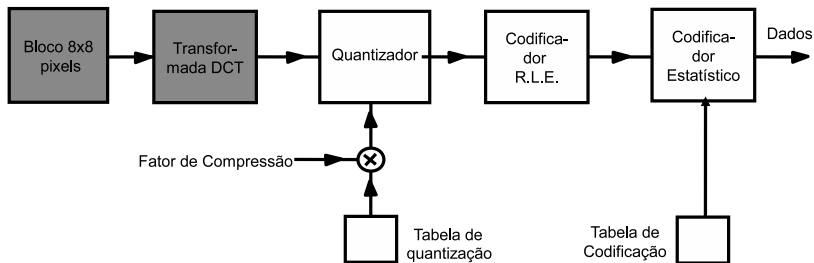
Esquema do codificador JPEG



Esquema do decodificador JPEG



Transformada do cosseno discreto



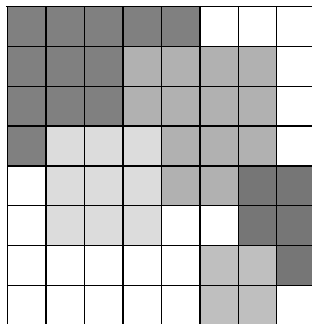
Equação

$$F(u, v) = \frac{C(u)}{2} \frac{C(v)}{2} \sum_{y=0}^7 \sum_{x=0}^7 f(x, y) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

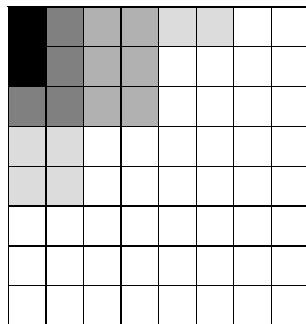
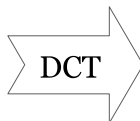
Onde:

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{se } k = 0 \\ 1 & \text{se } k > 0 \end{cases}$$

Transformada do cosseno discreto



$f(x,y) = 64$ elementos
de imagem (8x8 pixels)



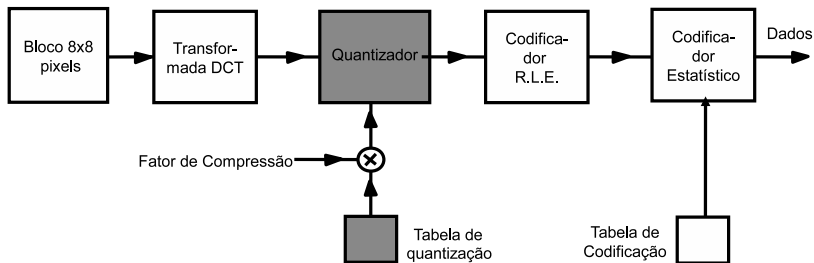
$F(u,v) = 64$ componentes
de frequências espaciais

Transformada do cosseno discreto

Objetivos do DCT:

- Descorrelacionar os elementos da imagem
- Representar a imagem em termos de componentes de frequência espacial para posterior processamento, de acordo com as características da visão humana
- Simplificar o processamento (valores reais)

Quantização



Quantização

A quantização é dada por:

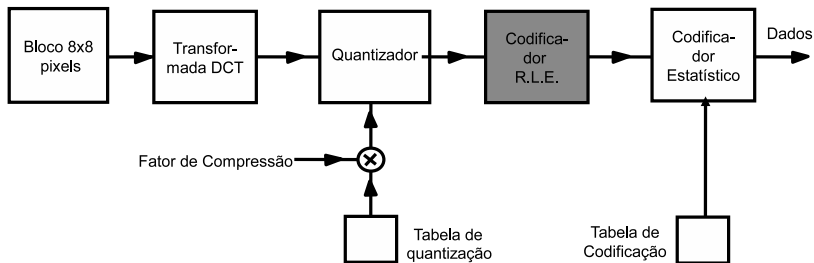
$$F_Q(u, v) = \text{round} \left[\frac{F(u, v)}{k \times Q(u, v)} \right]$$

onde k é o fator de compressão.

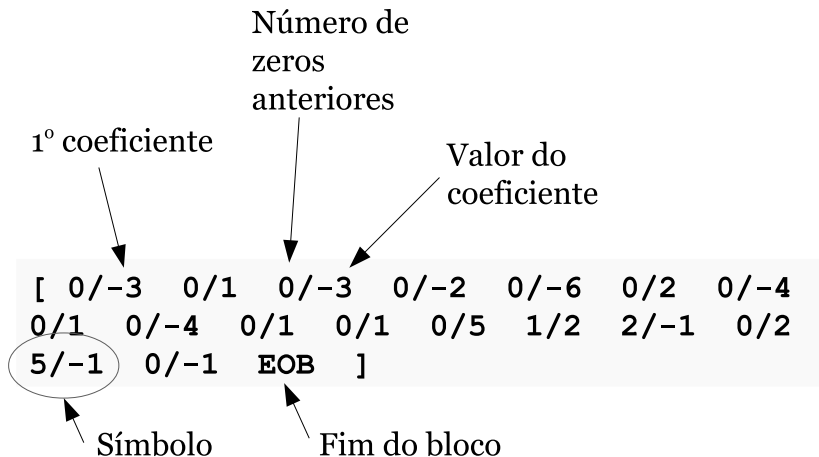
- O comitê deixou livre a escolha da matriz a ser utilizada, mas publicou juntamente com o padrão uma série de matrizes que foram classificadas de acordo com a taxa de compressão obtida após vários testes.
- Uma família de matrizes pode ser definida de acordo com um fator da seguinte forma:

$$Q(u, v) = 1 + [(i + i + j \times \text{fator})]; i, j = 1, 2, \dots, N - 1$$

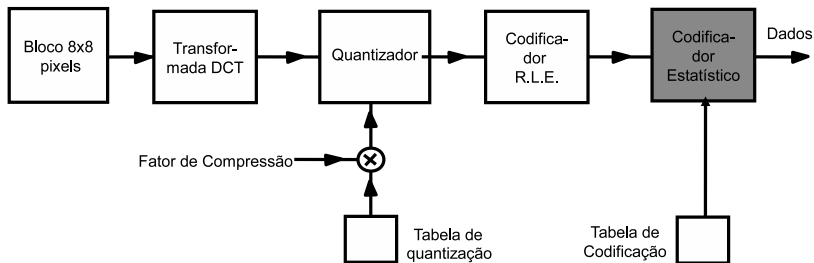
RLE = Run Length Encoding



RLE = Run Length Encoding



Codificação Estatística = Huffman



Codificação estatística

Características:

- Gera códigos de comprimento variável.
- Útil para aumentar a compressão.
- O código binário tem a forma **PREFIXO:MANTISSA**, onde o **PREFIXO** e **MANTISSA** tem tamanhos variáveis e são obtidos consultando tabelas de codificação de Huffman predefinidas.

Tabela de Categorias

Faixa de Valores	Categoria
0	0
-1; +1	1
-3, -2; +2, +3	2
-7 a -4; +4 a +7	3
-15 a -8; +8 a +15	4
-31 a -16; +16 a +31	5
-63 a -32; +32 a +63	6
-127 a -64; +64 a +127	7
-255 a -128; +128 a +255	8
-511 a -256; +256 a +511	9
-1023 a -512; +512 a +1023	A
-2047 a -1024; +1024 a +2047	B

Tabela de Prefixos DC

Categoria	Prefixo	Comprimento total (bits)	Comprimento Mantissa
0	010	3	0
1	011	4	1
2	100	5	2
3	00	5	3
4	101	7	4
5	110	8	5
6	1110	10	6
7	11110	12	7
8	111110	14	8
9	1111110	16	9
A	11111110	18	10
B	111111110	20	11

Tabela de prefixos AC padrão JPEG (luminância)

Zeros/ Categoria	Prefixo	Comprim. total
0/1	00	3
0/2	01	4
0/3	100	6
0/4	1011	8
0/5	11010	10
0/6	111000	12
0/7	1111000	14
0/8	111110110	18
0/9	111111110000010	25
0/A	1111111110000011	26
1/1	1100	5
1/2	111001	8
1/3	1111001	10

Zeros/ Categoria	Prefixo	Comprim. total
3/1	111010	7
3/2	111110111	11
3/3	11111110111	14
3/4	1111111110010000	20
...		
13/1	11111111010	12
13/2	1111111111100011	18
13/3	1111111111100100	19
...		
15/7	1111111111111011	23
15/8	1111111111111100	24
15/9	1111111111111101	25
15/A	1111111111111110	26

Exemplo da Codificação DC

Codificação DC

- Calcula-se e codifica-se a diferença do DC atual da subimagem (8x8) previamente processada.
 - Seja $DC_{atual} = -26$ e $DC_{antes} = -17$, então diferença = $-26 - (-17) = -9$.
 - Consultando a Tabela de Categorias, essa diferença está na categoria = 4. O prefixo é 101.
 - Como o comprimento total de bits é 7, a mantissa terá 4 bits. $-9 = 0110$, em binário.
 - O Coeficiente DC é portanto codificado como 101, concatenado com 0110 = 1010110.

Exemplo da Codificação AC

Codificação AC

- Os coeficientes AC não nulos do vetor reordenado devem ser codificados na forma PREFIXO:MANTISSA
- Com a informação do número de zeros antes do coeficiente e da categoria do coeficiente, busca-se na Tabela de Código AC, o código do prefixo.
 - Considere que o primeiro coeficiente não nulo do vetor ordenado seja -3 e que esse coeficiente não foi precedido de zeros.
 - A categoria desse coeficiente é 2, e o prefixo 01, conforme tabela.
 - Como o comprimento total de bits é 4, o comprimento da mantissa é 2 e o código para -3 = 00.
 - O coeficiente AC codificado é então 01 concatenado com 00 = 0100.

Exemplo com uma subimagem

Bloco 8x8 de uma imagem

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	8	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Exemplo com uma subimagem

Normalização

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-120	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

Exemplo com uma subimagem

DCT

-422	-34	-57	34	62	-21	-11	-6
0	-25	-58	16	16	-8	-13	0
-41	10	74	-30	-34	11	12	0
-39	20	29	-29	-19	9	13	14
18	-2	-16	-11	-8	3	6	9
-11	0	4	0	1	0	0	-3
-13	-8	7	12	11	-5	-12	-13
-9	-8	3	10	7	-3	-11	-10

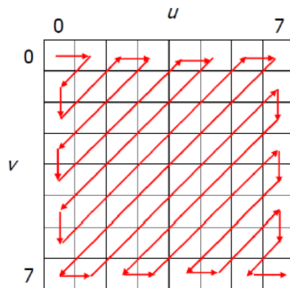
Exemplo com uma subimagem

Quantização

-26	-3	-5	2	2	0	0	0
0	-2	-4	0	0	0	0	0
-2	0	4	-1	0	0	0	0
-2	1	1	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Exemplo com uma subimagem

ZigZag



-26 -3 0 -2 -2 -5 2 -4 0 -2 1 1 4 0 2 0 0 -1 1 0 0 0 0 0 -1 0 0 0 0 0
 0

Exemplo com uma subimagem

Run Length Encoding

-26 -3 0 -2 -2 -5 2 -4 0 -2 1 1 4 0 2 0 0 -1 1 0 0 0 0 0 -1 0 0 0 0 0
0 0

RL [33]: -26 0 -3 1 -2 0 -2 0 -5 0 2 0 -4 1 -2 0 1 0 1 0 4 1 2 2 -1 0
1 5 -1 15 0 0 0

Exemplo com uma subimagem

Codificação Binária

RL [33]: -26 0 -3 1 -2 0 -2 0 -5 0 2 0 -4 1 -2 0 1 0 1 0 4 1 2 2 -1 0
1 5 -1 15 0 0 0

DC(-26 11000101) , 0 -3 0100 , 1 -2 11100101 , 0 -2 0101 , 0 -5
100010 , 0 2 0110 , 0 -4 100011 , 1 -2 11100101 , 0 1 001 , 0 1
001 , 0 4 100100 , 1 2 11100110 , 2 -1 110110 , 0 1 001 , 5 -1
11110100 , 15 0 111111110111 , 0 0 1010

11000101 0100 11100101 0101 100010 0110 100011 11100101 001
001 100100 11100110 110110 001 11110100 111111110111 1010

$8 \times 8 \times 8 = 512$ bits originais, 101 bits compactados.

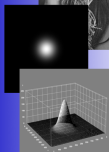
Agenda

- 1 Princípios da Compressão de imagens
- 2 Modelos de Compressão
- 3 Características da Compressão
- 4 Métodos de compressão
- 5 Codificação JPEG
- 6 Bibliografia**

Próxima aula...

Livro do Gonzalez

- Algumas definições básicas: Capítulo 9, Seção 9.1, Página 415
- Erosão e Dilatação: Capítulo 9, Seção 9.2, Página 417
- Abertura e Fechamento: Capítulo 9, Seção 9.3, Página 420
- A transformada hit-or-miss: Capítulo 9, Seção 9.4, Página 423
- Alguns algoritmos morfológicos básicos : Capítulo 9, Seção 9.5, Página 424



Compressão de Imagens

Luiz Eduardo da Silva

Processamento de Imagens

Ciência da Computação

UNIFAL-MG

1 de agosto de 2022