Compressão de Imagens

Fotografia Computacional - Lux.AI

INSTITUIÇÃO EXECUTORA

Centro de Informática UFPE

FADE UFPE

COORDENADORA

MCTI FUTURO

Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Tópicos

- Compressão e a Teoria da Informação;
- Redundância de informação
- Técnicas de compressão;
- Redundância de codificação;
- Código de Huffman;
- Compressão de imagem com perdas: JPEG;
- Compressão JPEG2000.

Motivação

- Armazenamento e transmissão de imagens: fotos, desenhos, gráficos, livros, vídeos etc.
- Câmera digitais
- Compressão de vídeo (de 4GB para 400MB)
- Vídeo sob demanda
  - youtube.com

Problema: como reduzir o espaço ocupado pela imagem?

257 KB

33 KB

Compressão de Imagens Digitais

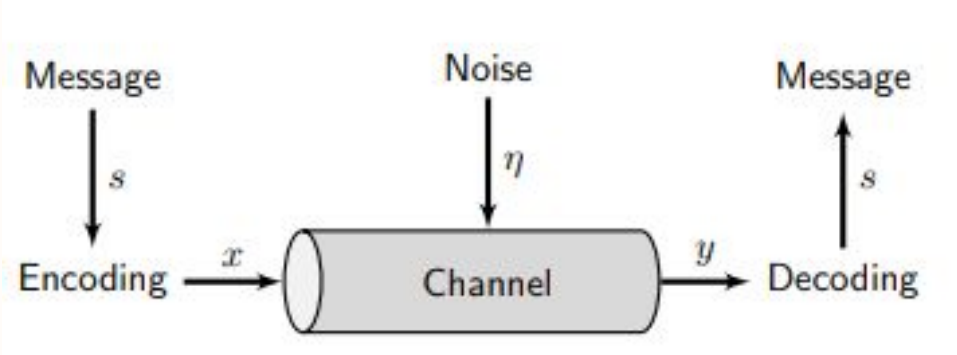
Aspectos relevantes:

- **Reduzir a quantidade de dados para representar uma imagem digital;**
- **Remover redundâncias;**
- **Transformar um array 2D de pixels em um conjunto de dados estatisticamente não correlacionados;**
- **Utilizar menor espaço de armazenamento e menor largura de banda ou tempo para transmissão;**
- **Reconstruir a imagem original ou uma aproximação dela;**

Fundamentos

- **Teoria da Informação**: "Uma teoria matemática de comunicação", 1948 - Claude Shannon (1916-2001)

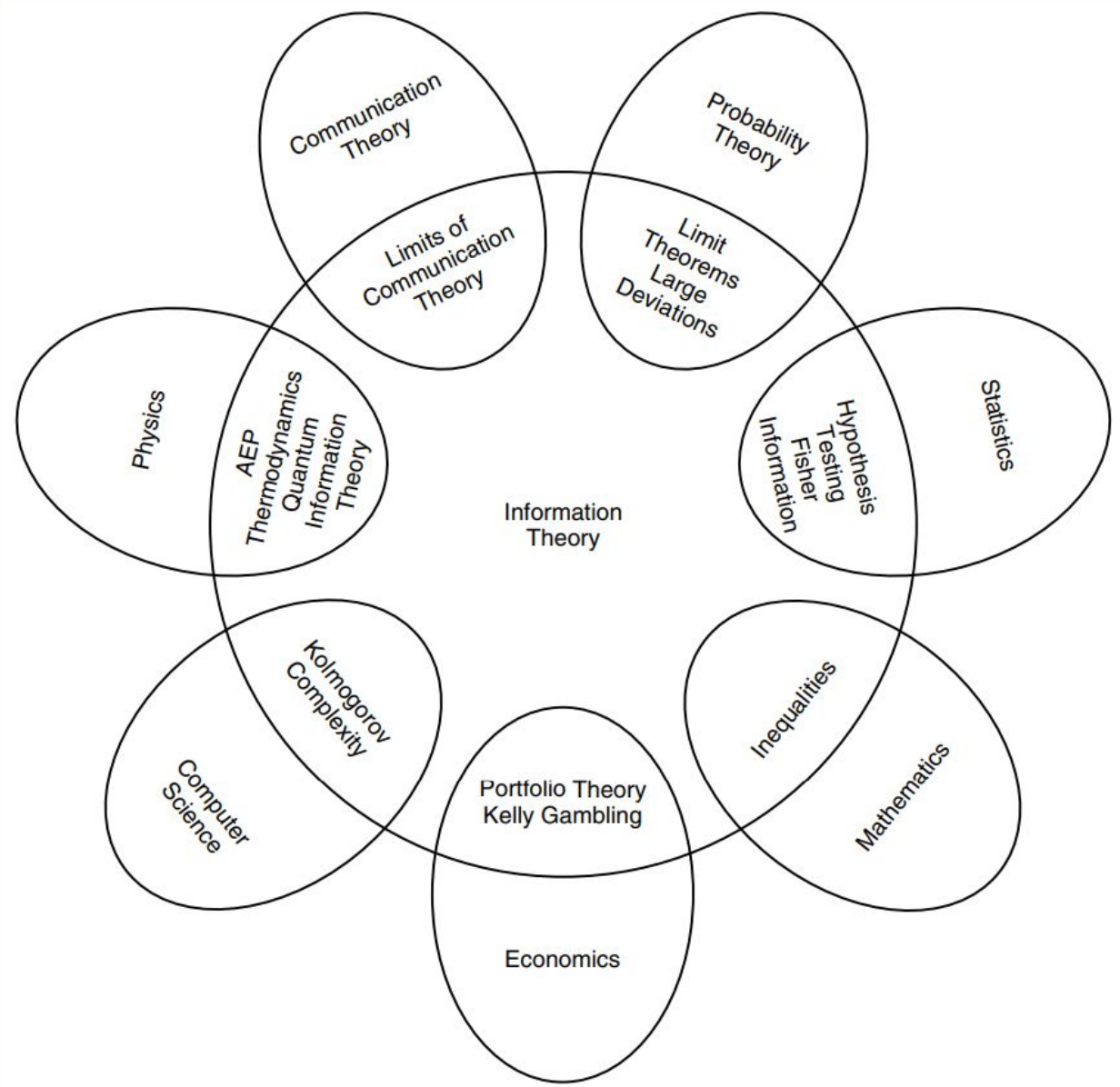
  Segundo Shannon: "*Uma ideia básica na teoria da informação é que a informação pode ser tratada de forma muito semelhante a uma quantidade física, como massa ou energia*".
- **Canal**: meio por onde a **informação** é transportada;
- **Dados**: meio de representação da informação;
- **Compressão de dados**: reduzir a quantidade de dados para representar a mesma informação



https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

Fundamentos

Relação da teoria da informação com outros campos: Cîencia da Computação, Física, Economia, Estatística, Matemática, etc.



Fonte:
Elements of Information Theory, Thomas Cover

Fundamentos

- Várias quantidades de dados podem ser usadas para transmitir a mesma informação. Exemplo: contadores de histórias P e C
  - P : é **prolixo**, demora muito e usa muitas palavras
  - C : é **conciso**, vai direto ao ponto e usa poucas palavras
  - P e C contam a mesma história: transmitem a mesma informação
  - P utiliza muitas palavras, muitos dados
  - C utiliza poucas palavras, poucos dados
  - P é mais **redundante**, repete mais

Fundamentos

- Qualquer conteúdo de uma mensagem pode ser comprimido?
  - Suponha que qualquer mensagem possa ser comprimida, resultando em uma mensagem menor;
  - Portanto, a saída de uma mensagem comprimida também pode sempre ser comprimida;
  - Se isso for verdade, qualquer mensagem pode ser comprimida infinitas vezes;
  - Segue-se então que qualquer quantidade de informação pode ser comprimida em um único bit;
  - Isso é claramente absurdo, então devemos rejeitar a premissa.

Redundância dos Dados

- Problema central na compressão de imagens
- **C_R: taxa de compressão**
- **R_D: redundância relativa dos dados**
- **n_1 e n_2 são as quantidades de dados** usadas em dois conjuntos de dados que transmitem a mesma informação onde n_1 é a quantidade de dados antes da compressão e n_2 após a compressão.
- $n_1 = n_2$, $C_R = 1$ e $R_D = 0$
- $n_1 >> n_2$, $C_R >> 1$ e $R_D >> 0$
- $n_1 << n_2$, $C_R \sim 0$ e $R_D << 0$
- **Exemplo: C_R = 10:1 e R_D = 90%**

$$C_R = \frac{n_1}{n_2} \qquad R_D = 1 - \frac{1}{C_R}$$

Redundância em Imagens

- Três tipos principais de redundância:
  - Redundância de **codificação**
  - Redundância **interpixel**
  - Redundância **psicovisual**
- **O objetivo da compressão é reduzir ou eliminar um ou mais desses tipos de redundância**

Redundância de Codificação

- Na redundância de código, o processo de codificação atribui código com tamanho variável (número de bits) de acordo com a probabilidade de ocorrência de determinado tom de cinza ou cor do pixel na cena;

- Como exemplo, os níveis de cinza ou de cor com maior frequência de ocorrência serão representados por um código com comprimento menor;

- De modo contrário, se um nível de cinza ou cor tem pouca presença na cena é representado por um código maior.

Símbolo	Frequência	Binário	Huffman
A	0,5	000	0
B	0,2	001	10
C	0,2	010	111
D	0,05	011	1100
E	0,05	100	1101
		3 bits/sb	1,9 bits/sb

Redundância Interpixel

- **Permite realizar previsões sobre o valor de um pixel a partir dos valores de seus pixels vizinhos;**

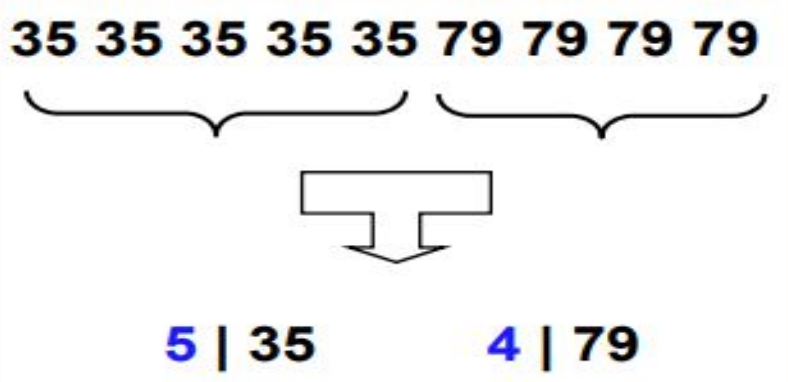
- Em algumas imagens há padrões de pixels que se repetem, o que significa que um pixel introduz pouca informação relativamente aos seus vizinhos;

- A redundância interpixel é normalmente removida, por exemplo, utilizando as diferenças entre pixels adjacentes para representar uma imagem;

Redundância Interpixel

- As transformações que removem redundância interpixel são chamadas **mapeamentos** e são consideradas **reversíveis** se os elementos da imagem original puderem ser **reconstruídos** a partir do conjunto de dados transformado.

RLE (Run-Length Encoding):



Redundância Psicovisual

- A redundância psicovisual está **relacionada à informação visual real ou quantificada em uma cena**; a redução ou a eliminação da redundância psicovisual, leva necessariamente a um processamento com perdas;

- **Algumas informações em imagens têm menos importância relativa do que outras no processamento visual normal**. Estas informações são ditas psicovisualmente redundantes e podem ser eliminadas sem prejudicar a qualidade de percepção da imagem;

Redundância Psicovisual

- Esta redundância é fundamentalmente diferente das redundâncias anteriores, pois está associada com a informação visual quantificável ou real.

Silhueta leve

Técnicas de compressão

- **Sem perda**
  - Mantém toda a informação contida no sinal original
  - Técnicas de compressão sem perdas são utilizadas em compressão de arquivos, através de formatos comprimidos como ZIP, ARJ ou GZ, e também em dispositivos de comunicação de dados, como os modems

- **Com perda**
  - É usada nos casos em que a perda de alguma informação é tolerável
  - Algumas perdas não são percebidas pela visão e audição humanas

Redundância de Codificação

- **Código**: sistema de símbolos (letras, números, bits etc.) usado para representar uma informação
- **Palavra**: seqüência de códigos

- Suponha uma imagem com **L** tons de cinza
- Cada tom de cinza r_k está no intervalo [0, 1], k = 0, 1, 2, … , L-1
- A probabilidade de ocorrência de cada r_k é $p(r_k)$
- O número de bits usados para representar r_k é $l(r_k)$
- L_{avg} é o número médio de bits por pixels

$$p_r(r_k) = \frac{n_k}{n}$$

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

Redundância de Codificação: exemplo

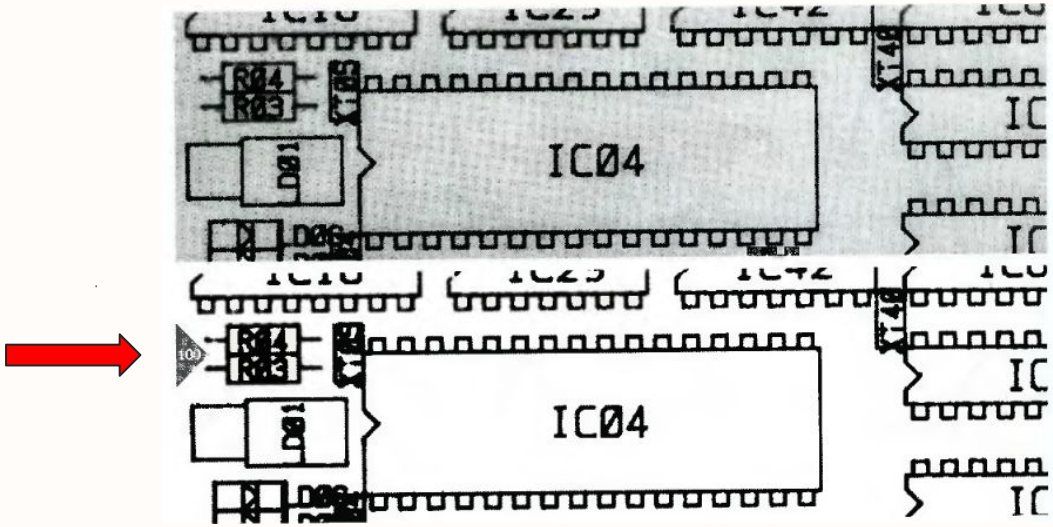
- L = 8
- Code 1: Lavg = 3.0 bpp (bits por pixel)
- Code 2: Lavg = 2.7 bpp

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

$$L_{\text{avg}} = \sum_{k=0}^{7} l_2(r_k)p_r(r_k)$$
$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08)$$
$$+ 5(0.06) + 6(0.03) + 6(0.02)$$
$$= 2.7 \text{ bits.}$$

Redundância Interpixel: exemplo

- Comprimindo a versão binarizada da imagem
- Passo 1: binarização (threshold)
- Passo 2: contar as repetições dos vizinhos na mesma linha (RLE)



Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

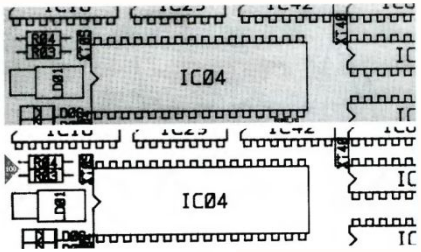
Redundância Interpixel: exemplo

Resultado da binarização e RLE:

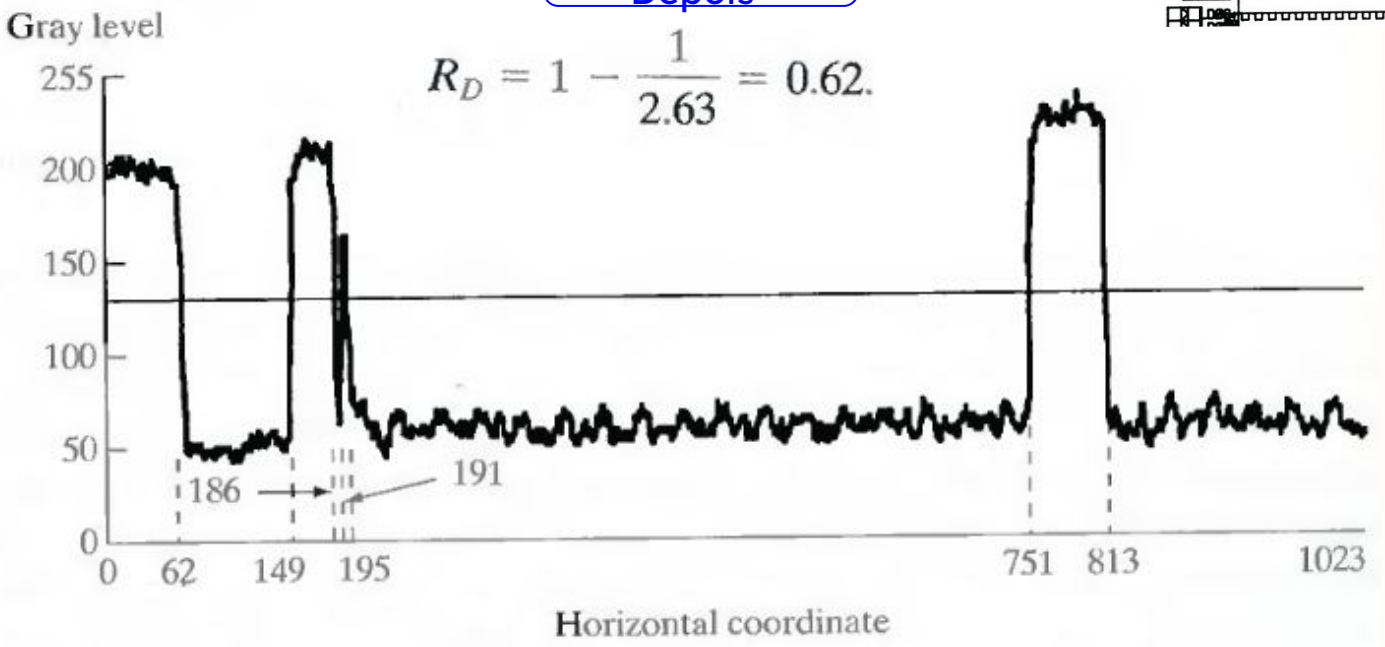
$$C_R = \frac{(1024)(343)(1)}{(12166)(11)} = 2.63$$

Antes

Depois



$$R_D = 1 - \frac{1}{2.63} = 0.62.$$



Line 100: (1, 63) (0, 87) (1, 37) (0, 5) (1, 4) (0, 556) (1, 62) (0, 210)

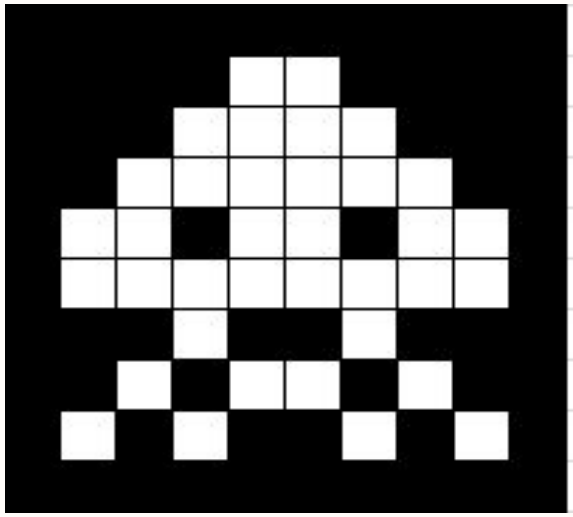
Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Método de Redução de Redundância Interpixel

RLE - código de comprimento de corrida (*run-length encoding*):

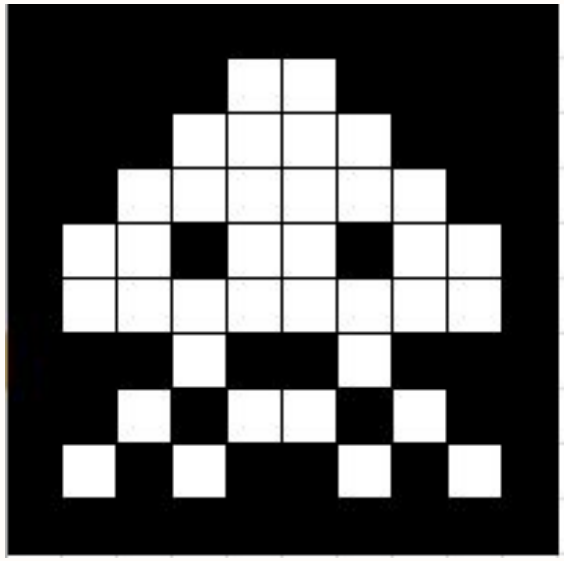
- Conta quantas vezes se repete determinado valor;
- (*run count, run value*)
- 555555444444000001101112223355444
- (6, 5)(7, 4)(5, 0)(2, 1)(1, 0)(3, 1)(3, 2)(2, 3)(2, 5)(3, 4)
- Poder ser aplicada outra codificação posteriormente (como o código Huffman)

Redundância Interpixel: exemplo (RLE) sobre imagem



1111111111
1111001111
1110000111
1100000011
1001001001
1000000001
1110110111
1101001011
1010110101
1111111111

$$C_R = \frac{n_1}{n_2}$$



101
412041
314031
216021
11201120112011
118011
3110211031
21101120111021
1110111021110111011
101

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

Fonte: https://mrdorancomputing.com/data-representation/rle/

Redundância Psicovisual

- A redundância psicovisual está **relacionada à informação visual real ou quantificada em uma cena**; a redução ou a eliminação da redundância psicovisual, leva necessariamente a um processamento com perdas;

- **Algumas informações em imagens têm menos importância relativa do que outras no processamento visual normal**. Estas informações são ditas psicovisualmente redundantes e podem ser eliminadas sem prejudicar a qualidade de percepção da imagem;

Redundância Psicovisual

- Redundância psicovisual é a informação que pode ser eliminada com pouco ou sem impacto para a percepção
- Quantização
    - Eliminação de algumas cores
    - Alto impacto na imagem
    - *error diffusion dithering*

Fonte: http://www.leptonica.org/color-quantization.html

Métodos de Redução de Redundância de Codificação

- **Huffman**
- Truncated Huffman
- B-Code
- Binary Shift
- Huffman Shift
- Arithmetic Coding

Algoritmos para compressão de dados:
https://paperswithcode.com/task/data-compression

Métodos de Redundância de Codificação

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	100000	C10C00	111 101	001110
a_{14}	0.03	01101	01111	100001	C10C01	111 110	001111
Block 3							
a_{15}	0.03	01110	01100	100010	C10C10	111 111 000	000010
a_{16}	0.02	01111	010000	100011	C10C11	111 111 001	000011
a_{17}	0.02	10000	010001	100100	C11C00	111 111 010	0000110
a_{18}	0.02	10001	001010	100101	C11C01	111 111 011	0000100
a_{19}	0.02	10010	001011	100110	C11C10	111 111 100	0000101
a_{20}	0.02	10011	011010	100111	C11C11	111 111 101	00001110
a_{21}	0.01	10100	011011	101000	C00C00C00	111 111 110	00001111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Código de Huffman

- Características gerais:
  - Codificação de **comprimento variável**, com códigos mais frequentes usando menos bits e códigos menos frequentes usando mais bits;
  - Codificação feita construindo uma **árvore de codificação**;
  - A árvore é construída de baixo para cima, com base nas frequências dos símbolos.

Código de Huffman

- A codificação de Huffman é **um algoritmo de compressão de dados sem perda**. A ideia é atribuir códigos de comprimento variável aos símbolos de entrada, sendo que os comprimentos dos códigos atribuídos são baseados nas frequências dos símbolos correspondentes;

- Os **códigos de comprimento variável** atribuídos aos caracteres de entrada são chamados de **códigos prefixo**, o que significa que os códigos (sequências de bits) são atribuídos de tal maneira que **o código atribuído a um símbolo não é o prefixo do código atribuído a qualquer outro símbolo**, evitando ambiguidades.

Código de Huffman

Contraexemplo de códigos prefixo:

- supor que haja quatro caracteres: **a, b, c e d**, e seus códigos de comprimento variável correspondentes sejam **00**, **01**, **0** e **1**;
- Essa codificação leva a ambiguidade porque o código atribuído a **c** é o prefixo dos códigos atribuídos a **a e b**;
- Se o fluxo de bits comprimido for **0001**, a saída descompactada pode ser "**cccd**", "**ccb**", "**acd**" ou "**ab**".

Código de Huffman: Vantagens

- **Eficiência de Compressão:** atribuindo códigos menores aos símbolos que aparecem mais frequentemente, resultando em uma alta taxa de compressão;

- **Esquema de Codificação de Prefixo**: método que simplifica a implementação e decodificação;

- **Amplamente Utilizado:** largamente adotado em compressão de dados, com suporte em muitas bibliotecas e ferramentas de software, facilitando a integração em sistemas existentes;

- **Compressão sem Perda**: permite que os dados originais sejam reconstituídos exatamente a partir dos dados comprimidos.

Código de Huffman: Desvantagens

- **Conhecimento Prévio das Frequências**: requer o conhecimento prévio das frequências de cada símbolo, o que a torna menos adequada para situações em que a distribuição de símbolos é desconhecida ou muda dinamicamente;

- **Complexidade das Árvores de Huffman**: As árvores de Huffman podem se tornar complexas e exigir recursos computacionais substanciais;

- **Eficiência Relativa**: pode haver outros métodos que oferecem melhores taxas de compressão para determinados conjuntos de dados;

- **Ineficácia em Casos Específicos**: pode ser menos eficaz em dados com poucos símbolos únicos ou quando os símbolos já estão altamente comprimidos.

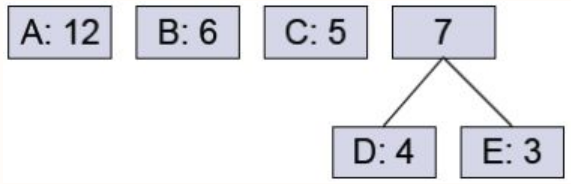
Código de Huffman: exemplo

Seja um alfabeto de cinco letras: A, B, C, D, E, com as seguintes frequências no texto: A: 12, B: 6, C: 5, D: 4, E: 3.

- Começar com cinco subárvores separadas:

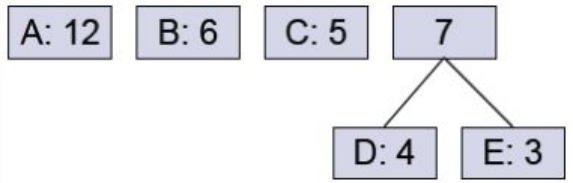
A: 12 B: 6 C: 5 D: 4 E: 3

- Subárvores para D e E possuem as frequências mais baixas, então são mescladas totalizando 7 ocorrências.



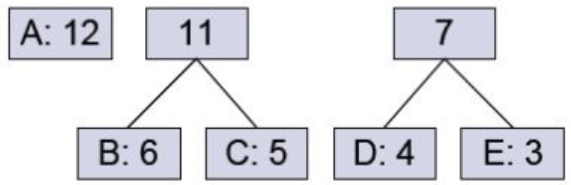
Código de Huffman: exemplo

Passo anterior:



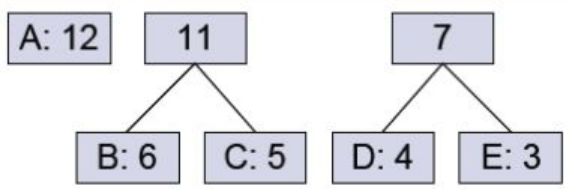
Próximo passo:

- Agora B e C possuem as frequências mais baixas, então são mescladas totalizando 11 ocorrências.



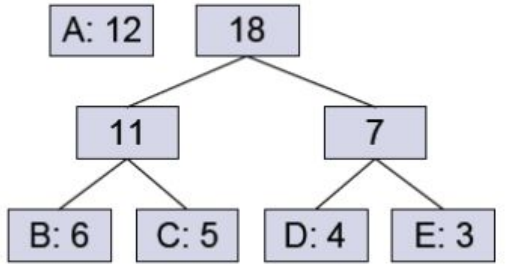
Código de Huffman: exemplo

Passo anterior:



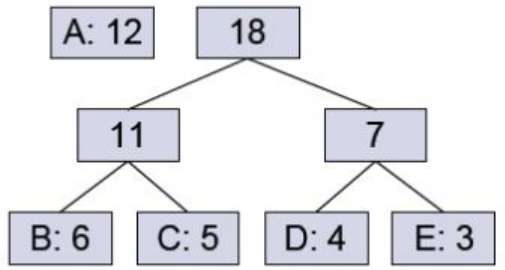
Próximo passo:

- Agora {B, C} e {D, E} possuem as frequências mais baixas, então são mescladas:

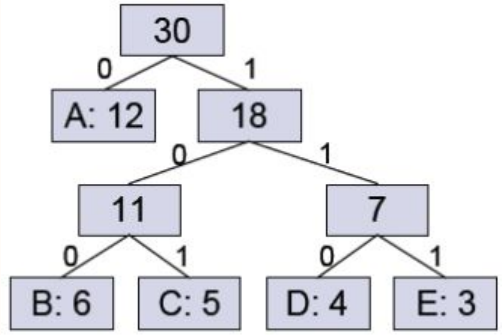


Código de Huffman: exemplo

Passo anterior:



Finalmente, para codificar um caractere, siga a árvore. Cada ramo à esquerda adiciona um 0 e cada ramo à direita adiciona um 1. Por exemplo: **A** é codificado como 0, **B** é codificado como 100, **E** é codificado como 111.



Código de Huffman

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Código de Huffman

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)
= 2.2 bits/pixel

Compressão com perdas

JPEG (Joint Photographic Experts Group):

- Bom para comprimir imagens fotográficas
- Mudanças graduais na cor;
- Não é bom para gráficos (mudanças bruscas na cor);
- Taxa de compressão de 10:1 é alcançável sem perda visível;
- Utiliza o formato de arquivo **JFIF**: o Formato de Intercâmbio de Arquivo JPEG (https://www.w3.org/Graphics/JPEG/)

Critérios de Fidelidade

Como medir perdas causadas (método de quantização, p. ex. em JPEG):

- **Critérios Objetivos**: PSNR (*Peak signal-to-noise ratio*), SNR (*means-square signal-to noise ratio*) etc.

- **Critérios Subjetivos**: análise por seres humanos (excelente, bom, razoável, marginal, inferior, indesejável)

Critério de fidelidade objetivo

Assim pode ser definida a **métrica PSNR**, a partir do **MSE** e do **nível máximo de cinza (L-1)**:

$$MSE = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2}{M.N}$$

$$PSNR = 10.\log_{10}\left(\frac{(L-1)^2}{MSE}\right) = 20.\log_{10}\left(\frac{255}{RMSE}\right)$$

Em que f e $\hat{f}$ representam as funções intensidade para o sinal original e restaurado, respectivamente

Critério de fidelidade objetivo

Em termos da **métrica SNR**, segue a definição:

$$SNR = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\hat{f}(x,y)^2\right)}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]^2}$$

Em que f e $\hat{f}$ representam as funções intensidade para o sinal original e restaurado, respectivamente

Desempenho do Padrão JPEG

bits/pixel	**qualidade da imagem reconstruída**
0,083	imagem reconhecível
0,25	imagem usável
0,75	imagem excelente
2,25	indistinguível da original

Principais passos envolvidos na compressão JPEG:

- **Conversão do Espaço de Cor**:
  - Se a imagem estiver em um espaço de cor diferente de YCbCr (como RGB), ela é convertida para o espaço de cor YCbCr. YCbCr separa a imagem em componentes de luminância (brilho) e crominância (cor);

- **Subamostragem** (**Redução de Crominância**):
  - No espaço de cor YCbCr, as informações de crominância (Cb e Cr) **podem ser subamostradas** para reduzir a quantidade de dados.

- **Divisão em Blocos e Transformação**:
  - A imagem é dividida em pequenos blocos, geralmente de 8x8 pixels cada. Cada bloco é então submetido a uma transformação matemática chamada Transformada Discreta de Cosseno (DCT).

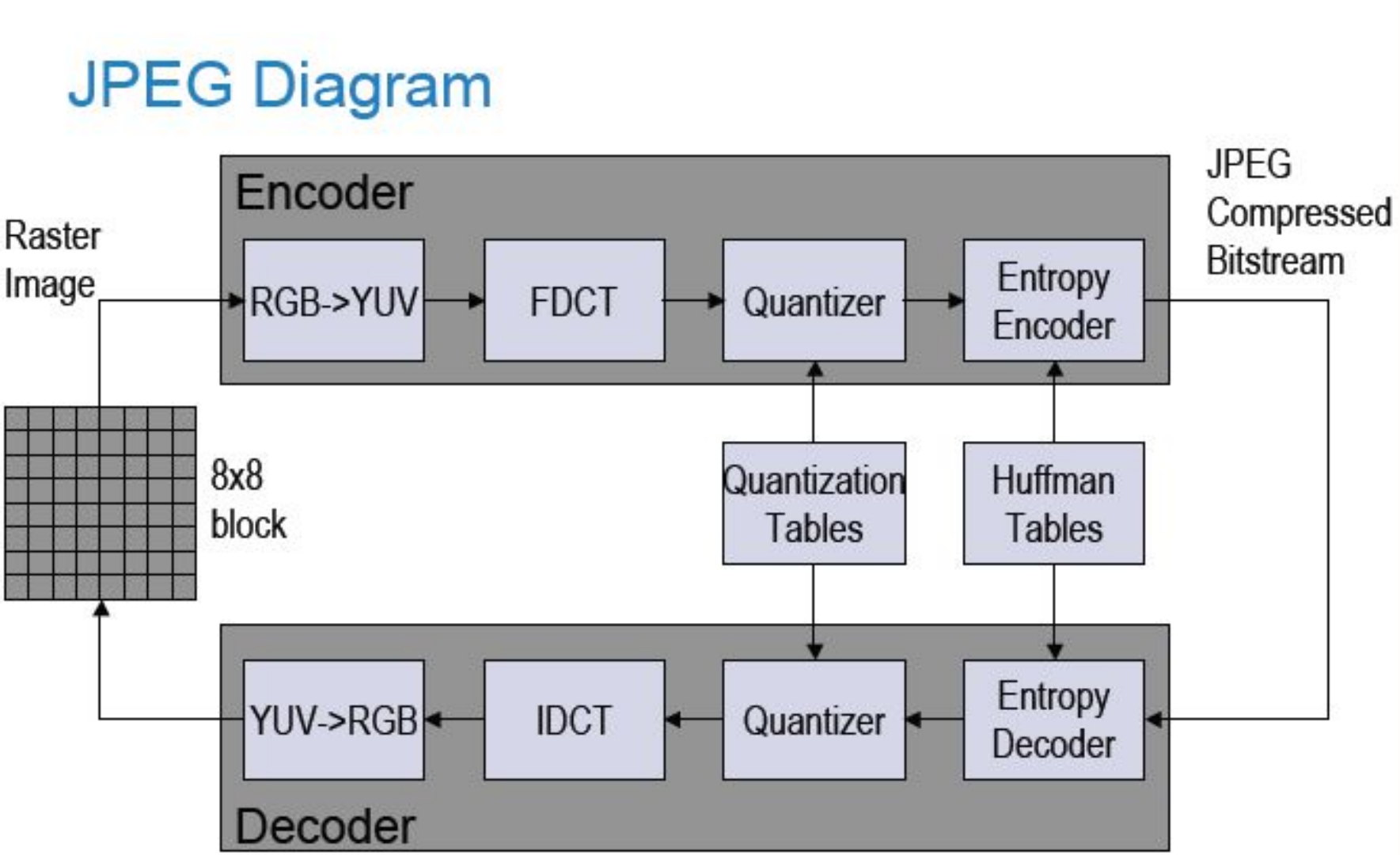
Continuação (passos envolvidos na compressão JPEG):

- **Quantização**:
  - Os coeficientes da DCT são quantizados, o que envolve dividir cada coeficiente por uma matriz de quantização predefinida. Esse passo introduz perda. Valores de quantização mais altos resultam em mais compressão, mas menor qualidade da imagem.

- **Codificação de Entropia (Codificação Huffman):**
  - Os coeficientes DCT quantizados são então submetidos à codificação de entropia, geralmente à codificação Huffman (e RLE).

- **Adição de Cabeçalho e Metadados:**
  - Os dados da imagem comprimida são encapsulados com cabeçalhos e metadados que fornecem informações sobre a imagem, como dimensões, proporções de subamostragem e tabelas de quantização.

Diagrama em blocos da compressão JPEG

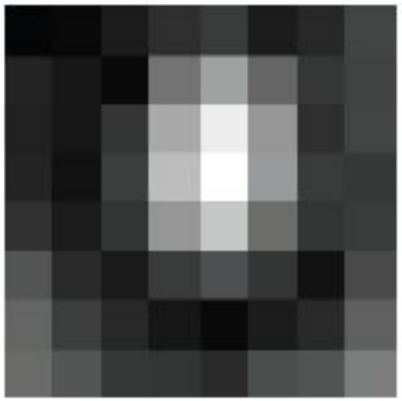


Fonte: https://slideplayer.com/slide/10403444/

Etapa para compressão JPEG:

JPEG Example

Original 8x8 luminance block



- Actual values:

52	55	61	66	70	61	64	73
64	59	55	90	109	85	69	72
62	59	68	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	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Discrete Cosine Transform (two-dimensional, 2D-DCT)

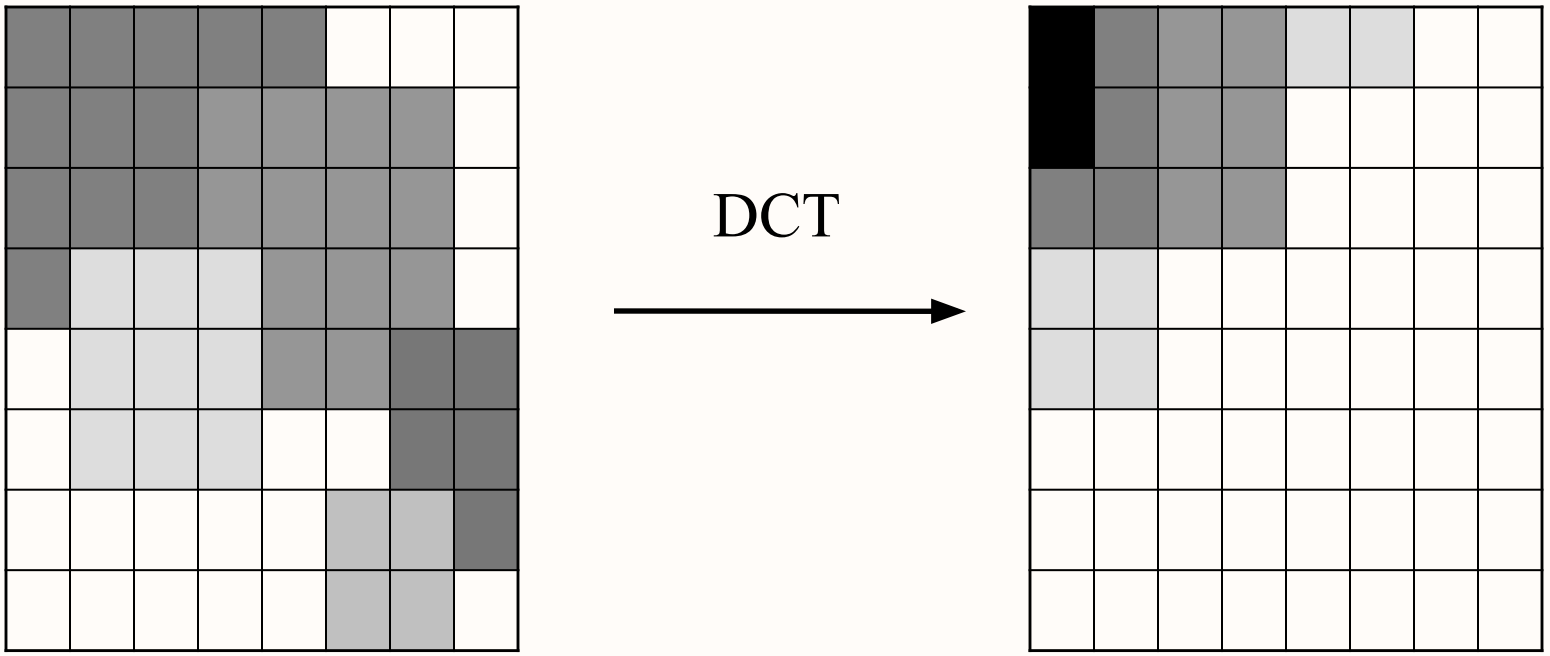
$$F[k.l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n]\alpha(k)\alpha(l)\cos\left(\frac{(2m+1)k\pi}{2M}\right)\cos\left(\frac{(2n+1)l\pi}{2N}\right)$$

$$k,l = 0,1,\ldots,N-1 \qquad \alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1,2,\ldots,N-1 \end{cases}$$

Inv. Discrete Cosine Transform (two-dimensional, 2D-DCT)

$$f[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l]\alpha(k)\alpha(l)\cos\left(\frac{(2m+1)k\pi}{2M}\right)\cos\left(\frac{(2n+1)l\pi}{2N}\right)$$

Etapa para compressão JPEG:



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

F(u,v) = 64 componentes de freqüências espaciais

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Etapa para compressão JPEG:

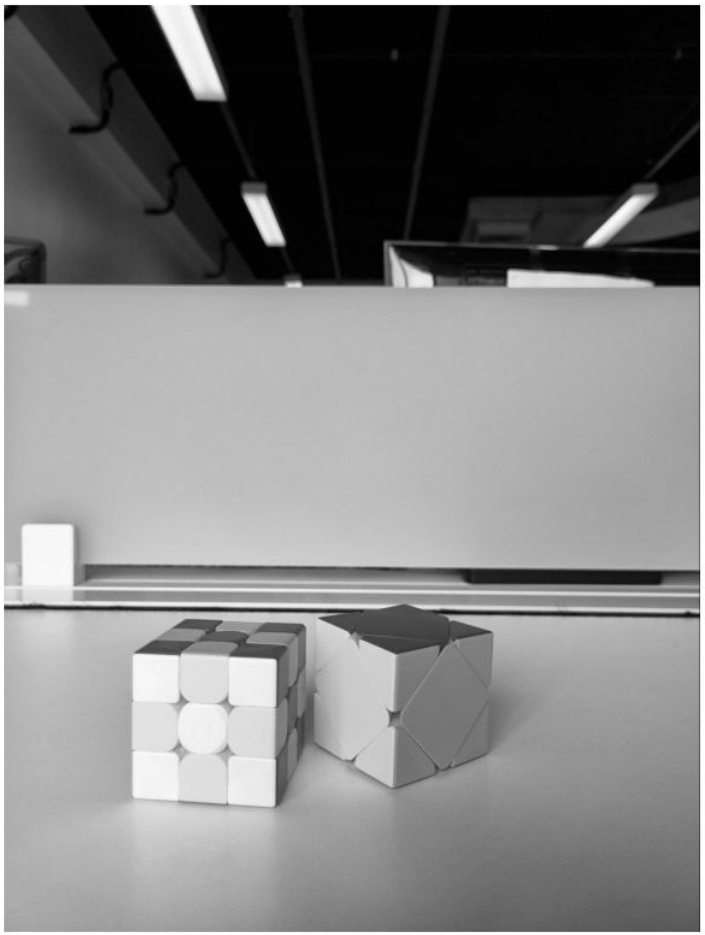


Imagem original

DCT

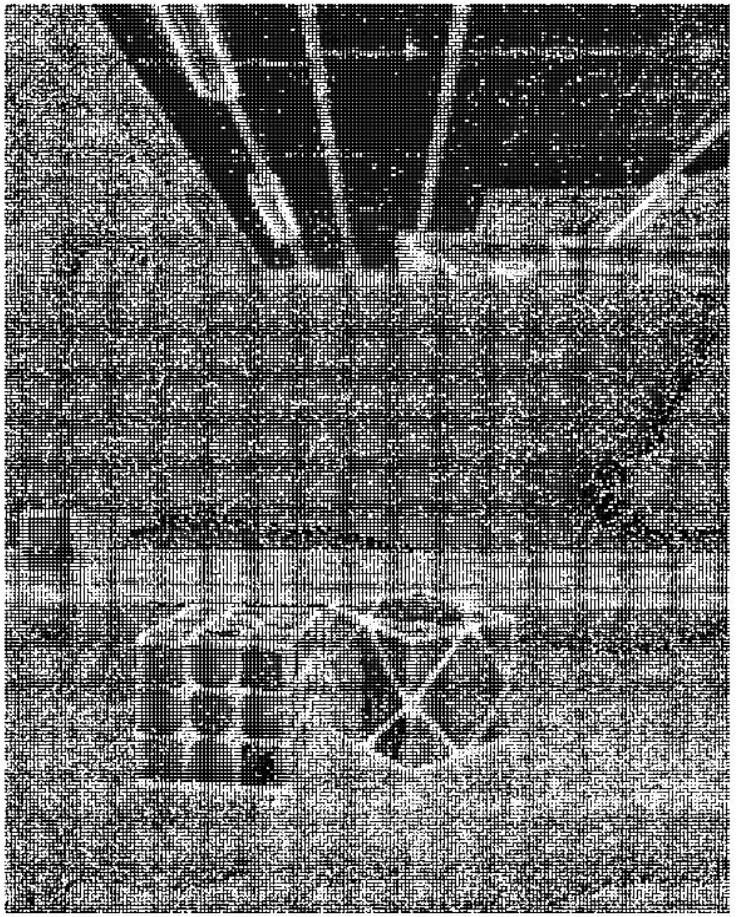


Imagem após a DCT decomposta em patches 8x8

Exemplo de Transformada DCT

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

-128

$$\begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

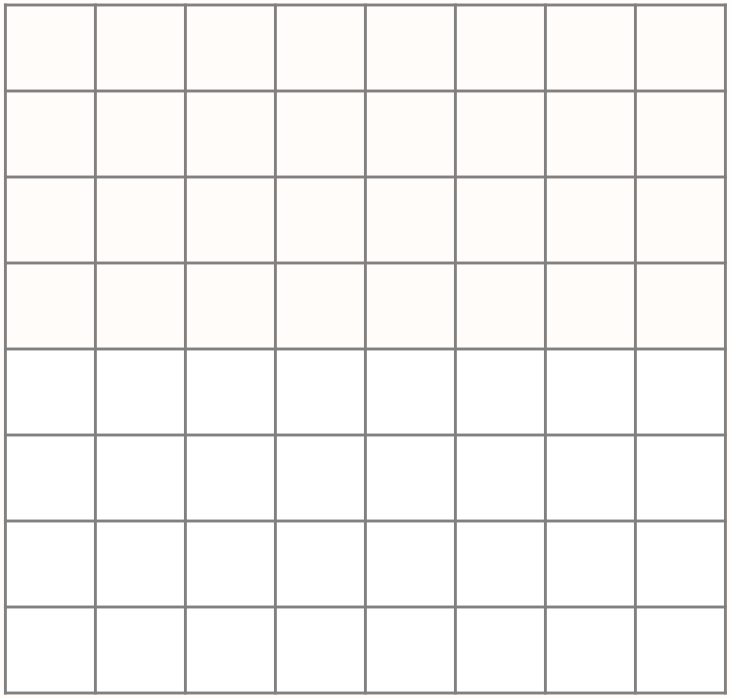
Exemplo de Transformada DCT

$$\begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

DCT

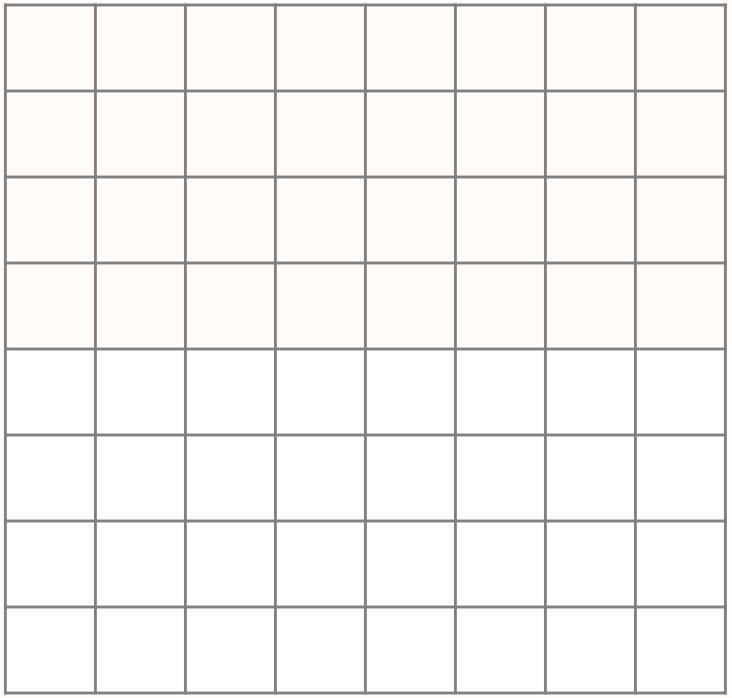
$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

Exemplo de Transformada DCT



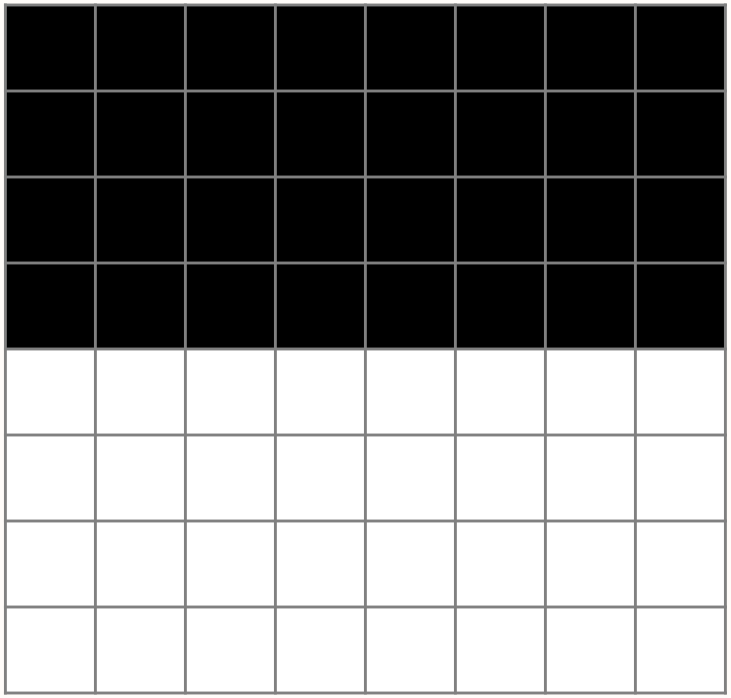
+1016.0	-0.6	+0.6	-0.6	+0.6	-0.7	+0.7	-0.8
-0.6	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0
+0.6	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0
-0.6	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0
+0.6	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0
-0.7	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0
+0.7	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0
-0.8	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0

Exemplo de Transformada DCT



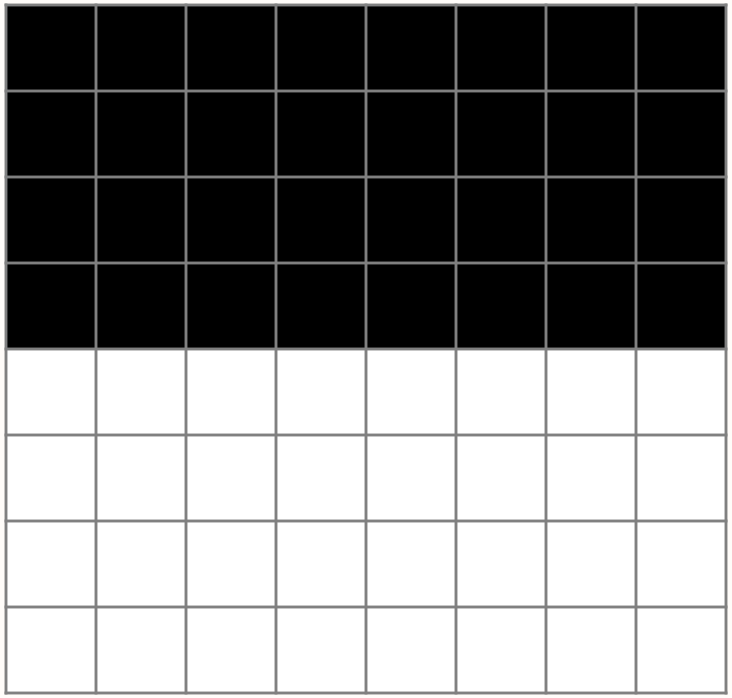
1016	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Exemplo de Transformada DCT



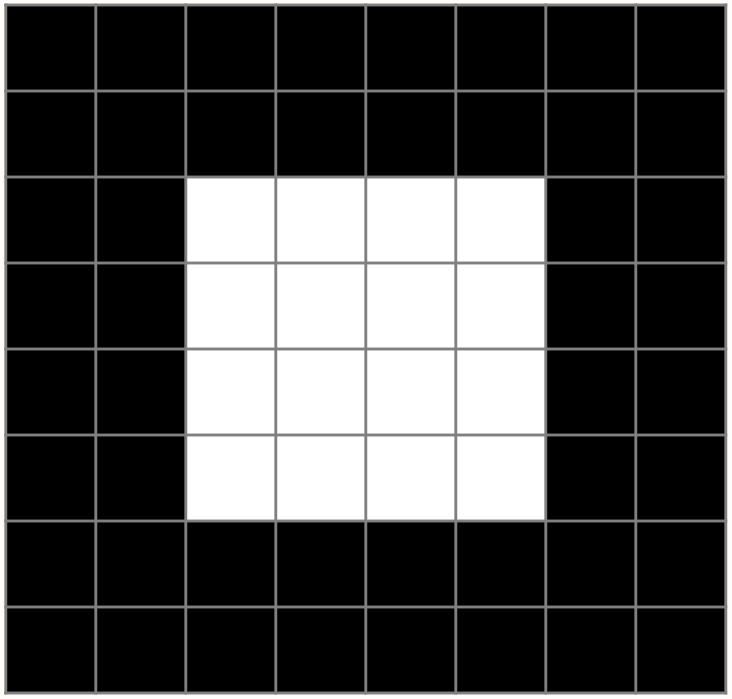
-4.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0
-924.5	+0.5	-0.5	+0.6	-0.6	+0.6	-0.7	+0.7
+1.2	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0
+323.8	-0.2	+0.2	-0.2	+0.2	-0.2	+0.2	-0.3
-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0
-217.5	+0.1	-0.1	+0.1	-0.1	+0.1	-0.2	+0.2
+1.5	-0.0	+0.0	-0.0	+0.0	-0.0	+0.0	-0.0
+183.0	-0.1	+0.1	-0.1	+0.1	-0.1	+0.1	-0.1

Exemplo de Transformada DCT



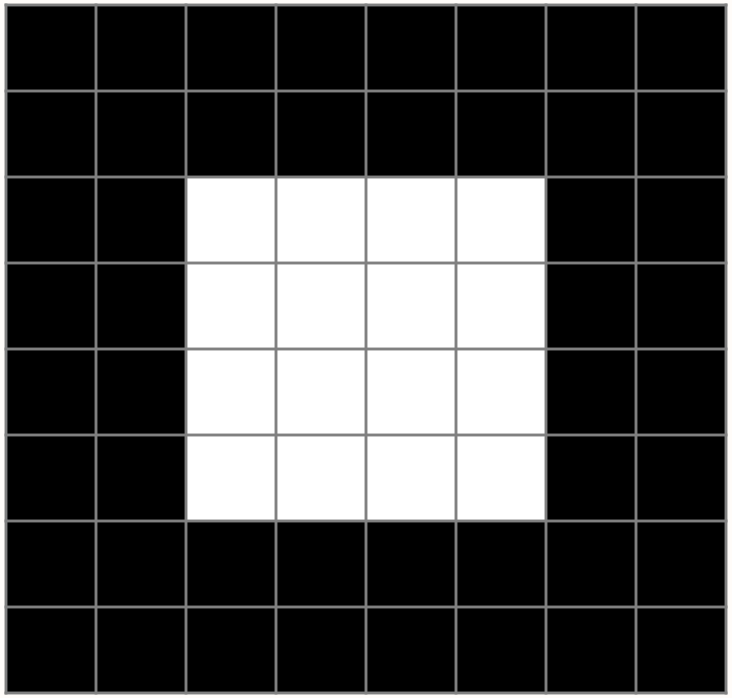
-4	0	0	0	0	0	0	0
-924	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
324	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-217	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
183	0	0	0	0	0	0	0

Exemplo de Transformada DCT



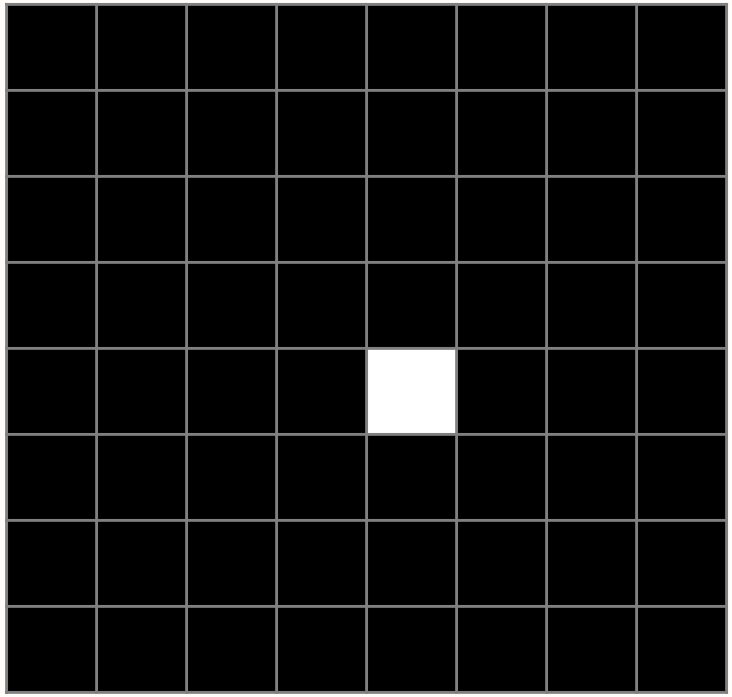
-514.0	+0.2	-471.6	+1.1	-1.0	+1.2	+194.4	+0.2
+0.2	-0.0	+0.4	-0.0	+0.0	-0.0	-0.2	-0.0
-471.6	+0.4	+435.0	-0.4	+0.3	-0.4	-180.2	+0.5
+1.1	-0.0	-0.4	-0.0	+0.0	-0.0	+0.2	-0.0
-1.0	+0.0	+0.3	+0.0	-0.0	+0.0	-0.1	+0.0
+1.2	-0.0	-0.4	-0.0	+0.0	-0.0	+0.2	-0.0
+194.4	-0.2	-180.2	+0.2	-0.1	+0.2	+74.7	-0.2
+0.2	-0.0	+0.5	-0.0	+0.0	-0.0	-0.2	-0.0

Exemplo de Transformada DCT



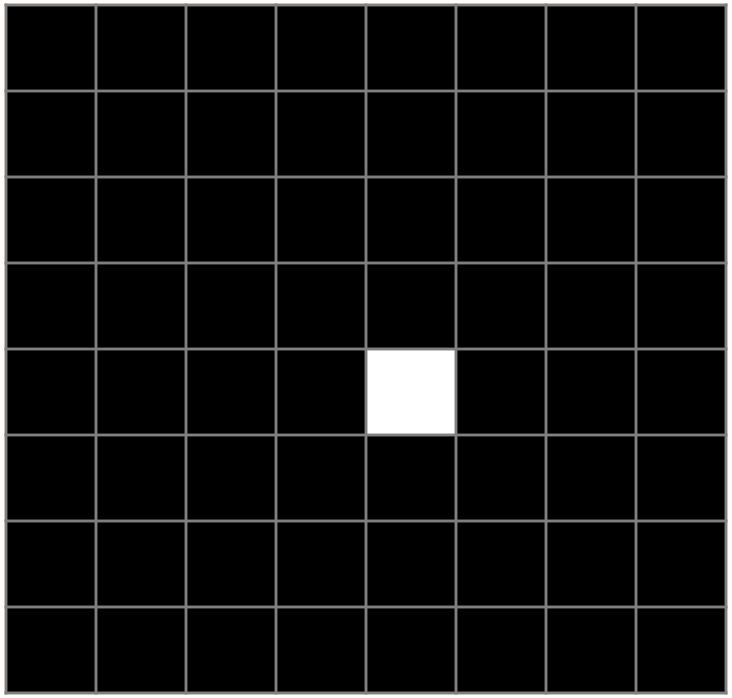
-514	0	-472	0	0	0	194	0
0	0	0	0	0	0	0	0
-471	0	435	0	0	0	-180	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
194	0	180	0	0	0	75	0
0	0	0	0	0	0	0	0

Exemplo de Transformada DCT



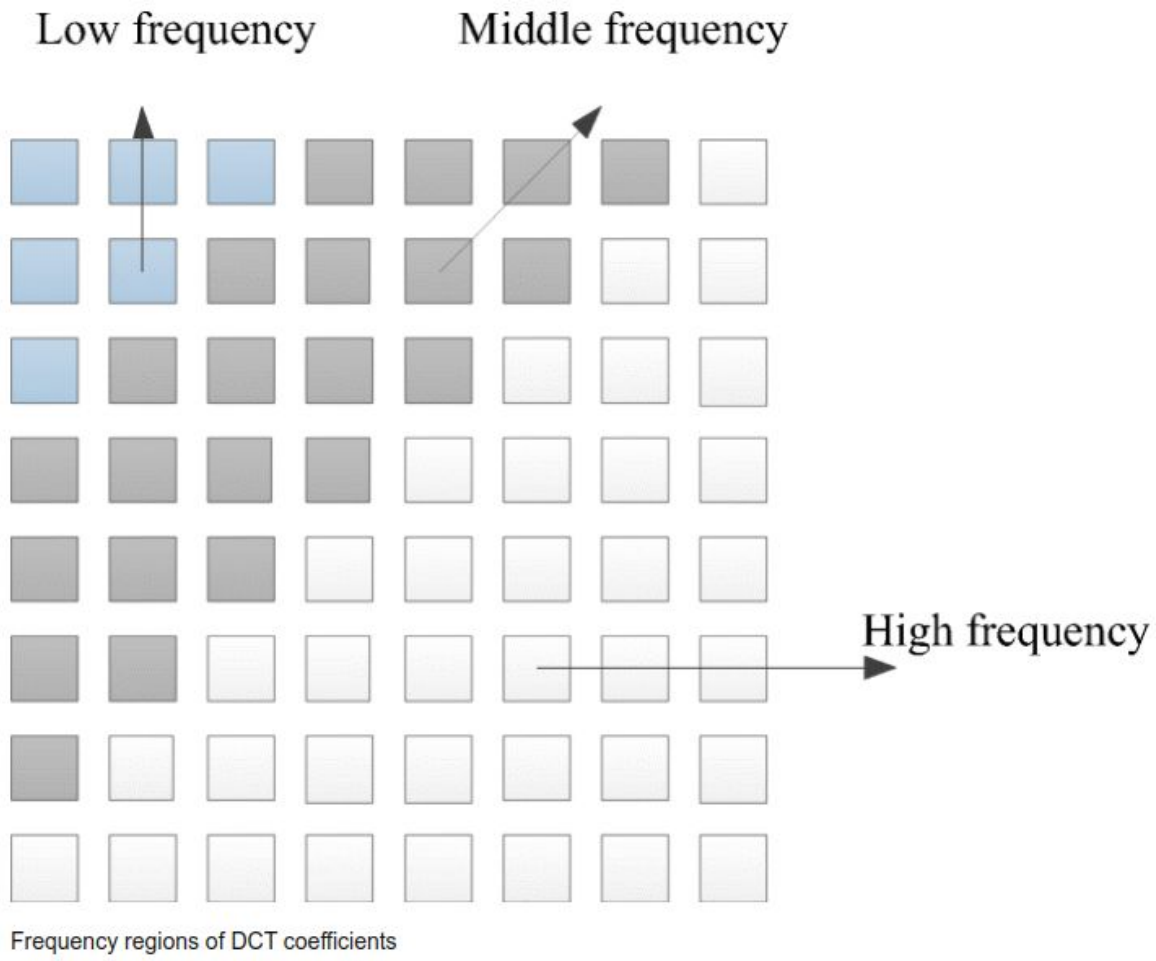
-992.1	-8.2	-42.2	+25.7	+31.1	-36.9	-17.8	+45.1
-8.2	+2.4	+11.5	-7.0	-8.8	+10.4	+4.7	-12.3
-42.2	+11.5	+54.3	-32.8	-41.5	+49.1	+22.3	-57.8
+25.7	-7.0	-32.8	+19.8	+25.1	-29.6	-13.5	+34.9
+31.1	-8.8	-41.5	+25.1	+31.7	-37.5	-17.0	+44.1
-36.9	+10.4	+49.1	-29.6	-37.5	+44.3	+20.1	-52.2
-17.8	+4.7	+22.3	-13.5	-17.0	+20.1	+9.1	-23.7
+45.1	-12.3	-57.8	+34.9	+44.1	-52.2	-23.7	+61.4

Exemplo de Transformada DCT



-992	-9	-42	25	32	-37	-17	44
-9	2	11	-7	-9	10	5	-12
-42	11	54	-33	-42	49	23	-58
25	-7	-33	20	25	-29	-14	35
32	-9	-42	25	32	-37	-17	44
-37	10	49	-29	-37	44	20	-52
-17	5	23	-14	-17	20	9	-24
44	-12	-58	35	44	-52	-24	61

Frequências em regiões na Transformada DCT



Frequency regions of DCT coefficients

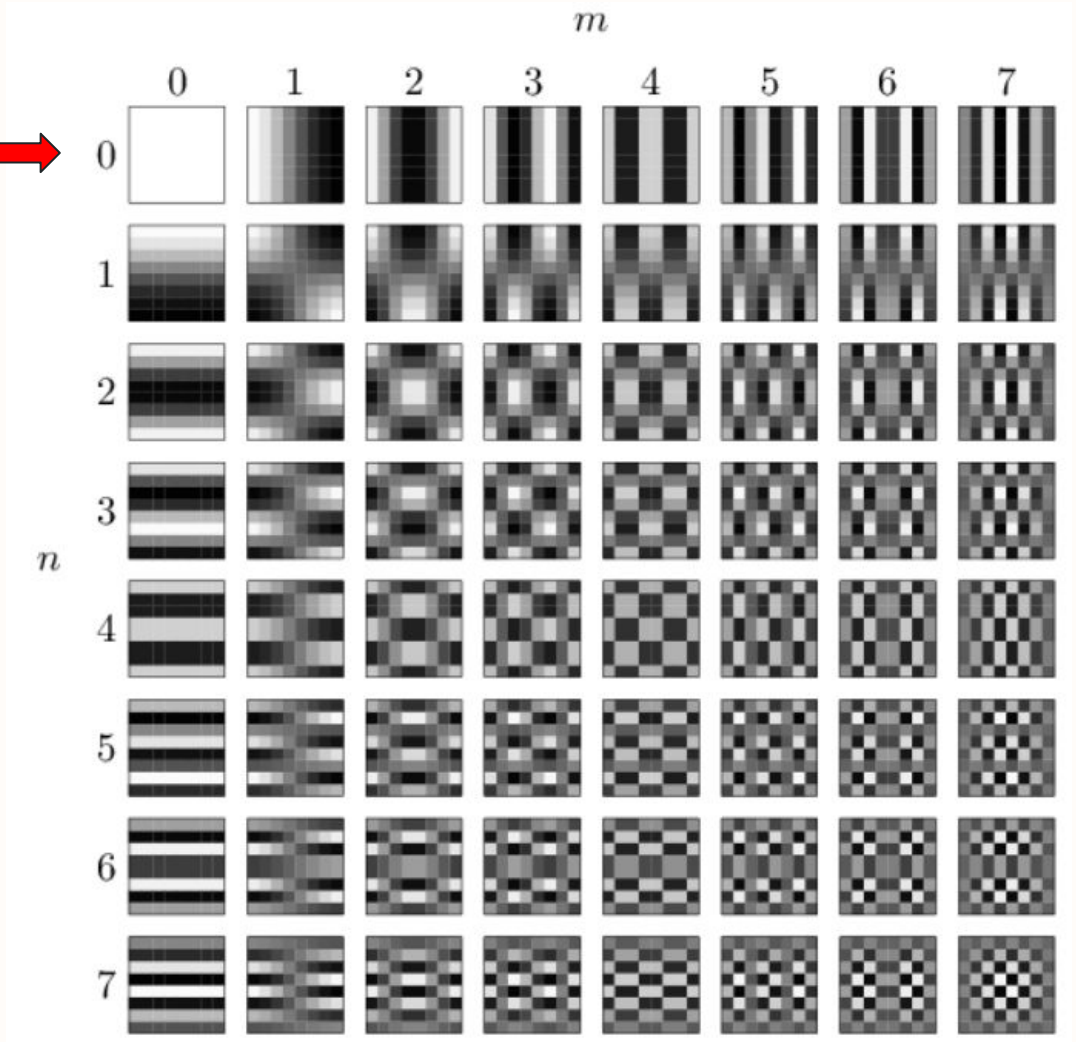
Fonte: Tian, Cheng & Wen, Ru-Hong & Zou, Wei-Ping & Gong, Li-Hua. (2020). Robust and blind watermarking algorithm based on DCT and SVD in the contourlet domain. Multimedia Tools and Applications. 79. 10.1007/s11042-019-08530-z.

Objetivos da DCT na Compressão JPEG

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

Exemplo da Transformada DCT para um bloco 8x8

Componente DC



Componentes em alta frequência

Fonte: https://sites.icmc.usp.br/frasson/jpeg/jpeg.html

Matriz base DCT

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases}$$

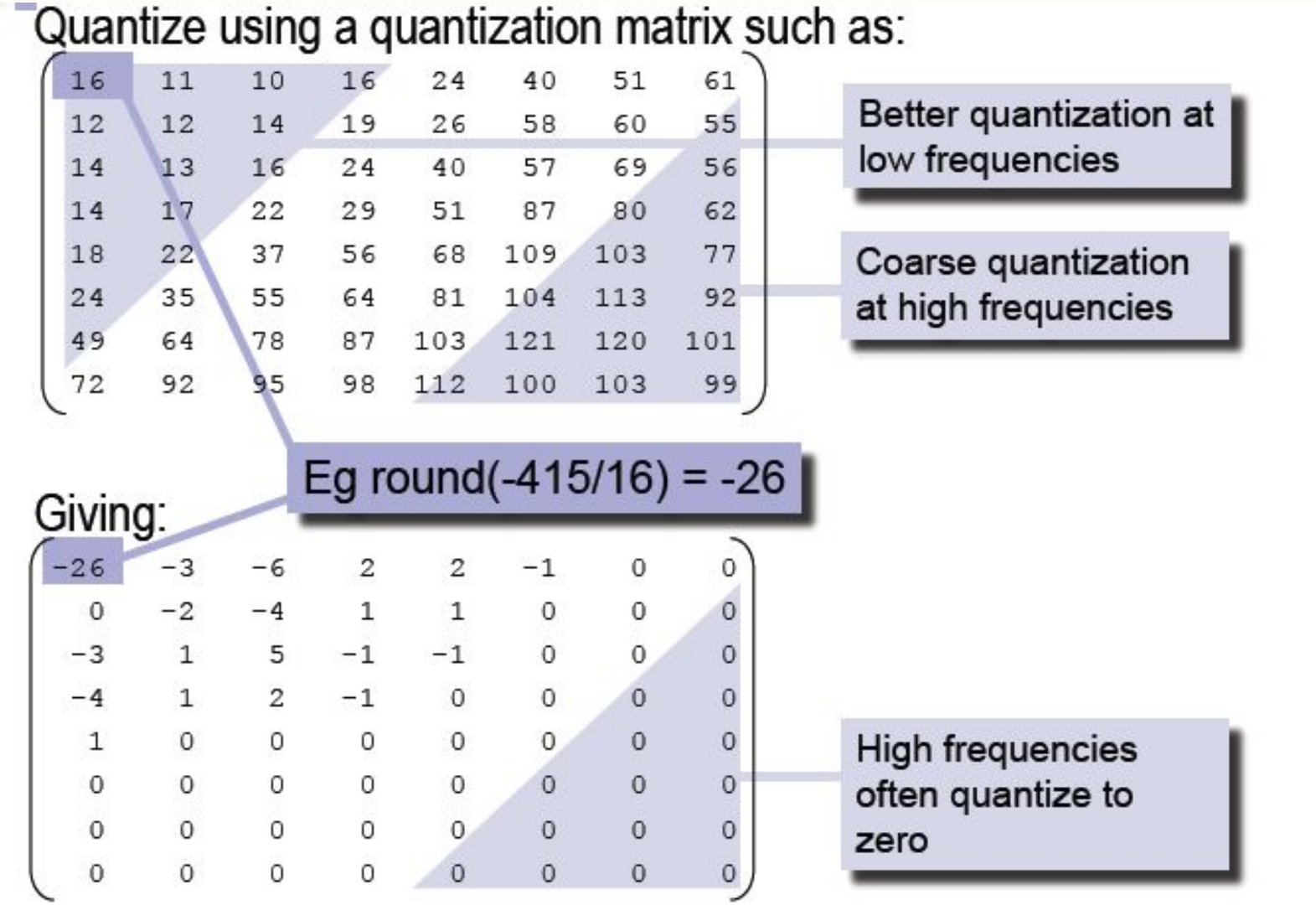
Considerando um bloco de 8 x 8 pixels, segue a matriz base ortonormal:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

$$D = T.M.T^{-1}$$

https://www.math.cuhk.edu.hk/~lmlui/dct.pdf

Aplicação de matriz de quantização



Fonte: https://slideplayer.com/slide/10403444/

The Baseline JPEG - Standard Quantization Matrix

- Determinado por testes subjetivos

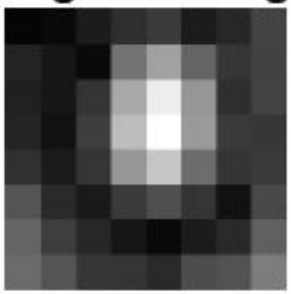
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantização dos coeficientes

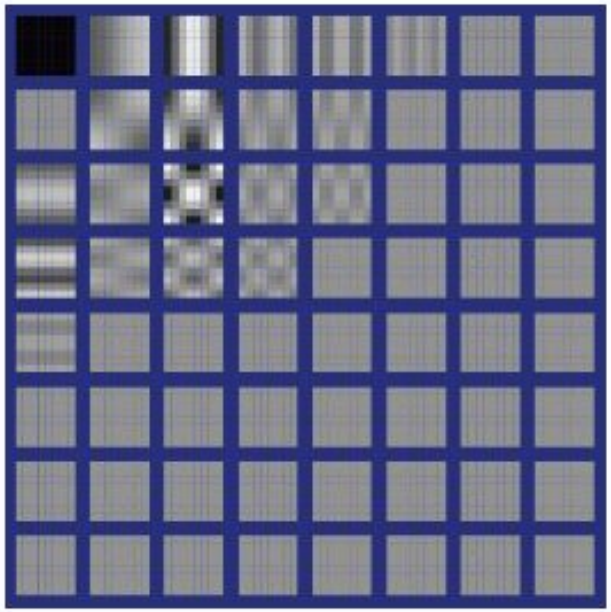
Quantized DCT coefficients:

$$\begin{pmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Original Image:



Scaled DCT basis functions that make up the (quantized) image

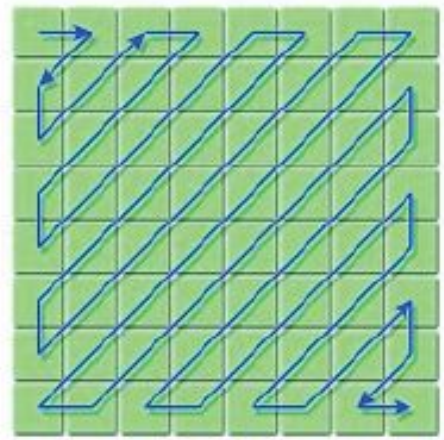


Fonte: https://slideplayer.com/slide/10403444/

Ordenação e compressão

Order the coefficients in zig-zag order:

−26	−3	−6	2	2	−1	0	0
0	−2	−4	1	1	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−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



−26, −3, 0, −3, −2, −6, 2, −4, 1, −4, 1, 1, 5, 1, 2, −1, 1, −1, 2, 0, 0, 0, 0, 0, −1, −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, 0, 0, 0, 0, 0, 0,

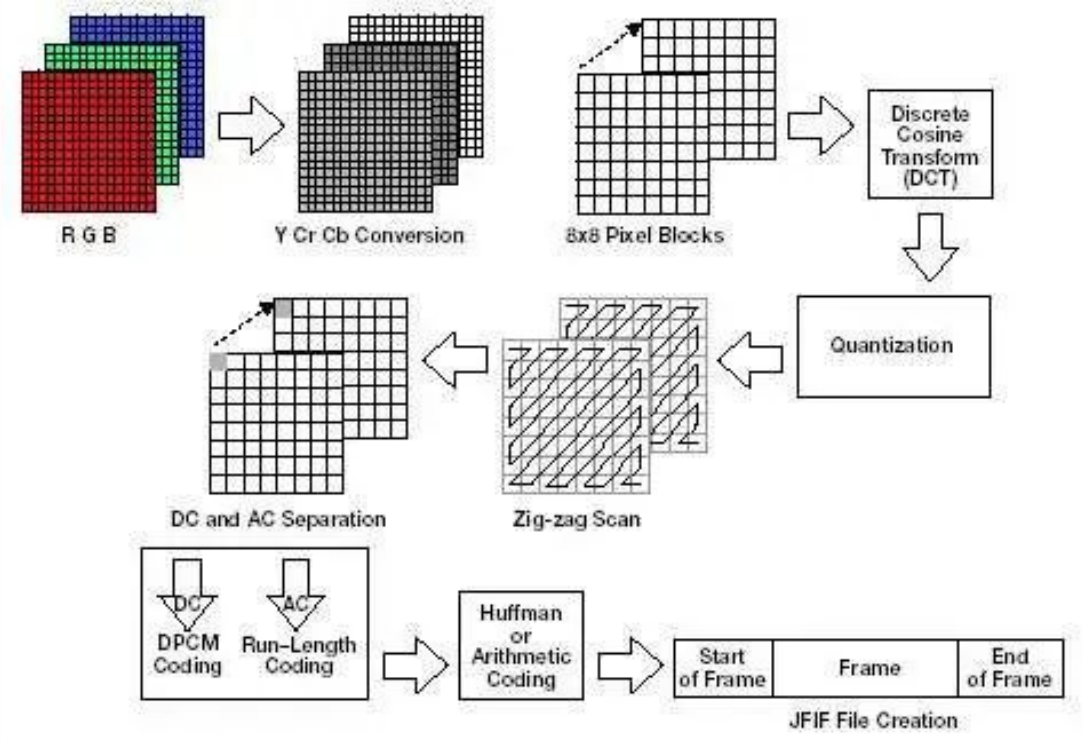
Run-length encode:

−26, −3, 0, −3, −2, −6, 2, −4, 1, −4, {2 x 1}, 5, 1, 2, −1, 1, −1, 2, {5 x 0} , −1, −1, EOB

Huffman code what remains.   Encoding is complete.

Fonte: https://slideplayer.com/slide/10403444/

Resumo da codificação JPEG



Fonte:
https://www.eetimes.com/baseline-jpeg-compression-juggles-image-quality-and-size/

Decodificação JPEG

Etapas:

- A decodificação é simplesmente o inverso da codificação
- Reverter as codificações de Huffman e RLE.
- Desquantizar;
- Aplicar a Inversa da Transformada Discreta do Cosseno (IDCT):

$$V(x,y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c_i c_j T(i,j) cos \frac{(2y+1)i\pi}{2N} \cos \frac{(2x+1)j\pi}{2N}$$

- Adicionar 128 para converter de volta para valores do tipo unsigned.

Comparação Pré x Pós Compressão

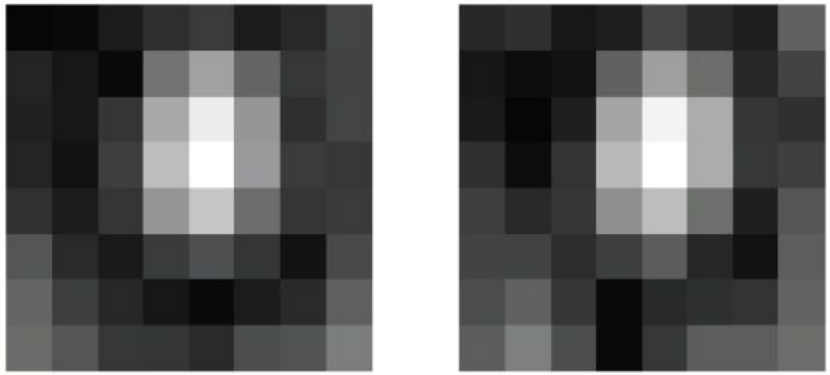


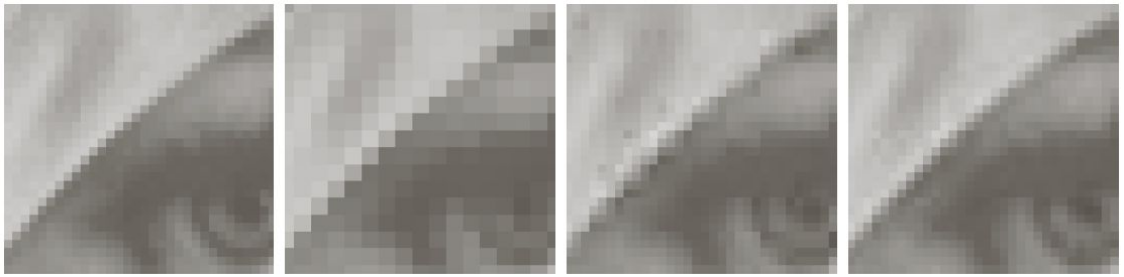
Imagem Original Imagem descomprimida

Fonte: https://slideplayer.com/slide/10403444/

Taxa de Compressão no formato JPEG

- A taxa de compressão depende da magnitude dos valores na matriz de quantização;
- 10:1 é alcançável sem perda perceptível;
- 100:1 é alcançável, mas defeitos são perceptíveis na imagem

Efeitos das subimagens na compressão(artefatos)



2 X 2 4 X 4 8 X 8

Aproximações usando 25% dos coeficientes da DCT em subimagens de 2 X 2 (segunda à direita), subimagens de 4 X 4 (próxima à direita) e subimagens (ultima à direita) de 8 X 8. A imagem original (à esquerda) é uma seção ampliada da imagem no topo.

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Efeitos da reconstrução sob diferentes razões de compressão

Razão de compressão: 25:1

Razão de compressão: 52:1

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Original (100KB)

JPEG 75% (18KB)

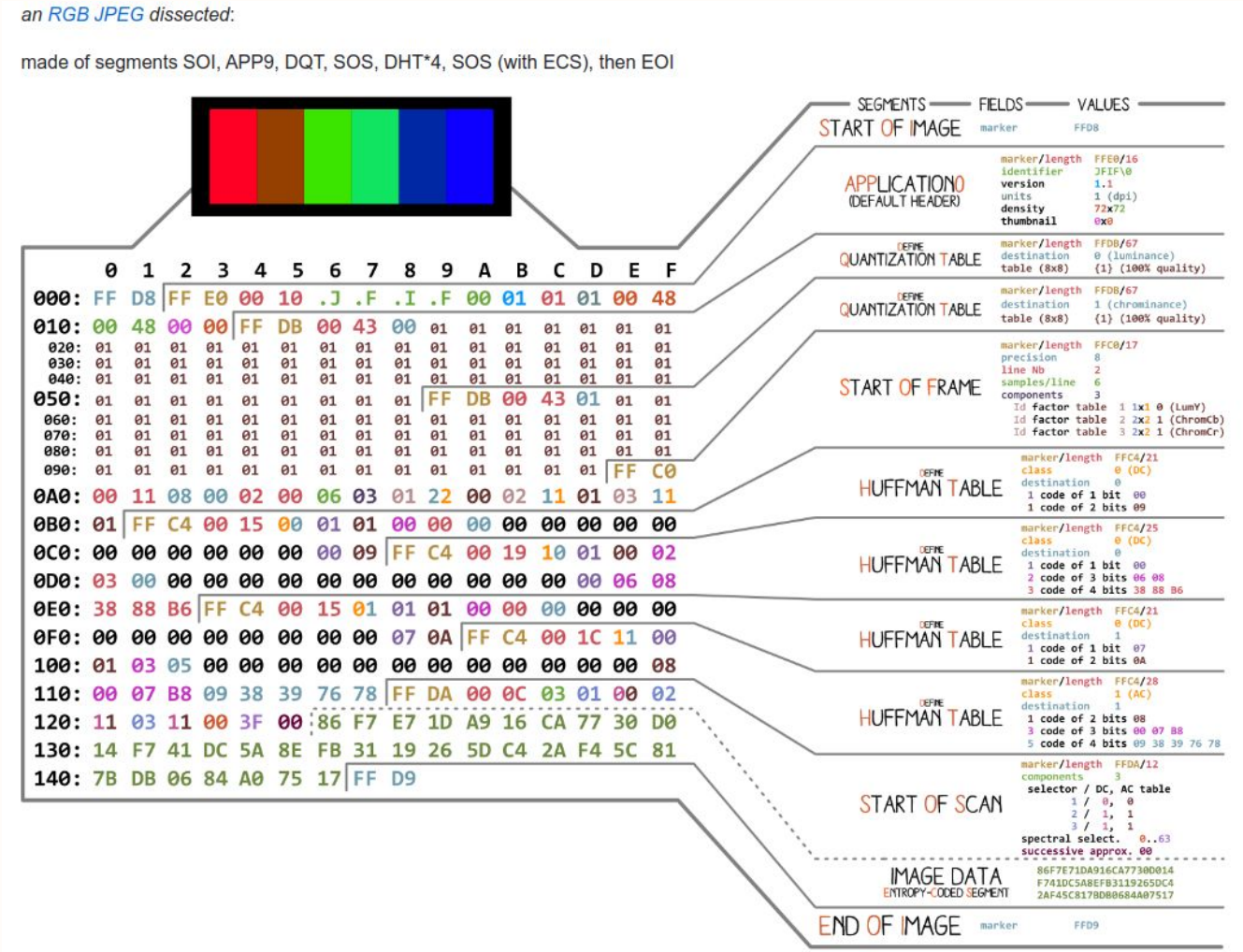
JPEG 50% (12KB)

JPEG 30% (9KB)

JPEG 10% (5KB)

Fonte: GONZALEZ, R. C.; WOODS, R. E. Digital Image Processing. Pearson, New York, NY.

Metadados: arquivos de imagens (JPEG)



https://github.com/corkami/formats/blob/master/image/jpeg.md

Outras Transformadas para Descorrelação Espacial

- Transformada de Walsh
- Transformada de Hadamard
- * Pesquisar o uso dessas transformadas para compressão de imagens!

JPEG2000

Características:

- Sistema de codificação de imagem: ISO/IEC 15444-1:2000
- **Compressão por “Wavelets”**
- Codificação Aritmética
- Segmentação: retângulos, planos, regiões de interesse (ROI)
- Desempenho: **arquivos 20-30% menores para mesma qualidade**
- Referência: **https://jpeg.org/jpeg2000/**

JPEG2000: Filtragem “Wavelet”

- **A DCT não permite propriedades de sinais não-estacionários;**

- Um sinal estacionário é aquele em que as propriedades estatísticas do sinal permanecem constantes ao longo do tempo, ou seja, a média e a variância do sinal não mudam;

- A transformação wavelet tem uma vantagem sobre as transformações DCT porque permite propriedades de sinais não-estacionários.

JPEG2000: Filtragem “Wavelet” (1 passo)

Fonte:https://www.lcs.poli.usp.br/~gstolfi/PPT/APTV0616.pdf

JPEG2000: Filtragem “Wavelet” (2 passos)

Fonte:https://www.lcs.poli.usp.br/~gstolfi/PPT/APTV0616.pdf

JPEG2000: Filtragem “Wavelet” (3 passos)

Cada sub-imagem obtida em cada etapa será então quantizada, e as amostras não nulas resultantes são compactadas e codificadas em símbolos de comprimento variável.

Fonte:https://www.lcs.poli.usp.br/~gstolfi/PPT/APTV0616.pdf

Comparação com JPEG

JPEG

JPEG2000

0,25 bits/pixel

Fonte:https://www.lcs.poli.usp.br/~gstolfi/PPT/APTV0616.pdf

Muito obrigado!

INSTITUIÇÃO EXECUTORA

Centro de Informática UFPE

FADE UFPE

COORDENADORA

MCTI FUTURO

Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO