

Tópico 11 - Formatos de Imagens

João B. Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
jbflorindo@ime.unicamp.br

Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão
- 8 Conclusões

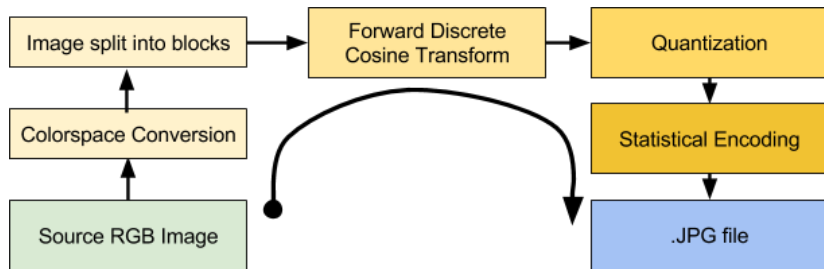
Formato JPG

Surge em 1992.

Um dos maiores avanços da história da Computação.

Padrão dominante na Internet.

Percepção de cores e contornos pelo olho humano.



Outline

- 1 Formato JPG
- 2 **JPG - Conversão de cores**
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão
- 8 Conclusões

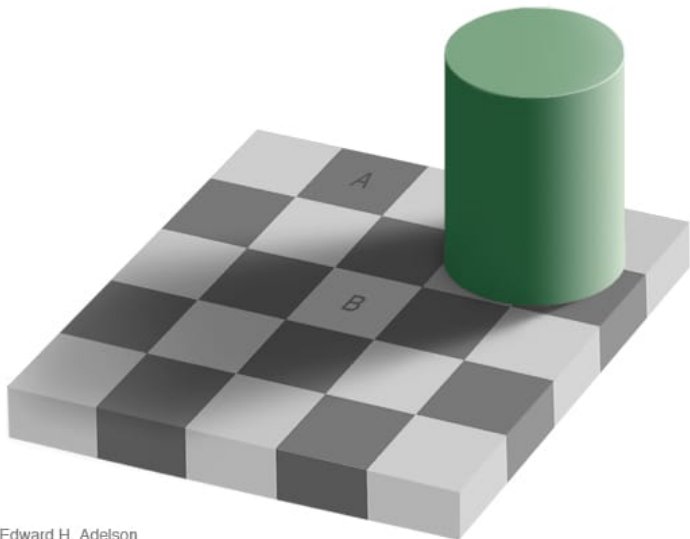
JPG - Conversão de cores

Compressão com perda (*lossy*) - olho humano não é tão preciso quanto a máquina.

Distingue “apenas” 10 milhões de cores.

Percepção facilmente enganada.

JPG - Conversão de cores

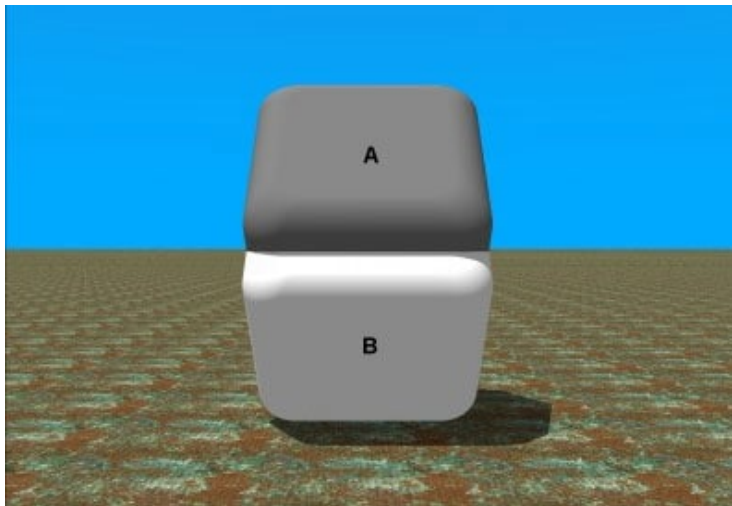


Edward H. Adelson

JPG - Conversão de cores



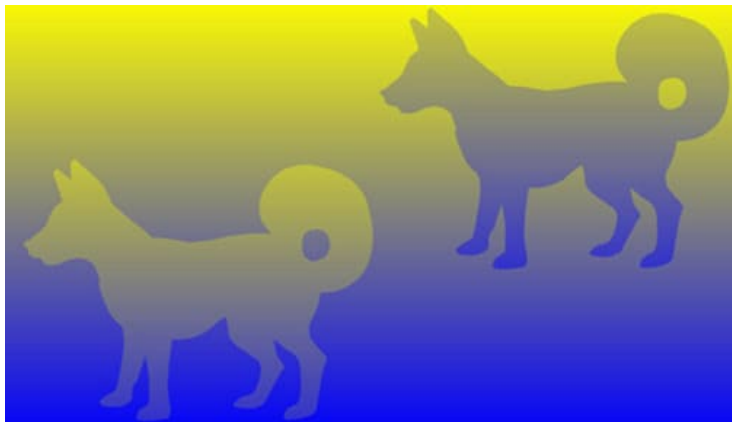
JPG - Conversão de cores



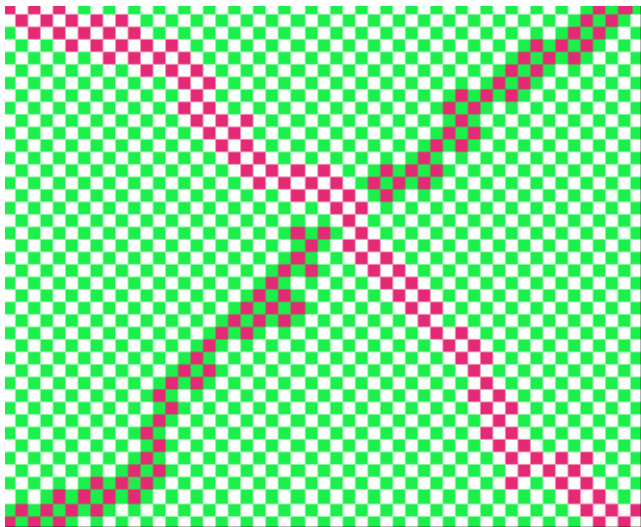
JPG - Conversão de cores



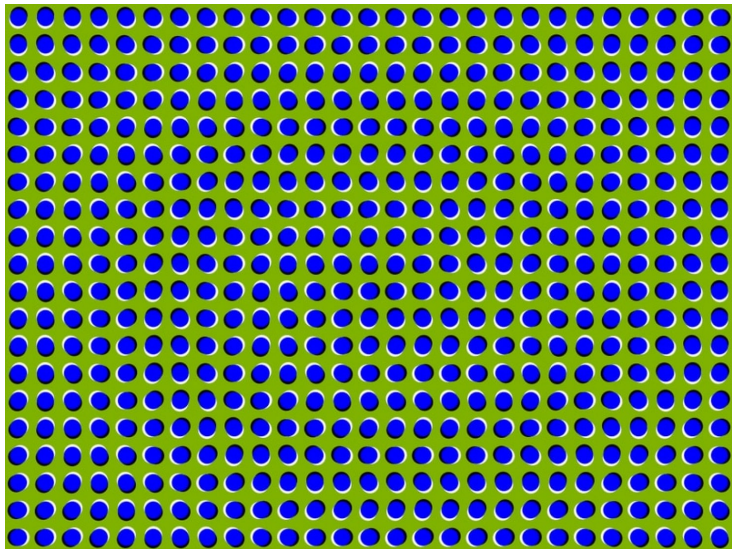
JPG - Conversão de cores



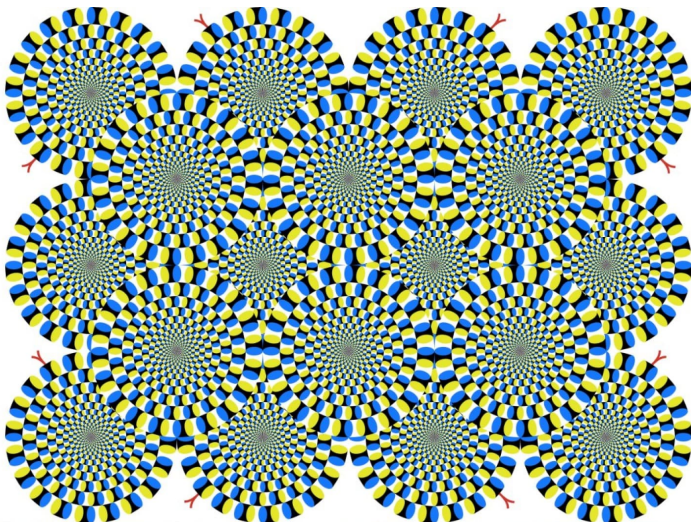
JPG - Conversão de cores



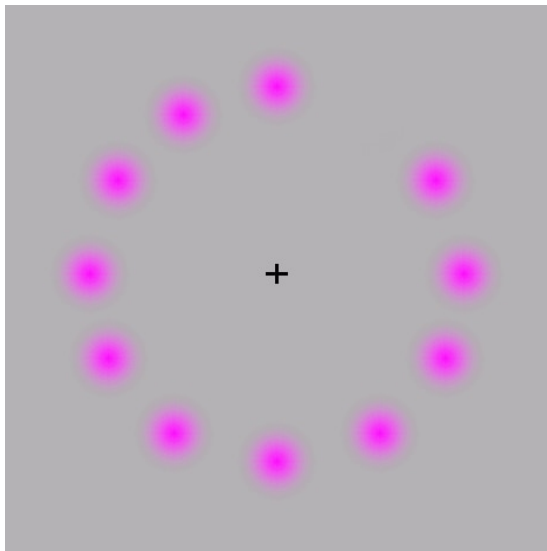
JPG - Conversão de cores



JPG - Conversão de cores



JPG - Conversão de cores



JPG - Conversão de cores

JPG - Conversão de cores



JPG - Conversão de cores

Mente pode ser manipulada.

Diminuir a quantização é uma estratégia na compressão *lossy*.

Porém, JPG faz mudança do espaço de cores.



R: 233
G: 30
B: 99



C: 0
M : 0.877
Y : 0.593
K : 0.0471

JPG - Conversão de cores

Converte RGB para Y (*Luminance*), Cb (*Chroma Blue*), Cr (*Chroma Red*).



Experimentos psico-visuais: olho mais sensível a luminância do que crominância.

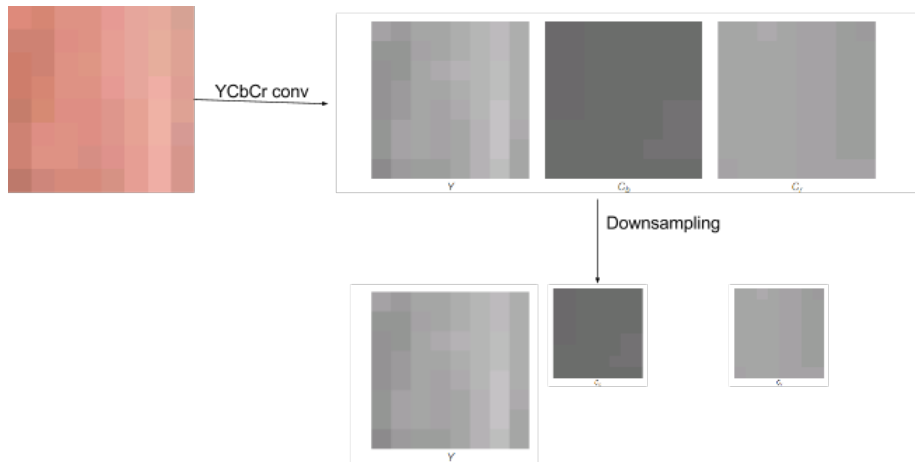
Liberdade para manipular canais Cb e Cr.

Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem**
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão
- 8 Conclusões

Subamostragem

Canais Cb e Cr reduzidos para 1/4 do tamanho original (*downsampling*).



Subamostragem

JPG e o “vazamento de cores” nas bordas.



Canais de crominância (cor) recebem a média de cada grupo com 4 *pixels*.

Podem vazar através de bordas mais agudas.

Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos**
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão
- 8 Conclusões

Divisão em Blocos

Divisão em blocos 8×8 de *pixels*.

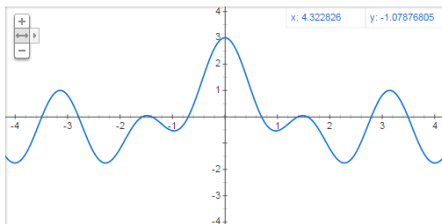
Bloco pequeno: pouca variância, auto-similaridade local.

Outline

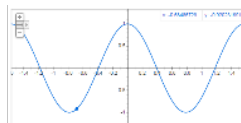
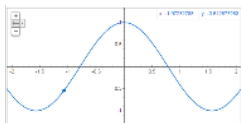
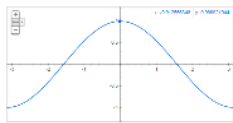
- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno**
- 6 Quantização
- 7 Compressão
- 8 Conclusões

Transformada do Cosseno

Sinais podem ser reconstruídos por soma de cossenos (semelhante a Fourier). Exemplo:

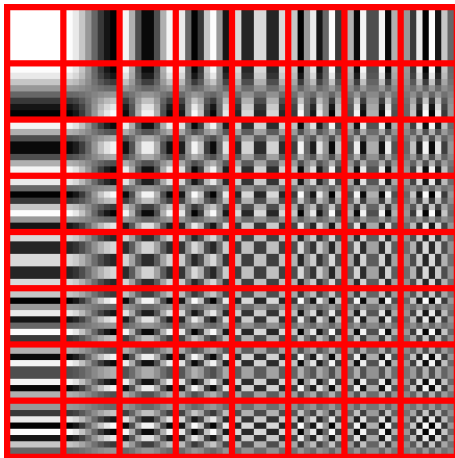


pode ser $\cos(x) + \cos(2x) + \cos(4x)$:



Transformada do Cosseno

Ideia similar em 2D usando uma *base de cossenos*.



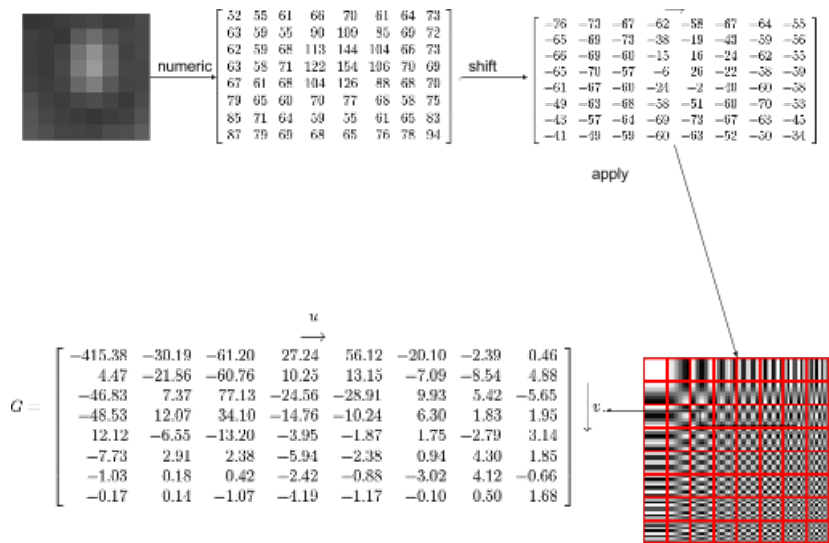
Transformada do Cosseno

Cada bloco 8×8 transformado em uma soma ponderada de cossenos.

Coeficientes (pesos) dados por

$$G(u, v) = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right].$$

Transformada do Cosseno



Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização**
- 7 Compressão
- 8 Conclusões

Quantização

Apenas a matriz G (dos pesos) será usada daqui para frente.

Troca de inteiros de 1 *byte* por valores reais (*float*) “incha” os dados (1 *byte* trocado por 4).

Quantização é a volta para o espaço dos inteiros de 1 *byte* ($[0,255]$ ou $[-127,127]$).

Poderia fazer normalização simples:

$$\frac{x - \min}{\max - \min} * 255,$$

mas geraria uma distribuição irregular de valores e perda de precisão.

Quantização

JPG então usa uma matriz de fatores de quantização para escalar os valores.

Matriz Q_l para o canal Y e Q_c para os canais Cb e Cr :

$$Q_l = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

A fórmula para a matriz quantizada B é

$$B_{j,k} = \text{round} \left(\frac{G_{j,k}}{Q_{j,k}} \right), \quad j = 0, 1, 2, \dots, 7; k = 0, 1, 2, \dots, 7.$$

Quantização

Supondo que G no exemplo anterior seja de um bloco do canal Y , temos, por exemplo:

$$B[0,0] = \text{round} \left(\frac{-415.37}{16} \right) = -26.$$

Ao final:

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note o grande número de zeros facilitando a compressão.

Matriz Q pode ser alterada e é ela que controla o parâmetro “qualidade” do JPG: qualidade maior gera matriz B com menos zeros.

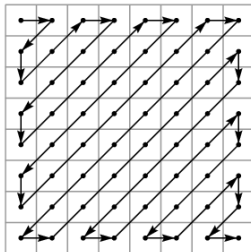
Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão**
- 8 Conclusões

Compressão

Do canto superior esquerdo para o inferior direito aumenta o número de zeros em B .

Faz-se então uma varredura em ziguezague:



resultando no nosso caso em

$-26, -3, 0, -3, -2, -6, 2, -4, 1, -3, 1, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0,$
 $-1, -1, 0,$
 $0, 0, 0, 0, 0, 0, 0, 0$

Compressão

Técnicas de compressão *lossless* (sem perda) são então aplicadas.

Exemplo é *Run-Length Encoding* (RLE).

Marca caractere repetido e número de vezes. Ex.:

“AAAAHHHHHHHHHHHHHHH” é comprimido para “5A14H”.

Este algoritmo também é usado em formatos *lossless* como BMP, TIFF e PCX.

Outline

- 1 Formato JPG
- 2 JPG - Conversão de cores
- 3 Subamostragem
- 4 Divisão em Blocos
- 5 Transformada do Cosseno
- 6 Quantização
- 7 Compressão
- 8 Conclusões**

Conclusões

Escolha do parâmetro de qualidade adequado é fundamental.

Artefatos surgem em blocos (devido à compressão em blocos) ou em aneis.

Uso de blocos impede compressão de regiões redundantes em blocos diferentes (novo padrão WebP tenta resolver isso).