

Tecnologias Multimédia

Manuela Pereira

Departamento de Informática Universidade da Beira Interior

2017

Resumo

- 1 Introdução
- 2 Digitalização
- 3 Operações sobre Imagem
- 4 Operações sobre Áudio
- 5 Cor e Codificação da cor
- 6 Imagem Bitmap vs. Imagem Vetorial
- 7 Representação de Áudio
- 8 Comprimir
- 9 Compressão de Imagem
- 10 Compressão de Imagem
- 11 Compressão de Vídeo

Multimédia

- Digitalização.
- Compressão.
- Sincronização.
- Interação.

Digitalizar

Representação Digital dos Dados

- Texto.
- Som.
- Imagem.
- Vídeo.

Digitalizar

Processo de digitalização

- ① Amostragem.
- ② Quantificação ou quantização.
- ③ Codificação.

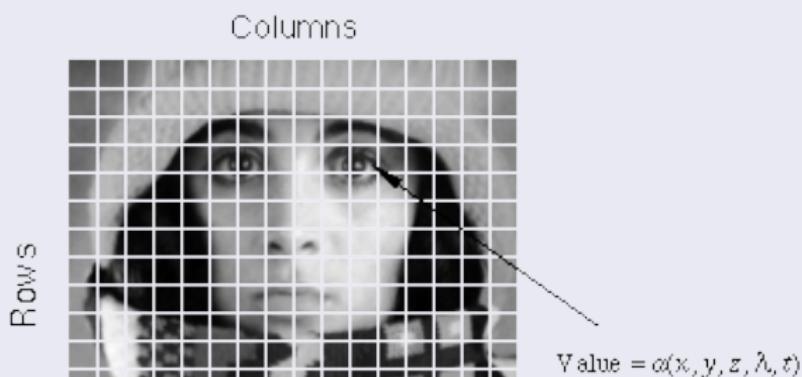
Digitalizar

Processo de digitalização

- ① Amostragem: Transformação do contínuo ao discreto.
- ② Quantização ou Quantificação: Definição e atribuição de uma escala de medição.
- ③ Codificação: Correspondência entre um conjunto de pontos na escala de medição e o conjunto de valores código (aqui, sequências binárias de tamanho uniforme). Esta codificação tem de ser uma função injectiva.

Processo de digitalização Imagen (bitmap)

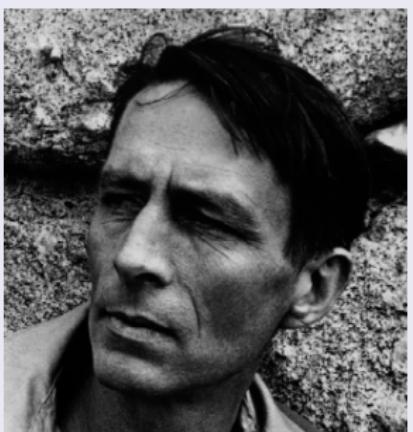
Amostragem.



Processo de digitalização.

Imagen (bitmap)

Amostragem.



Importante

Relação: Número de amostras - qualidade - espaço ocupado e tempo de tratamento.

Processo de digitalização Imagen (bitmap)

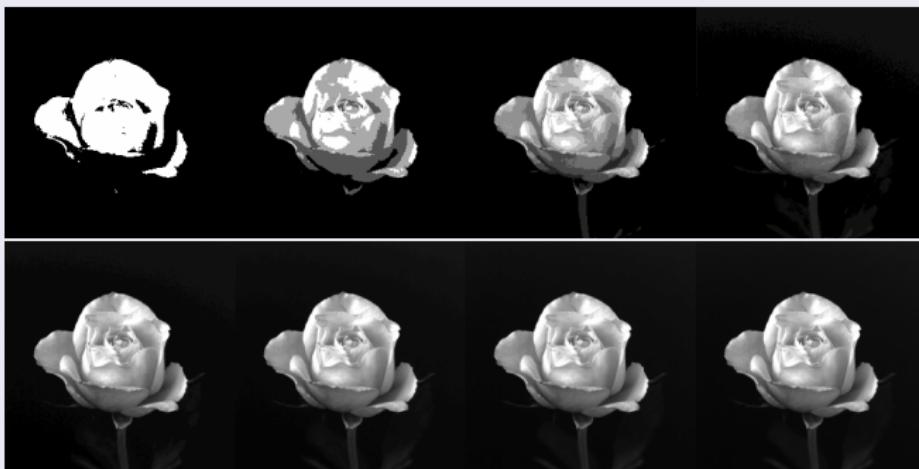
Quantização.



Processo de digitalização.

Imagen (bitmap)

Quantização.



Importante

Relação: Profundidade de quantificação - qualidade - espaço ocupado e tempo de tratamento.

Processo de digitalização

Imagen (bitmap)

Codificação.

- 1 Designar as diferentes graduações, isto é, dar a cada uma um nome que a distinga de todas as outras.
- 2 Não implica perdas.

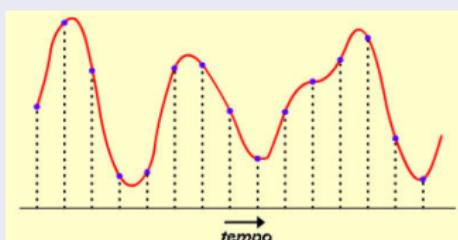
Importante

A frequência de amostragem e a profundidade de quantificação são os dois fatores que determinam a qualidade de um processo de digitalização.

Processo de digitalização.

Som

Amostragem.

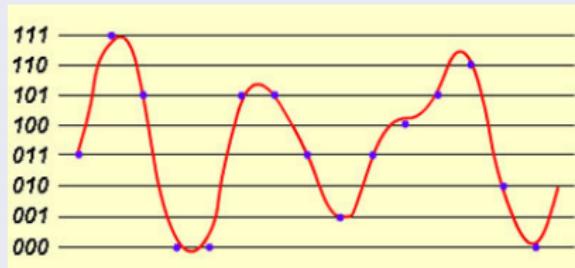


- A imagem elétrica da onda sonora vai ser partida em amostras segundo o eixo do tempo.
- O número de amostras por segundo define a frequência da amostragem.
- Exemplo: telefone numérico: 8KHz (i.e. 8000 amostras por segundo); CD de música: 44.1 KHz (i.e. 44100 amostras por segundo).

Processo de digitalização

Som

Quantização e Codificação.



- Exemplo: CD de música: recorre a uma codificação sobre 16 bits permitindo distinguir $2^{16} = 65536$ níveis diferentes de amplitude.

Processo de digitalização

Texto

- 1832 Código de Morse (comprimento variável).
- 1874 Código de Baudot (comprimento constante de 5 bits, 2 caracteres especiais).
- 1963 Código ASCII (7 bits + 1 para controlo de erro).
- Código ASCII estendido: são utilizados os oito bits para determinar cada carater. Já engloba os caracteres acentuados e um maior número de caracteres especiais e de controlo. É este que é utilizado atualmente nos nossos computadores.
- 1991 Unicode (16 bits -> $2^{16} = 65536$ símbolos): código muito mais rico, integrando outros alfabetos além do latim (árabe, grego, hebraico, etc.).

Processo de digitalização

Texto

Amostragem, Quantização e Codificação.

No caso do texto a amostragem já está realizada pelo alfabeto e a profundidade de quantificação imposta pela quantidade de símbolos a distinguir. Assim no caso do texto a digitalização limita-se a uma codificação.

Digitalização

Conclusão

Sinal analógico <-> Sinal digital

Operações de imagem

- Edição de imagem;
- Operações sobre pontos;
- Transformações geométricas
- Operações de filtragem
- Operações de composição

Operações de imagem

Edição de imagem

Estas operações permitem alterar pixels da imagem. Por exemplo:

- Retocar imagens;
- Airbrushing;
- Texturing;
- Cortar, copiar e colar blocos de pixels;
- Cortar, copiar e colar grupos de pixels;

Exercício

Crie uma imagem B contendo o bloco central da imagem original A;

Operações de imagem

Edição de imagem

Exercício: Crie uma imagem B contendo o bloco central da imagem original A;

```
A = imread('Rosa1024.pgm');  
B = A(257:768,257:768);
```

Operações de imagem

Operações sobre pontos

Por aplicação de uma função a cada pixel produzindo um novo valor para o pixel. Por exemplo:

- thresholding;
- correção de cores, do brilho ou do contraste;
- atenuação de certas cores;

Exercícios

Crie uma imagem B que seja o negativo da imagem original A;
Crie uma imagem B a partir da imagem original A, mas apenas com branco e preto;

Operações de imagem

Operações sobre pontos

Exercício: Crie uma imagem B que seja o negativo da imagem original A;

```
A = imread('Rosa1024.pgm');
```

```
B = 255-A;
```

Operações de imagem

Operações sobre pontos

Exercício: Crie uma imagem B a partir da imagem original A, mas apenas com branco e preto;

```
A = imread('Rosa1024.pgm');
```

```
B = A > 128;
```

Operações de imagem Transformações geométricas

- Rotação;
- escala;
- inversão

Exercícios

Crie uma imagem B contendo a imagem original A invertida horizontalmente;

Crie uma imagem B contendo a imagem original A reduzida para metade;

Operações de imagem Transformações geométricas

Exercício: Crie uma imagem B contendo a imagem original A invertida horizontalmente;

```
A = imread('Rosa1024.pgm');  
B = A(end:-1:1,:);
```

Operações de imagem Transformações geométricas

Exercício: Crie uma imagem B contendo a imagem original A reduzida para metade;

```
A = imread('Rosa1024.pgm');  
B = A(1:2:end, 1:2:end);
```

Operações de imagem

Operações de filtragem

Aplicação de uma função a cada pixel, mas que usa informação dos vizinhos do pixel. Por exemplo:

- blur;
- sharpen;
- distorção.

Operações de imagem

Operações de filtragem

Exemplos de filtros: gaussian; laplacian; sobel; prewitt; log; average; etc

Blur

```
A = imread('Rosa1024.pgm');
PSF = fspecial('gaussian',12,10);
Blurred = imfilter(A,PSF,'symmetric','conv');
figure(1);
subplot(1,2,1);imshow(A);
subplot(1,2,2);imshow(Blurred);
```

Operações de imagem

Operações de composição.

Composição de grupos de pixeis de duas ou mais imagens. A imagem final obtida é o resultado de uma soma de várias imagens.

Operações de Áudio

Edição de Áudio

- Cortar, copiar e colar segmentos;
- Junção de segmentos diferentes (cross-fade);

Exercício

Crie um audio B contendo o tempo entre os segundos 5 e 8 do áudio A;

Operações de Áudio

Edição de Áudio

Exercício: Crie um audio B contendo o tempo entre os segundos 5 e 8 do áudio A;

```
[A,Fs] = audioread('bass.wav');
```

```
B = A(Fs*5+1:Fs*8);
```

```
sound(B,Fs);
```

Operações de Áudio

Filtragem

- Redução de ruído;
- atrasos (adição de ecos ou reverberações);
- equalização (enfatizar, reduzir, alterar várias bandas de frequências);
- normalizar (de forma que o pico não ultrapasse o máximo permitido);
- compressão/expansão temporal;
- alteração da tonalidade, sem alteração da duração.
- ...

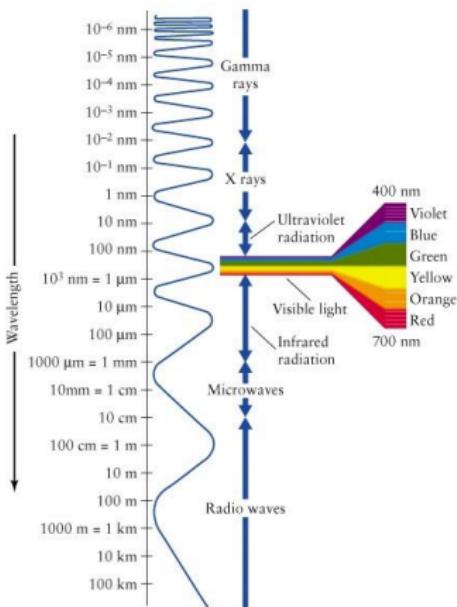
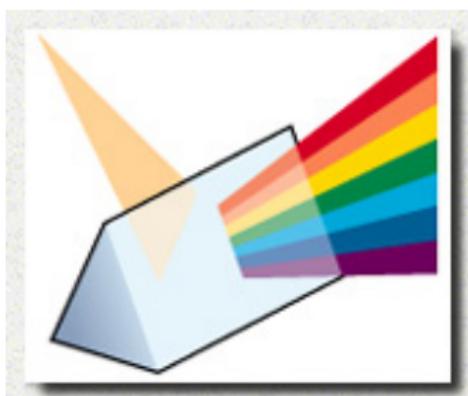
Cor e Codificação da cor

- O olho e a luz.
- As cores.
- Síntese aditiva e síntese subtractiva.
- Os modelos de cor.
- Gamas e paletes de cor.

A luz e a Cor

- Luz: onda eletromagnética.
- Luz visível: entre os 400nm e os 700 nm.
- Cor: comprimento de onda da luz.
- Luz branca: todas as cores do arco íris.

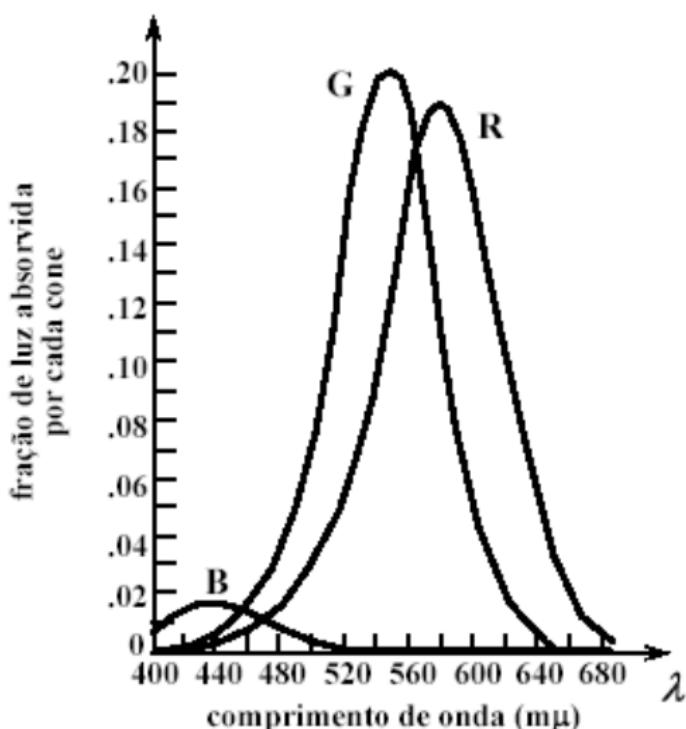
A luz e a Cor



O olho e a Cor

- 1801 Thomas Young: três nervos na retina que são estimulados respectivamente pelo vermelho, pelo verde e pelo azul.
- 1851 Hermann Ludwig Ferdinand von Helmholtz: a síntese das três cores primárias que permite ao cérebro humano realizar todas as matizes de cores do espectro da natureza.
- Atualmente: cones; bastonetes.

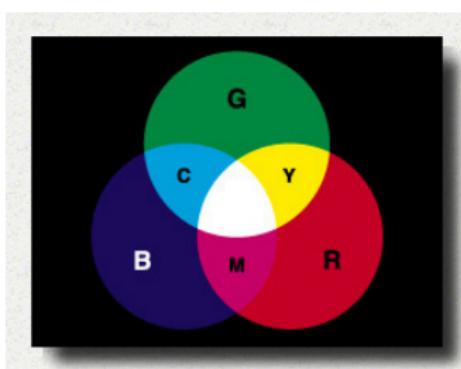
O olho e a Cor



Os modelos de cor

Modelo RGB

- Três cores primárias: vermelho, verde e azul.
- Ecrãs de vídeo (computador, televisão, ...) que restituem os tons pela adição das três cores primárias.
- Um tom é obtido através de uma combinação linear $\theta = \rho \times R + \gamma \times G + \beta \times B$, com $(\rho, \gamma, \beta) \in [0, 1]^3$.



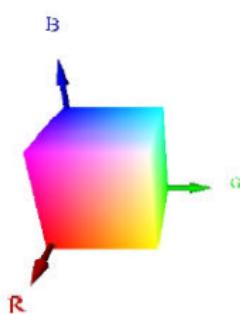
Os modelos de cor

Modelo RGB

- Considerando uma base de um byte para cada cor podemos modelar $2^8 \times 2^8 \times 2^8 = 2^{24} = 16777216$ cores. Uma imagem em modo 16 milhões de cores é conhecida por imagem em "true color".

Os modelos de cor

Modelo RGB



- Origem: ponto onde os três valores são iguais a 0 e que corresponde ao preto.
- Vértice oposto: ponto onde os valores das três coordenadas são máximos e por isso corresponde ao branco.
- Vértices do cubo equidistantes de duas cores primárias: encontram-se as cores secundárias: amarelo, magenta e cião.

Este modelo é dependente dos periféricos utilizados na aquisição e na representação.

Os modelos de cor

Modelo RGB

Exercício

Visualize a informação referente a cada um dos canais, para o caso da imagem baboon;

```
Im = imread('baboon.png');
```

Os modelos de cor

Modelo RGB

Exercício (Matlab): Visualize a informação referente a cada um dos canais, para o caso da imagem baboon;

```
Im = imread('baboon.png');

figure;
subplot(1,4,1); imshow(Im); title('RGB information');
subplot(1,4,2); imshow(Im(:,:,1)); title('Red information');
subplot(1,4,3); imshow(Im(:,:,2)); title('Green information');
subplot(1,4,4); imshow(Im(:,:,3)); title('Blue information');
```

Os modelos de cor

Modelo RGB

Exercício

Qual será a informação referente a cada um dos canais, para o caso do cubo do modelo RGB?

```
Im = imread('Fig6_08.png');
```

Os modelos de cor

Modelo RGB

Exercício (Matlab): Qual será a informação referente a cada um dos canais, para o caso do cubo do modelo RGB?;

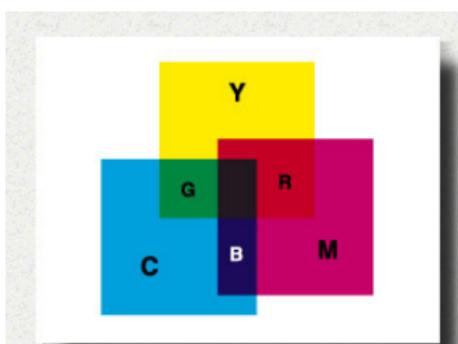
```
Im = imread('Fig6_08.png');
```

```
figure;  
subplot(1,4,1); imshow(Im); title('RGB information');  
subplot(1,4,2); imshow(Im(:,:,1)); title('Red information')  
subplot(1,4,3); imshow(Im(:,:,2)); title('Green information')  
subplot(1,4,4); imshow(Im(:,:,3)); title('Blue information')
```

Os modelos de cor

Modelo CMY

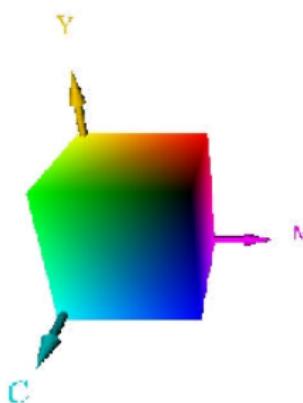
- Próprio aos objetos que refletem luz: para obter as matizes desejadas procede-se por subtração das cores refletidas por uma superfície branca.
- Seja uma fonte luminosa branca emitindo em direção de três filtros cião, magenta e amarelo:
 - Cião(verde, azul): opõe-se à passagem do vermelho.
 - Magenta (vermelho, azul): opõe-se à passagem do verde.
 - Amarelo (verde, vermelho): opõe-se à passagem da cor azul.



Os modelos de cor

Modelo CMY

- Impressão sobre suportes físicos.
- Cada cor é obtida modificando as proporções na mistura das cores primárias.
- O valor máximo em todas as cores dá-nos o preto.
- As três cores nulas temos o branco.



Os modelos de cor

Modelo CMYK

- A sobreposição das três cores cião, magenta e amarela, com saturações máximas, não é suficiente, geralmente, para absorver todo o espectro luminoso e o preto é assim difícil de produzir.
- Um quarto plano é introduzido no modelo, correspondendo à cor preta.

$$K = \min\{C, M, Y\}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \Rightarrow \begin{bmatrix} C - K \\ M - K \\ Y - K \end{bmatrix}$$

Os modelos de cor

Modelo CMY

Exercício (Matlab):

Qual será o negativo do cubo do modelo RGB?

```
Im = imread('Fig6_08.png');
```

Os modelos de cor

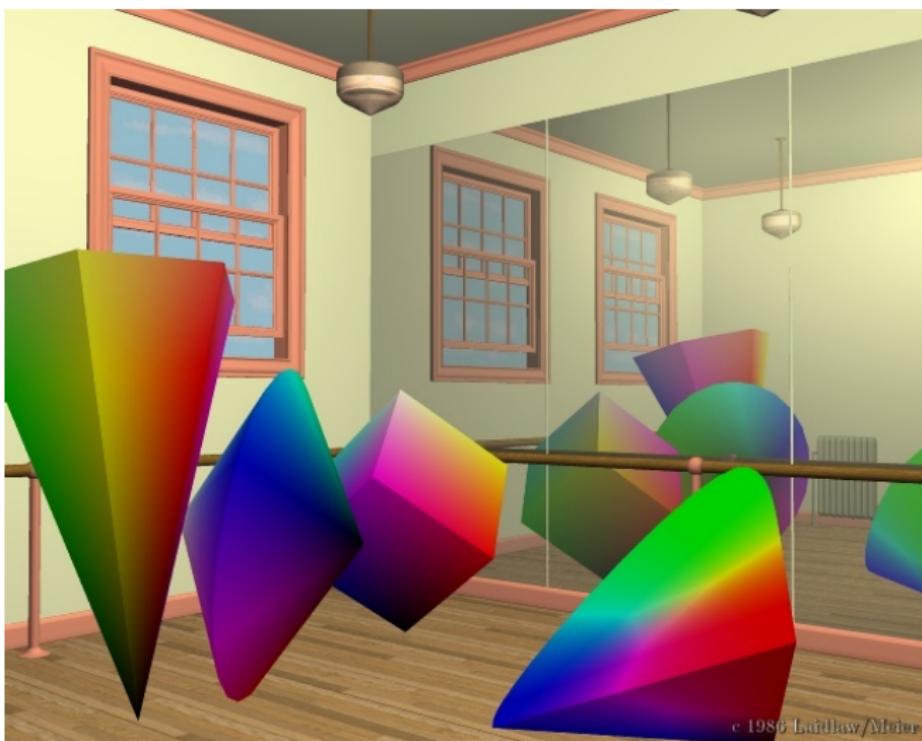
Modelo CMY

Exercício (Matlab): Qual será o negativo do cubo do modelo RGB?

```
Im = imread('Fig6_08.png');

figure;
subplot(1,2,1); imshow(Im); title('RGB cube');
subplot(1,2,2); imshow( 255-Im); title('CMY cube');
```

Os modelos de cor



Os modelos de cor

Modelo HSV (Hue, Saturation, Value)

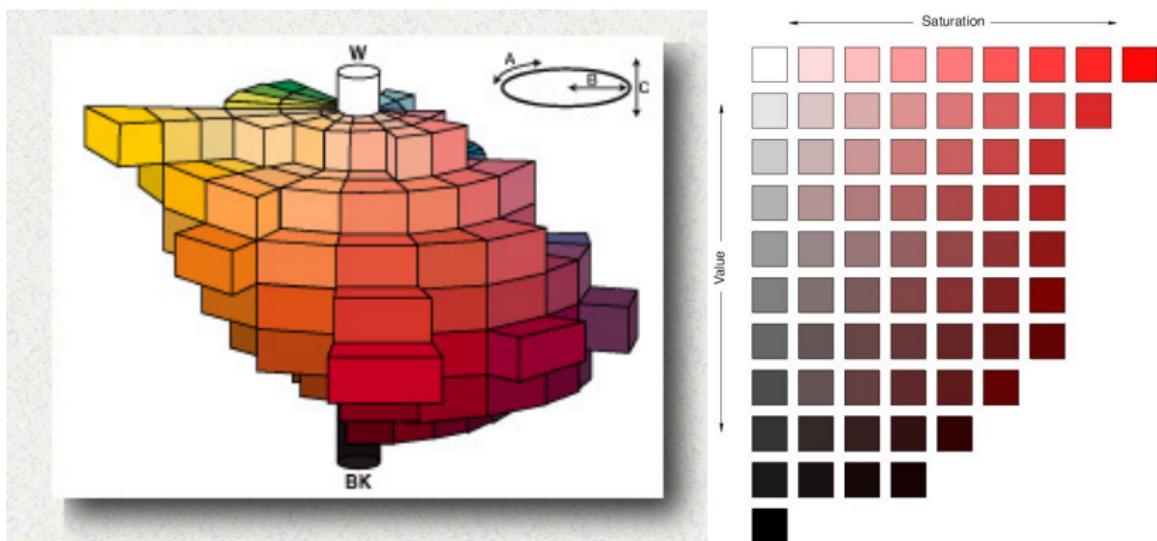
H - Hue Representa o tom.

S - Saturation S a saturação da cor. Quanto mais baixa for a saturação, mais cinzenta é a cor. Quando a saturação é zero a cor é cinzenta.

V - Value Representando a intensidade luminosa. O brilho é determinado pelo grau de refletividade da superfície física que recebe a luz. Quanto maior for o brilho mais clara é a cor.

Os modelos de cor

Modelo HSV (Hue, Saturation, Value)



Os modelos de cor

Modelo HSV

Exercício (Matlab):

Visualize cada um dos canais H, S e V da imagem HSV.png.

```
Im = imread('hsv.png');
```

```
ImHSV = rgb2hsv(Im);
```

Os modelos de cor

Modelo HSV

Exercício (Matlab): Visualize cada um dos canais H, S e V da imagem HSV.png.

```
Im = imread('hsv.png');
ImHSV = rgb2hsv(Im);
figure;
subplot(1,4,1); imshow(Im); title('HSV model');
subplot(1,4,2); imshow(ImHSV(:,:,1)); title('Hue')
subplot(1,4,3); imshow(ImHSV(:,:,2)); title('Saturation')
subplot(1,4,4); imshow(ImHSV(:,:,3)); title('Value')
```

Os modelos de cor

Modelo HSV

Exercício (Matlab):

Altere os tons da imagem images16.jpg que se encontrem no intervalo [0.5, 0.7[.

```
Im = imread('hsv.png');
```

```
ImHSV = rgb2hsv(Im);
```

Os modelos de cor

Modelo HSV

Exercício (Matlab): Altere os tons da imagem images16.jpg que se encontrem no intervalo [0.5, 0.7[.

```
I1 = imread('images16.jpg');
hsv = rgb2hsv(I1);
tmp = hsv(:,:,1) >= 0.5 & hsv(:,:,1) < 0.7;
figure; subplot(4,3,1); imshow(I1);
for ind = 1:11
    hsv(:,:,1) = hsv(:,:,1) + tmp*0.1;
    tmp2 = hsv(:,:,1)>1;
    hsv(:,:,1) = hsv(:,:,1) - tmp2;
    rgbsv = hsv2rgb(hsv);
    subplot(4,3,ind+1); imshow(rgbsv);
end
```

Os modelos de cor

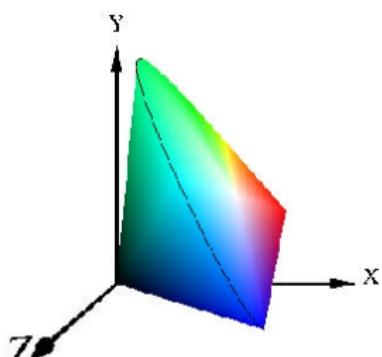
Modelo CIE

Os modelos de cor CIE (Comission Internationale de l'Eclairage) são uma família de modelos matemáticos que descrevem a cor em termos de tom (hue), brilho (value) e saturação (saturation). Os modelos de cor CIE incluem os modelos CIE XYZ, CIELAB, CIELUV.

Os modelos de cor

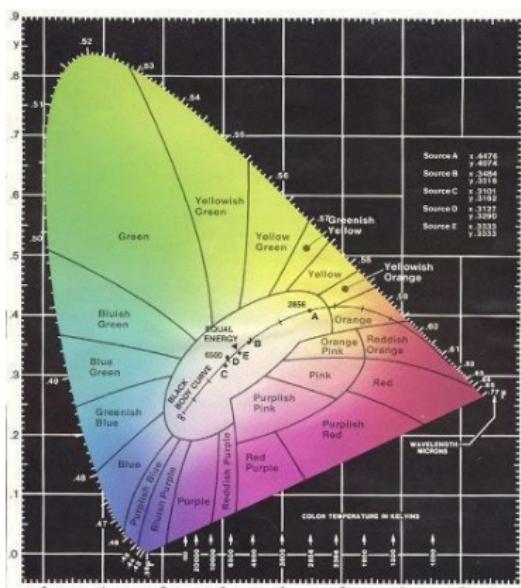
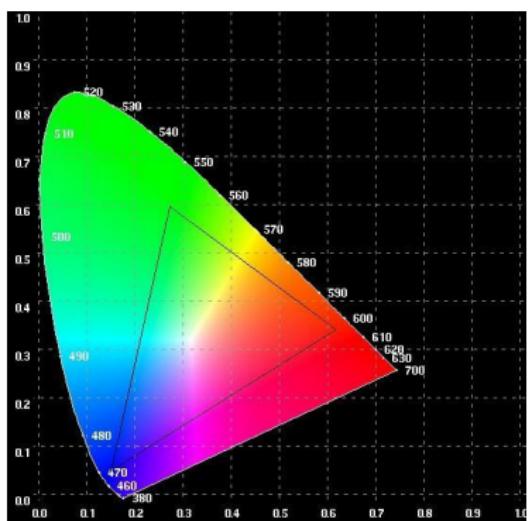
Modelo CIE XYZ

Este modelo apareceu em 1931 e define um espaço de cor no qual se inscrevem todas as cores do espectro visível. Cada uma das cores é definida pela mistura em proporções específicas de três cores primárias X, Y e Z.



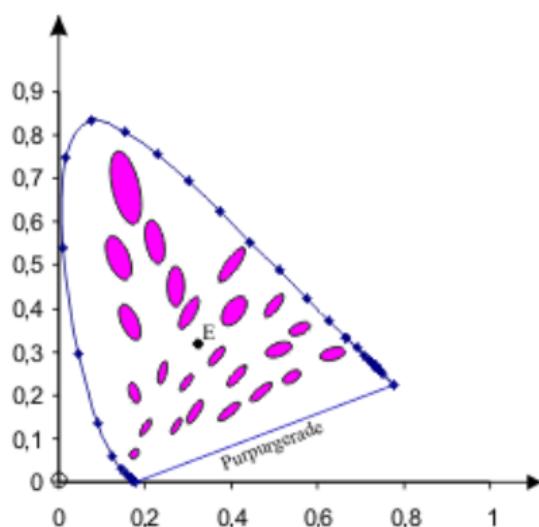
Os modelos de cor

Modelo CIE XYZ



Os modelos de cor

Não uniformidade perceptual do espaço CIE XYZ



Os modelos de cor

Modelo CIE LAB

Como vimos atrás a percepção da cor não obedece a um mecanismo linear. A CIE tomou em conta esta não linearização da percepção visual humana propondo em 1960 uma escala uniforme de cor e depois estandardizando em 1976 a luminosidade (L^*).

A partir deste parâmetro a CIE normalizou dois sistemas: CIE LUV e CIE LAB. O CIELUV é geralmente usado para calibrar monitores.

Os modelos de cor

Modelo CIE LAB

Quanto ao espaço LAB ele apresenta o interesse de quantificar as cores segundo as classificações realizadas pelo pintor Albert H. Munsell e apresentadas no seu famoso atlas. Este espaço é o mais uniforme dos espaços CIE.



Os modelos de cor

Modelo CIE XYZ

Normalizando os três valores XYZ podemos obter dois valores cromáticos:

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad (1)$$

Claro que está definido outro valor cromático, mas este é redundante. Realmente, sabendo que $x + y + z = 1$, é suficiente saber dois valores para poder representar o terceiro como: $z = 1 - x - y$.

Os modelos de cor

Modelo CIE xyY

Para caracterizar a luminância, CIE escolheu por convenção a coordenada Y: uma cor é sempre definida pelas suas coordenadas x, y e Y.

Transformação XYZ - xyY

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad Y = Y \quad (2)$$

Transformação xyY - XYZ

$$X = \frac{x}{y} Y \quad Y = Y \quad Z = (1 - x - y) \frac{Y}{y} \quad (3)$$

Os modelos de cor

Modelos que separam luminância de crominância

Nestes modelos é usado um plano Y para a luminância e dois planos para a crominância. A luminância é a soma das três componentes básicas (R, G, B) moduladas seguindo a sensibilidade da nossa percepção visual.

Para calcular a luminância a fórmula utilizada é a seguinte:

$$Y = 0.2999R + 0.58G + 0.114B.$$

Os modelos de cor

Modelos que separam luminância de crominância

Os coeficientes de ponderação de cada cor R, G e B são calculados usando a relação seguinte:

$$\text{Coeff}_{cor_X} = \frac{\text{sensibilidade}_{cor_X}}{\text{sensibilidade}_{cor_R} + \text{sensibilidade}_{cor_G} + \text{sensibilidade}_{cor_B}}$$

Por exemplo:

$$\text{Coeff}_{cor_R} = \frac{0.5}{1 + 0.5 + 0.2} = 0.2999$$

$$\text{Coeff}_{cor_G} = \frac{1}{1 + 0.5 + 0.2} = 0.58$$

$$\text{Coeff}_{cor_B} = \frac{0.2}{1 + 0.5 + 0.2} = 0.114$$

Os modelos de cor

Modelos que separam luminância de crominância

- NTSC - 1953, Americano.

$$I = 0.74(R - Y) - 0.27(B - Y)$$

$$Q = 0.48(R - Y) - 0.49(B - Y).$$

- PAL - 1967 Alemão.

$$U = 0.147R - 0.289G + 0.436B = 0.493(B - Y)$$

$$V = 0.615R - 0.515G - 0.100B = 0.877(R - Y).$$

- SECAM - 1967 Francês.

$$D_r = -1.9(R - Y)$$

$$D_b = 1.5(B - Y).$$

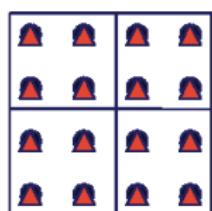
- CCIR (Imagem fixa)

$$C_r = 128 + 128 \frac{1}{0.701}(R - Y)$$

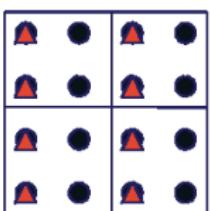
$$C_b = 128 + 128 \frac{1}{0.886}(B - Y).$$

Compressão de Imagem

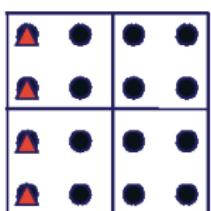
Transformada de cor



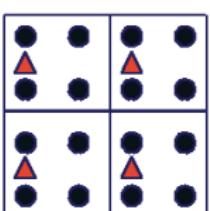
4:4:4
For every 2x2 Y Pixels
4 Cb & 4 Cr Pixel
(No subsampling)



4:2:2
For every 2x2 Y Pixels
2 Cb & 2 Cr Pixel
(Subsampling by 2:1 horizontally only)



4:1:1
For every 4x1 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 4:1 horizontally only)



4:2:0
For every 2x2 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 2:1 both horizontally and vertically)

● Y Pixel

▲ Cb and Cr Pixel

Os modelos de cor

Modelo HSV

Exercício (Matlab):

Vamos verificar se existem alterações com a sub-amostragem da crominância.

```
Im = imread('baboon.png');
```

```
Icbcr = rgb2ycbcr(Im);
```

Os modelos de cor

Modelo HSV

Exercício (Matlab): Altere os tons da imagem images16.jpg que se encontrem no intervalo [0.5, 0.7[.

```
Im = imread('baboon.png');
```

```
% Code
```

```
YCbCr = rgb2ycbcr(Im);
```

```
Y = Icbc(:,:,1);
```

```
Cb = Icbc(:,:,2);
```

```
Cr = Icbc(:,:,3);
```

```
figure; imshow(Y);
```

```
Cbhalf = imresize(Cb,0.5);
```

```
Crhalf = imresize(Cr,0.5);
```

Os modelos de cor

Modelo HSV

Exercício (Matlab): Altere os tons da imagem images16.jpg que se encontrem no intervalo [0.5, 0.7[.

```
% DecCode  
Cbdouble = imresize(Cb,2);  
Crdouble = imresize(Cr,2);  
Inew = Im;  
Inew(:,:,2) = Cbdouble;  
Inew(:,:,3) = Crdouble;  
rgbnew = ycbcr2rgb(Inew);  
subplot(1,2,1); imshow(Im);  
subplot(1,2,2); imshow(rgbnew);
```

Os modelos de cor

Gamas e paletes de cor

- Chamamos paletes de cores ao sub-conjunto das gamas disponíveis por um sistema particular.
- Uma imagem original RGB em milhões de cores terá de ser traduzida utilizando somente as cores disponíveis na paleta do sistema de visualização.
- As cores da imagem original que não estão disponíveis na paleta de cores do sistema terão de ser aproximadas.
- Esta técnica, de construção de tramas, consiste em criar pixéis contíguos com a ajuda das diferentes cores disponíveis na paleta para dar, por confusão, a impressão visual da cor de origem.

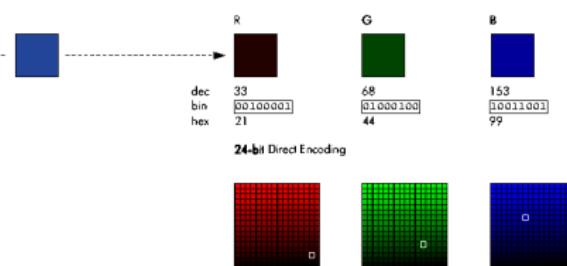
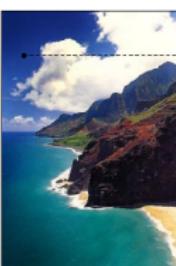
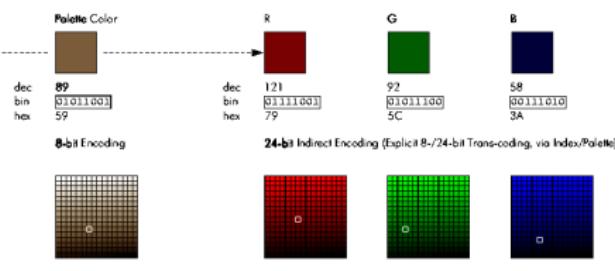
Os modelos de cor

Imagens em cor indexada

- Definem-se 256 cores cada uma delas codificada sobre 8 bits por cor primária.
- Cada cor selecionada é designada por um índice numa tabela de 256 cores.
- Assim uma imagem em 256 cores é chamada em cor indexada.
- Numa paleta cada parâmetro é codificado numa tabela da paleta sobre um número de bits idêntico ao utilizado na gama das cores. Para uma paleta de 256 valores, é assim necessário uma tabela de dimensão 256×24 . Dispomos assim de uma paleta de 256 cores sobre os 16 milhões de cores disponíveis.

Os modelos de cor

Imagens em cor indexada



Os modelos de cor

Imagens em cor indexada

- paleta exata: são registadas as cores exatas do original;
- paleta sistema: paleta por defeito de um sistema de exploração;
- paleta uniforme: obtidas por tomada uniforme de amostras de cores no cubo RGB; o número total de cores obtidas será igual ao cubo do número de tomas (com cinco níveis de tomas, por exemplo, obtemos $5^3 = 125$ cores);
- paleta web: constituída de 216 cores (seis intervalos, $6^3 = 216$).

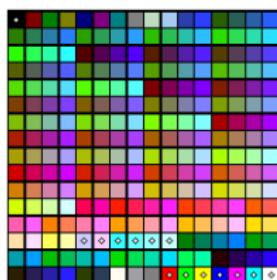
Os modelos de cor

Imagens em cor indexada

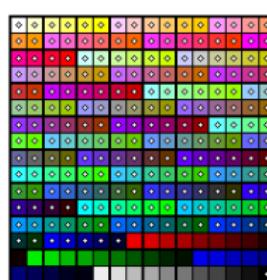
- paleta perceptiva: favorece, entre as cores presentes na imagem, aquelas às quais o olho humano é mais sensível;
- paleta seletiva: similar à paleta **perceptiva** valorizando as zonas de cores mais importantes e preservando as cores **web**;
- paleta adaptativa: favorece os **tons dominantes** na imagem; uma imagem RGB na qual dominam as matizes de verde e de azul conduzirá à criação de uma paleta composta principalmente de verde e azul;

Os modelos de cor

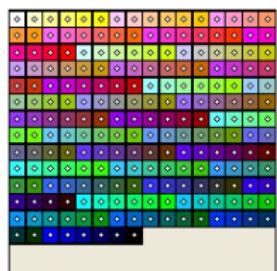
Imagens em cor indexada



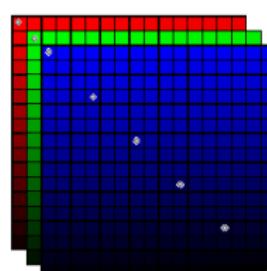
Paleta do Windows;



Paleta do Macintosh



Paleta do WEB;



3 Canais RGB

Os modelos de cor

Conclusões

A representação numérica da cor é um tema muito vasto do qual acabamos de ver uma pequena parte.

Imagen Bitmap vs. Imagen Vetorial

- Imagem bitmap: o ficheiro é constituído pelos dados da cor de cada ponto;
- Imagem vetorial: o ficheiro contem a descrição matemática das figuras elementares que o constitui.

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap

- Conjunto de pontos alinhados horizontal e verticalmente como linhas e colunas de uma matriz.
- Um pixel é o elemento mais pequeno da imagem.
- Podemos dividir as imagens bitmap em várias categorias:
 - imagens em dois níveis (ou preto e branco), com 1 bpp;
 - imagem em níveis de cinzento, com 8 bpp;
 - imagens em cor indexada, com 8 bpp;
 - imagens "true color", com 24 bpp (+ componente alpha).
- Todas as imagens reproduzidas num ecrã são imagens bitmap, e o mesmo acontece para as imagens obtidas através de um scanner ou de um aparelho fotográfico numérico.

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap

- Sabendo a largura e a altura, em pixels, de uma imagem e a profundidade de quantificação podemos calcular o tamanho do ficheiro de uma imagem bitmap (não comprimido) como:

$$T(KB) = \frac{\text{largura (pixel)} \times \text{altura (pixel)} \times \text{profundidade (bits/pixel)}}{8(\text{bits/byte}) \times 1024(\text{bytes/KB})}$$

- Resolução (ppp): número de pontos digitalizados, afixados ou imprimidos por unidade de largura.

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - Entrelaçamento

- As imagens bitmap podem permitir uma visualização progressiva através da técnica de entrelaçamento. Mesmo com a redução de tamanho permitida pela indexação de cores uma imagem pode demorar um tempo considerável até poder ser visualizada.
- A técnica de entrelaçamento consiste em reordenar as linhas das imagens, organizando-as em vários grupos.
- A transmissão da imagem grupo a grupo permite que o utilizador comece a formar uma ideia da imagem após apenas algumas linhas terem sido transmitidas.
- Esta característica pode ser observada durante o carregamento de imagens transmitidas pela web quando o fluxo de dados é baixo.

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - Vantagens

- São capazes de produzir graduações de nuances e de cores muito finas.
- São constituídas por pixels, o que permite o tratamento ponto a ponto e são adaptadas a representação sobre ecrã ou impressão;
- São directamente guardadas na memória, logo são representadas muito rapidamente no ecrã (muito mais depressa do que as imagens vectoriais que, como veremos de seguida, devem ser reconstituídas).

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - Desvantagens

- A qualidade destas imagens está directamente dependente ao material de aquisição e de reprodução (por exemplo a resolução do ecrã, da impressora, do scanner);
- O facto de serem imagens de pontos exige uma grande quantidade de espaço em memória (grande quantidade de pontos cujas características de cor são definidas individualmente).

Imagen Bitmap vs. Imagen Vetorial

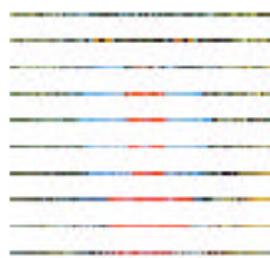
Imagen bitmap - GIF

- Concebido em 1980 pela Compuserve ;
- Utiliza compressão sem perdas LZW;
- Profundidade de pixel não superior a 8 bits;
- Optimizado para compressão de imagens contendo poucas cores diferentes e apresentando grandes quantidades de pixels da mesma cor (logos, esquemas, diagramas a preto e branco, etc);
- É adaptado a imagens com forte contraste e a texto mas não à fotografia realista;
- A variante GIF87a aceita entrelaçamento e a versão 89a junta ainda a possibilidade de transparência e animação;
- Permite entrelaçamento.

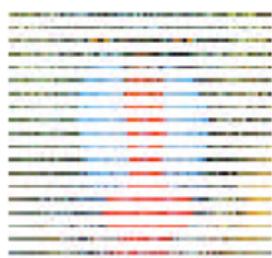
Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - GIF

- Entrelaçamento em 4 passagens pelas linhas das imagens:
 - 1 0, 8, 16,
 - 2 4, 12, 20, ...
 - 3 2, 6, 20, 24, ...
 - 4 1, 3, 5, 7,



16 s



31 s



62 s



123 s

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - PNG

- Formato mais recente que reúne as principais qualidades dos seus predecessores, eliminando a maioria dos defeitos;
- Utiliza compressão sem perdas;
- Permite codificações até 48 bpp;
- Atinge níveis de compressão próximos dos JPEG (mas sem perdas);
- Permite transparência por canal alpha;
- Este formato é livre (não utiliza qualquer algoritmo de domínio privado);

Imagen Bitmap vs. Imagen Vetorial

Imagen bitmap - PNG

- Permite entrelaçamento (Adam7 - representação da imagem em 7 passagens);

1 6 4 6 2 6 4 6
7 7 7 7 7 7 7 7
5 6 5 6 5 6 5 6
7 7 7 7 7 7 7 7
3 6 4 6 3 6 4 6
7 7 7 7 7 7 7 7
5 6 5 6 5 6 5 6
7 7 7 7 7 7 7 7

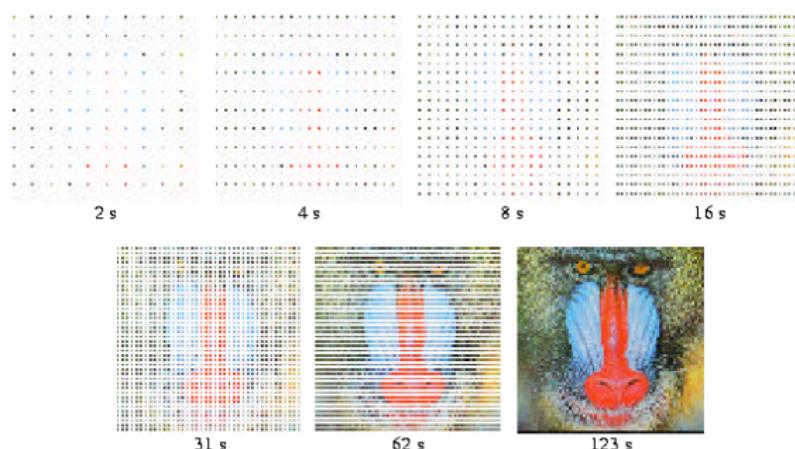


Imagen Bitmap vs. Imagen Vetorial

Imagen vectorial

- A imagem vectorial é constituída por um conjunto de figuras elementares descrita por dados matemáticos.
- Descrevem as diferentes figuras como objectos gráficos independentes uns dos outros.
- Os objectos podem ser manipulados e transformados independentemente.

Imagen Bitmap vs. Imagen Vetorial

Imagen vectorial - Ciração (XML)

Para criar um ficheiro SVG qualquer editor de texto serve. Encontramos porém alguns softwares concebidos para a criação do SVG:

- Amaya - <http://www.w3c.org> (open source W3C);
- Sodipodi - <http://wwwsodipodi.com> (GNU/Linux);
- Inkscape - <http://www.inkscape.org> (open source).

```
<svg id="drapeau France" viewBox="0 0 300 200">
<title> Test SVG </title>
<g id="France" stroke ="blue">
  <path d="M 50 50 h 50 v 100 h -50 z" fill="blue"/>
  <path d="M 100 50 h 50 v 100 H 100 z" fill="white"/>
  <path d="m 150 50 H 200 V 150 H 150 v -100" fill="red"/>
</g>
</svg>
```

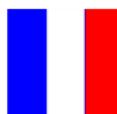


Imagen Bitmap vs. Imagen Vetorial

Imagen vectorial - Vantagens

- As informações são descritas textualmente logo eficazmente comprimidas;
- O tamanho da memória é independente do tamanho da imagem;
- É possível aplicar facilmente e sem perda de precisão transformações geométricas: deslocamentos, translações, alteração de tamanho, rotações, ...;
- Os diferentes objectos podem ser manipulados e transformados independentemente e com grande precisão;
- São independentes dos periféricos e da resolução, assim são automaticamente colocadas na escala para ser imprimidas de forma precisa sobre qualquer periférico de saída.

Imagen Bitmap vs. Imagen Vetorial

Imagen vectorial - Desvantagens

- O tamanho do ficheiro varia em função da complexidade da imagem;
- Não podemos usar o formato vectorial para descrever uma imagem muito complexa, por exemplo fotografia;
- O tempo de representação de uma imagem vectorial é superior em relação a uma imagem bitmap (aumenta com a complexidade da imagem);
- Qualquer perda ou corrupção no ficheiro leva à perda da imagem na totalidade;

Compressão de Audio

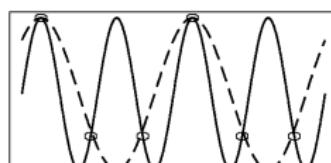
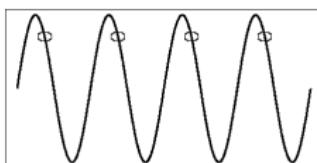
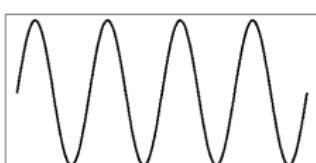
Compressão de Audio

Compressão de Áudio Digital

- Pulse Code Modulation (PCM);
- Algoritmos de compressão de som.
- Formatos de ficheiros de áudio digital.

Pulse Code Modulation (PCM)

- Em áudio as fases de amostragem e quantização denominam-se PCM (Pulse Code Modulation);
- O teorema de Nyquist determina: para uma amostragem correta devemos usar uma frequência de amostragem pelo menos igual a duas vezes a máxima frequência contida no sinal.



Pulse Code Modulation (PCM)

- Assumindo uma largura de banda para discurso de aproximadamente 50 Hz a 10 kHz, a taxa de Nyquist irá exigir uma amostragem de 20 kHz.
 - Quantização uniforme: o débito mínimo que conseguiremos será de aproximadamente 12 bits, para discurso em mono teríamos um débito de 240 kbps (30kB/sec).
 - Quantização não uniforme: podemos reduzir o débito para aproximadamente 8 bits por amostra guardando o nível de qualidade e reduzindo o débito para 160 kbps (20kB/sec).
- O standard para a telefonia assume que a maior frequência que pretendemos reproduzir é de 4Khz. Desta forma a frequência de amostragem é de 8KHz e usando quantização não uniforme reduz este débito para 64 kbps (8kB/sec).

Pulse Code Modulation (PCM)

Quality	Sample Rate (KHz)	Bits per Sample	Mono/ Stereo	Data Rate (uncompressed) (kB/sec)	Frequency Band (KHz)
Telephone	8	8	Mono	8	0.200-3.4
AM Radio	11.025	8	Mono	11.0	0.1-5.5
FM Radio	22.05	16	Stereo	88.2	0.02-11
CD	44.1	16	Stereo	176.4	0.005-20
DAT	48	16	Stereo	192.0	0.005-20
DVD Audio	192 (max)	24 (max)	6 channels	1,200.0 (max)	0-96 (max)

Algoritmos de compressão de som

Modulações diferenciais

Este tipo de método propõe reduzir o número de bits necessários à quantização de cada amostra codificando apenas a diferença entre duas amostras sucessivas, em vez de codificar o valor de cada amostra.

Modulação delta

- Substituir o valor que define uma amplitude de uma amostra por um único sinal (positivo ou negativo) representando a variação do sinal em relação ao seu valor precedente. Esta informação necessita apenas de um único bit por amostra (1=diferença positiva, 0=diferença negativa).
- Esta técnica, chamada modulação delta, supõe que cada valor quantificado só difere do valor precedente de mais ao menos 1 incremento de quantização.
- Esta técnica é assim limitada aos sinais cujas variações são relativamente lentas. Variações rápidas provocam saturação de declive.

DPCM (Diferencial Pulse Code Modulation)

- Para tentar limitar a incidência da saturação de declive, vamos aqui codificar não apenas o sinal da variação, mas também vamos consagrar alguns bits suplementares à expressão da diferença entre o valor de uma amostra com o valor da mostra precedente.
- Em geral a expressão desta diferença requer menos bits que a expressão do valor absoluto da amostra.: se por exemplo, consideramos que numa série de amostras diferenças de +-16 incrementos são muito pouco prováveis, uma codificação de 4 bits pode chegar para codificar as diferenças.

ADPCM (Adaptative Differential Pulse Code Modulation)

- Vamos não somente codificar a diferença entre duas amostras sucessivas (geralmente sobre 4 bits), mas também vamos tomar em consideração a evolução do sinal sobre as amostras precedentes, para prever o valor óptimo do passo de quantização a utilizar, sempre com o objectivo de aumentar as capacidades de codificação do modulador e evitar assim a saturação de declive, ou ao menos limitar a sua incidência.
- Teremos neste caso de guardar as variações dos passos de quantização numa tabela de passos.
- Este tipo e modulação diferencial é usado na telefonia digital sem fio doméstica.
- Este método permite uma muito boa reprodução sonora com um débito inferior ao conseguido com um modulador PCM clássico.

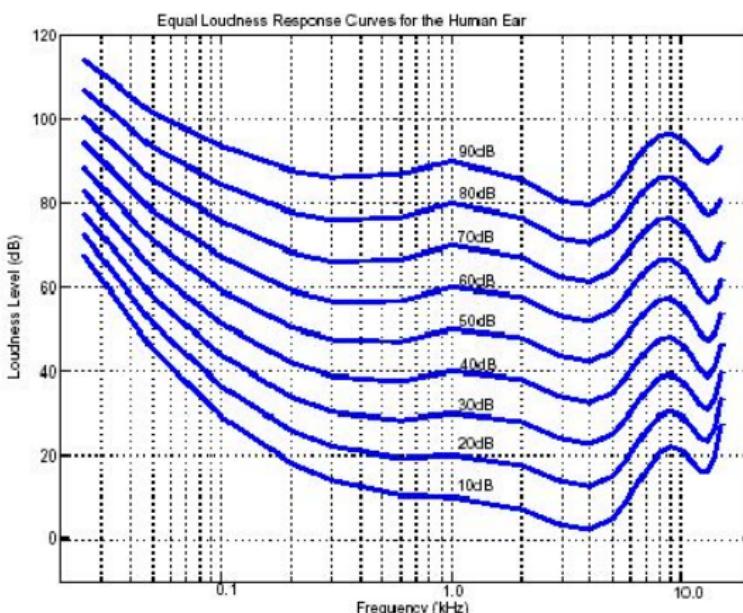
Algoritmos generalistas não destrutivos

- Os algoritmos generalistas do tipo WinZip, WinRaR, WinAce apenas conseguem uma redução de volume do ficheiro - dependendo da complexidade da fonte- de 5% a 35%.
- Os algoritmos não destrutivos especificamente dedicados ao som fazem bastante melhor: WavArc (.wa), RKAU (RK AUdio, .rka) ou Perfect Clarity Audio (.pca) do célebre Sound Forge, permitem atingir ou mesmo passar, uma redução de volume de 50%.
- No entanto estes resultados estão ainda longe de satisfazer as restrições de débitos impostas pela difusão em tempo real sobre a Internet, ou para as televisores ou rádios digitais sobre redes por cabo. Assim, temos necessidade de técnicas de compressão impondo desta vez uma inevitável perda de informação.

Algoritmos destrutivos: a codificação Psico-acústica

O princípio base da codificação Psico-acústica é simples: existe em todo o registo sonoro uma quantidade não negligenciável de sons que o auditivo não pode perceber e ruídos que ele não quer ouvir. Suprimir as modulações "inúteis" e o ruído incomodo diminuirá o peso dos ficheiros sem, no entanto, degradar a qualidade de reprodução.

Algoritmos destrutivos: a codificação Psico-acústica



Propriedades do ouvido humano

- Em regra geral, suprimimos os sinais situados em frequências consideradas inúteis, e da mesma forma todos os sinais, qualquer que seja a sua frequência situados abaixo do limite de percepção.
 - Sabemos por exemplo, que a banda das frequências ditas audíveis, isto é 20Hz-20kHz.
 - Juntemos que neste intervalo de 20Hz - 20kHz, o intervalo absoluto de percepção não é uniforme. É inútil registar os sons relativamente fracos com frequências inferiores a 100 Hz.
 - Se juntarmos a esta constatação a pobre performance da maior parte dos sistemas de reprodução sonora de baixa e média gama quando se trata de reproduzir som no extremo grave, negligenciaremos as frequências superiores a 10 ou 11 KHz.

Propriedades do ouvido humano

- Existe outra limitação, esta relativa, que consiste na incapacidade de distinguir um som "coberto" por um som de frequência vizinha e de amplitude superior. É o famoso efeito de máscara.
 - Se, num grupo de frequências idênticas ou vizinhas, algumas têm uma amplitude muito superior às outras, apenas elas serão percebidas
 - Podemos assim, sem grande sacrifício da qualidade de reprodução, suprimir as frequências mascaradas.

Propriedades do ouvido humano

Juntemos ainda as limitações da percepção auditiva, mas dizendo respeito desta vez à localização do som no espaço de difusão.

- Sabemos que a estereofonia, graças a um registro e a uma difusão específicas dos canais direito e esquerdo, permitem ao auditor localizar a origem dos sons.
- Não se trata realmente de uma capacidade do ouvido humano mas do cérebro humano, capaz de interpretar as muito pequenas diferenças que distinguem os sinais percebidos pelo ouvido esquerdo e pelo ouvido direito.
- Esta capacidade é limitada às frequências superiores a algumas centenas de Hz.

Propriedades do ouvido humano

- é inútil, pelo menos nas frequências baixas, distinguir um canal de direita e um canal de esquerda.
- Vamos registar apenas a parte grave do espectro em mono-fonia, realizando assim uma economia de 50% relativamente a um registo estereofónico dos mesmos sons graves.
- É o que se chama de estéreo junto (em Inglês, joint stereo).

MP3

- O brevete original do algoritmo de compressão MP3 foi deposto em 1996 pelo instituto de pesquisa Alemão Fraunhofer, que faz parte do grupo Thomson Multimedia.
- Desenvolvido no quadro de recomendações do Moving Picture Experts Group, ele tornou-se o algoritmo de compressão de som na norma MPEG e o seu verdadeiro nome é *MPEG – layer3*.
- MP3 utiliza todas as técnicas descritas anteriormente, ao qual junta em primeiro uma codificação RLC (que permite evitar a codificação de dados sucessivos idênticos ou muito vizinhos) para terminar pela aplicação aos dados finais do algoritmo de Huffman.

MP3

- A 64 Kbits/s um minuto de música ocupa somente 469 KB para uma taxa de compressão de 22:1.
- O mesmo minuto de música não comprimido ocupa um pouco mais do de 10 MB.
- A forte compressão do MP3 a 64 kbits/s é obtida à custa de uma qualidade medíocre (as frequências compreendidas entre 11 e 20 kHz são suprimidas).
- Um débito binário de 96 Kbits/s melhora, um pouco as coisas, mas é apenas a partir 128 Kbits/s e com uma taxa de compressão de 11:1 que nos aproximamos realmente da alta fidelidade.
- Finalmente a 192 Kbits/s, podemos considerar que a qualidade auditiva do CD de áudio é globalmente preservada; mas a taxa de compressão não ultrapassa 7:1.

Enquadramento

Teoria da codificação

- **Codificação fonte;**
- Codificação canal;
- Criptografia;

Sistema multimédia

Representação - Armazenamento - Transmissão - Processamento - Reprodução

Enquadramento

Teoria da codificação

O estudo de um **sistema de comunicações digitais** envolve dois aspectos cruciais:

- **A eficiência da representação da informação gerada pela fonte;**
- A taxa de transmissão à qual é possível enviar a informação com fiabilidade através de um canal ruidoso.

A teoria da informação estabelece os limites fundamentais associados às questões acima referidas. A saber:

- **O número mínimo de unidades de informação binária (bit) por símbolo necessário para representar completamente a fonte;**
- O valor máximo da taxa de transmissão que garante fiabilidade da comunicação através de um canal ruidoso.

Compressão Redundância

- Quantificar a redundância
- Incerteza vs. Informação
- Em 1948, Claude Shannon enunciou uma fórmula matemática que permite calcular a quantidade média de informação de uma experiência tendo em conta o nível de incerteza próprio a essa experiência:

Compressão

- O ganho de informação resultante da ocorrência do acontecimento certo é nulo;
- Excepto no caso do acontecimento certo, a ocorrência de um qualquer acontecimento conduz a um ganho de informação;
- quanto menor for a probabilidade de ocorrência de um acontecimento maior é o ganho de informação que lhe está associado.

Considere-se uma fonte discreta sem memória. A informação associada à ocorrência de um símbolo desta fonte é definida por:

$$I(m_i) = \log_2\left(\frac{1}{p_i}\right) = -\log_2(p_i), i = 1, \dots, M.$$

Compressão

Redundância

Entropia

Seja uma experiência X suscetível de tomar os estados x_1, x_2, \dots, x_N com probabilidades p_1, p_2, \dots, p_N , respectivamente.

$$H(p_1, p_2, \dots, p_n) = \sum_{i=1}^N p_i \times \log_2 \left(\frac{1}{p_i} \right) = - \sum_{i=1}^N p_i \times \log_2(p_i) \quad (4)$$

Se os estados possíveis são equiprováveis, isto é, se eles têm todos a mesma probabilidade ($p_i = 1/N$, para $i = 1, 2, \dots, N$), a equação 4 pode-se escrever como:

$$H(p_1, p_2, \dots, p_n) = \sum_{i=1}^N \frac{1}{N} \times \log_2(N) = \log_2(N) \quad (5)$$

Compressão

Redundância

Exercícios

Seja uma imagem em níveis de cinzento com profundidade de pixel de 8 bits. Os valores representáveis com 8 bits vão de 0 a 255.

Suponhamos agora duas situações diferentes:

Primeira situação Todos os valores entre 0 e 255 são equiprováveis.

Segunda situação O valor 128, por exemplo, tem uma probabilidade $p_{128} = 0.99$ e todos os outros valores tem uma probabilidade idêntica $p_i = 0.01/255$.

Compressão Redundância

Entropia

A entropia H exprime o número médio de bits por símbolo necessários para a codificação ideal de um determinado alfabeto.

Compressão

Parâmetros de compressão

- Taxa de compressão;
- Simetria e assimetria;
- Com ou sem perdas.

Técnicas de base

Run Length Coding (RLC)

- Método simples que pode ser usado em caso de existência de muitas repetições.
- $aaaa.... < - > \#Na$.

Exercícios

- A sequência "AAAABQEEEEFFFF" quando codificada usando 8 bits por símbolo requer um total de 14 bytes. E usando a codificação RLC?
- Codifique usando a codificação RLC:
"0000000001111000000".

Técnicas de base

Codificação relativa

- Método simples que pode ser usado na codificação de longas séries de valores próximos.
- 01011100 01011111 01011101 01011110...
#6 010111 4 00 11 01 10

Exercícios

Codifique "00000001 00000000 00000001 00000001
00000001 00000000 00000000 00000000 00000001
00000001".

Algoritmos de codificação estatística

Algoritmos que utilizam uma análise estatística ou entrópica que melhora consideravelmente a sua eficiência.

- Método de Huffman;
- Método de Huffman canónico;
- Método de Huffman adaptativo;
- Método de Shannon-Fano;
- Codificação aritmética.

Algoritmos de codificação estatística

Método de Shannon-Fano

- 1 Leitura do ficheiro e cálculo das frequências de cada símbolo;
- 2 Classificação dos símbolos em função das suas frequências;
- 3 Divisão das frequências em dois sub-grupos, de tal forma que a relação da soma das frequências de cada sub-grupo seja o mais próxima possível de 1: $\sum_1 f_i / \sum_2 f_j \approx 1$. Esta operação é repetida até que todas as frequências de origem sejam encontradas.
- 4 Atribuição de um código a cada símbolo;
- 5 Codificação final.

Algoritmos de codificação estatística

Método de Shannon-Fano

Exemplo

Usando o mesmo exemplo usado nos códigos de Huffman teremos:

Símbolo i	Frequência f_i
E	4
R	4
O	3
Espaço	3
M	2
A	1

Algoritmos de codificação estatística

Método de Huffman

- Codificar com o menor número de bits os símbolos em que a frequência de ocorrência é a maior.
- Os primeiros elementos de uma palavra código (o prefixo) não pode constituir uma outra palavra de código (código prefixo). Isto permite-nos garantir uma descodificação instantânea.
- Obriga à criação de tabelas de frequências que terão de ser transmitidas visto que são necessárias para a descodificação.

Algoritmos de codificação estatística

Método de Huffman

Algoritmo

- Etapa 1** Leitura do ficheiro - cálculo da frequência de cada símbolo.
- Etapa 2** Classificação dos símbolos em função da sua frequência.
- Etapa 3** Reagrupamento sequencial dos pares de símbolos de menor frequência, voltando a executar a classificação, se necessário.
- Etapa 4** Atribuição de um código a cada símbolo.
- Etapa 5** Codificação final.

Algoritmos de codificação estatística

Método de Huffman

Exemplo

Vamos codificar a sequência:

AMOREMOREORERE

Algoritmos de codificação estatística

Método de Huffman

Conclusões

- Os códigos obtidos não são únicos.
- O código obtido com o algoritmo de Huffman é ótimo, visto que não existe nenhum outro código de prefixo cuja largura média seja inferior.
- São instantâneos.

Algoritmos de codificação estatística

Medidas

- Largura média de um código

$$I_{media} = \sum_i p_i l_i \quad (6)$$

l_i - a largura do código i .

- Redundância do código: $I_{media} - H$.

Exercícios

- Calcule a largura média do código do exemplo. Indique qual a redundância do código.
- Use árvores binárias para calcular o código do exemplo dado

Algoritmos de codificação estatística

Desvantagens/Soluções

Conclusões

- Leitura do ficheiro para calculo das frequências;
- Necessidade de guardar as tabelas de códigos;
- Uma solução para estes problemas pode passar pela utilização de tabelas estáticas juntando as probabilidades médias representativas para certos tipos de ficheiros.

Algoritmos de codificação estatística

Método de Huffman Canónico

- Alternativa que pode ser distinguida pois segue regras simples (códigos mais pequenos têm valores superiores (considerando os zeros à direita); Dentro do mesmo comprimento os valores aumentam com o alfabeto.)
- O código mais longo contém apenas zeros, e cada código difere 1 bit do anterior.
- O cabeçalho pode ser simplificado. Aqui só precisamos de saber o número de bits a usar para codificar cada símbolo.

Algoritmos de codificação estatística

Método de Huffman Canónico

Código de Huffman

Símbolo <i>i</i>	Código
E	11
Espaço	100
R	00
M	1010
O	01
A	1011

Forma canónica

Símbolo <i>i</i>	Código
A	0000
M	0001
Espaço	001
E	01
O	10
R	11

Algoritmos do tipo dicionário

- Não necessitarem de conhecer a estatística dos dados a comprimir.
- Não usarem códigos de comprimento variável.
- Utilizarem sequências de símbolos de comprimento variável.
- Muitos dos programas de compressão de dados mais conhecidos baseiam-se nestas técnicas: pkzip, zoo, arj, compress, gzip, ...
- LZ77, LZ78: Jacob Ziv e Abraham Lempel.
- LZH, LZSS,LZW (Lempel, Ziv, Welch).
- Pode ser usado para comprimir texto, código binário, código executável, imagens (GIF, TIFF, PostScript).

Algoritmos do tipo dicionário

Algoritmo de codificação LZW

- 1 Os símbolos do alfabeto são inseridos no dicionário.
- 2 Inicializa uma sequência (S) a vazio;
- 3 Lê símbolo (x) da mensagem a codificar;
- 4 Se símbolo é o EOF vai para ponto 7.
- 5 Gera sequência (Sx) da concatenação de S com x ($Sx = S+x$);
- 6 Se Sx puder ser encontrada no dicionário:
 - 1 $S = SX$;
 - 2 vai para ponto 2.
- 7 Se Sx não puder ser encontrada no dicionário:
 - 1 é emitido o código de S;
 - 2 Sx é colocada no dicionário;
 - 3 e S é inicializada com o símbolo x.
 - 4 Vai para ponto 2;
- 8 é emitido o código de S;

Algoritmos do tipo dicionário

Algoritmo de codificação LZW

Exemplo

Mensagem a codificar:

aaabaaaadaabaado

Dicionário inicial:

0	...	97	98	99	100	...	255
nul	...	a	b	c	d	...	

Algoritmos do tipo dicionário

Algoritmo de codificação LZW

Exemplo

- No final o ficheiro comprimido contém os códigos: 97 256 98 256 97 100 257 256 100 111.
- O tamanho do ficheiro será: número de códigos $\times \log_2 TamanhoDicionario$.
- neste caso $10 \times \log_2 TamanhoDicionario$.

Algoritmos do tipo dicionário

Algoritmo de descodificação LZW

- 1 Os símbolos do alfabeto são inseridos no dicionário.
- 2 Inicializa um contador COUNTD = 256;
- 3 Lê código (CARCODE);
- 4 D[COUNTD] = D[CARCODE];
- 5 escreve no ficheiro D[CARCODE];
- 6 Lê código (CARCODE); Se EOF termina, senão continua;
- 7 D[COUNTD] = D[COUNTD] + D[CARCODE][0];
- 8 COUNTD++;
- 9 Volta ao ponto 3

Algoritmos do tipo dicionário

Algoritmo de descodificação LZW

Exemplo

Descodificação de:

97 256 98 256 97 100 257 256 100 111.

Dicionário inicial:

0	...	97	98	99	100	...	255
nul	...	a	b	c	d	...	

Algoritmos do tipo dicionário

Conclusões

- A criação do dicionário e a compressão do ficheiro são realizadas à medida que o ficheiro é lido, sem necessitar de uma leitura prévia (assim o tempo de compressão é reduzido).
- O dicionário não é adicionado ao ficheiro comprimido (maior taxa de compressão).

Os métodos mistos: estatísticos e de dicionário

- Num primeiro passo, aplicamos aos dados de origem um algoritmo do tipo dicionário, que explora, como acabamos de verificar, as sequências repetitivas. Num segundo tempo, o resultado obtido é submetido a um algoritmo de compressão estatístico que permite obter uma segunda redução de volume do ficheiro.
- Esta técnica mista permite obter bons resultados (50 a 65 % sobre a maioria dos ficheiros).
- ARJ, LHA, PKZIP, UC, GZIP, WINZIP, ou ZOO.

Compressão

Conclusões

- Por vezes o tamanho dos ficheiros é tal que temos de aceitar perder alguma informação menos importante no que diz respeito à percepção visual e auditiva humana.
- A compressão com perdas é capaz de reduzir o tamanho dos ficheiros em proporções muito superiores àquelas conseguidas com os métodos de compressão tradicionais.
- Notemos que estas práticas de compressão com perdas, não excluem a utilização complementar destes métodos de compressão sem perdas.

Frequências

Representação no espaço das frequências.

Frequências

Aplicação

- Filtragem
- Remoção de ruído
- Análise de sinais/imagens
- Implementação simples da convolução
- Processamento de efeitos em áudio ou em imagem.
- Restauração de sinais/imagens - e.g. Deblurring
- Compressão de sinais/imagens - MPEG (Audio e Video), JPEG.
- etc

Frequências

Introdução ao espaço das frequências - 1D

Considerando um exemplo em 1D (ex. audio) vejamos o que os diferentes domínios significam:

Consideremos um acorde tocado num piano ou numa guitarra. Este pode ser descrito de duas formas:

Domínio temporal Amostrar as amplitudes do sinal várias vezes num segundo, o que permite obter uma aproximação do sinal em função do tempo.

Dominío das frequências Analisar o sinal em termos dos tons das notas (pitches), ou frequências. Que parte do sinal está dentro de cada banda de frequência dada uma gama de frequências.

Frequências base: $D_b : 554.40 \text{ Hz}$; $F : 698.48 \text{ Hz}$; $A_b : 830.64 \text{ Hz}$; $C = 1046.56 \text{ Hz}$, mais frequências hamónicas/parciais.

Frequências Domínio temporal

Um sinal consistindo numa onda sinusoidal a 8Hz:

```
A=1; %amplitude  
Fw=8; %frequencia da onda  
ph = 0; %phase  
Fa = 44100;  
t=0:1/Fa:1-1/Fa; %tempo de amostragem  
  
y= A*sin(2*pi*t*Fw+ph);  
figure(1); plot(t,y);
```

Frequências

Domínio das frequências

No domínio da frequência podemos ver que o nosso sinal contém um pico na frequência de 8Hz. A magnitude/fração de 1.0 correspondendo à totalidade do sinal.

```
Y = abs(fft(y) / (Fa/2));  
figure(2); stem(0:1:19, Y(1:20));
```

Frequências

Introdução ao espaço das frequências -2D

O que significam as frequências numa imagem?

- O brilho ao longo de uma linha pode ser gravado como um conjunto de valores medidos a distâncias igualmente espaçadas.
- Ou, de forma equivalente, num conjunto de valores de frequência espacial.
- Cada um destes valores de frequência é uma componente de frequência.
- Uma imagem é uma matriz 2D de medições de pixels.
- Formamos uma grelha 2D de frequências espaciais. Um dado componente de frequência agora especifica qual a contribuição feita pelos dados que mudam com as frequências espaciais de direção x e y especificadas.

Frequências

Introdução ao espaço das frequências -2D

O que significam as frequências numa imagem?

- Grandes valores nas componentes de altas frequências indicam que os dados estão a mudar rapidamente.
 - Uma página de texto;
 - Contornos de uma imagem;
 - O ruído também contribui muito para as altas frequências;
- Valores altos nas componentes de baixa frequência indicam que as características de grande escala da imagem são mais importantes.
 - Um objeto simples que ocupe a maior parte da imagem;

Frequências

Transformada de Fourier

- Qualquer sinal digital (função) pode ser decomposta em componentes puramente sinoidais.
 - Ondas seno de diferentes tamanhos e formas - variando a amplitude, a frequência e a phase;
- Adicionando as componentes sinoidais reconstruímos o sinal original.
- A transformada de Fourier é a ferramenta que permite efetuar estas operações. Esta deve-se ao matemático Baron Jean-Baptiste-Joseph Fourier.

Frequências

Transformada de Fourier

Exemplo: Se tocar um acorde de piano pressionando três teclas. Produz três notas diferentes, todas com frequências bem definidas (pitches). Quando falamos de áudio podemos representá-las por ondas seno.

- Transformar um sinal no domínio das frequências permite-nos verificar que ondas seno compõem o nosso sinal.

Frequências

Domínio das frequências

```
Fa=100;
t=0.0:1/Fa:1-1/Fa;
A1=1; A2=0.3; A3=0.2; A4=0.1;
Fw1=1; Fw2=10; Fw3=15; Fw4= 30;
x=zeros(4,Fa);
x(1,:) = A1 * sin(2*pi*t * Fw1);
x(2,:) = A2 * sin(2*pi*t * Fw2);
x(3,:) = A3 * sin(2*pi*t * Fw3);
x(4,:) = A4 * sin(2*pi*t * Fw4);

xt=x(1,:)+x(2,:)+x(3,:)+x(4,:);
```

Frequências

Domínio das frequências

```
figure(2);
tmp=0;
for(i=1:4)
    subplot(2,1,1); plot(t,xt, 'r');
    hold on;
    tmp = tmp+x(i,:);
    plot(t,tmp,'b');
    legend('Original', 'Reconstruida');
    hold off;

    subplot(2,1,2); plot(t,x(1:i,:),'b');
    hold on;
    plot(t,x(i,:),'r'); ylim([-1 1]);
    hold off;
    pause();
end
```

Frequências

Transformada de Fourier

Considerando a função contínua $f(x)$ de variável x , que pode representar uma distância (ou tempo).

A transformada de Fourier dessa função é denominada $F(u)$, onde u representa a frequência espacial (ou temporal) é definida por:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi xu} dx.$$

Note que no geral $F(u)$ será uma quantidade complexa, mesmo sendo os dados originais reais (significa que tanto a frequência como a fase são informações importantes).

O termo $e^{-j2\pi xu}$ é uma representação matemática da sinusóide.

Frequências

Transformada de Fourier inversa

A transformada de Fourier inversa para recuperar $f(x)$ a partir de $F(u)$ é dada por

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi xu} du$$

Que é muito semelhante à transformada (direta) de Fourier, excepto que o termo exponencial tem o sinal contrário;

Frequências

Transformada de Fourier Discreta - 1D

Dado que todos os nossos dados de áudio e imagem são digitalizados, necessitamos de uma formulação discreta da transformada de Fourier.

Em 1D, e considerando que x varia em passos de 1, e que existem N amostras, a valores de x de 0 a $N - 1$.

A DFT toma a forma:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi xu/N}$$

enquanto a inversa da DFT é

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{j2\pi xu/N}$$

Frequências

Transformada de Fourier Discreta - 2D

Em 2D, a DFT funciona de forma semelhante. Assim, para uma grelha de $N \times M$ em x e em y , temos:

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi(xu/N+yv/M)}$$

e

$$f(x) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{-j2\pi(xu/N+yv/M)}$$

Frequências

Transformada de Fourier Discreta

O MATLAB fornece as funções 1D e 2D da transformada de Fourier Discreta.

fft(x) é a transformada de Fourier 1D, onde x é um vetor.

fft2(x) é a transformada de Fourier 2D, onde X é uma matriz.

ifft(x) é a transformada de Fourier inversa 1D, onde x é um vetor.

ifft2(x) é a transformada de Fourier inversa 2D, onde X é uma matriz.

Frequências

Transformada de Fourier Discreta

Relembremos que a transformada de Fourier de dados reais é complexa.

Como podemos visualizar estes vetores/matrizes de dados complexos?

Espetro de amplitude $|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$.

$R(u, v)$ é a parte real e $I(u, v)$ é a parte imaginária de $F(u, v)$.

O espectro de amplitude é simétrico relativamente ao eixo vertical

Em matlab $A = abs(fft(X))$

Frequências

Domínio das frequências

```
figure(3);
subplot(5,2,1); plot(t, xt,'k'); ylim([-2 2]);
subplot(5,2,3); plot(t, x(1,:),'r'); ylim([-2 2]);
subplot(5,2,5); plot(t, x(2,:),'g'); ylim([-2 2]);
subplot(5,2,7); plot(t, x(3,:),'b'); ylim([-2 2]);
subplot(5,2,9); plot(t, x(4,:),'b'); ylim([-2 2]);

XT = abs(fft(xt)/(Fa/2));
X1 = abs(fft(x(1,:))/(Fa/2));
X2 = abs(fft(x(2,:))/(Fa/2));
X3 = abs(fft(x(3,:))/(Fa/2));
X4 = abs(fft(x(4,:))/(Fa/2));

subplot(5,2,2); plot(t(1:Fa/2), XT(1:Fa/2),'k'); ylim([0 1]);
subplot(5,2,4); plot(t(1:Fa/2), X1(1:Fa/2),'r'); ylim([0 1]);
subplot(5,2,6); plot(t(1:Fa/2), X2(1:Fa/2),'g'); ylim([0 1]);
subplot(5,2,8); plot(t(1:Fa/2), X3(1:Fa/2),'b'); ylim([0 1]);
subplot(5,2,10); plot(t(1:Fa/2), X4(1:Fa/2),'b'); ylim([0 1]);
```

Frequências

Transformada de Fourier Discreta

Quando desenhamos os gráficos do espetro de amplitude, é muitas vezes mais interessante representar o eixo do x em Hz (frequência) em vez de número de pontos.

Existe uma relação simples entre os dois:

- Os pontos da amostra variam em passos de $k = 0, 1, \dots, N - 1$
- Para um dado ponto de amostra k a frequência relacionada com este é dada por:

$$f_k = k \frac{f_s}{N}$$

onde f_s é a frequência de amostragem e N é o número de amostras.

- Temos assim passos de frequência equidistantes de $\frac{f_s}{N}$ a variar de 0 Hz até $\frac{N-1}{N} f_s$ Hz

Frequências

Filtrar no espaço das frequências

```
Fs = 1000; % Sampling frequency
T = 1/Fs; % Sample time
L = 1000; % Length of signal
t = (0:L-1)*T; % Time vector

x1 = 0.7*sin(2*pi*50*t); %50 Hz sinusoid
x2 = sin(2*pi*120*t); %120 Hz sinusoid

x = x1+x2; % Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
y = x + 2*randn(size(t)); % Sinusoids plus noise

figure(5);
subplot(2,2,1); plot(Fs*t(1:50),x1(1:50));
title('Amplitude: 0.7; Frequencia: 50');
xlabel('time (milliseconds)');

subplot(2,2,2);plot(Fs*t(1:50),x2(1:50));
title('Amplitude: 1; Frequencia: 120');
xlabel('time (milliseconds)');

subplot(2,2,3);plot(Fs*t(1:50),x(1:50));
title('Soma das duas anteriores');
xlabel('time (milliseconds)');

subplot(2,2,4);plot(Fs*t(1:50),y(1:50));
title('Signal Corrupted with Zero-Mean Random Noise');
xlabel('time (milliseconds)');
```

Frequências

Filtrar no espaço das frequências

```
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y1 = fft(x1,NFFT)/L;
Y2 = fft(x2,NFFT)/L;
Y = fft(x,NFFT)/L;
Y0 = fft(y,NFFT)/L;

f = Fs/2*linspace(0,1,NFFT/2+1);

figure(6);
subplot(2,2,1);plot(f,2*abs(Y1(1:NFFT/2+1)));
title('Single-Sided Amplitude Spectrum of y(t)');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|');
subplot(2,2,2);plot(f,2*abs(Y2(1:NFFT/2+1)));
title('Single-Sided Amplitude Spectrum of y(t)');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|');
subplot(2,2,3);plot(f,2*abs(Y(1:NFFT/2+1)));
title('Single-Sided Amplitude Spectrum of y(t)');
xlabel('Frequency (Hz)'); ylabel('|Y(f)|');
subplot(2,2,4);plot(f,2*abs(Y0(1:NFFT/2+1)));
title('Single-Sided Amplitude Spectrum of y(t)');
xlabel('Frequency (Hz)'); ylabel(|Y(f)|');
```

Frequências

Filtrar no espaço das frequências

Filtrar no espaço das frequências envolve atenuar ou remover certas frequências:

- Filtro passa-baixo: ignora as componentes de alta frequência (coloca a zero ou um valor baixo); Apenas guarda as componentes de baixa frequência.
- Filtro passa-alto: inverso do anterior;
- Filtro passa banda: apenas permite frequências num determinado intervalo.

Frequências

Filtrar no espaço das frequências

```
%High Pass Filtering in Fourier Domain
a=imread('cameraman.tif'); %Rosa1024.png cameraman.tif
b=size(a);
af=fftshift(fft2(a));

figure;
subplot(2,3,1); imshow(a);
subplot(2,3,4); fftshow(af);

[x,y]=meshgrid(-b(1,1)/2:b(1,1)/2-1,-b(1,2)/2:b(1,2)/2-1);
z=sqrt(x.^2 + y.^2);

c=z>15;
hp = af.* c;
hpi = ifft2(hp);
subplot(2,3,2); fftshow(hp);
subplot(2,3,5); ifftshow(hpi);

c=z>50;
hp = af.* c;
hpi = ifft2(hp);
subplot(2,3,3); fftshow(hp);
subplot(2,3,6); ifftshow(hpi);
```

Frequências

Filtrar no espaço das frequências

```
%Low Pass Filtering in Fourier Domain
a=imread('cameraman.tif'); %Rosa1024.png cameraman.tif
b=size(a);
af=fftshift(fft2(a));

figure;
subplot(2,4,1); imshow(a);
subplot(2,4,5); fftshow(af);

[x,y]=meshgrid(-b(1,1)/2:b(1,1)/2-1,-b(1,2)/2:b(1,2)/2-1);
z=sqrt(x.^2+y.^2);

c=z<40;
af1=af.*c;
afli=ifft2(af1);
subplot(2,4,2); fftshow(af1);
subplot(2,4,6); ifftshow(afli);

c=z<15;
af1=af.*c;
afli=ifft2(af1);
subplot(2,4,3); fftshow(af1)
subplot(2,4,7); ifftshow(afli)

c=z<5;
af1=af.*c;
afli=ifft2(af1);
subplot(2,4,4); fftshow(af1)
subplot(2,4,8); ifftshow(afli)
```

Frequências

Filtrar no espaço das frequências

Filtrar no espaço de Fourier consiste em calcular:

$$G(u, v) = H(u, v)F(u, v);$$

onde:

- $F(u, v)$ é a transformada de Fourier da imagem original.
- $H(u, v)$ é uma função de filtragem.
- $G(u, v)$ é a transformada de Fourier da imagem filtrada.
- Efetuando a transformada de Fourier inversa de $G(u, v)$ recuperamos $g(x, y)$, a imagem filtrada.

Frequências

Filtrar no espaço das frequências

Muitos dos efeitos em áudio e em imagem podem ser descritos em termos de convoluções.

- Filtrar - A filtragem de Fourier apresentada antes é a aplicação de convoluções de um filtro passa-baixo onde as equações são transformadas de Fourier de equivalentes no espaço real.
- Deblurring - filtragem passa-alto..
- Reverb - convolução da resposta impulsional.

Frequências

Filtrar no espaço das frequências

A convolução de duas funções $f(x)$ e $h(x)$,

$$f(x) * h(x) = \int_{-\infty}^{\infty} f(u)h(x - u)du.$$

- $*$ é a notação matemática da convolução

Frequências

Filtrar no espaço das frequências

Existe ainda uma definição de convolução para funções de domínio discreto, dada por

$$(f * h)(n) = \sum_{k=-\infty}^{\infty} f(k) \cdot h(n - k) \quad (2a)$$

Frequências

Filtrar no espaço das frequências

Teorema da convolução Se $f(x)$ e $h(x)$ são duas funções com transformadas de Fourier $F(u)$ e $H(u)$, então a transformada de Fourier da convolução $f(x) * h(x)$ é simplesmente o produto das transformadas de Fourier das duas funções, $F(u)H(u)$.

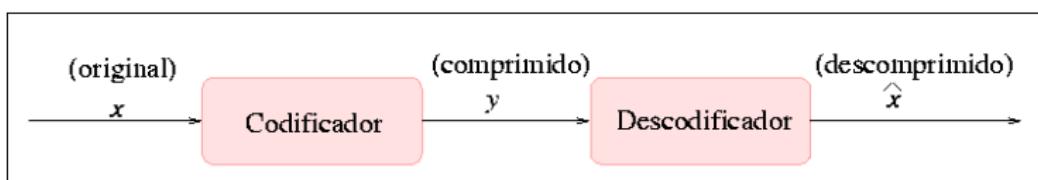
$$f(x) * h(x) \Leftrightarrow H(u)F(u)$$

ou em 2D

$$f(x, y) * h(x, y) \Leftrightarrow H(u, v)F(u, v)$$

Compressão de Imagem

Processo de compressão com perdas



Compressão de Imagem

Medidas de distorção

MSE Erro médio quadrado (Mean square Error):

$$MSE = \frac{1}{N} \sum_{n=1}^N (x_n - \hat{x}_n)^2$$

SNR Signal to Noise Ratio:

$$SNR = 10 \log_{10} \frac{\sigma_x^2}{MSE} dB$$

PSNR Peak Signal to Noise Ratio,

$$PSNR = 10 \log_{10} \frac{m^2}{MSE}$$

Compressão de Imagem

Outras medidas

$$\text{Taxa de compressão} = \frac{N_1 N_2 B}{||c||}$$

N_1 e N_2 : dimensões da imagem.

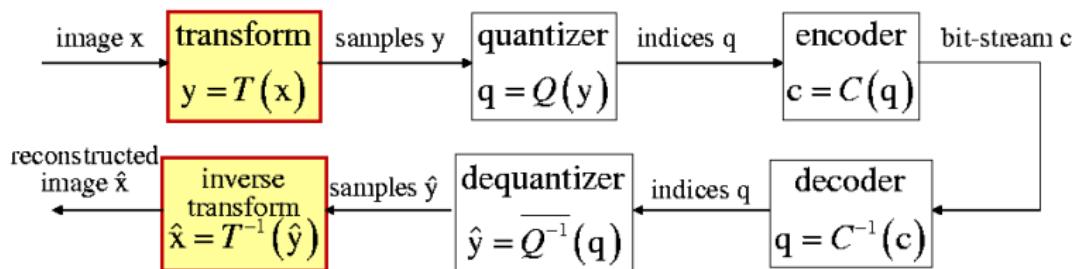
B : número de bits usados para representar cada pixel.

$||c||$ representa o tamanho da trama binária comprimida c .

$$\text{Débito binário} = \frac{||c||}{N_1 N_2}$$

Compressão de Imagem

Sistema de compressão



Compressão de Imagem

Sistema de compressão

Transformada Representa os dados da imagem de forma a eliminar as redundâncias estatísticas. Neste passo os valores de entrada são transformados em coeficientes. Muitos destes coeficientes são zero. Assim, a energia está compactada em poucos coeficientes.

A transformada é normalmente invertível.

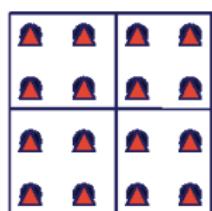
Quantização Reduzir o número de valores de amplitude possíveis para codificação.

A quantização não é invertível, introduzindo erro de quantização.

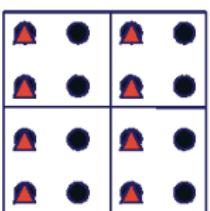
Codificação Explorar a não uniformidade da distribuição de probabilidade dos índices de quantização.
Este passo não implica perdas.

Compressão de Imagem

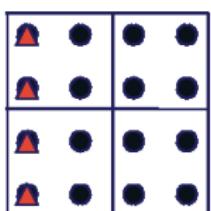
Transformada de cor



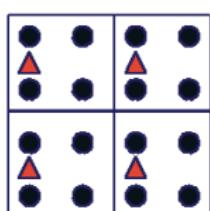
4:4:4
For every 2x2 Y Pixels
4 Cb & 4 Cr Pixel
(No subsampling)



4:2:2
For every 2x2 Y Pixels
2 Cb & 2 Cr Pixel
(Subsampling by 2:1 horizontally only)



4:1:1
For every 4x1 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 4:1 horizontally only)



4:2:0
For every 2x2 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 2:1 both horizontally and vertically)

● Y Pixel

▲ Cb and Cr Pixel

Compressão de Imagem Transformadas

Algumas das transformadas usadas no processamento de imagens são por exemplo:

- KLT - Karhunen-Loëve Transform;
- DFT - Discrete Fourier Transform;
- DCT - Discrete Cosine Transform (JPEG, MPEG, JVT, ...);
- Wavelets.

Compressão de Imagem

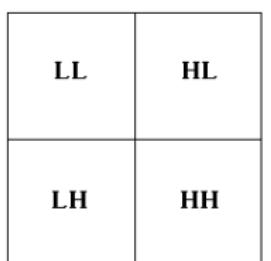
Transformada DCT

O papel da DCT é decompor o sinal original em componentes DC e AC. O coeficiente DC é, geralmente, o mais importante, enquanto os coeficientes AC, em geral, tornam-se menos importantes à medida que se distanciam do DC; Propriedades das DCT que têm um valor particular para as aplicações de processamento de imagem:

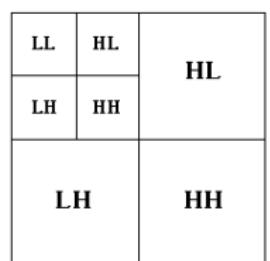
- Decorrelação
- Compactação de energia
- Separabilidade
- Simetria
- Ortogonalidade

Compressão de Vídeo

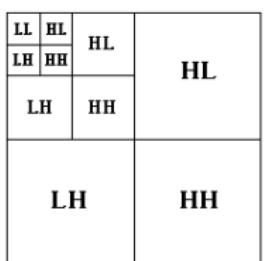
Transformada wavelets



(a) Single Level Decomposition



(b) Two Level Decomposition



(c) Three Level Decomposition

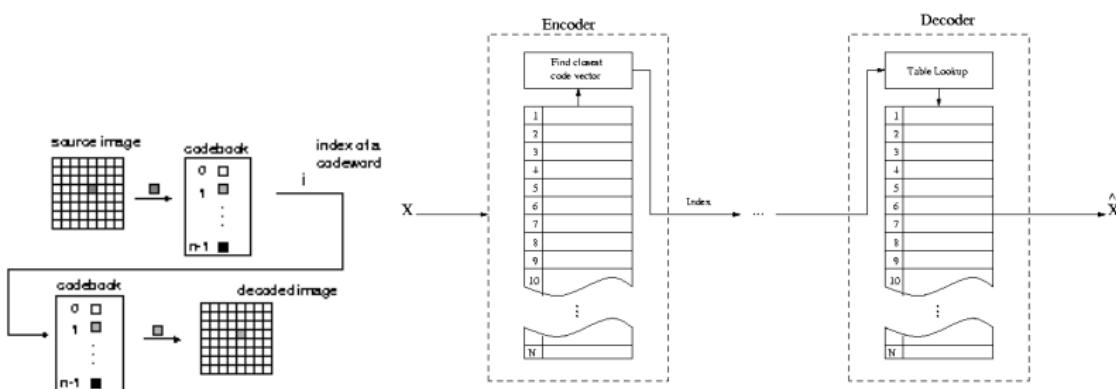
Compressão de Vídeo

Transformada wavelets



Compressão de Imagem

Quantização



Compressão de Imagem

Quantização escalar

$$I_q = [t_q, t_{q+1}), q = 0, 1, \dots, M - 1$$

com

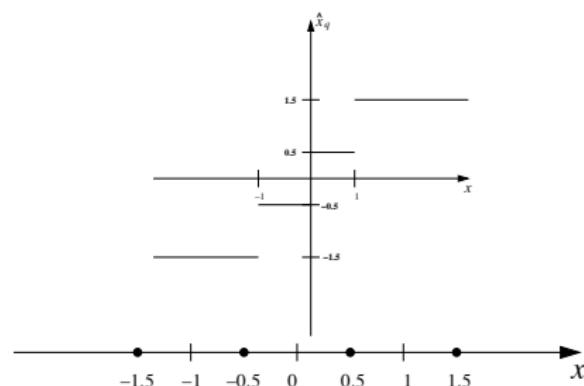
$$-\infty = t_0 < t_1 < \dots < t_M = +\infty.$$

- O quantizador, $Q(x)$, é o índice q do intervalo I_q que contém x .
- O dequantizador é dado por

$$\overline{Q^{-1}}(q) = \hat{x}_q.$$

Compressão de Imagem

Quantização escalar



Compressão de Imagem

Quantização escalar uniforme

- O valor de saída correspondente a cada intervalo é o ponto médio do intervalo ou então o centróide do intervalo ($\text{round}(\sum x p_x)$).
- O tamanho de cada intervalo é chamado tamanho do passo e denotado por Δ .
- Podemos ter:
 - Quantizadores midrise possuem número par de níveis de saída;
 - Quantizadores midtread possuem número ímpar de níveis de saída, incluindo o zero como um deles.

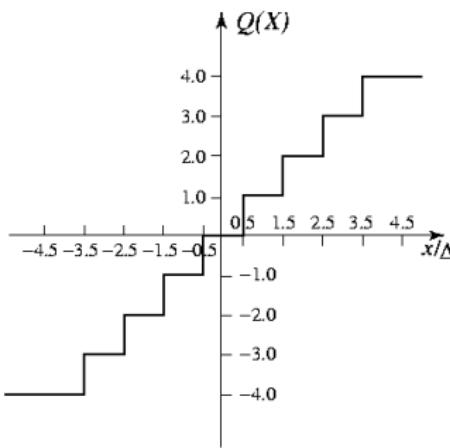
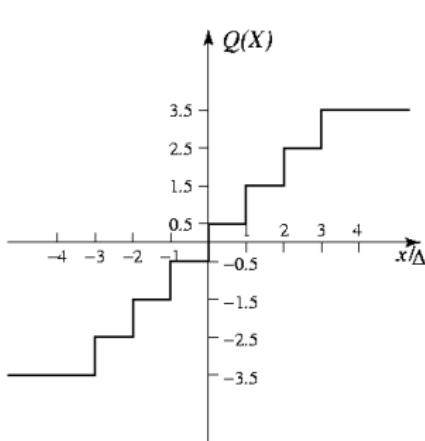
Compressão de Imagem

Quantização escalar uniforme

Para o caso especial em que $\Delta = 1$, podemos simplesmente calcular os valores de saída do quantizador como:

$$Q_{\text{midrise}}(x) = \lceil x \rceil - 0.5$$

$$Q_{\text{midtread}}(x) = \lfloor x + 0.5 \rfloor$$



Compressão de Imagem

Quantização escalar não uniforme

Na quantização escalar não uniforme:

- Para zonas com maior quantidade de valores são definidos passos de quantização menores
- Para zonas com menor quantidade de valores são definidos passos de quantização maiores.
- O Algoritmo de Loyd (1957) tem por objectivo encontrar os intervalos que implicam o menor erro de quantização.

Compressão de Imagem

Quantização escalar não uniforme

Condições de Lloyd-Max:

$$t_q = \frac{\hat{x}_{q-1} + \hat{x}_q}{2} \quad q = 1, 2, \dots, M - 1.$$

$$\hat{x}_q = \frac{\int_{t_q}^{t_{q+1}} xf_X(x)dx}{\int_{t_q}^{t_{q+1}} f_X(x)dx} \quad q = 0, 1, \dots, M - 1.$$

ou

$$\hat{x}_q = \text{round} \left(\frac{\sum_{x=t_q}^{t_{q+1}} x p_X(x)}{\sum_{x=t_q}^{t_{q+1}} p_X(x)} \right) \quad q = 0, 1, \dots, M - 1.$$

Os níveis de quantificação ótimos são dados pelos centroides e os limites de quantificação de cada região são dados pelos pontos médios entre cada par de centroides.

Compressão de Imagem

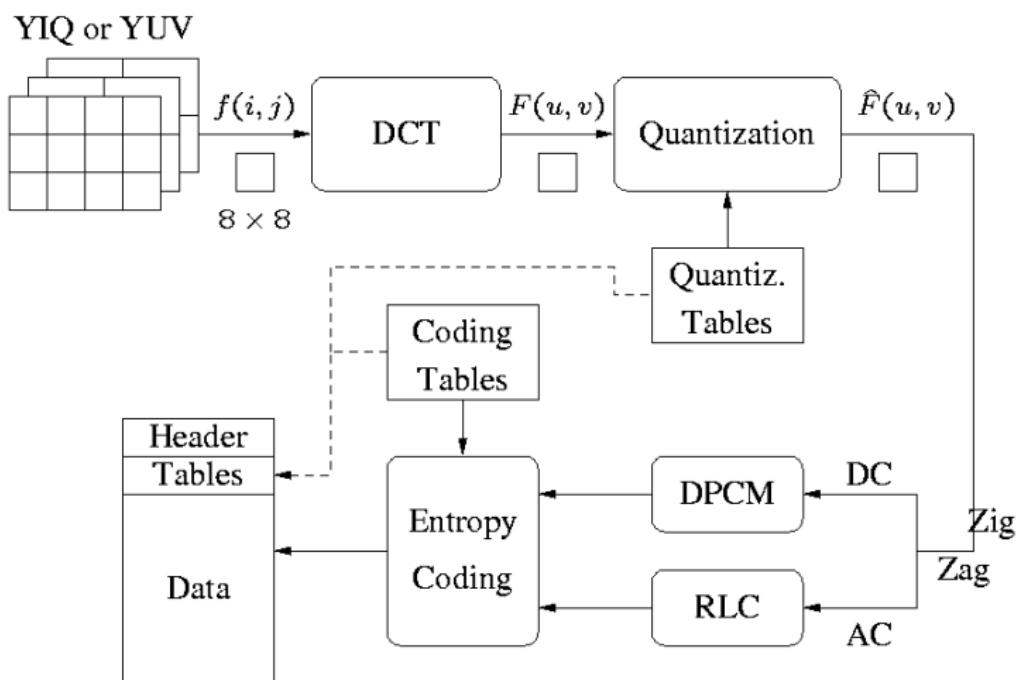
A norma JPEG

- Codificação sequencial: cada componente da imagem é codificado numa única passagem da esquerda para a direita e de cima para baixo;
- Codificação progressiva: A imagem é codificada em várias passagens para aplicações em que o tempo de transmissão é longo e o utilizador prefere ver construir a imagem de mais grosseira para mais clara;
- Codificação sem perdas;
- Codificação hierárquica: a imagem é codificada com múltiplas resoluções, de forma a que uma versão com menor resolução possa ser acedida antes de descomprimir a imagem na sua resolução final.

A maior parte das implementações do mercado possui apenas o codec sequencial de base ("baseline").

Compressão de Imagem

Principais passos da compressão JPEG



Compressão de Imagem

Principais passos da compressão JPEG

- Transformação de cor: o objectivo deste passo é o de transformar as imagens representadas em RGB em YIQ ou YUV. De seguida pode-se fazer uma sub - amostragem dos planos de cor;
- Transformação espacial: neste passo é executada a DCT em blocos da imagem (para imagens em 8 ou 12 bits - o modo de base apenas executa a DCT para 8);
- Quantização;
- Ordenação em zig-zag e codificação run-length;
- Codificação entrópica (ou estatística).

Compressão de Imagem

DCT em blocos

- Cada imagem é dividida em blocos de 8×8 .
- A 2D DCT é aplicada a cada um desses blocos.
- O facto de usar blocos neste passo cria o chamado efeito de bloco quando a taxa de compressão especificada pelo utilizador é elevada.

Compressão de Imagem

Quantização

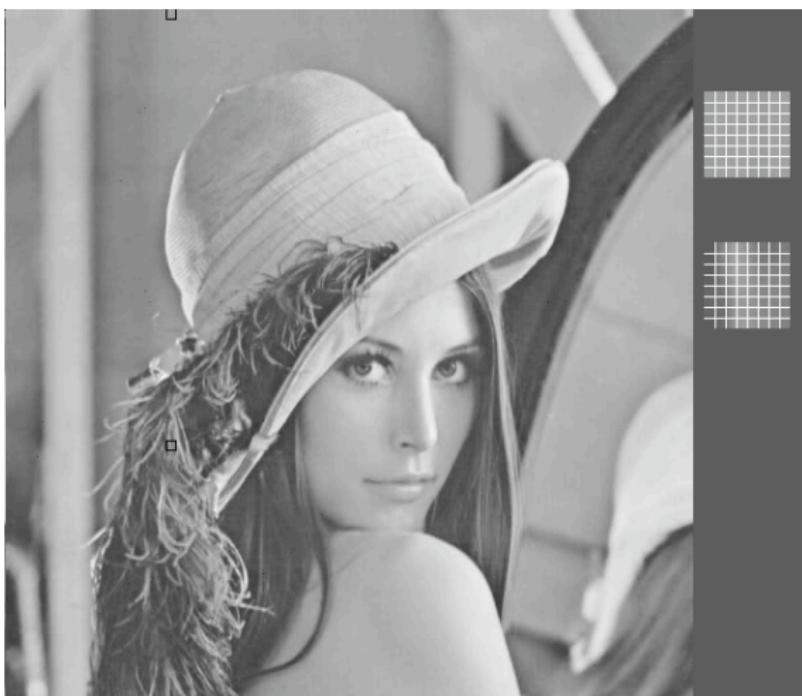
- Cada conjunto de 64 coeficientes é quantizado utilizando tabelas de quantização.
- Os valores destas tabelas são valores inteiros que definem o passo de quantização.
- A quantização corresponde à divisão inteira de cada coeficiente pelo correspondente na tabela de quantização.

$$F^Q(u, v) = \text{Integer Round} \frac{F(u, v)}{Q(u, v)}$$

- As tabelas usadas podem ser as dos standard ou indicadas pelo utilizador.
- Os valores que constituem a tabela de quantização determinam a qualidade final da compressão.

Compressão de Imagem

Funcionamento da DCT e da Quantização



Compressão de Imagem

Funcionamento da DCT e da Quantização

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

Compressão de Imagem

Funcionamento da DCT e da Quantização

Os valores da imagem são deslocados de forma a ficarem centrados em zero. Como neste caso o intervalo inicial é [0, 255] para centrar em zero teremos de subtrair a todos os elementos o valor 128.

72	74	61	60	61	47	47	47
72	75	70	60	61	54	50	47
75	72	72	67	72	59	57	47
72	72	72	72	69	59	59	59
72	77	72	72	67	60	59	47
72	72	72	72	72	62	59	47
77	72	71	72	63	59	59	47
82	72	72	72	60	57	59	58

Compressão de Imagem

Funcionamento da DCT e da Quantização

DCT separável. Note que os valores de maior energia encontram-se no canto superior esquerdo.

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

Compressão de Imagem

Funcionamento da DCT e da Quantização

Tabela de quantização típica do standard JPEG:

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

Compressão de Imagem

Funcionamento da DCT e da Quantização

32	6	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
-1	0	1	0	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
0	0	0	0	0	0	0	0

Compressão de Imagem

Funcionamento da DCT e da Quantização

Procedendo agora à quantização inversa e à DCT inversa, obtemos:

512	66	-10	0	0	0	0	0
-12	0	0	0	0	0	0	0
-14	0	16	0	0	0	0	0
-14	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

Compressão de Imagem

Funcionamento da DCT e da Quantização

71	68	63	58	54	50	49	48
73	71	68	64	60	55	52	50
75	75	74	72	67	61	55	52
74	75	76	75	70	63	55	51
72	73	74	73	68	61	54	49
72	72	71	69	64	58	53	49
76	74	71	67	62	58	55	53
79	76	72	66	62	59	57	56

Compressão de Imagem

Funcionamento da DCT e da Quantização

Finalmente colocando novamente os valores no intervalo [0,255] obtemos:

199	196	191	186	182	178	177	176
201	199	196	192	188	183	180	178
203	203	202	200	195	189	183	180
202	203	204	203	198	191	183	179
200	201	202	201	196	189	182	177
200	200	199	197	192	186	181	177
204	202	199	195	190	186	183	181
207	204	200	194	190	187	185	184

Compressão de Imagem

Funcionamento da DCT e da Quantização

O erro de final é apresentado na tabela:

1	6	-2	2	7	-3	-2	-1
-1	4	2	-4	1	-1	-2	-3
0	-3	-2	-5	5	-2	2	-5
-2	-3	-4	-3	-1	-4	4	8
0	4	-2	-1	-1	-1	5	-2
0	0	1	3	8	4	6	-2
1	-2	0	5	1	1	4	-6
3	-4	0	6	-2	-2	2	2

Compressão de Imagem

Funcionamento da DCT e da Quantização

O bloco, cujas transformações estão representadas nas imagens seguintes é um bloco onde já existem variações bruscas dos coeficientes. Repare no resultado da DCT e como este se reflecte depois no erro final.

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

Compressão de Imagem

Funcionamento da DCT e da Quantização

Efectuando a DCT separável:

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

Compressão de Imagem

Funcionamento da DCT e da Quantização

Usando a tabela de quantização:

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Compressão de Imagem

Funcionamento da DCT e da Quantização

Procedendo agora à quantização inversa e à DCT inversa:

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

Compressão de Imagem

Funcionamento da DCT e da Quantização

O erro de final é:

0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

Compressão de Imagem

Funcionamento da DCT e da Quantização

DCT coding with increasingly coarse quantization, block size 8x8



quantizer stepsize
for AC coefficients: 25



quantizer stepsize
for AC coefficients: 100



quantizer stepsize
for AC coefficients: 200

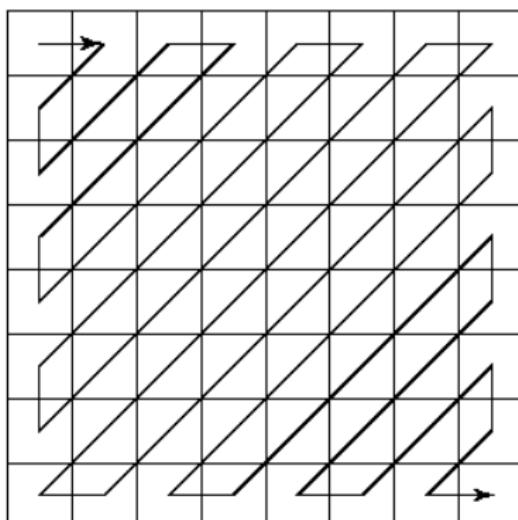
Compressão de Imagem

Codificação entrópica

- Os coeficientes DC e os 63 coeficientes AC são primeiro preparados para a codificação. Eles serão codificados em separado.
- O valor DC, como tem o valor mais elevado não será codificado tal e qual, mas codificaremos apenas a sua diferença com o DC do bloco precedente (DPCM Differential Pulse Code Modulation);
- Os coeficientes AC são linearizados e lidos segundo uma sequência em zigzag.

Compressão de Imagem

Codificação entrópica



Compressão de Imagem

Codificação entrópica

- Depois de linearizados é aplicado o método Run length que transforma a sequência em conjuntos do tipo: (# zeros a saltar, próximo valor não zero).
- Depois desta preparação aplicamos à sequência obtida uma codificação de Huffman ou codificação aritmética.
- No modo sequencial de base apenas existem duas tabelas de Huffman uma para os coeficientes DC e outra para os coeficientes AC.

Compressão de Imagem

Codificação entrópica: Primeiro passo (Run Length)

símbolo-1 (RUNLENGTH, SIZE)	símbolo-2 (AMPLITUDE)
--------------------------------	--------------------------

RUNLENGTH número de zeros consecutivos que precedem um AC diferente de zero. Aceita valores entre 0 e 15. Se o RUNLENGTH é maior do que 15 temos o (15,0) que representa 16 zeros consecutivos. Podemos ter no máximo 3 (15,0) seguidos. No caso de termos zeros até ao fim usamos o (0,0) que funciona como *EOB - fim de bloco*.

SIZE Número de bits usado para codificar AMPLITUDE usando a codificação de Huffman do JPEG

AMPLITUDE O valor do AC diferente de zero

Compressão de Imagem

Codificação entrópica: Primeiro passo (Run Length)

Relação SIZE - AMPLITUDE:

SIZE	AMPLITUDE
1	-1,1
2	-3,-2,2,3
3	-7,...,-4,4,...,7
4	-15,...,-8,8,...,15
5	-31,...,-16,16,...,31
6	-63,...,-32,32,...,63
7	-127,...,-64,64,...,127
8	-255,...,-128,128,...,255
9	-511,...,-256,256,...,511
A	-1023,...,-512,512,...,1023

Compressão de Imagem

Codificação entrópica: Primeiro passo (Run Length)

- Quanto aos coeficientes DC a estrutura final é similar aos AC.
- Neste caso o *símbolo-1* representa apenas o SIZE:

símbolo-1	símbolo-2
(SIZE)	(AMPLITUDE)
- Neste caso o SIZE pode ir até 11 assim à tabela acima acrescentamos uma linha.

Compressão de Imagem

Codificação entrópica: Segundo passo (Codificação de Huffman)

- Neste passo primeiro é codificado o coeficiente DC e depois os 63 AC's.
- Para AC's e DC o processo é semelhante. Cada *símbolo-1* é codificado usando a tabela de Huffman (VLC's).
- Cada *símbolo-2* é codificado usando também códigos de tamanho variável (VLI), mas não são códigos de Huffman.
- As tabelas de Huffman terão de ser especificadas externamente.

Compressão de Imagem

Codificação entrópica: Segundo passo (Codificação de Huffman)

Diferença dos Coef. DC	SIZE	Código	Tamanho final
0	0	00	2
-1,1	1	010	4
-3,-2,2,3	2	011	5
-7,...,-4,4,...,7	3	100	6
-15,...,-8,8,...,15	4	101	7
-31,...,-16,16,...,31	5	110	8
-63,...,-32,32,...,63	6	1110	10
-127,...,-64,64,...,127	7	11110	12
-255,...,-128,128,...,255	8	111110	14
-511,...,-256,256,...,511	9	1111110	16
-1023,...,-512,512,...,1023	A	11111110	18
-2047,...,-1024,1024,...,2047	B	111111110	20

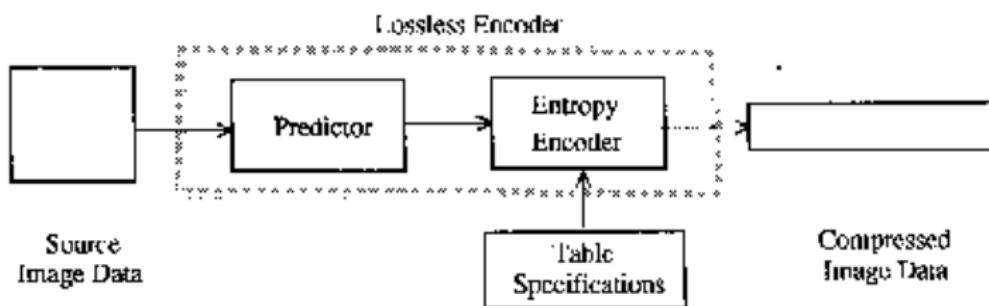
Compressão de Imagem

JPEG: Modo de codificação progressivo

- O passo de DCT e da quantização mantêm-se.
- A principal diferença é que cada componente da imagem é codificado em passagens múltiplas, em vez de uma única passagem.
- Existem dois métodos complementares de executar estas passagens:
 - No primeiro apenas um grupo de coeficientes da sequência zigzag são codificados em cada passagem.
 - No segundo método, são codificados primeiro os bits mais significativos de cada coeficiente, e nos passos seguintes vão-se descodificando os outros.

Compressão de Imagem

JPEG: Modo de codificação sem perdas



Compressão de Imagem

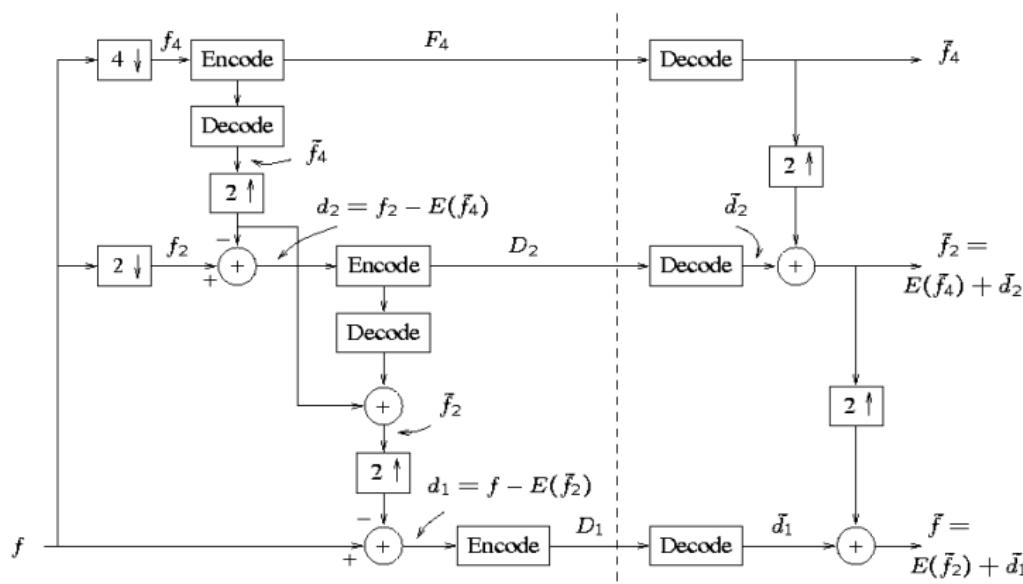
JPEG: Modo de codificação sem perdas

C	B			
A	X			

Valor Selecionado	Predição
0	Sem predição
1	A
2	B
3	C
4	$A+B-C$
5	$A+(B-C)/2$
6	$B+(A-C)/2$
7	$(A+B)/2$

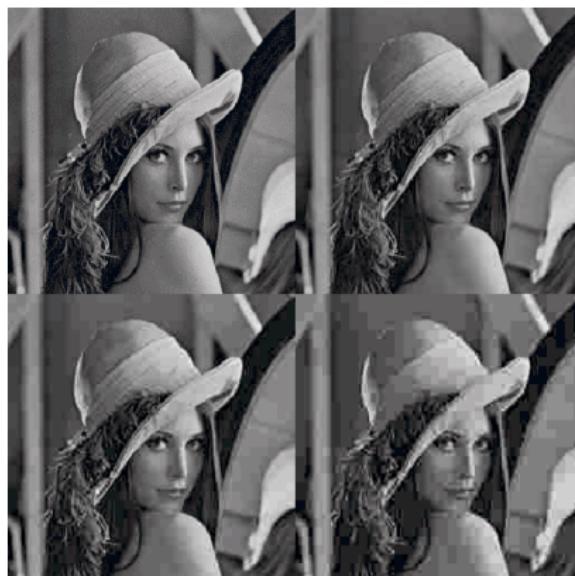
Compressão de Imagem

JPEG: Modo de codificação hierárquico



Compressão de Imagem

JPEG: Relação Débito-Qualidade



65536 Bytes (8 bpp);
3037 Bytes (0.37 bpp);

4839 Bytes (0.59 bpp);
1818 Bytes (0.22 bpp).

Compressão de Imagem

JPEG: Relação Débito-Qualidade



Compressão de Imagem

MJPEG (Motion JPEG)

- MJPEG é uma extensão do JPEG.
- Aplica-se a cada imagem as mesmas técnicas usadas para a imagem fixa.
- Este tipo de compressão foi adotado pelos novos formatos de DV (Vídeo Digital).

Compressão de Vídeo

- Tipos de sinais de vídeo.
- Vídeo analógico.
- Vídeo digital.
- Compressão de vídeo.
- Norma MPEG.
- Formatos de ficheiros de vídeo digital.

Tipo de sinais de vídeo

- Vídeo componente: utilizam três sinais de vídeo separados, um para o vermelho, um para o verde e um para para o azul;
- Vídeo composto - 1 sinal: crominância e luminância são misturados num único sinal (usa um único fio para transmissão);
- S-Vídeo - 2 sinais: usa dois sinais, um para a luminância e outro para a composição das duas crominâncias.

Vídeo analógico - Normas

- *PAL (Ocidental, China e Índia) e SECAM (França)*
- distinguem 625 (576 são destinadas à imagem).
- modo entrelaçado;
- 25 fps (frames por segundo);
- cor em YUV (Em PAL temos o U e V representados em sinais diferentes e em SECAM temos-los juntos num único sinal);
- *NTSC (América do Norte e Japão)*
- afeta apenas 480 (525 linhas para a representação da imagem).
- modo entrelaçado;
- 30 fps (frames por segundo);
- cor em YIQ (um único sinal);

Vídeo analógico - Normas

Modo progressivo vs interlaçado

50i



Only half of the image data is reproduced every 1/50 sec.

50p



All image data is reproduced in a picture every 1/50 sec.

Vídeo digital

Norma ITU-R-601 (1970)

- o número de pontos ativos por linha é de 720 para todos os standards;
- usa vídeo entrelaçado;
- foi adotado pelo SDTV (Standard Definition Television, formato 4:3).

Sub-amostragem da cor - ITU-R-601

- Luminância, reduzida de [0, 255] para [16, 235] (220 níveis, os restantes valores são utilizados para sincronização e tratamento do sinal):

$$Y' = 219(0.299R' + 0.58G' + 0.114B') + 16$$

- Crominância (225 níveis), definida por:

$$C_B = 128 + 112 \times \frac{1}{0.886} \times (B' - Y')$$

$$C_R = 128 + 112 \times \frac{1}{0.701} \times (R' - Y')$$

- Norma 4:2:2 - produção; normas 4:1:1 e 4:2:0 - difusão.

Outras normas

- SIF (Source Input Format):
 - PAL, SECAM: luminância 352×288 ; Crominância 176×144 ; 25 imagens/s;
 - NTSC: luminância 352×240 ; Crominância 176×120 ; 30 imagens/s;
- QSIF (Quarter SIF):
 - PAL, SECAM: luminância 176×144 ; Crominância 88×72 ; 25 imagens/s;
 - NTSC: luminância 176×120 ; Crominância 88×60 ; 30 imagens/s;

Outras normas

- Para permitir a difusão nas duas zonas, de 525 e de 625 linhas, foi definido um formato único: CIF (Common Intermediate Format):
 - não entrelaçado;
 - luminância 352×288 ; Crominância 176×144 ; 30 imagens/s;
 - Para um CIF não comprimido temos um débito binário de $(352 \times 288 + 176 \times 144 \times 2) \times 30 \times 8 = 36,5\text{ Mbits/s}$. Estes débitos são ainda excessivos para aplicações de vídeo conferência, vídeo sobre PC etc.

Outras normas

Outros formatos de base CIF

	Luminância Pixel por linha	Luminância Linha por imagem	Crominância Pixel por linha	Crominância Linha por imagem
SQCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

Compressão de vídeo

Na compressão de vídeo vamos tirar proveito das redundâncias:

- espaciais;
- psico-sensoriais;
- estatísticas;
- temporais.

Compressão temporal

- Explora as semelhanças existentes entre imagens sucessivas.
- Identifica as informações redundantes no tempo, mesmo se elas mudaram de lugar no espaço.
- Estas semelhanças podem ser constatadas entre
 - análise unidireccional: a imagem corrente e a imagem precedente;
 - análise bidireccional: entre a imagem corrente e as duas imagens que a englobam (a precedente e a seguinte).

Compressão temporal unidireccional

- Depois da primeira imagem de uma sequência podemos codificar apenas as diferenças que distinguem uma dada imagem da imagem anterior.
- Este processo não poderá ser efectuado para a primeira imagem da sequência.
- As imagens que são codificadas apenas espacialmente chamamos imagem *intra* ou *key frame*.
- Às imagem obtida da diferença de uma imagem com a precedente chamamos imagem *delta*.
- As imagens *delta* podem de seguida ser submetidas a uma compressão espacial.

Compressão temporal unidireccional

- Temos de dispor simultaneamente de duas imagens sucessivas.
- Para descomprimir a sequência o descompressor vai primeiro proceder à descompressão espacial antes de juntar as diferenças à imagem precedente para obter a imagem corrente.
- Poderíamos imaginar: a primeira imagem *intra* seguida de todas as outras *deltas*?

Compressão temporal unidireccional

- É obrigatória a introdução frequente de imagens *intra*.
- A frequência de imagens *intra* é determinada pelo dinamismo da sequência. Assim, uma sequência muito animada deverá incluir muito mais imagens *intra* que uma entrevista apresentando uma pessoa estática sobre um fundo fixo.
- Esta técnica é utilizada pelos algoritmos de compressão do tipo Cinepak, Indeo (Intel Video).

Compressão temporal bidireccional

- Um movimento lateral da câmara faz aparecer sobre a imagem elementos novos sobre os quais as imagens anteriores não continham qualquer informação.
- Neste caso, se procedemos a um cálculo diferencial a partir da imagem $n - 1$, o tamanho da imagem *delta* correspondente à imagem n é frequentemente grande, pois apenas alguns dos elementos são comuns às duas imagens.
- Mas se compararmos a imagem n não apenas com $n - 1$ mas também com $n + 1$, é possível aplicar uma codificação diferencial eficaz à imagem n na qual uma parte de elementos é comum com $n - 1$ e outra parte é comum com $n + 1$.

Compensação de movimento

- O movimento dos elementos de uma imagem à seguinte aumenta consideravelmente as diferenças.
- Limitar o volume das informações diferenciais a codificar.
- Não vamos calcular as diferenças de um bloco com o mesmo bloco na imagem anterior, ou seguinte, mas com os blocos que mais se aproximam do bloco que estou a tratar.
- Temos de guardar os movimentos em forma de vectores (vectores de movimento).
- A compressão bidireccional e a compensação de movimento são muito utilizadas pelos algoritmos de compressão de vídeo MPEG.

Normas

- IUT-T , em 1989 define a norma H.261: que já utiliza técnicas híbridas de compressão intra trama e de compressão inter trama graças ás primeiras técnicas elaboradas para estimar a compensação do movimento).
- MPEG ("Moving Picture Experts Group"): criou a norma de vídeo e áudio em fins de 1992 sob a referência ISO 11172 e batizada com o nome do grupo que a desenvolveu: MPEG 1.

Norma MPEG

- Classe geral de algoritmos de compressão híbridos do tipo predição-transformação:
 - a predição é aplicada para a compressão temporal;
 - a transformada DCT seguida da quantização dos coeficientes transformados assegura uma codificação espacial;
 - ao fim, uma codificação entrópica optimiza o código final.
- Assimétrica: a codificação é muito mais complexa (4 vezes mais longa) do que a descodificação.

Especificações MPEG

- MPEG-1: áudio e vídeo digital num único ficheiro, para armazenamento ou transmissão. Atinge débitos da ordem dos 1.5 Mbits/s (CD).
- MPEG-2: adaptação a débitos binários maiores (de 6 a 40 Mbits/s) e a maior número de formatos (4 vezes mais do que MPEG-1). Com uma representação de 50 tramas/s, MPEG-2 é destinado ao vídeo de qualidade broadcast e às aplicações HDTV (débitos binários de 20 a 40 Mbits/s).
- MPEG-4: para aplicações que dispõe de débitos muito baixos. Para aplicações gráficas interativas, multimédia interactiva (WWW, distribuição e acesso a conteúdos)
- MPEG-4 Parte 10 ou H264 - Codificação avançada de vídeo;
- MPEG-7 - Interface para descrição de conteúdos multimédia;

MPEG-1

- Objetivo: permitir a restituição, a partir de um CD de 72 minutos de vídeo digital no ecrã todo (imagem e som) ao ritmo de 25 ou 30 imagens por segundo.
- A primeira comercialização de produtos correspondentes a esta norma foi realizada pela Philips (CDI - Compact Disc Interactive) no início de 1994: consegui-se passar o vídeo digital para 1.44 Mbits/s, para a imagem e o som.
- Taxa de compressão de aproximadamente 140:1. Registamos no disco de suporte menos de 1% dos dados originais do vídeo.
- A qualidade proposta é comparável à obtida a partir do VHS.

MPEG-1

- Para conseguir baixos débitos a norma MPEG1 toma em conta todos os métodos de compressão:
 - Em primeiro usa a redução de formato usando o formato SIF que como já vimos propõe 352×288 pixéis de luminância e 176×144 pixéis em crominância.
 - Depois só toma uma imagem sobre duas do vídeo original (usando a representação entrelaçada).
 - Quanto à compressão espacial ela funciona tal como já funcionava o MJPEG.
 - É no tratamento das redundâncias temporais que MPEG propõe soluções novas.

Tratamento das redundâncias temporais

MPEG-1 define três tipos diferentes de imagem que serão tratadas usando procedimentos específicos:

- As imagens *intra* ou imagens de referência são simbolizadas pela letra **I**;
- As imagens preditas ou imagens **P**;
- As imagens bidirecccionais ou imagens **B**.

Tratamento das redundâncias temporais

- **Imagens I:**

- são comprimidas segundo um algoritmos JPEG.
- Estas imagens servem de referência para o calculo das imagens **P** e **B** e para a descodificação. É por isso que uma imagem em cada 12, ou menos deverá ser uma imagem intra.

Tratamento das redundâncias temporais

- **Imagens preditas, P:**

- São constituídas a partir da imagem I ou P precedente, usando vectores de movimento e cálculo de diferenças.
- Para diminuir o erro de predição, uma compensação de movimento é aplicada aos macro-blocos;
- Se o macro bloco é encontrado, ele é codificado como a sua diferença com os valores na imagem precedente e depois quantizado; finalmente, aplica-se uma codificação estatística aos valores obtidos depois da quantização.
- Se o macro-bloco não é encontrado, ele será codificado seguindo a norma JPEG como se se tratasse de uma imagem I;
- Os vetores de movimento são calculados segundo o deslocamento de cada bloco de pixéis duma imagem à outra, e depois codificados;

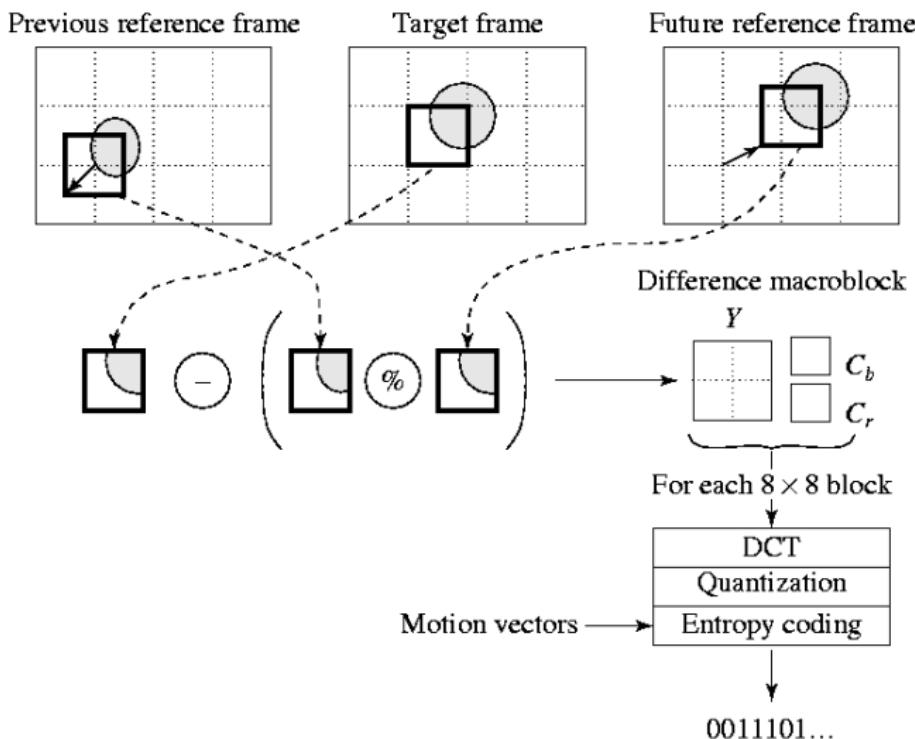
Tratamento das redundâncias temporais

- Evidentemente, e contrariamente às imagens **I** que são descodificadas como imagens JPEG, a descodificação das imagens **P** necessita que o descodificador guarde a imagem precedente. A descodificação da imagem atual é efetuada a partir dos macro-blocos codificados em direto e os macro-blocos codificados pelas diferenças.
- Na compensação de movimento a norma MPEG não especifica nem como é feita a procura, nem até que ponto a fazer, nem mesmo limite de proximidade a esperar. Estes parâmetros deverão ser definidos por cada implementação do algoritmo.

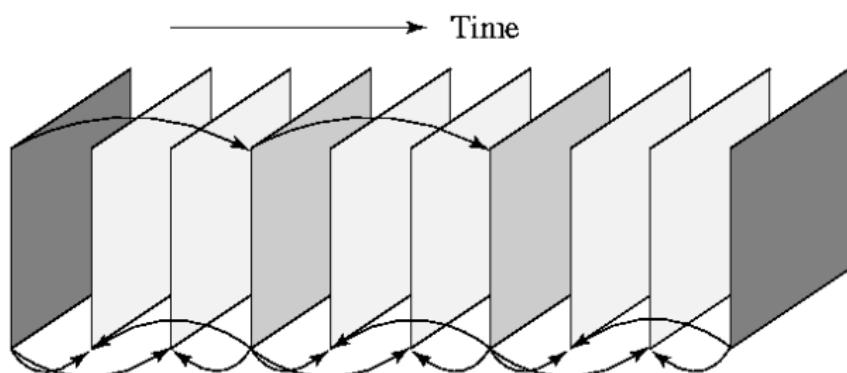
Tratamento das redundâncias temporais

- **Imagens do tipo B:**
 - são construídas a partir das imagens I ou P precedentes e seguintes.
 - Neste caso teremos vetores de movimento para a frente e para trás.
 - Cada bloco de pixels de uma imagem **B** toma como valor a média entre o mesmo bloco na imagem precedente e na imagem seguinte.
 - Para descodificar este tipo de imagens o descodificador terá de ter simultaneamente três imagens em memória. A **I** ou **P** precedentes, a **I** ou **P** seguintes e a **B** actual.

Tratamento das redundâncias temporais



Tratamento das redundâncias temporais



Display order

I B B P B B P B B I B

Coding and
transmission order

I P B B P B B I B B

Tipos de imagens e taxas de compressão

Tipos de imagem	Tamanho (em Ko)	Compressão
Intra (I)	18	7:1
Preditas (P)	6	20:1
Bidireccionais (B)	2.5	50:1
Média	4.8	27:1

O som na norma MPEG-1

- Uma sequência de vídeo integra necessariamente uma banda de som.
- O algoritmo de compressão é fundado sobre standard MUSICAM (Masking-pattern Universal Sub-band Integrated Coding ans Multiplexing).
- Permite obter um sinal estéreo com amostragem 48 kHz e quantificado sobre 16 bits (ou seja um débito de 2×768 Kbits/s).
- Permite quatro taxas de compressão (taxa 8 : 1 para obter uma qualidade final do tipo Hi-Fi).
- O princípio geral desta compressão áudio é relativamente simples: suprimimos do sinal sonoro as redundâncias psicosensoriais, o que quer dizer, tudo o que o ouvido humano não percebe ou percebe mal.

Sincronização

- Funcionando os codecs de áudio e de vídeo de forma completamente independente no tratamento dos seus dados específicos, será necessário sincronizar o fluxo de áudio e de vídeo;
- É por isso que o MPEG-1 é constituído por três partes: áudio, vídeo e sistema. Este último assegura a integração dos dois primeiros.
- MPEG-1 dispõe de um relógio de sistema a 90 Mhz que indica a hora aos codificadores áudio e vídeo. Os marcadores temporais gerados são incluídos no código e enviados ao destinatário que os utiliza para sincronizar o som e a imagem.

MPEG-2

- É uma extensão de MPEG-1 desenvolvido com a finalidade de obter uma qualidade superior de imagem e som.
- Os domínios de aplicação visados são a difusão de canais de televisão digital por cabo ou satélite, assim como a distribuição de alta qualidade sobre DVD (Digital Versatile Disc).
- Os procedimentos de compressão espaciais e temporais usados por MPEG-2 são diretamente tomados do MPEG-1.
- A diferença essencial entre estes dois standards consiste na definição própria ao MPEG-2 de vários níveis de qualidade e perfis de codificação específicos.

Os níveis de MPEG-2

Quatro níveis definem a resolução da imagem em MPEG-2:

- ① Nível baixo: Resolução 352×288 ;
- ② Nível médio: Resolução 720×576 ;
- ③ Nível alto-1440: Resolução 1440×1152 ;
- ④ Nível alto: Resolução 1920×1152 .

Os perfis de MPEG-2

Propõe 5 perfis diferentes:

- ① Perfil simples: apenas de imagens I e P;
- ② Perfil médio: esquema herdado de MPEG-1;
- ③ Perfil escalável em SNR: permite uma receção sobre dois níveis graças à organização de duas tramas distintas de dados. Permite assim que se descodifique apenas a primeira trama. A segunda vem melhorar o snr.
- ④ Perfil escalável espacialmente: Idêntico ao anterior, mas dispondo de uma trama de dados suplementares permitindo obter uma imagem em alta definição a partir de um formato TV 720×576 .
- ⑤ Perfil alto: dispões de todas as técnicas dos perfis anteriores, mas codifica o sinal vídeo 4:2:2 (em vez de 4:2:0 como nos outros perfis).

Combinação níveis - perfis no MPEG-2

Níveis	Perfis	Simples	Médio	Escalável em SNR	Escalável espacialmente	Alto
Alto	Não retido	80	Não retido	Não retido	100	
Alto-1440	Não retido	60	Não retido	60	80	
Médio	15	15	15	Não retido	20	
Baixo	Não retido	4	4	Não retido	Não retido	

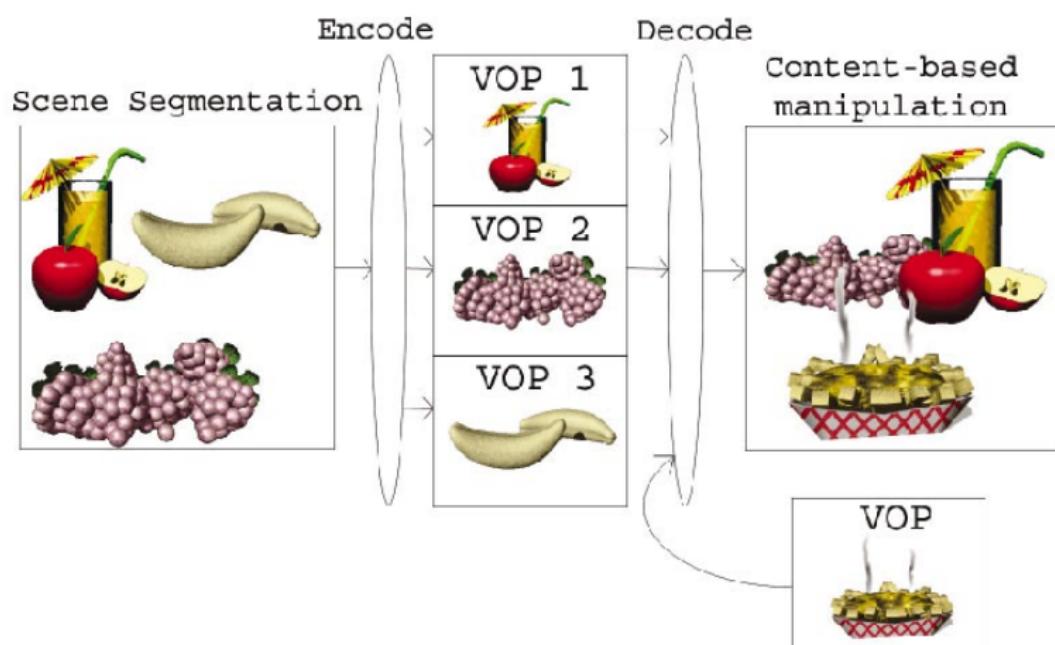
MPEG-4

- muito mais do que uma evolução de MPEG-1 e MPEG-2;
- nova abordagem da codificação e da compressão de imagem animada e do som, fundada sobre uma análise do conteúdo em objetos (visuais ou sonoros) estruturados hierarquicamente e susceptíveis de tratamentos específicos.
- permite interacção com o utilizador;
- permite uma melhor gestão e protecção dos direitos de autor.
- Exemplos de implementações: RelPlayer, QuickTime, GPAC/Osmo4, XviD, MPEG4IP, iTunes, VLC, 3ivx, Nero Digital, etc.

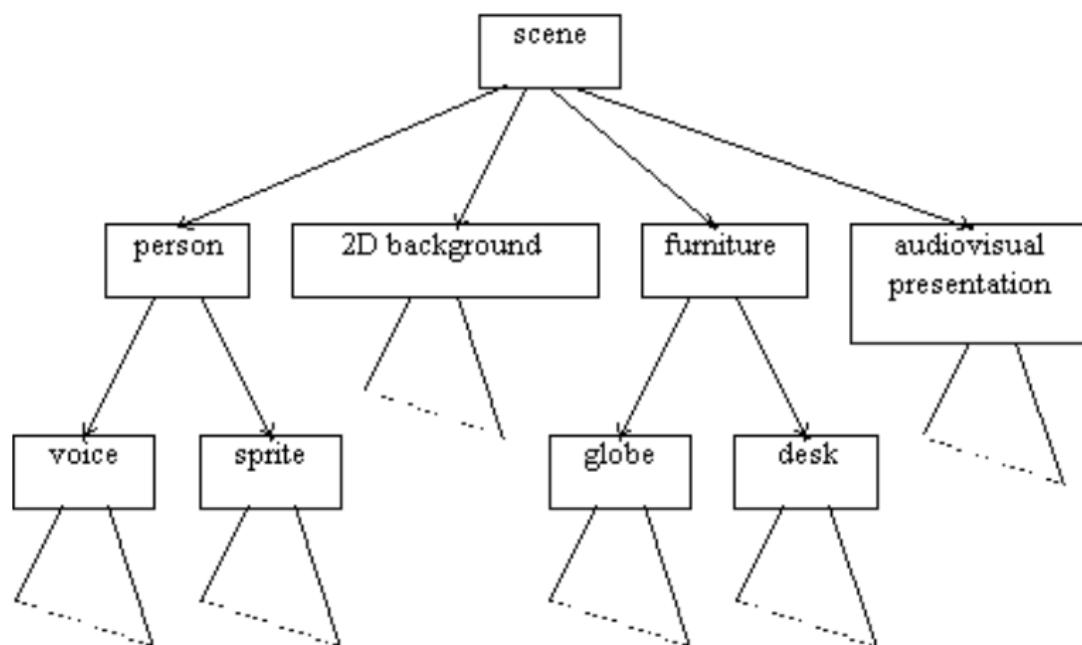
Decomposição da cena em objetos

- MPEG-4 propõe uma abordagem orientada ao objeto: uma cena de vídeo é decomposta em objetos que podem ser:
 - Imagens fixas, por exemplo um fundo fixo;
 - Objetos de vídeo, por exemplo uma pessoa que fala, sem o fundo;
 - Objetos áudio, por exemplo a voz associada com a pessoa, ou música de fundo;
 - Texto e gráficos;
 - Sons sintetizados;
 - rostos que falam, sintetizados, associados a texto usado para sintetizar um discurso e animar o rosto; corpos animados para juntar aos rostos;
- Estes objetos podem ainda ser compostos e depois estruturados e hierarquizados em árvore. Nas folhas destas árvores temos os objetos simples.

Decomposição da cena em objetos



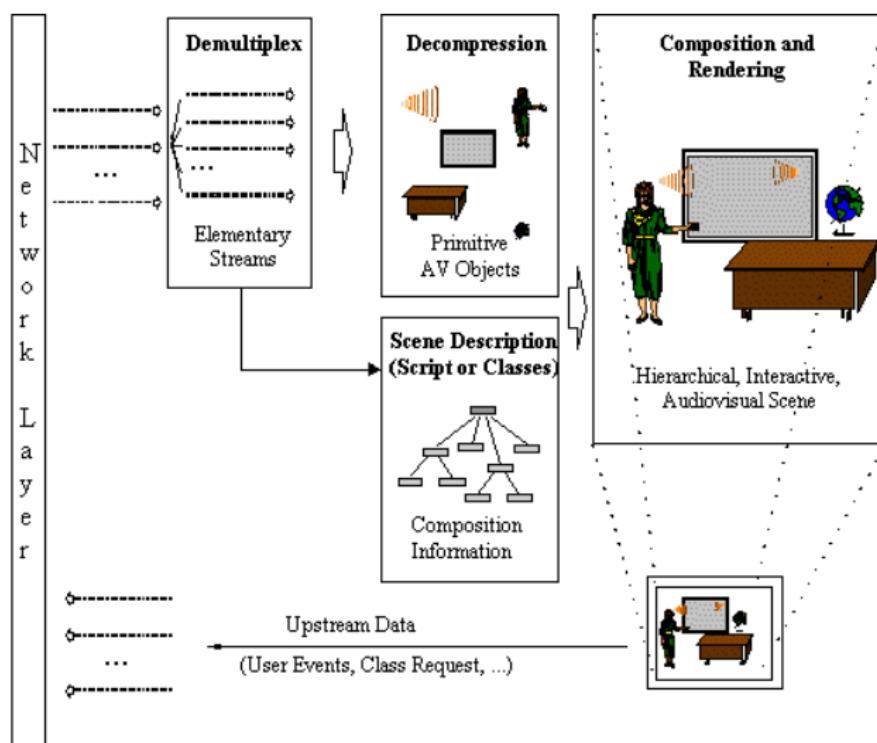
Composição de objetos



Descrição da cena

- MPEG-4 fornece uma normalização da descrição da cena. Esta permite por exemplo:
 - colocar objetos em qualquer lado num determinado sistema de coordenadas;
 - aplicar transformadas para alterar a aparência geométrica ou acústica de um objeto;
 - agrupar objetos simples para formar objetos compostos;
 - modificar atributos dos objetos;
 - alterar interativamente, os pontos de visualização e audição para qualquer ponto da cena.

Descrição da cena



Adaptação dimensional

- A exploração desta descrição detalhada da cena não é usada unicamente para a otimização da compressão objeto a objeto.
- A adaptação dimensional (scalability) permite otimizar a difusão do fluxo de dados em função do débito e/ou do sistema de representação que possui o utilizador final.

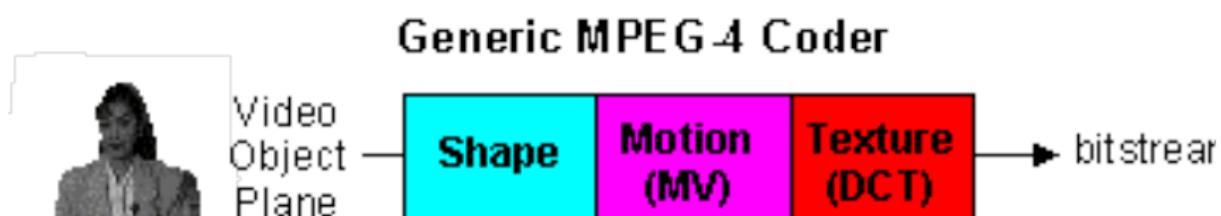
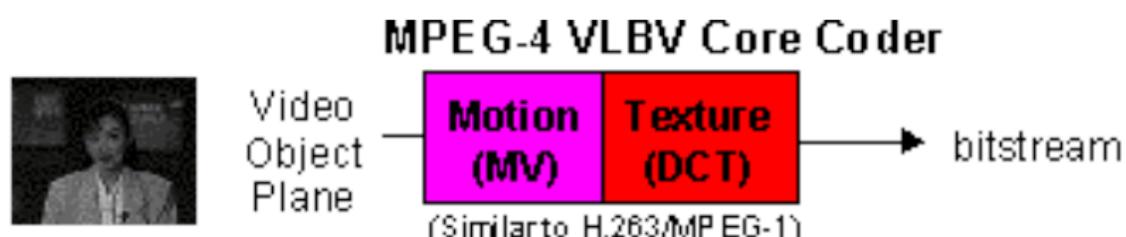
Interacção com os objetos

Dependendo do grau de liberdade fornecida pelo autor, o utilizador tem possibilidades de interacção com a cena.

Algumas operações que o utilizador poderá efectuar incluem:

- Mudar o ponto de visualização ou audição da cena;
- Mudar objetos da cena para outros locais;
- alterar a cor de um objeto apresentado sobre um site de comércio electrónico;
- Alterar o tamanho de texto sintetizado;
- Despoletar um conjunto de acontecimentos, clicando num objeto específico, por exemplo começar ou terminar uma sequência de vídeo;
- Escolher uma determinada linguagem quando várias estão disponíveis;
- etc.

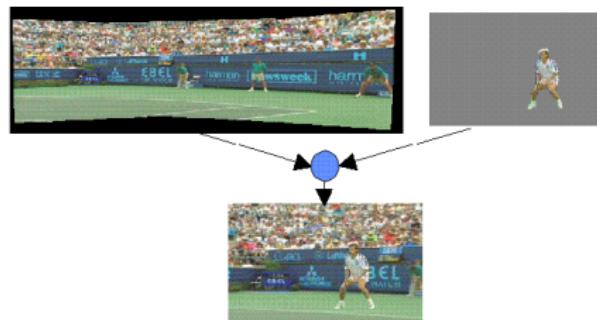
Codificação baseada nos objetos de vídeo (VOP)



Codificação baseada nos objetos de vídeo (VOP)

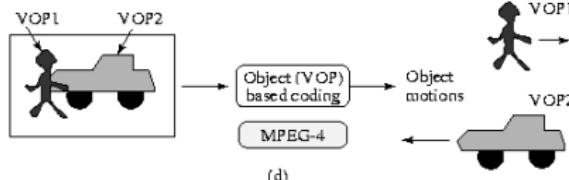
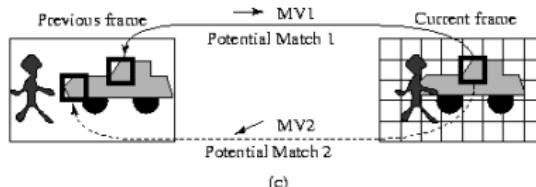
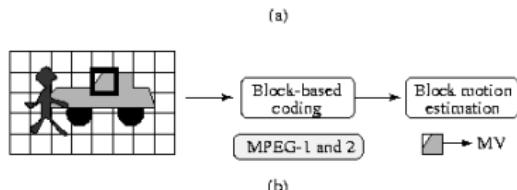
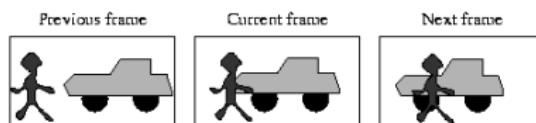
Algumas técnicas específicas da codificação baseada em conteúdos são:

- Compensação de movimento global para objetos de vídeo;
- Compensação global de movimento baseada em "sprites" estáticos.



- DCT adaptada à forma SA-DCT.

Codificação baseada nos objetos vs. codificação baseada em blocos



Codificação das texturas dos VOP

- DCT: As partes dos macro-blocos fronteira fora do VOP são preenchidas com zeros antes de serem enviados para a DCT;
- Shape Adaptive DCT (SA-DCT): é uma 2D DCT e é calculada como uma transformada 2D separável em duas iterações da 1D DCT-N.

Codificação de objetos sintetizados

- Animação de rostos;
- Animação de corpos;
- Malhas 2D animadas;
- Malhas 3D;



Perfis do MPEG-4

- perfil visual: cinco níveis de exigência para a codificação de dados naturais; quatro níveis para a codificação de dados sintetizados ou mistos;
- perfil áudio: com quatro níveis definidos pelo número de técnicas integradas e o débito requerido;
- perfis gráficos: com três níveis de tratamento de elementos gráficos e textuais;
- perfis de descrição da cena: quatro níveis determinam os tipos de informação suscetíveis de ser integradas na descrição de uma cena (desde apenas informações áudio até ao 3D);
- perfis de descrição de objetos: um nível único propondo as técnicas seguintes: descritor de objetos; sincronização; informação sobre os objetos, propriedade intelectual e proteção.

H.264 - Codificação avançada de vídeo

- H.261, H.263, H.263+ e H.26L.
- Em Dezembro de 2001 VCEG e MPEG formam o JVT (Joint Video Team) cuja missão é de finalizar um novo standard de codificação: H.264 ou AVC (Advanced Vide Coding), igualmente designado por MPEG-4 parte 10.
- mantém os elementos funcionais de base (predição, transformação, quantização, codificação entrópica); as alterações mais importantes dizem respeito aos detalhes funcionais de cada um dos elementos.
- é entre 2 a 3 vezes mais complexo em descodificação e 4 a 5 vezes mais complexo em codificação que o MPEG-2. Esta complexidade permite um aumento de eficiência que permite reduzir 39 % o débito binário relativamente ao MPEG-4 e 65% relativamente ao MPEG-2.

Características gerais do standard H.264/AVC

- codificação de vídeo no formato 4:2:0, progressivo ou entrelaçado;
- cada macro-bloco é transformado, quantizado e codificado;
- A estimação de movimento é enriquecida: é realizada no interior do macro-bloco, cuja resolução pode variar de 16×16 pixels a 4×4 pixels, passando por várias combinações intermédias.
- A escolha do tamanho dos sub-blocos tem um grande impacto na eficiência da compressão e sobre o tempo de cálculo.
- A transformação e a quantização operam sobre blocos de tamanho variável e não apenas blocos de 8×8 pixels;

Características gerais do standard H.264/AVC

- A transformação é baseada na DCT que se pode resumir a uma multiplicação matricial;
- Usa 52 quantizadores escalares, não lineares (a série começa em 0.625 e termina em 224 com um crescimento de 12.5% de um valor a outro);
- A codificação entrópica é enriquecida (CABAC - Context Adaptive Binary Arithmetic Coding);
- É usado um filtro na fase de descompressão para melhorar a qualidade visual subjectiva por redução do efeito de mosaico.

MPEG-7

Característica característica do dado.

Ferramenta de Descrição conjunto de Ds e DSs que descrevem a informação estrutural e conceptual do conteúdo.

D (Descriptor) definição (sintáctica e semântica) das características.

DS (Esquema de descrição) especificação da estrutura e relação entre os seus componentes.

DDL (linguagem de definição de descrição) regras sintáticas para exprimir e combinar DSs e Ds (XML extendido).

Objectivo do MPEG-7: normalizar os Ds, DSs e DDL para a descrição dos conteúdos.