

EULLER HENRIQUE BANDEIRA OLIVEIRA  
11821BSI210

Resumo :  
Aula 03




## Representação da imagem digital

### Imagem

- $f(x,y) = i(x,y) \cdot r(x,y)$ 
  - Função de intensidade luminosa ->  $f(x,y)$ 
    - Fonte de iluminação ->  $i(x,y)$ 
      - Energia eletromagnética
        - Ex: Luz
          - É percebida pelo olho humano
            - Violeta
            - Azul
            - Verde
            - Amarela
            - Laranja
            - Vermelho
            - Monocromática/acromática
              - Intensidade/Nível de cinza
              - Radiância (energia emitida)
              - Luminância (energia percebida)
              - Brilho
    - Ultrassom etc..
  - Reflexão de energia ou Absorção de energia ->  $r(x,y)$

### Aquisição da imagem: Sensores

#### Sensor

- Energia elétrica  

- Filtro do sensor
  - Melhora a seletividade do sensor
    - Ex: Enfatiza uma banda do espectro  

- Material do sensor
  - Detecta um tipo particular de energia  

- Voltagem
  - Proporcional a luz que entra

## Faixa de sensores

- Obtém slices de objetos 3d
- Utilizado em: medicina e indústria
- Algoritmos de reconstrução transformam a saída do sensor em imagens seccionais (slices)
- Ex: PET, MRI e CT

## Array de sensores

- Sensor: CCD (Charge Coupled Device)
  - Resposta proporcional a integral da energia gerada pela luz que incidiu na superfície do sensor.
  - É utilizado em câmaras digitais

### Digitalização de uma imagem

- Passo 1

- Amostragem

- Teoria:

- Extrai o valor presente em cada elemento imagem (pixel: picture element)
- Armazena tais valores em uma matriz  $M$  (quantidade de pixels presentes no eixo  $x$ )  $\times$   $N$  (quantidade de pixels presentes no eixo  $y$ )
- O seguinte cálculo é feito para determinar o quanto uma imagem precisa para armazenar uma imagem:  $M \times N \times k$  bits ou  $N^2 \times k$  bits para matrizes quadradas
- Obs: O tamanho da imagem não define a sua qualidade
  - O que define é a resolução espacial e a resolução de intensidade

- Resolução espacial

- Medida do menor detalhe discernível em uma imagem
- É medida em partes de linhas por unidade de distância
- Pontos (pixels) por unidade de distância (dpi)

- Ex:

- jornal: 75 dpi
- revista: 133 dpi
- livros: 2400 dpi

- Resolução de intensidade

- Medida da menor alteração discernível nos níveis de intensidade ( $2^k$ )

## Prática:

### Reamostragem de pixels

#1250 dpi  
#3692x2812

```
img1 = imread('relogio.tif');  
subplot(2,2,1);  
imshow(img1);  
title("1250dpi");
```

#300dpi  
#888x675

```
img2 = imresize(img1, [888 675]);  
subplot(2,2,2);  
imshow(img2);  
title("300dpi");
```

#150dpi  
#444x337

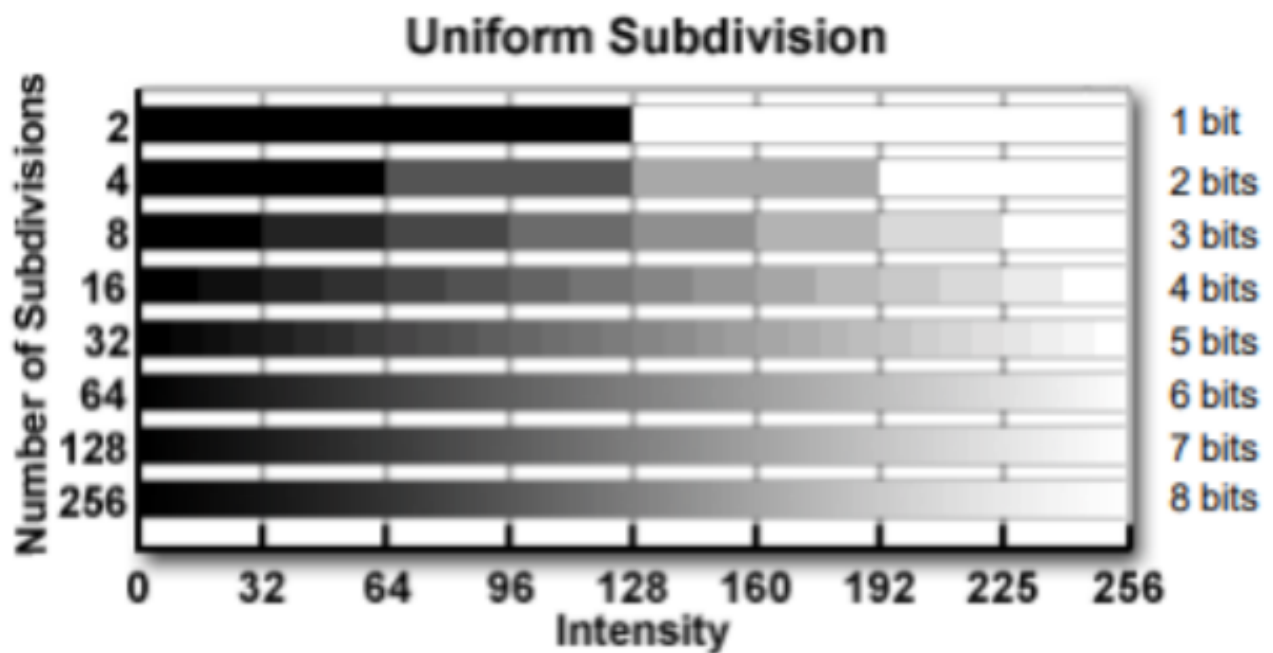
```
img3 = imresize(img1, [444 337]);  
subplot(2,2,3);  
imshow(img3);  
title("150dpi");
```

#72dpi  
#213x162

```
img4 = imresize(img1, [213 162]);  
subplot(2,2,4);  
imshow(img4);  
title("72dpi");
```



- Passo 2:
  - Quantização
    - Altera o nível de cinza ( $L=2^k$ ) de uma imagem monocromática
    - Teoria:
      - Divida a intensidade máxima presente em uma imagem monocromática (256) pela quantidade desejada de níveis de cinza
      - O resultado da divisão determina os intervalos e os valores que serão inseridos dentro de cada intervalo
      - Por exemplo:
        - Se  $k = 4$
        - $256/4 = 64$
        - Intervalo 1: 1-64
        - Valor que o intervalo receberá: 0
        - Intervalo 2: 64-128:
        - Valor que o intervalo receberá: 64
        - Intervalo 3: 128-192
        - Valor que o intervalo receberá: 128
        - Intervalo 4: 192-256
        - Valor que o intervalo receberá: 192



**Figure 1**

- Prática:

```

q2_img = 0;
nbits = 0;
pos = 0;

function quant(q2_img, nbits, pos)

    [x, y] = size(q2_img);

    #Número de bits
    nb = nbits;

    #Número de níveis
    quantificadores
    nq = 2^nb;

    lim_inf = 0;
    lim_sup = 256/nq;

    for i = 1: nq

        if(i > 1)

            limites_superiores(i) =
            limites_superiores(i-1) + lim_sup;
            limites_inferiores(i) =
            limites_inferiores(i-1) + lim_sup;

        else

            limites_superiores(i) = lim_sup;
            limites_inferiores(i) = lim_inf;

        end

    end

end

for i = 1:x

    for j = 1:y

        xy = q2_img(i,j);

        for k = 1:nq

            if(xy >=
            limites_superiores(nq-1) && xy <=
            256)

                q2_img(i,j) = 256;
                break

            elseif(xy <=
            limites_superiores(k))

                q2_img(i,j) =
            limites_inferiores(k);
                break

            end

        end

    end

    subplot(2,2, pos);
    imshow(q2_img);
    title([num2str(nbits) " bits"]);

end

q2_img = imread('ctskull-256.tif');

figure;

#1 bit
quant(q2_img, 1, 1);

#2 bits
quant(q2_img, 2, 2);

#3 bits
quant(q2_img, 3, 3);

#4bits
quant(q2_img, 4, 4);

figure;

#5 bits
quant(q2_img, 5, 1);

#6 bits
quant(q2_img, 6, 2);

# 7 bits
quant(q2_img, 7, 3);

#8 bits

subplot(2,2,4);
imshow(q2_img);
title("8 bits")

```

