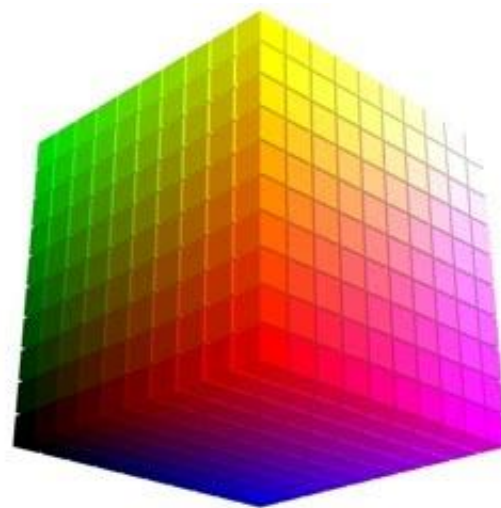




# PROCESSAMENTO DIGITAL DE IMAGENS



Projeto de Ensino Material didático sobre processamento digital de imagens  
Departamento de Ciências Exatas e Tecnológicas - DCET  
Discente - Luciana Roncarati - Ciência da Computação

# SUMÁRIO

- Definição
- Algoritmo Interface *Processing*
- Histograma da imagem original.
- Histograma da imagem equalizada.
- Referências Bibliográficas

# DEFINIÇÃO

- Também conhecida como “Linearização de Histograma”, esta técnica tem a finalidade de obter um histograma uniforme, através do espalhamento da distribuição dos níveis de cinza. Aparentemente comum, esta operação é muito poderosa, conseguindo, muitas vezes, recuperar imagens consideradas perdidas. Há vários métodos empregados para a realização da equalização, sendo que muitos deles são baseados em distribuições estatísticas.

# DEFINIÇÃO

$$I = \frac{m \times n}{g}$$
$$q = \max \left\{ 0, \text{arred} \left( \frac{\sum_{j=0}^k n_j}{I} \right) - 1 \right\} \quad 0 \leq k \leq g$$

onde:

- ***I*** é a quantidade ideal de pixels em cada nível de cinza da escala usada
- ***n*** é a quantidade de linhas da imagem
- ***m*** é a quantidade de colunas da imagem
- ***g*** é a quantidade total de níveis de cinza da escala
- ***q*** é o novo nível de cinza procurado
- ***n<sub>j</sub>*** é a quantidade de pixels no nível de cinza *j*
- ***arred*** indica o arredondamento do resultado obtido pela expressão que constitui seu argumento.

# ALGORITMO DE EQUALIZAÇÃO DO HISTOGRAMA

```
size(500, 500);

//Carrega uma imagem do diretório de dados
PImage img = loadImage("Toyokawa.jpg");
image(img, 50, 50);
int[] hist = new int[256];
float quantideal;
int g = 256;
float q = 0;
int aux = 0;
float somafreq = 0;

PrintWriter output;
output = createWriter("equalizacao.txt");
// calcula o histograma
for (int i = 0; i < img.width; i++) {
    for (int j = 0; j < img.height; j++) {
        int bright = int(brightness(get(i, j)));
        hist[bright]++;
    }
}
output.println( "\nFREQUENCIA DE CADA TOM DE PIXEL\n");
for (int i = 0; i < 256; i++) {
    output.println(i + "\t" + hist[i]);
}
```

```
quantideal= (img.width *img.width)/g;

output.println( "\nFREQUENCIA ACUMULADA\n");
for (int i = 0; i < 255; i++) {
    somafreq += hist[i];
    output.println(i + "\t = " + somafreq);
    q = ((somafreq/quantideal) - 1);
    aux = int (q);
    output.println("NOVO NIVEL DE CINZA PROCURADO " + q );
}

output.println("\nQUANTIDADE IDEAL 'I' = " + quantideal);
//output.println(quantideal);

output.flush();
output.close();
exit();
```

Fig. 1 – interface *Processing*

# HISTOGRAMA DA IMAGEM ORIGINAL

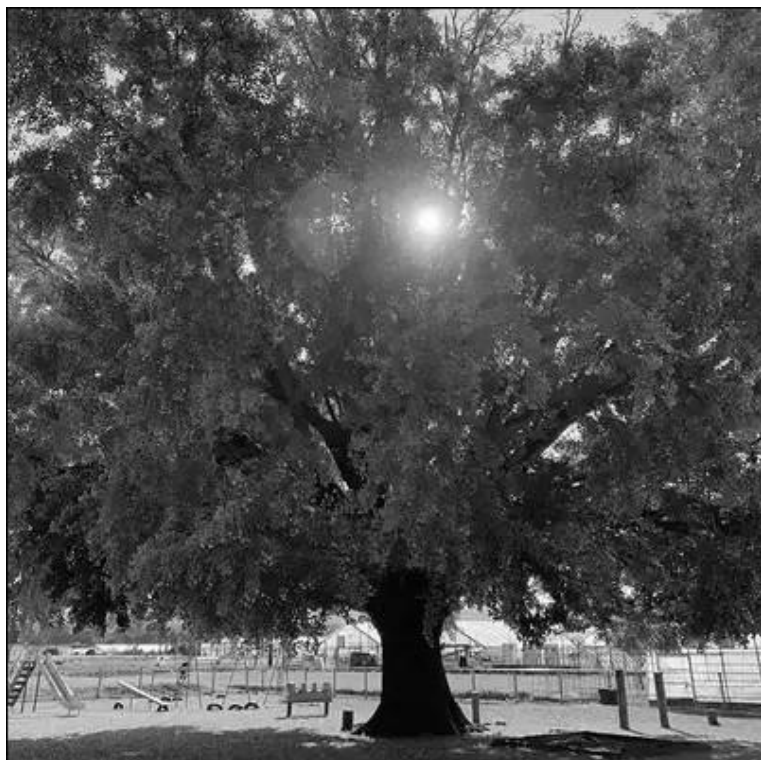


Fig. 2 – Imagem original

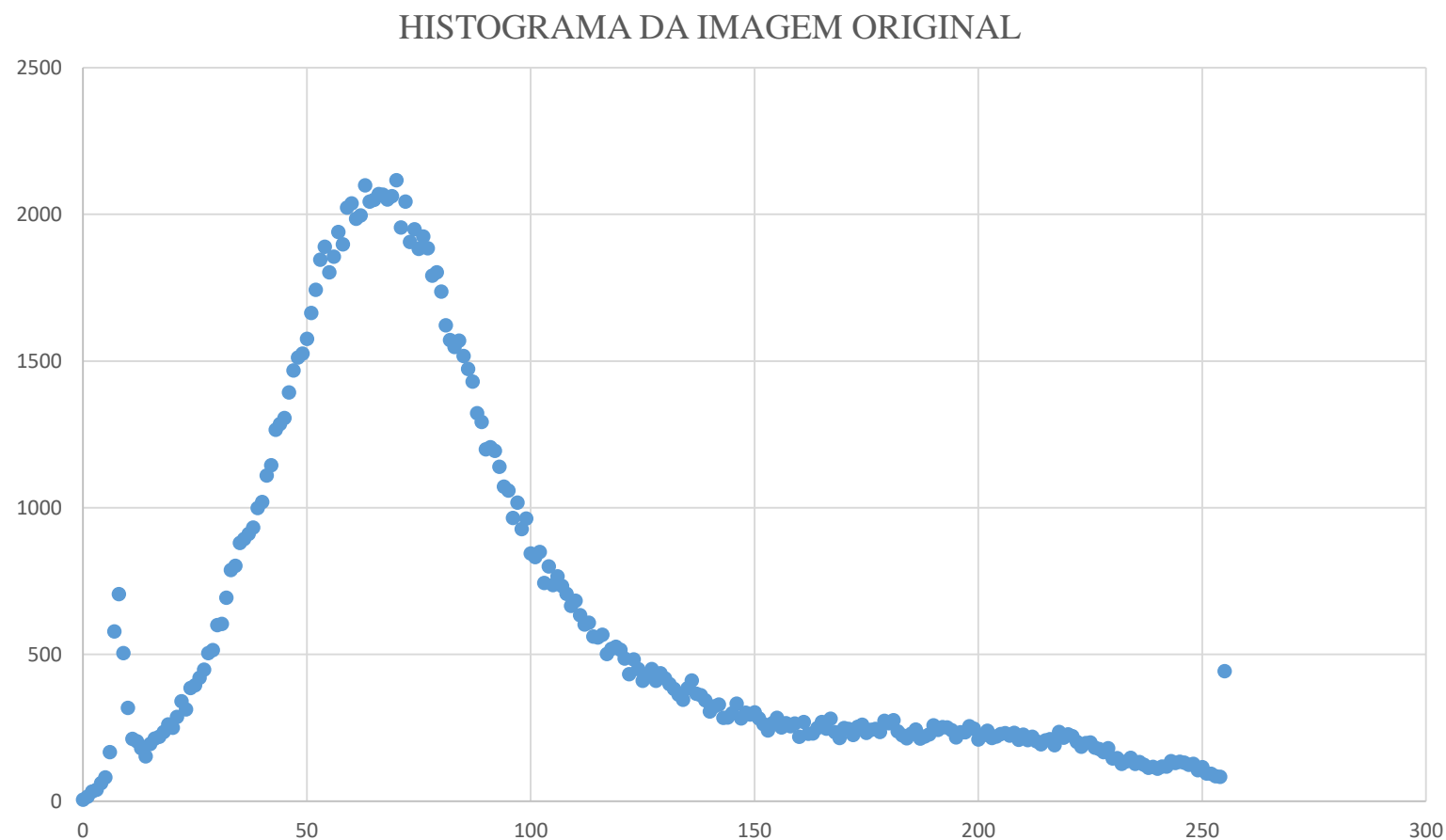


Fig. 3 – Histograma da imagem original

# HISTOGRAMA DA IMAGEM EQUALIZADA

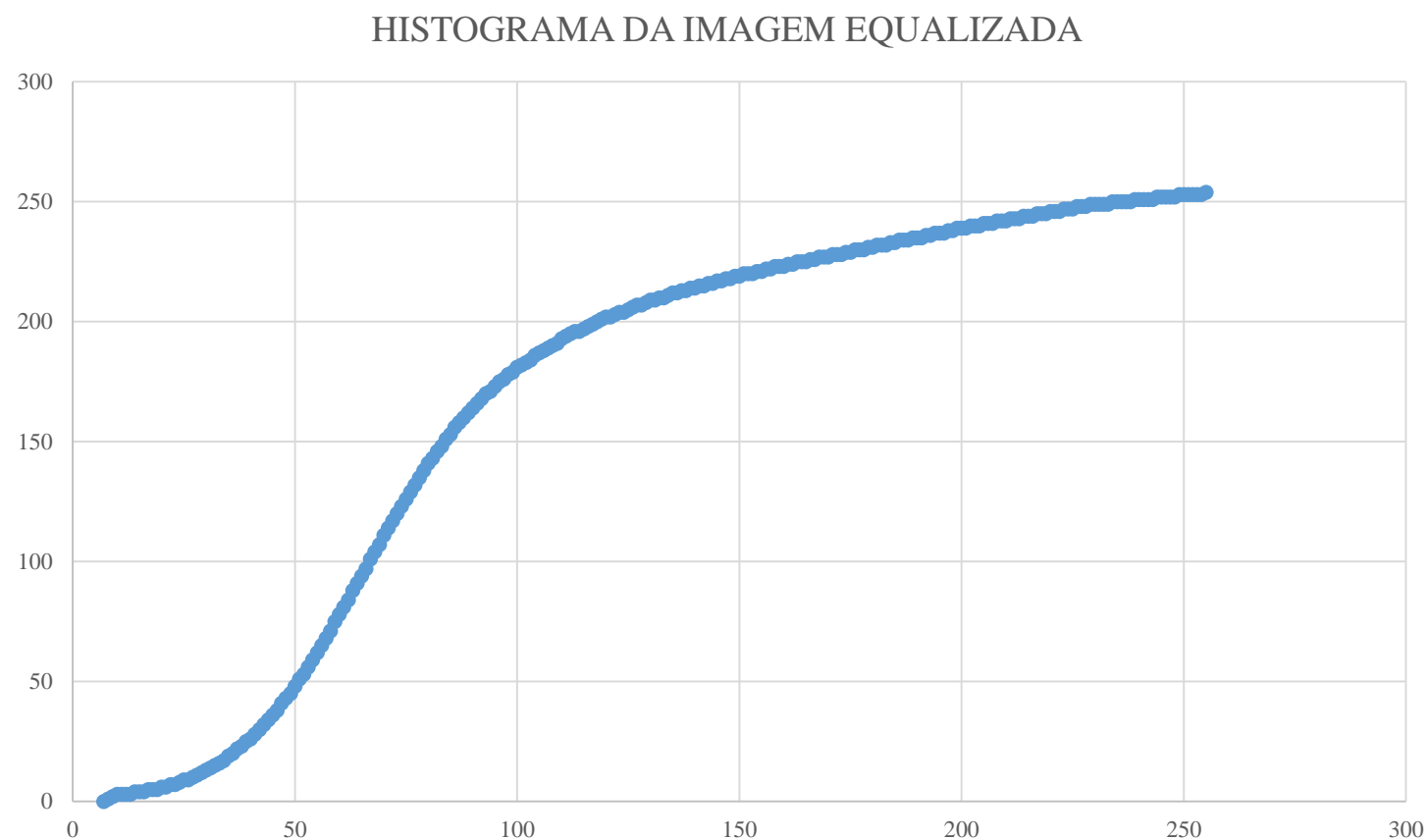


Fig. 3 – Histograma da imagem equalizada

# REFERÊNCIAS BIBLIOGRÁFICAS

- NUNES L. S, Fátima - Introdução ao processamento de imagens médicas para auxílio ao diagnóstico - uma visão prática, capítulo 2.
- GONZALEZ C, Rafael. e WOODS, Richard - Processamento digital de imagens - 3. Ed. Pearson Prentice hall, São paulo,2010.