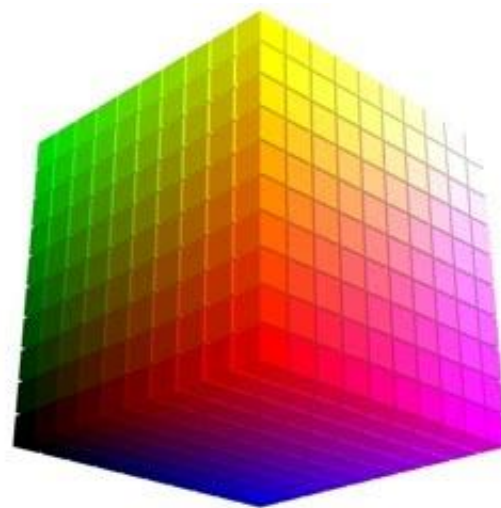




# PROCESSAMENTO DIGITAL DE IMAGENS



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*
- Exemplo

# DEFINIÇÃO

- O método de Otsu é um algoritmo de limiarização de imagem que busca maximizar a variação entre as classes binárias (claros e escuros) da imagem para determinar o melhor limiar. É baseado na teoria de otimização e é fácil de implementar. É amplamente utilizado em processamento de imagem, mas pode não ser apropriado para imagens com distribuições de intensidade de cor complexas.

# ALGORITMO DE LIMIAÇÃO - OTSU

```
size(400, 400);
PImage img = loadImage("Toyokawa.jpg");
//PImage imgh = createImage(400, 400, RGB);
image(img, 0, 0);

// Converter a imagem para escala de cinza
img.filter(GRAY);

// Calcular o histograma da imagem em escala de cinza
int[] hist = new int[256];
for (int i = 0; i < img.pixels.length; i++) {
    int p = (int) brightness(img.pixels[i]);
    hist[p]++;
}
// Calcular a probabilidade acumulada
int[] cumulativeHist = new int[256];
cumulativeHist[0] = hist[0];
for (int i = 1; i < 256; i++) {
    cumulativeHist[i] = cumulativeHist[i - 1] + hist[i];
}
```

Fig. 1 – interface Processing

# ALGORITMO DE LIMIAÇÃO - OTSU

```
// Calcular a média das classes
float[] mean = new float[256];
for (int i = 0; i < 256; i++) {
    mean[i] = (float) (i * hist[i]) / cumulativeHist[255];
}

// Calcular a variação inter-classe
float[] interClassVariance = new float[256];
float overallMean = 0;
for (int i = 0; i < 256; i++) {
    overallMean += mean[i] * hist[i];
}
overallMean /= img.pixels.length;
for (int i = 0; i < 256; i++) {
    interClassVariance[i] = (float) (cumulativeHist[i] * (img.pixels.length - cumulativeHist[i])) / (img.pixels.length * img.pixels.length);
    interClassVariance[i] *= (mean[i] - overallMean) * (mean[i] - overallMean);
}
```

Fig. 2 – interface Processing

# ALGORITMO DE LIMIARIZAÇÃO - OTSU

```
// Encontrar o limiar de Otsu
int threshold = 0;
float maxVariance = 0;
for (int i = 0; i < 256; i++) {
    if (interClassVariance[i] > maxVariance) {
        maxVariance = interClassVariance[i];
        threshold = i;
    }
}
// Aplicar a limiarização
for (int i = 0; i < img.pixels.length; i++) {
    int p = (int) brightness(img.pixels[i]);
    if (p < threshold) {
        img.pixels[i] = color(0);
    } else {
        img.pixels[i] = color(255);
    }
}
image(img, 0, 0);
save("Toyokawa_otsu.jpg");
exit();
```

Fig. 3 – interface Processing

# EXEMPLO



Fig. 4 – Imagem original

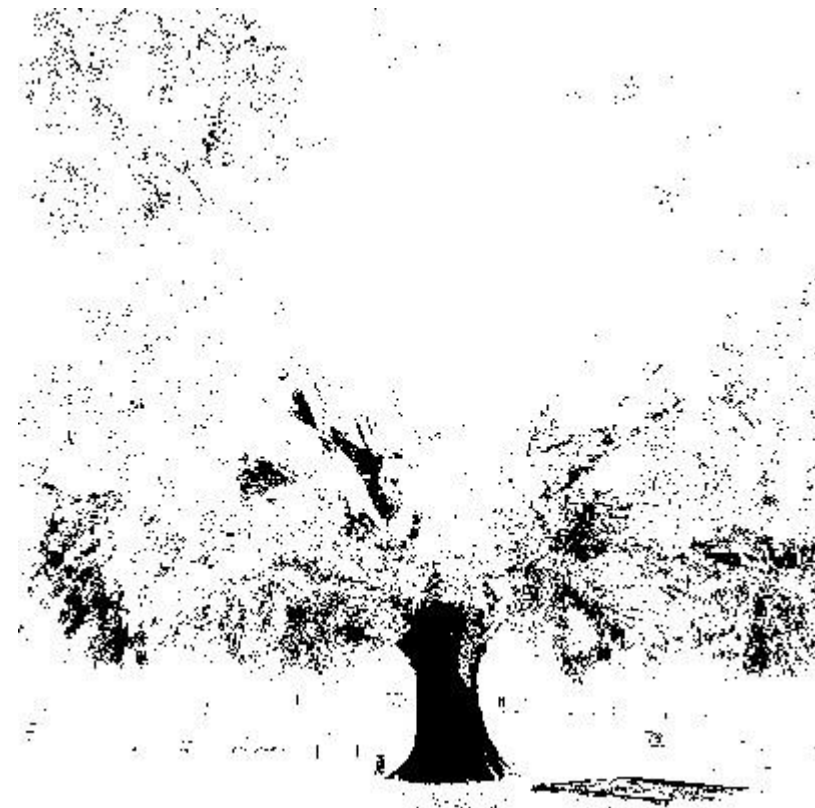


Fig. 5 – Imagem método limiarização - OTSU



# 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.
- [https://pt.wikipedia.org/wiki/Limiariza%C3%A7%C3%A3o\\_por\\_equil%C3%ADbrio\\_do\\_histograma](https://pt.wikipedia.org/wiki/Limiariza%C3%A7%C3%A3o_por_equil%C3%ADbrio_do_histograma)
- <https://chat.openai.com/chat>