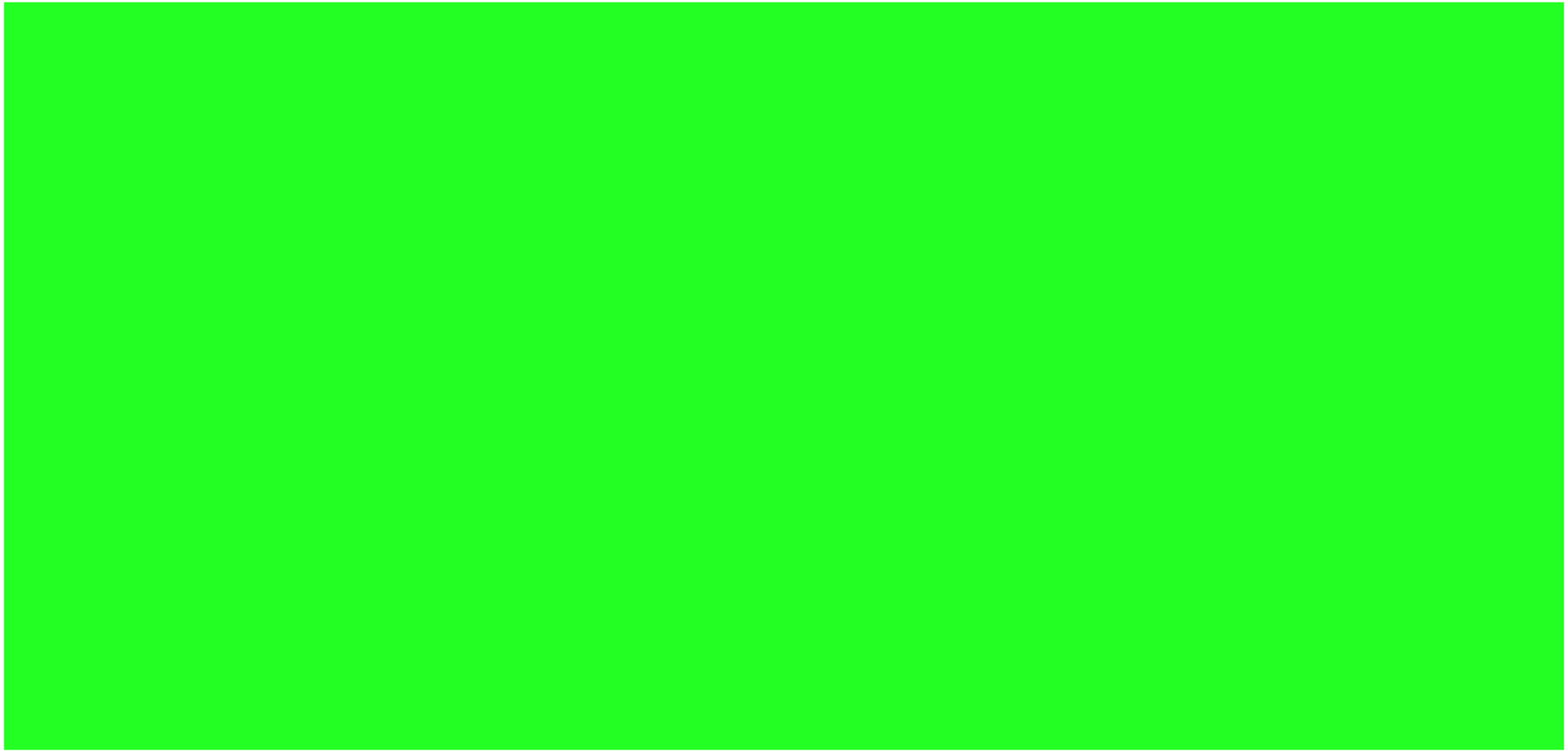
Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu



UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Hoje

- Algoritmo de Otsu.
- *Chroma key*.

Algoritmo de Otsu

•Objetivo: definir automaticamente um limiar para binarização de uma imagem em escala de cinza.

•Conceito: o melhor limiar é aquele que, separando os pixels em duas classes, *maximiza a variância inter-classe*.

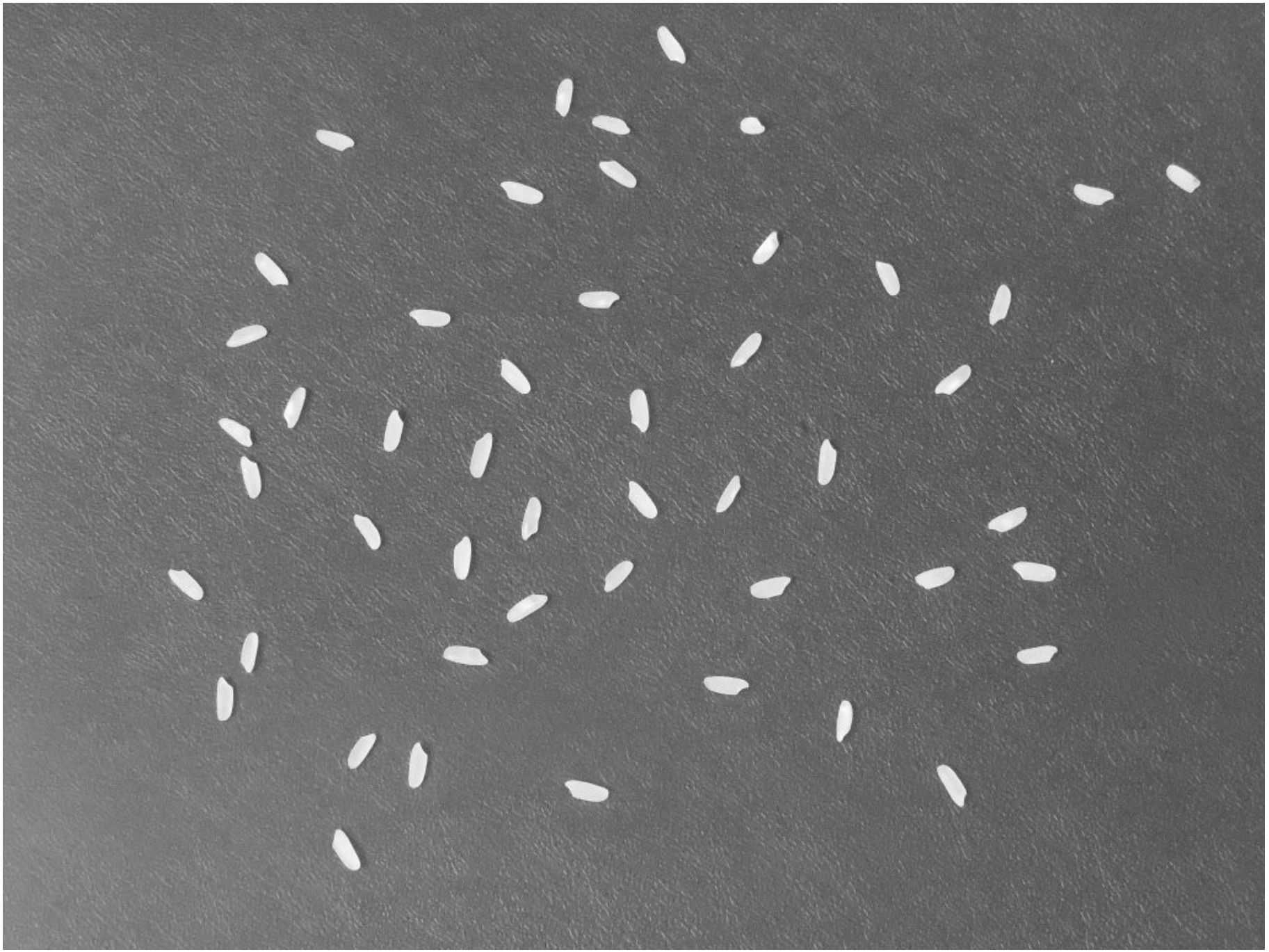
- ???!!!

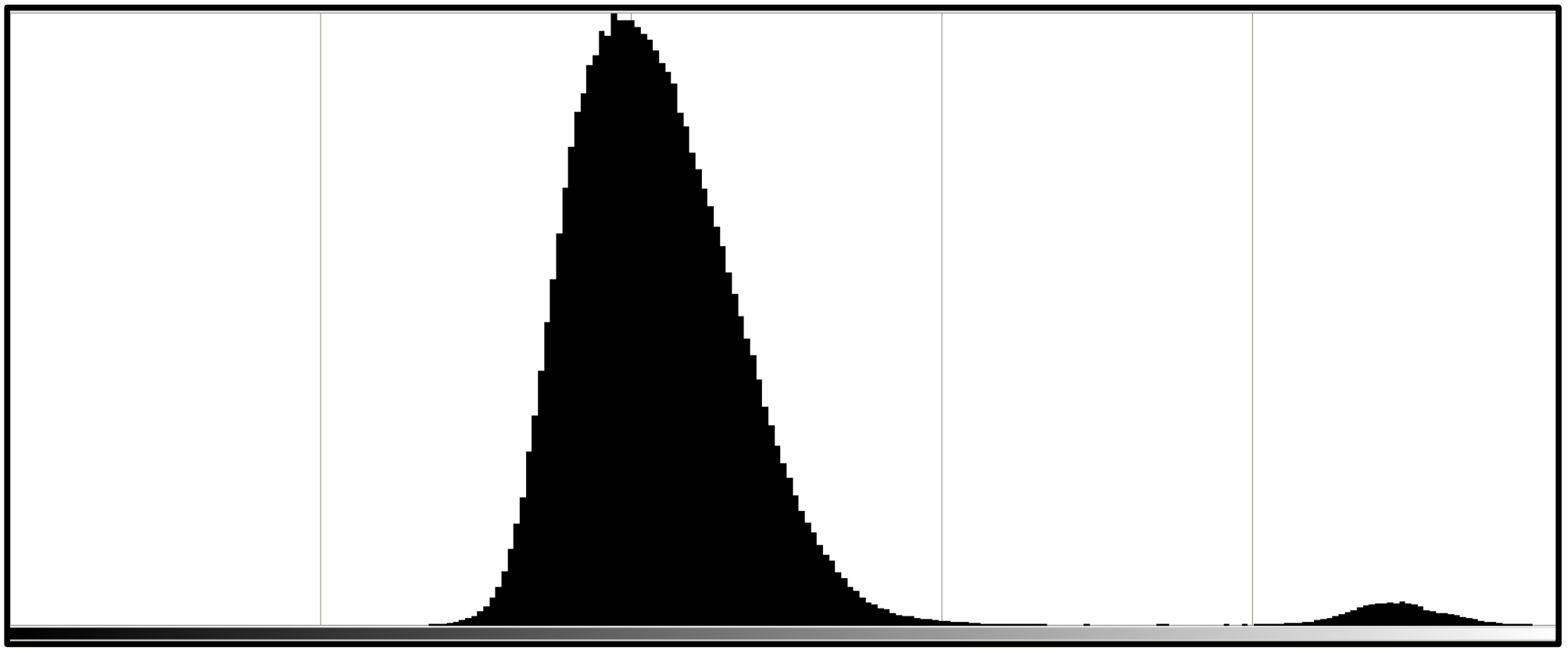
•Versão alternativa: o melhor limiar é aquele que, separando os pixels em duas classes, *minimiza a variância intra-classe*.

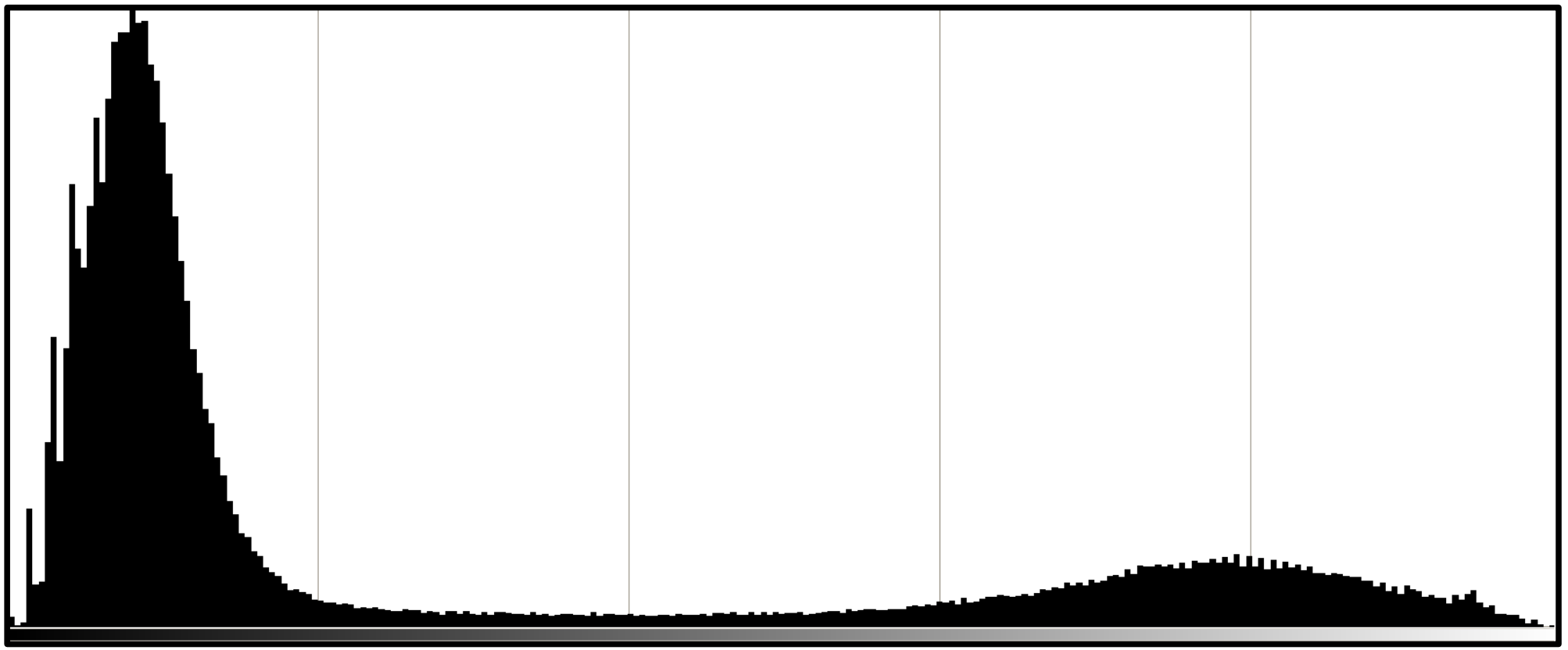
- ???!!!

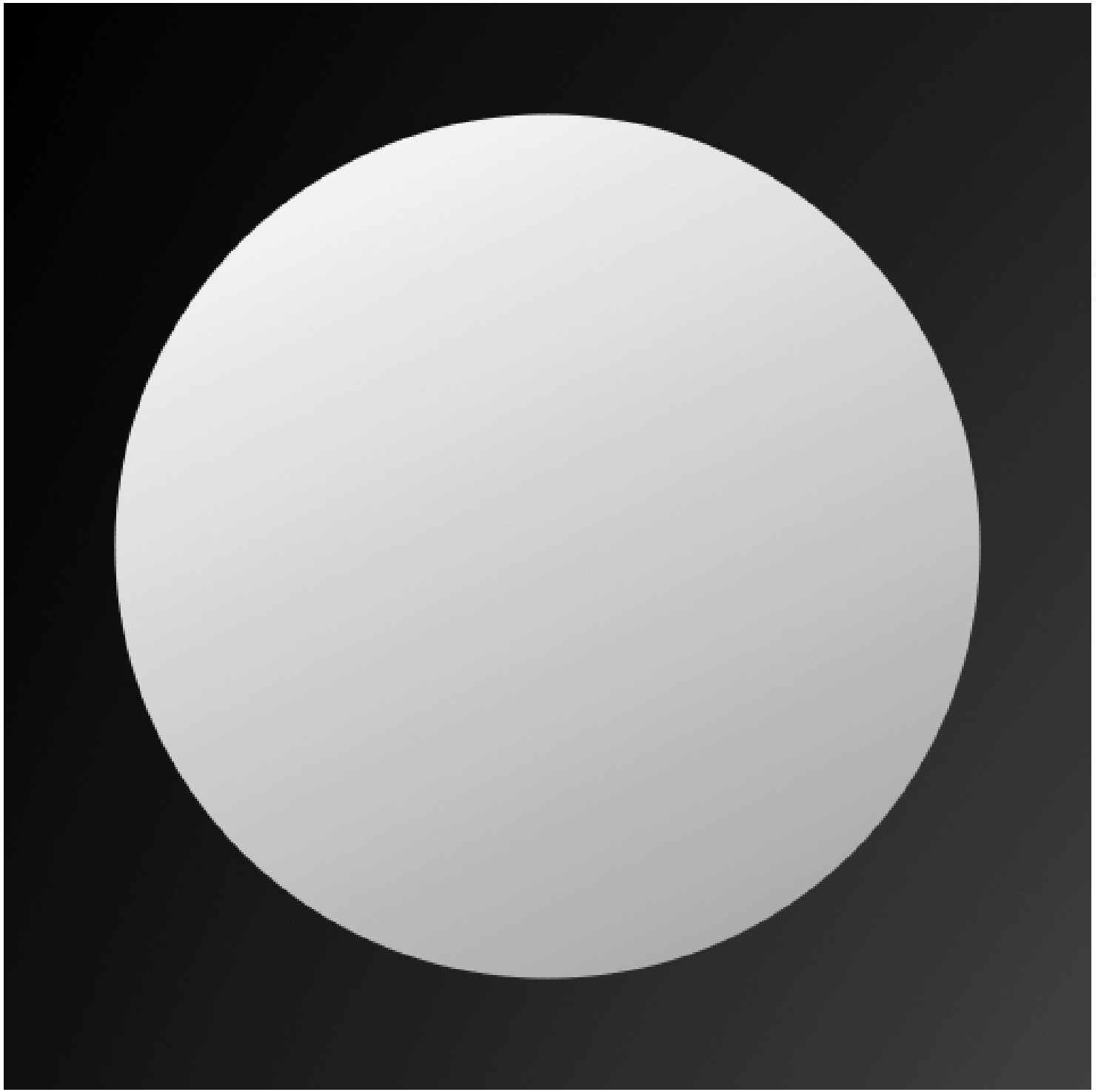
Algoritmo de Otsu

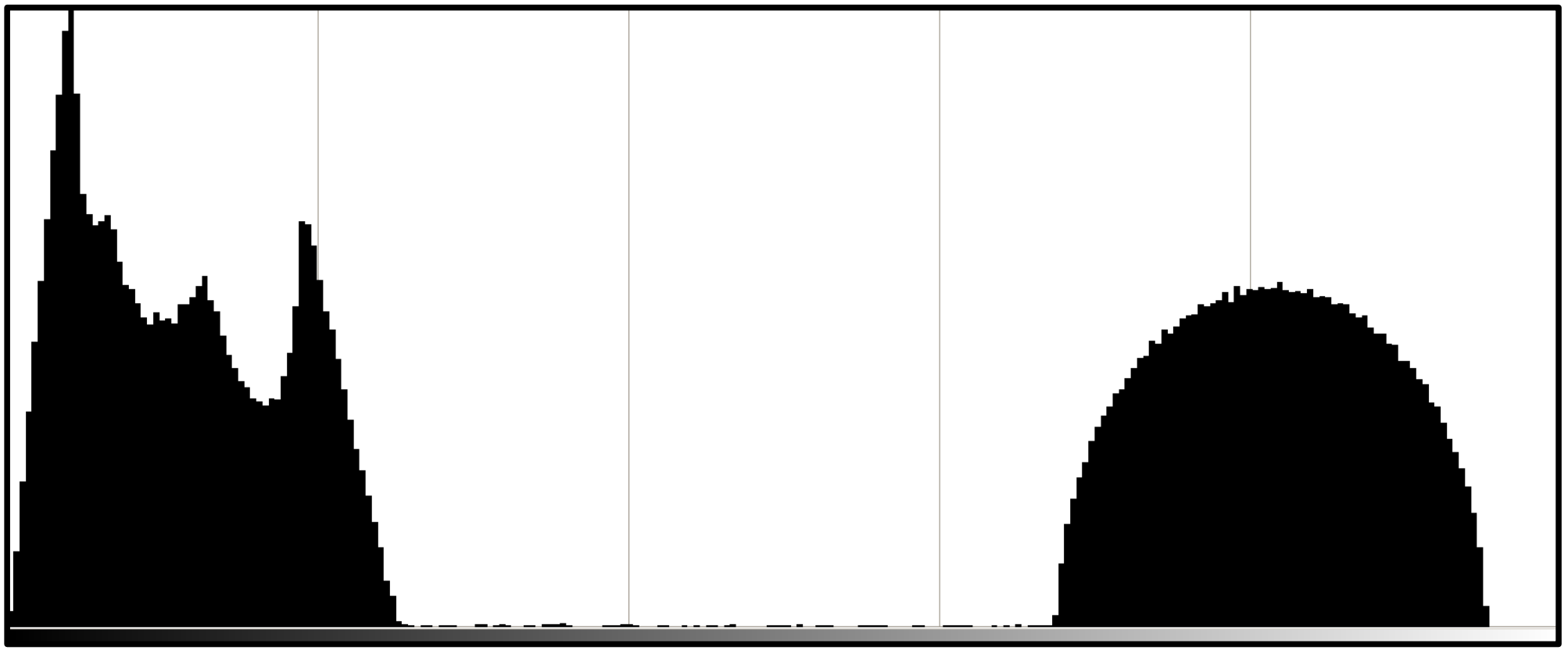
•É mais fácil de entender a estratégia de Otsu observando o histograma de algumas imagens em escala de cinza.











Histogramas

- O que todos os histogramas têm em comum?

Histogramas

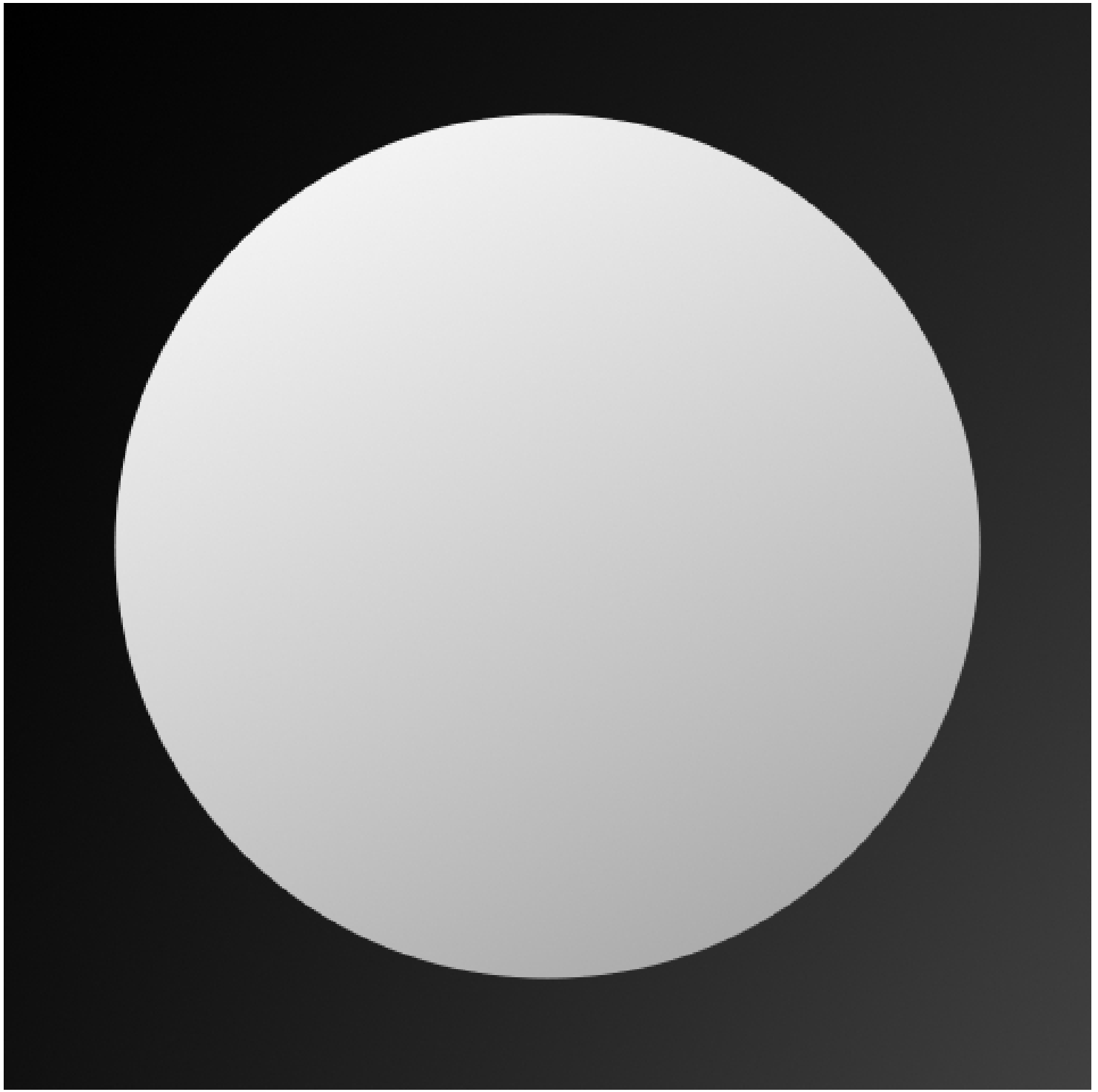
- O que todos os histogramas têm em comum?
  - R: eles são aproximadamente *bimodais*.
  - Informalmente, existem dois “morros” distintos.

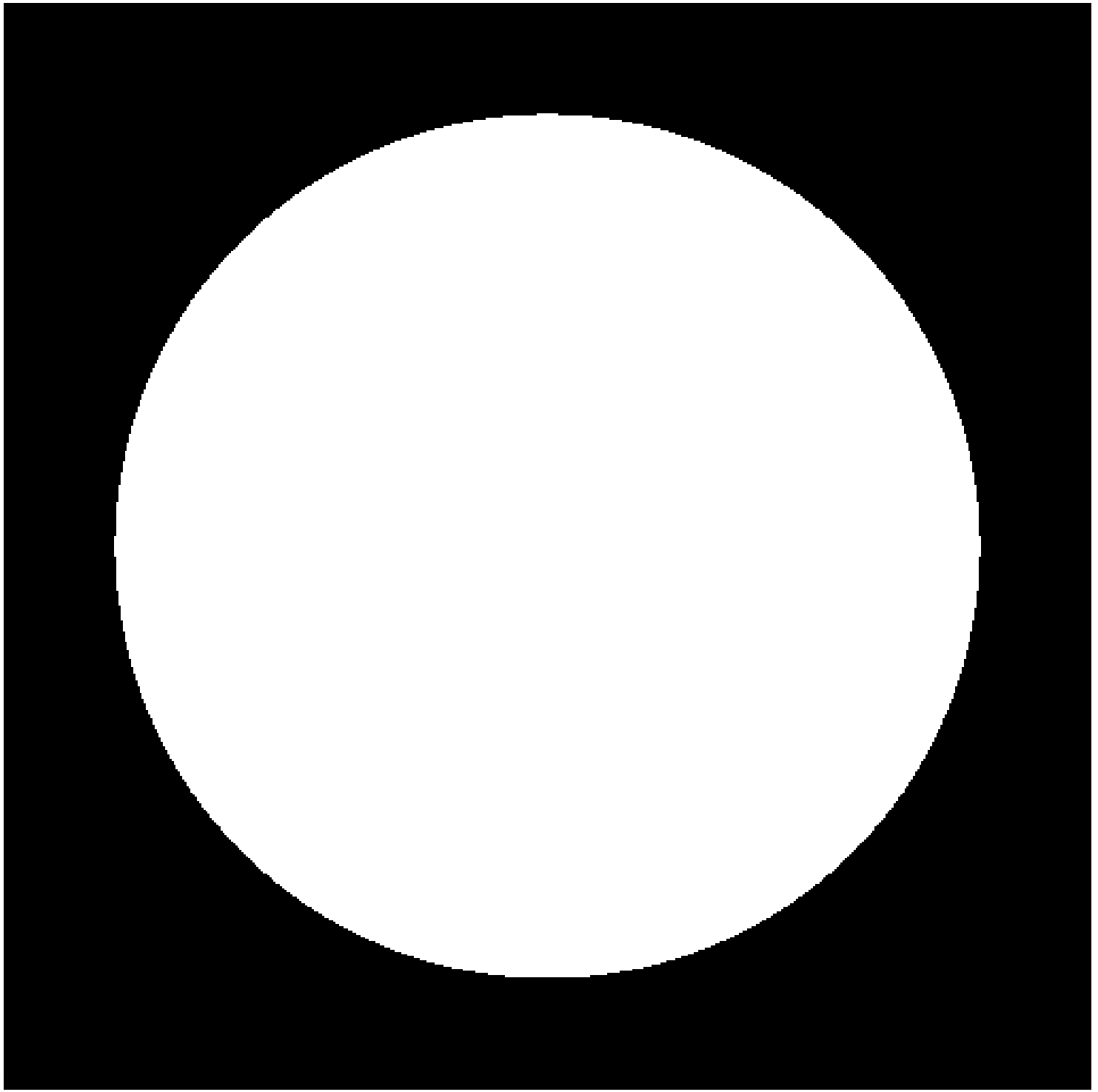
Algoritmo de Otsu

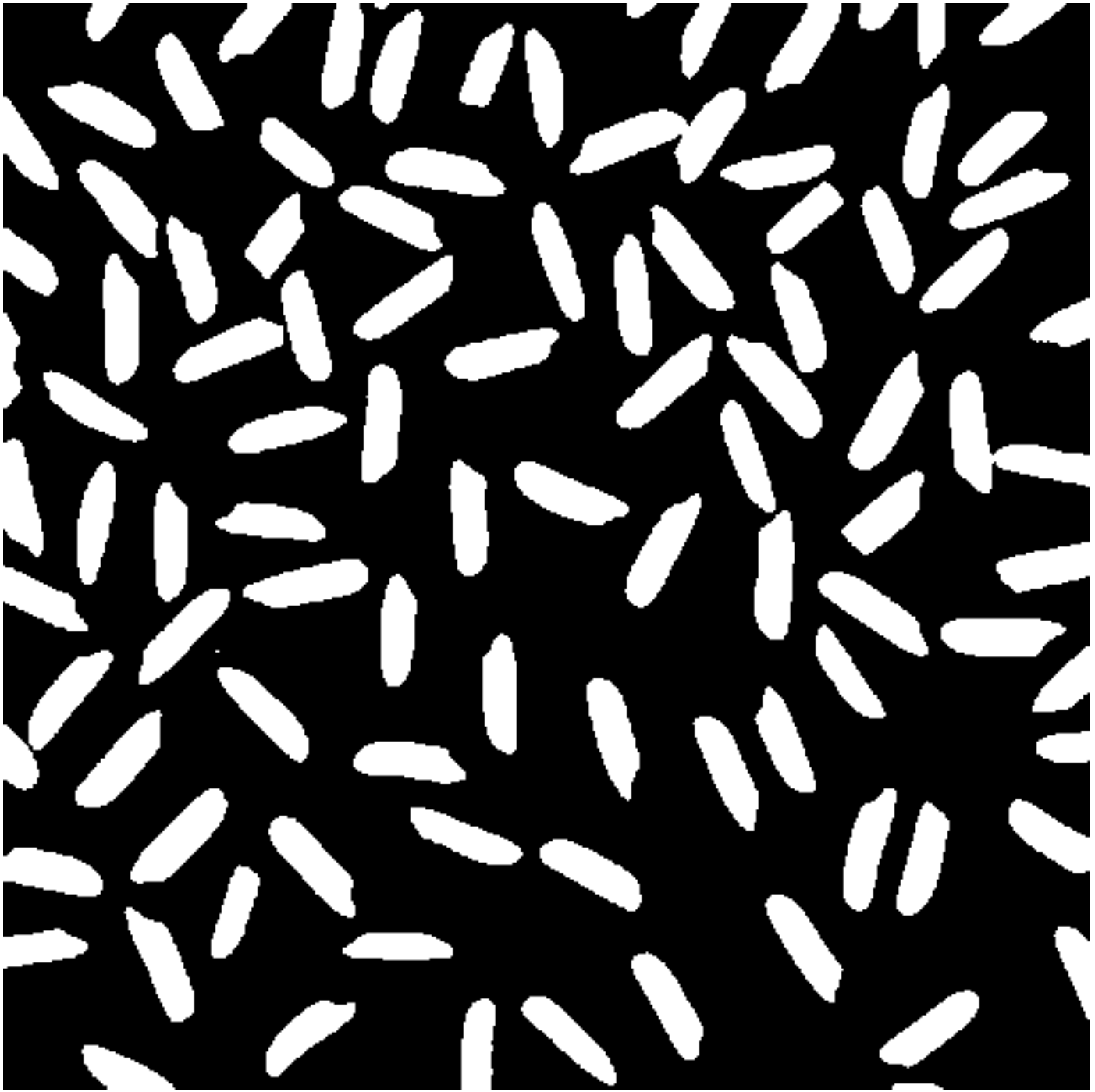
- O algoritmo de Otsu:
  - Testa todos os limiares possíveis para binarização.
  - Escolhe aquele que melhor divide o histograma em duas modas.
    - Para isso, somamos a variância do histograma à esquerda e à direita do limiar.
    - O melhor limiar será aquele que produzir a menor soma.
      - Você consegue entender o porquê?
    - (Na implementação rápida, em vez de minimzar a variância intra-classe, maximizamos a variância inter-classe).

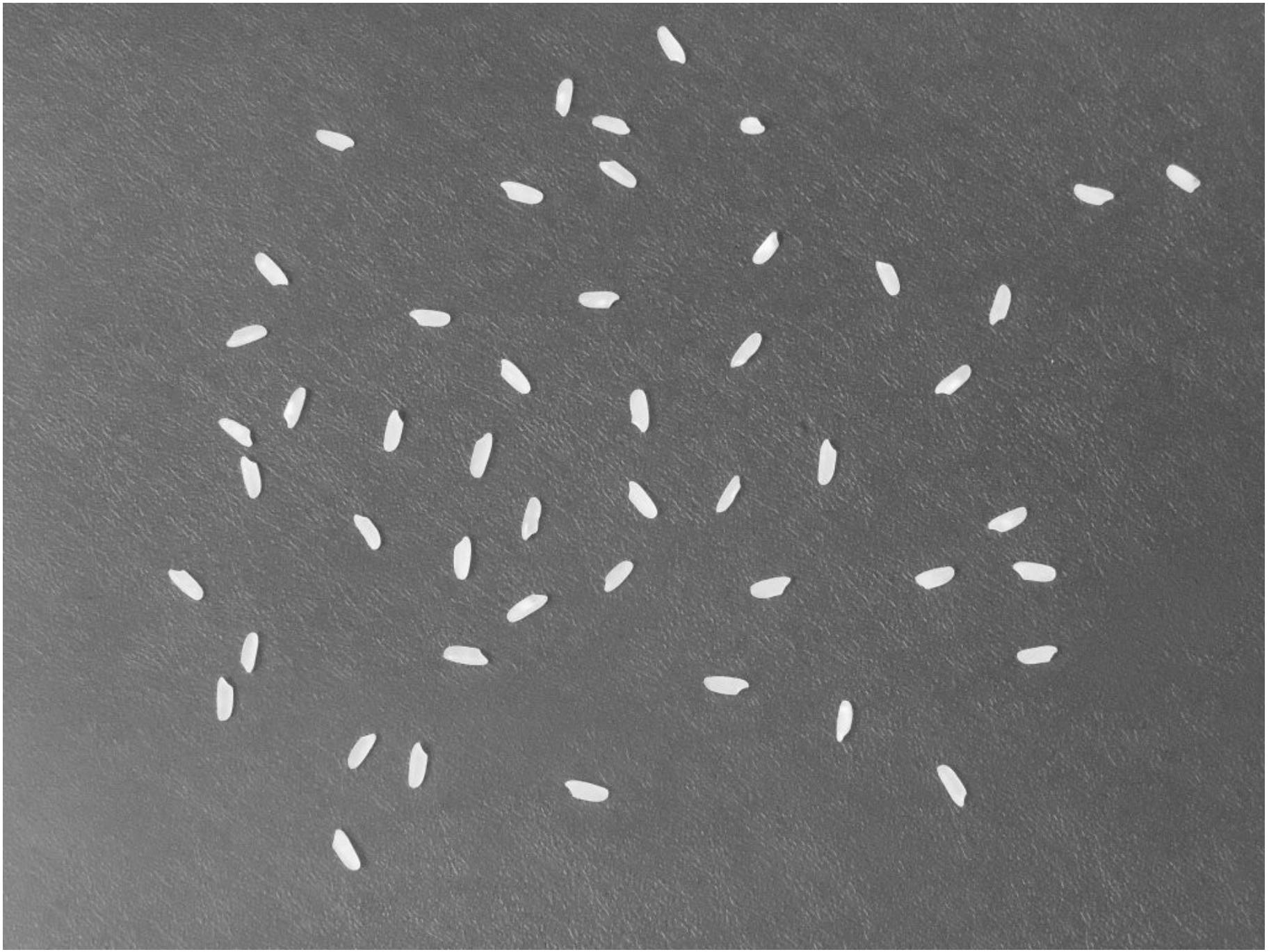
- Nota: o algoritmo de Otsu pode ser generalizado para outras tarefas de classificação baseadas em distribuições bimodais.

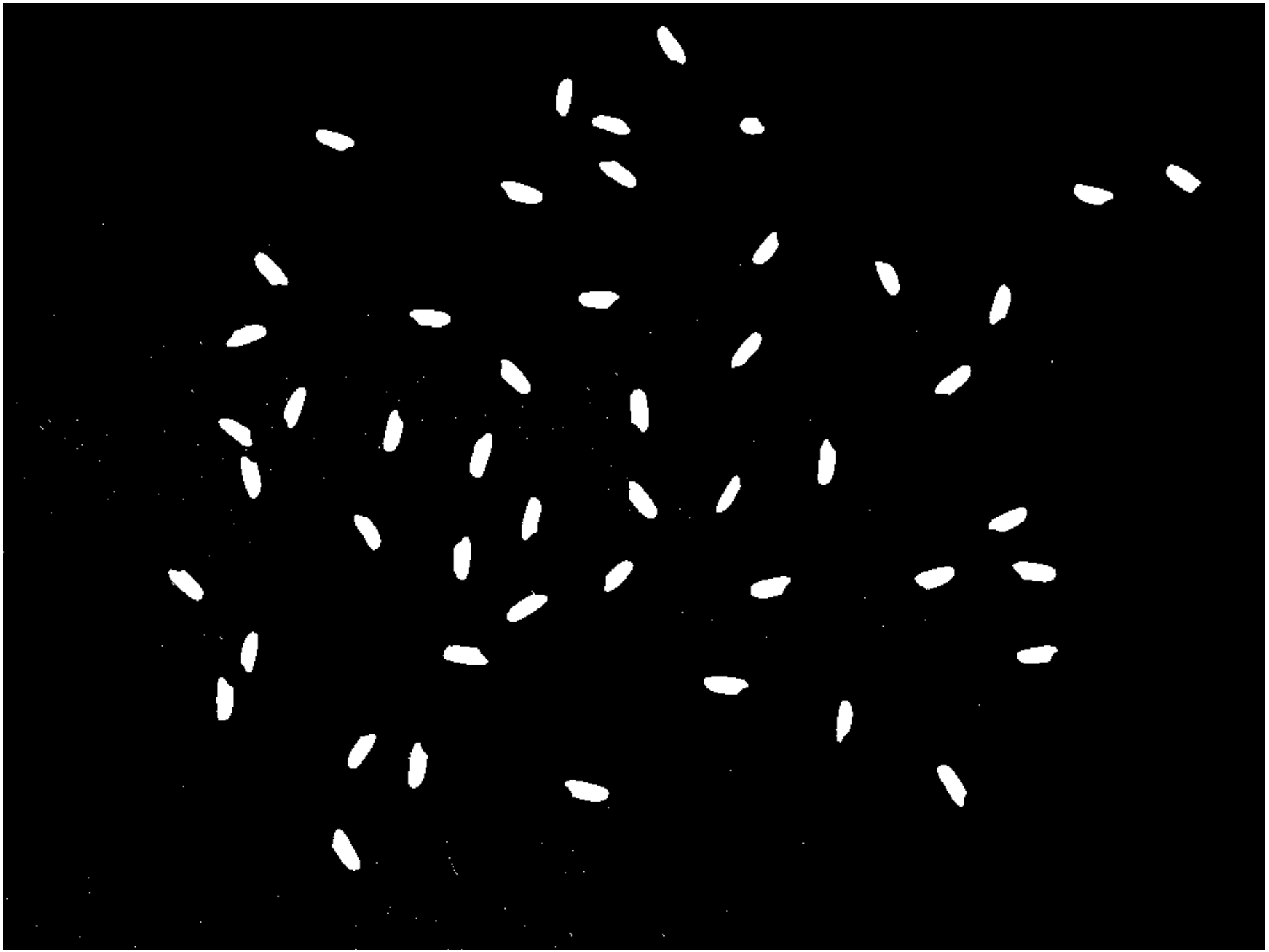
```
limiarOtsu (histograma H (normalizado))

peso1 ← H[0]
soma1 ← 0
peso2 ← 1-peso1
soma2 ← Σ H[i]·i, para todas as faixas i do histograma

melhor_limiar ← 0
melhor_score ← 0

for (cada faixa i de H, pulando a primeira)
    peso1 ← peso1 + H[i]
    if (peso1 = 0) vai para a próxima iteração (nada à esquerda!)
    peso2 ← 1-peso1
    soma1 ← soma1 + H[i]·i
    soma2 ← soma2 – H[i]·i
    media1 ← soma1 / peso1
    media2 ← soma2 / peso2

    score ← peso1·peso2·(media1-media2)²
    if (score > melhor_score)
        melhor_score = score
        melhor_limiar = i

retorna melhor_limiar
```











Trabalho 5

- Prazo: 23/10, 16:00.
- Peso: 1.6 (de 10).
- Escreva um programa para criar cenas usando *chroma key*.
  - Considere fundo verde!
  - Deve funcionar para imagens diferentes sem mudar parâmetros.
  - Soluções excessivamente simples serão penalizadas.