

# Threshold selection by clustering gray levels of boundary 论文复现

徐恺阳 523030910085

## 算法复现

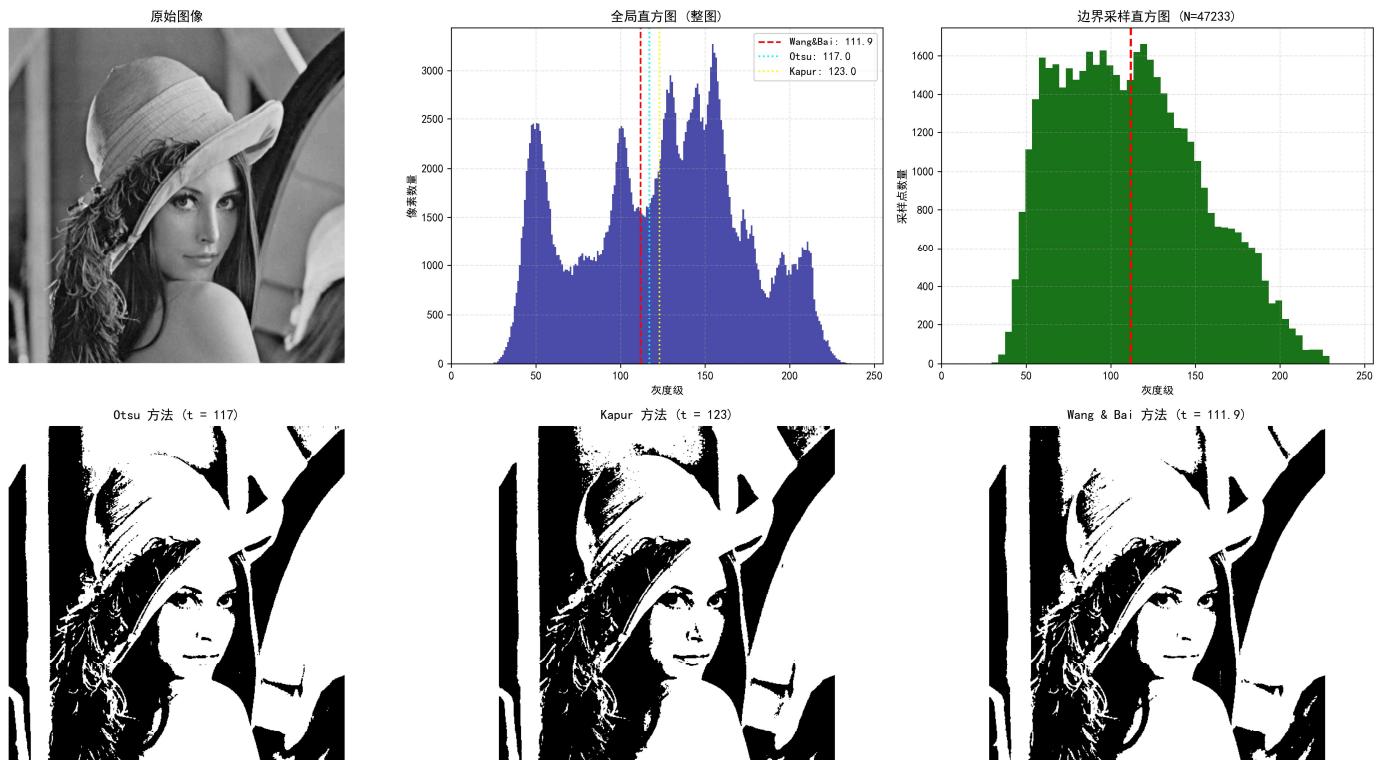
### 双阈值选择

双阈值情况下，全局直方图会出现多个峰，而按照论文算法，选取适当的梯度阈值  $T$ ，得到的**边界采样直方图**只会出现清晰的单峰。

选取单峰的**平均值**作为**最优阈值**  $r$ ，进行图像分割。

girl

"girl" 图像阈值分割方法对比



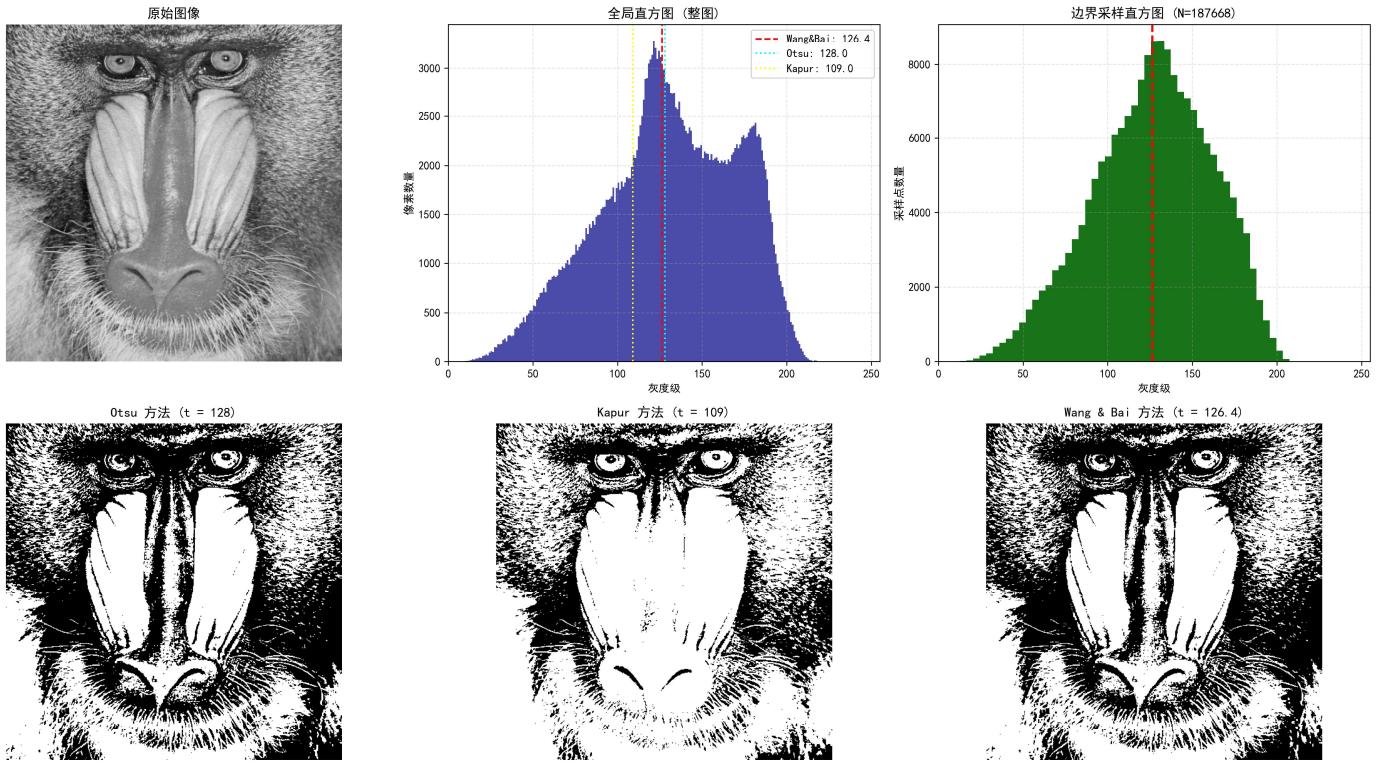
梯度阈值  $T$  选择 90.0，得到第三幅图所示的边界采样直方图。分别用 Otsu 方法、Kapur 方法及论文算法计算阈值  $r$ ，结果如下：

	Otsu 方法	Kapur 方法	论文算法
阈值 $r$	117	123	111.9

用三种方法得到的阈值  $r$  分割图像，结果分别如第四、五、六幅图所示。可以看出，本文算法的分割效果显著优于 Otsu 方法和 Kapur 方法。

## baboon

"baboon" 图像阈值分割方法对比



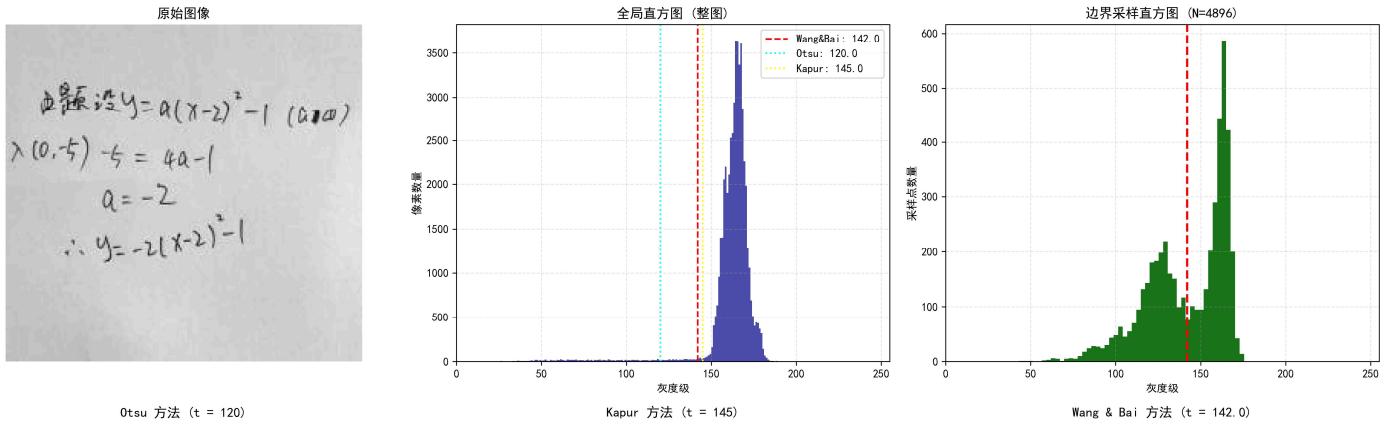
梯度阈值  $T$  选择 60.0, 得到第三幅图所示的边界采样直方图。分别用 Otsu 方法、Kapur 方法及论文算法计算阈值  $r$ , 结果如下:

	Otsu 方法	Kapur 方法	论文算法
阈值 $r$	128	109	126.4

用三种方法得到的阈值  $r$  分割图像, 结果分别如第四、五、六幅图所示。可以看出, 本文算法的分割效果显著优于 Kapur 方法, 与 Otsu 方法效果近似。

## characters

"character" 图像阈值分割方法对比



$\text{设 } y = a(x-2)^2 - 1 \text{ (原图)}$   
 $\lambda(0, -\frac{1}{4}) - \frac{1}{4} = 4a - 1$   
 $a = -2$   
 $\therefore y = -2(x-2)^2 - 1$

$\text{设 } y = a(x-2)^2 - 1 \text{ (原图)}$   
 $\lambda(0, -\frac{1}{4}) - \frac{1}{4} = 4a - 1$   
 $a = -2$   
 $\therefore y = -2(x-2)^2 - 1$

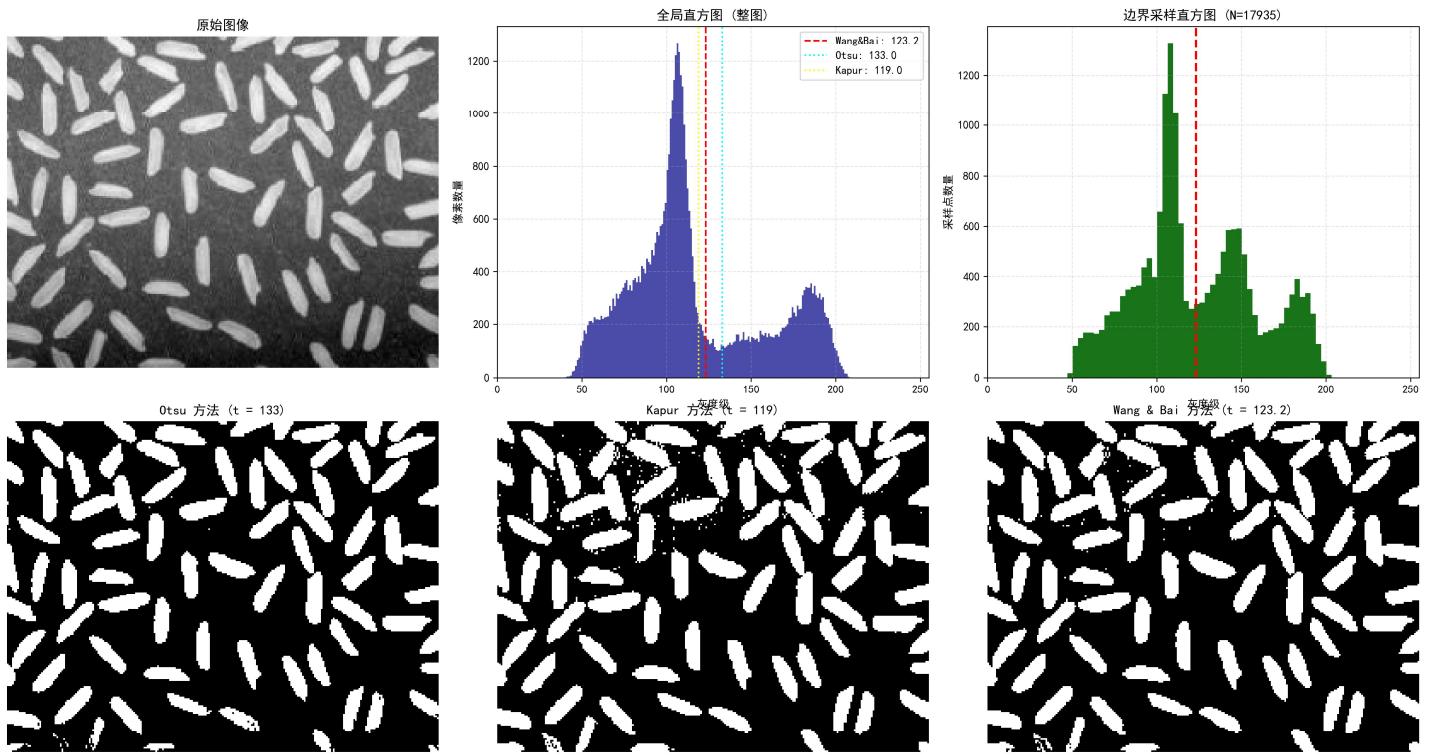
$\text{设 } y = a(x-2)^2 - 1 \text{ (原图)}$   
 $\lambda(0, -\frac{1}{4}) - \frac{1}{4} = 4a - 1$   
 $a = -2$   
 $\therefore y = -2(x-2)^2 - 1$

梯度阈值  $T$  选择 60.0, 得到第三幅图所示的边界采样直方图。分别用 Otsu 方法、Kapur 方法及论文算法计算阈值  $r$ , 结果如下:

	Otsu 方法	Kapur 方法	论文算法
阈值 $r$	120	145	142.0

用三种方法得到的阈值  $r$  分割图像, 结果分别如第四、五、六幅图所示。可以看出, 本文算法的分割效果显著优于 Otsu 方法, 与 Kapur 方法效果近似。

"rice" 图像阈值分割方法对比



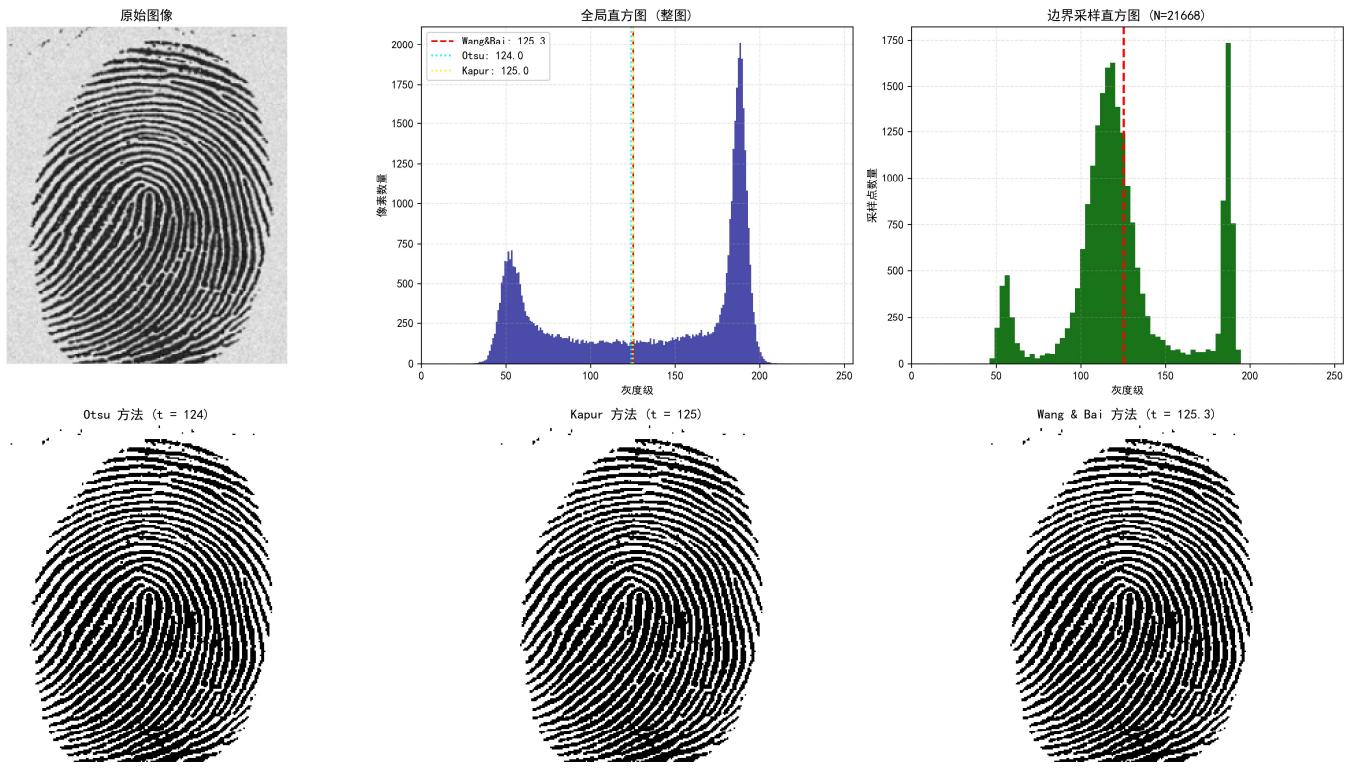
梯度阈值  $T$  选择 60.0, 得到第三幅图所示的边界采样直方图。分别用 Otsu 方法、Kapur 方法及论文算法计算阈值  $r$ , 结果如下:

	Otsu 方法	Kapur 方法	论文算法
阈值 $r$	133	119	123.2

用三种方法得到的阈值  $r$  分割图像, 结果分别如第四、五、六幅图所示。可以看出, 本文算法的分割效果优于 Kapur 方法, 但相比 Otsu 方法略有不足。

# fingerprint

"fingerprint" 图像阈值分割方法对比



梯度阈值  $T$  选择 60.0, 得到第三幅图所示的边界采样直方图。分别用 Otsu 方法、Kapur 方法及论文算法计算阈值  $r$ , 结果如下:

	Otsu 方法	Kapur 方法	论文算法
阈值 $r$	124	125	125.3

用三种方法得到的阈值  $r$  分割图像, 结果分别如第四、五、六幅图所示。可以看出, 本文算法与 Otsu 方法、Kapur 方法均取得较好的效果。

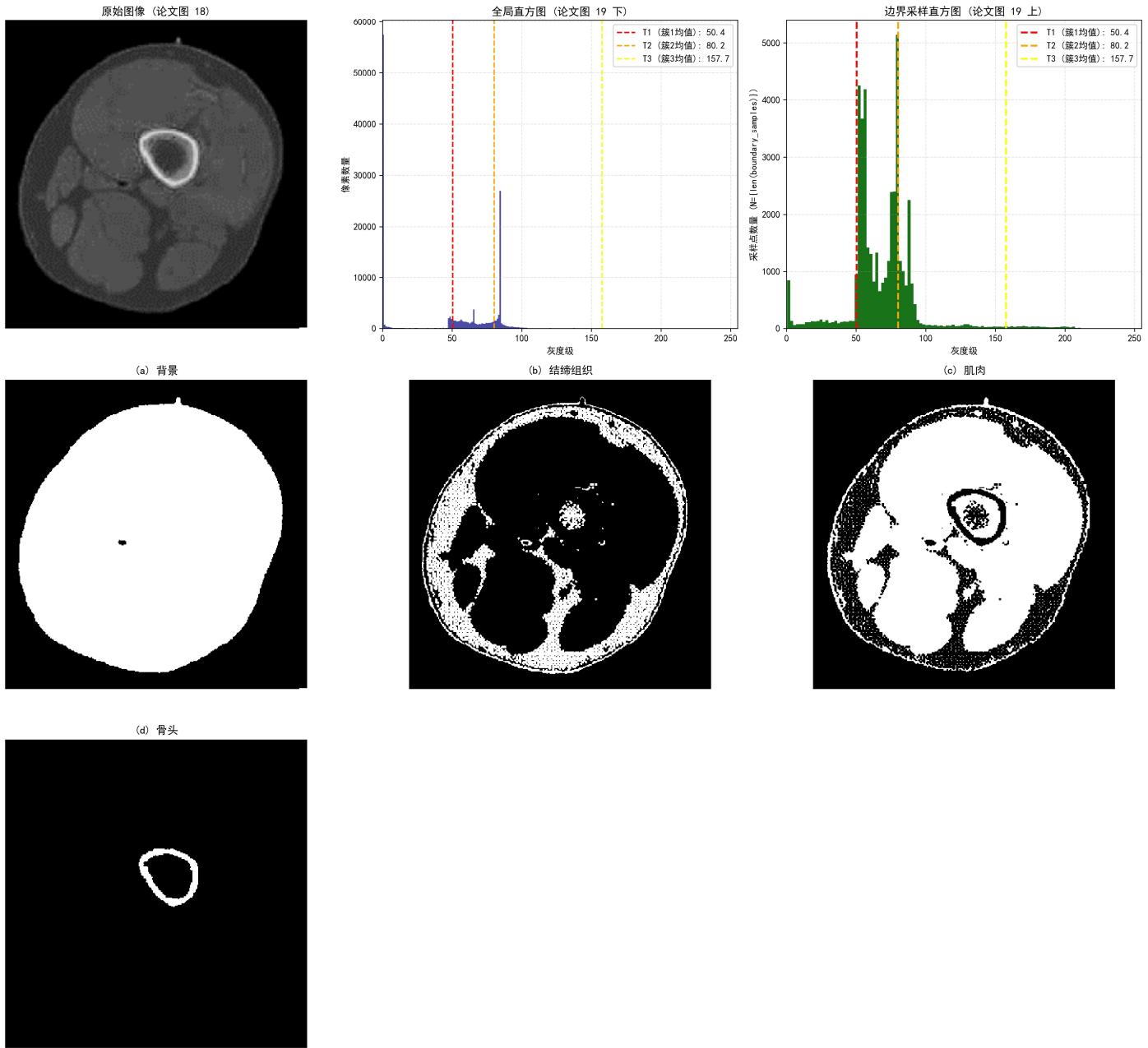
## 多阈值选择

多阈值情况下, 按照论文算法, 选取适当的梯度阈值  $T$ , 得到的边界采样直方图会聚类成多峰。

通过 K-Means 算法, 选取每个峰的平均值作为最优阈值  $r$ , 进行图像分割。

## leg

"leg" 图像多阈值分割 (Wang & Bai, K-Means法)



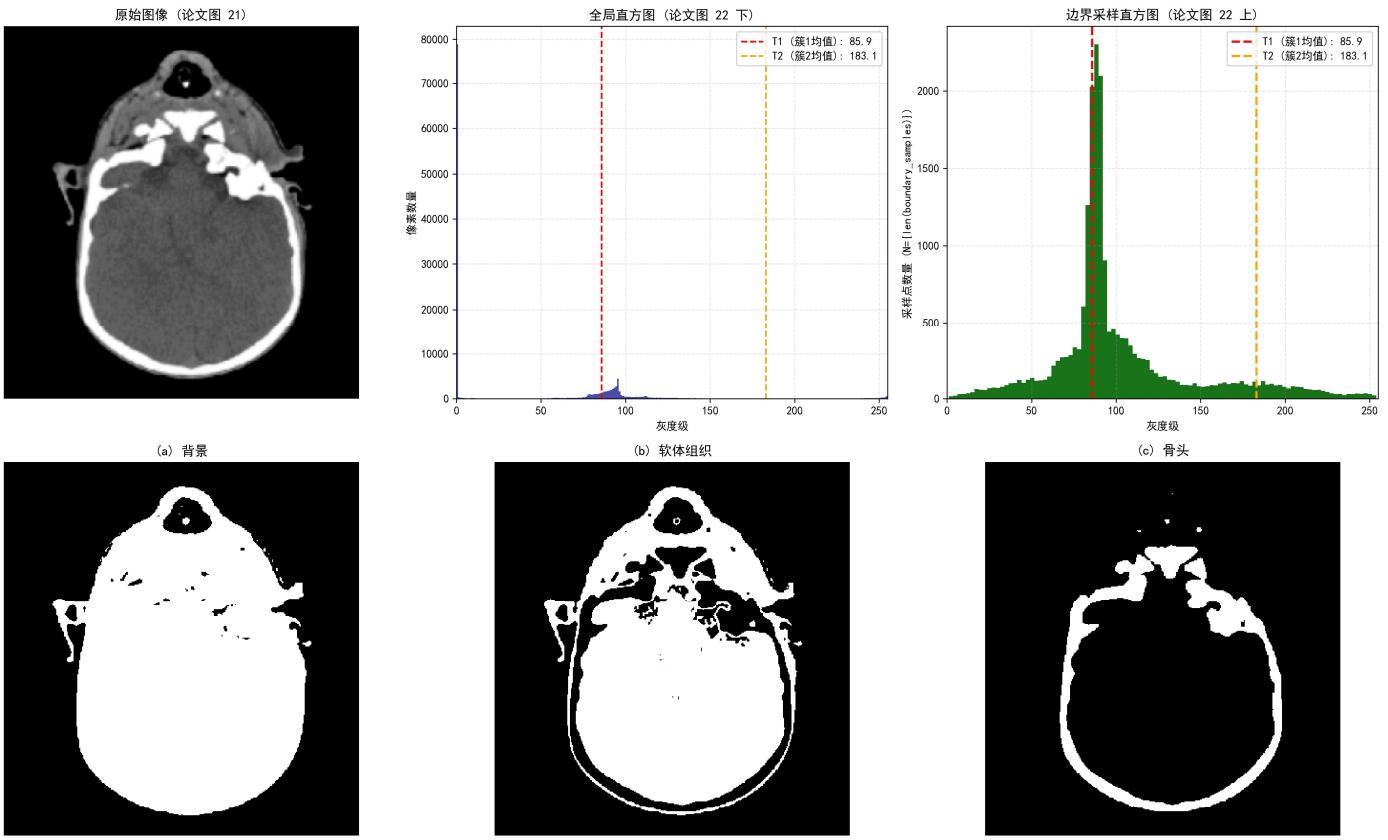
梯度阈值  $T$  选择 15.0, 得到第三幅图所示的边界采样直方图。用论文算法计算阈值  $r$ , 结果如下:

阈值 $r_1$	阈值 $r_2$	阈值 $r_3$
50.4	80.2	157.7

用阈值  $r_1, r_2, r_3$  分割图像, 结果分别如第四、五、六、七幅图所示。可以看出, 本文算法成功分割出腿部 CT 图像中的**背景**、**结缔组织**、**肌肉**和**骨头**, 效果较优。

# head

"head" 图像多阈值分割 (Wang & Bai, K-Means法)



梯度阈值  $T$  选择 40.0, 得到第三幅图所示的边界采样直方图。用论文算法计算阈值  $r$ , 结果如下:

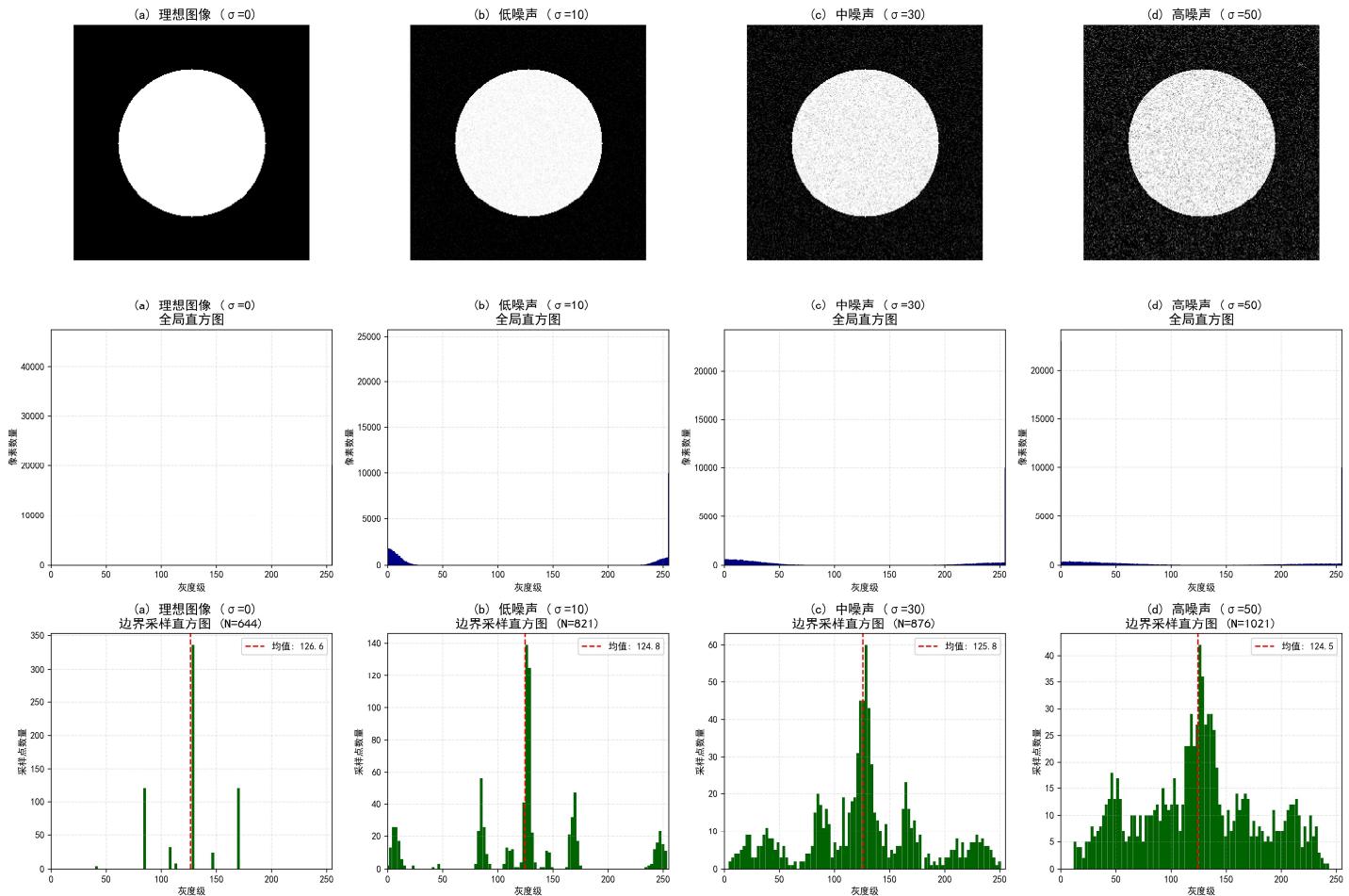
阈值 $r_1$	阈值 $r_2$
85.9	183.1

用阈值  $r_1, r_2$  分割图像, 结果分别如第四、五、六幅图所示。可以看出, 本文算法成功分割出脑部 CT 图像中的**背景**、**软体组织**和**骨头**, 效果较优。

# 实验复现

## sensitivity to noise

噪声敏感性分析



对理想图像添加三种不同强度 ( $\sigma = 10, 30, 50$ ) 的高斯噪声，得到第一行的四幅原始图像，其全局直方图如第二行的四幅图所示。可以看出，理想图像的全局直方图在 0 和 255 处有两个尖峰，而随着噪声增加，这两个峰迅速展宽、合并，最终变为一个无法分辨的、混乱的宽峰。

梯度阈值  $T$  选择 750.0，得到第三行的四幅边界采样直方图。随着噪声增加，这个峰虽然也变“胖”了，但其中心位置几乎没有发生偏移。用论文算法计算阈值  $r$ ，结果如下：

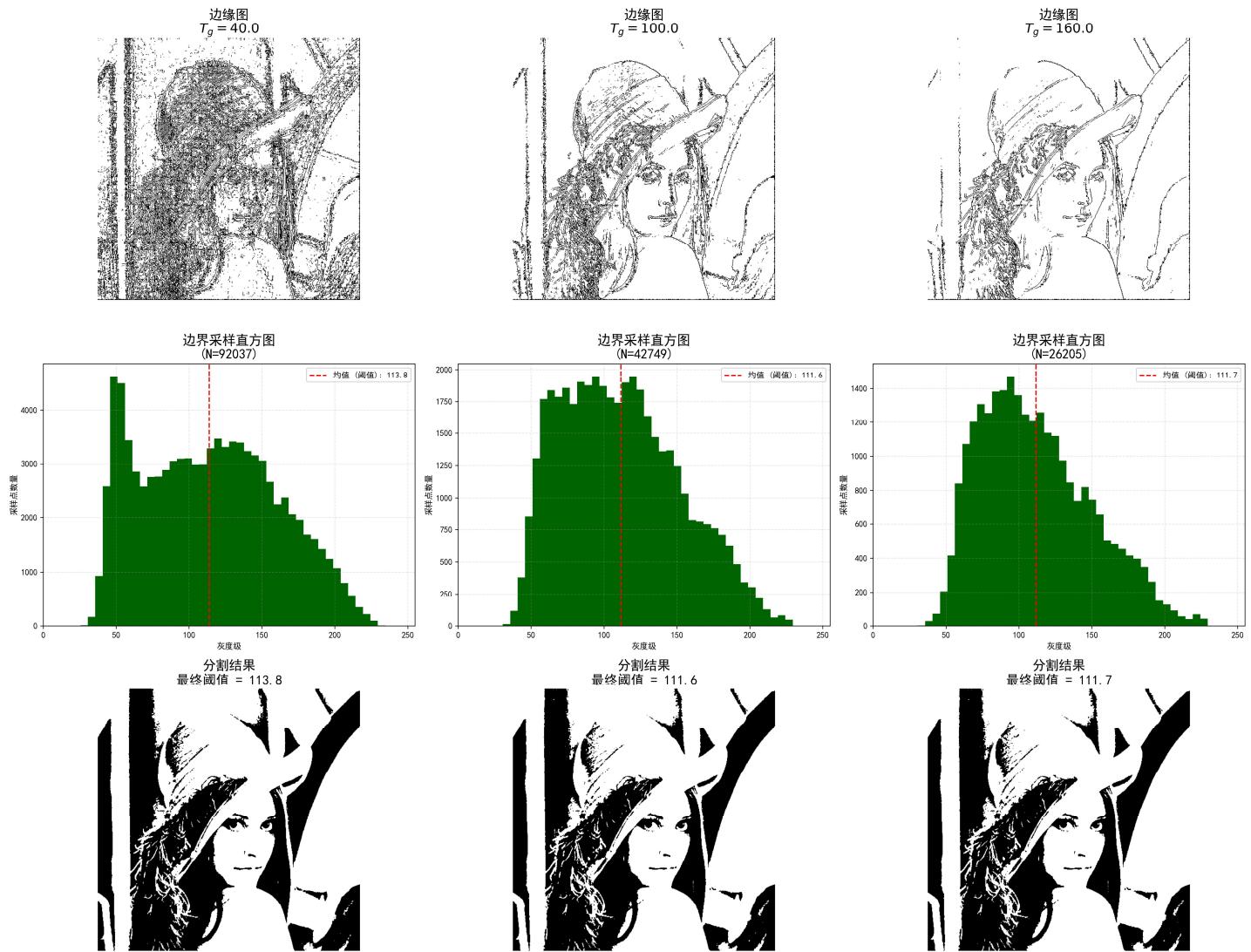
	理想图像	低噪声	中噪声	高噪声
阈值 $r$	126.6	124.8	125.8	124.5

可以看出，倘若适当地选择  $T$ ，则本文算法对噪声具有极高的鲁棒性。

## sensitivity to the gradient threshold $T$

girl

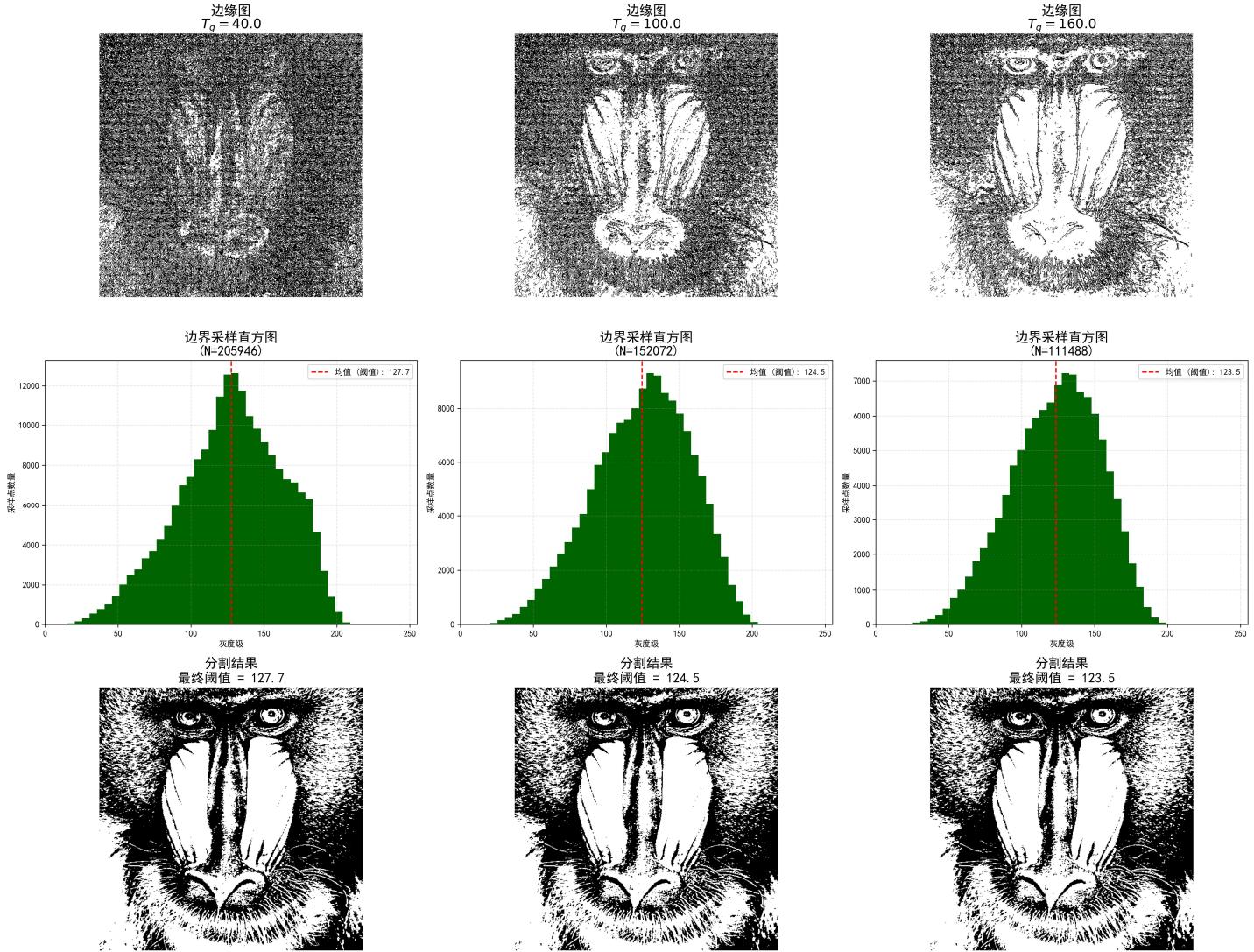
梯度阈值  $T$  敏感性分析



梯度阈值  $T$  分别选择 40.0, 100.0, 160.0, 得到第一行的三幅采样点位置图, 随着  $T$  的增大, 被算法筛选通过的采样点数量急剧减少, 采样点位置图从浓密变为稀疏, 边界直方图的形态和峰高也发生巨大变化。用论文算法计算阈值  $r$ , 结果如下:

梯度阈值 $T$	40.0	100.0	160.0
阈值 $r$	113.8	111.6	111.7

可以看出, 尽管采样点数量完全不同, 但计算出的阈值  $r$  却几乎保持不变, 说明本文算法对梯度阈值具有极高的鲁棒性。

梯度阈值  $T$  敏感性分析

梯度阈值  $T$  分别选择 40.0, 100.0, 160.0, 得到第一行的三幅采样点位置图, 随着  $T$  的增大, 被算法筛选通过的采样点数量急剧减少, 采样点位置图从浓密变为稀疏, 边界直方图的形态和峰高也发生巨大变化。用论文算法计算阈值  $r$ , 结果如下:

梯度阈值 $T$	40.0	100.0	160.0
阈值 $r$	127.7	124.5	123.5

可以看出, 尽管采样点数量完全不同, 但计算出的阈值  $r$  却几乎保持不变, 说明本文算法对梯度阈值具有极高的鲁棒性。