

厦門大學

课 程 作 业 报 告

常用边缘检测算法的分析

姓 名：张韦妮

学 号：23020181154257

学 院：信息科学与技术学院

专 业：计算机科学

2018 年 10 月 16 日

目录

| | |
|--------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 图像的边缘 | 1 |
| 1.2 图像边缘检测的基本步骤 | 1 |
| 1.3 本文工作 | 2 |
| 第 2 章 边缘检测算法 | 3 |
| 2.1 Prewitt 算子 | 3 |
| 2.2 Sobel 算子 | 4 |
| 2.3 Canny 算子 | 4 |
| 2.4 阈值自适应 | 5 |
| 第 3 章 实验结果及分析 | 7 |
| 3.1 Prewitt 算子检测结果 | 7 |
| 3.2 Sobel 算子检测结果 | 9 |
| 3.3 Canny 算子检测结果 | 11 |
| 3.4 性能对比分析 | 14 |
| 3.4.1 抗噪性能 | 14 |
| 3.4.2 计算复杂度 | 15 |
| 第 4 章 总结 | 16 |
| 参考文献 | 18 |

第 1 章 绪论

1.1 图像的边缘

边缘是图像的一种重要特征，包括了位置、轮廓等重要信息，图像的边缘是图像信息最集中、最丰富的地方。图像上物体的边缘可能对应不同的物理意义，如：空间曲面的不连续点，前景与背景的分界线，不同材料、颜色的交界处，阴影的边缘等等。这些边缘点都是在图像上灰度不连续的点，或灰度变化剧烈的点^[1]。如图所示1.1，根据灰度变化的剧烈程度，通常将边缘模型分为阶跃模型、斜坡模型和屋顶模型。针对阶跃模型，边缘点对应于灰度变化曲线的一阶导数极大值点和二阶导数过零点。

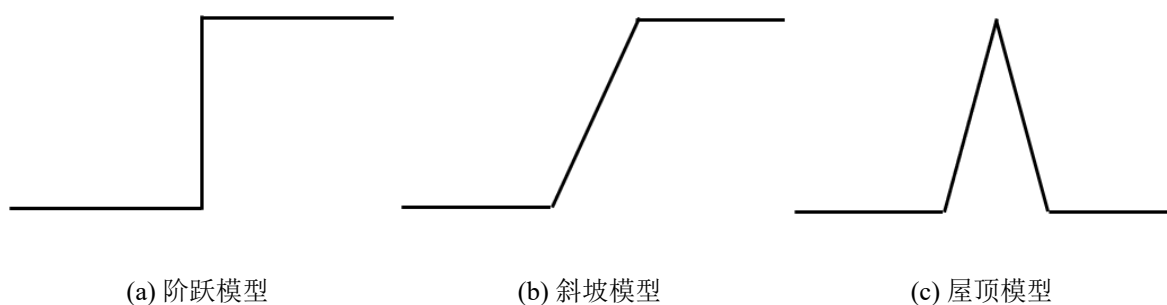


图 1.1: 边缘模型

1.2 图像边缘检测的基本步骤

图像边缘检测是一种重要的技术，其检测结果可以为后续的图像处理提供重要的依据。由边缘的定义可知，边缘是灰度变化剧烈的地方，即边缘信号具有空间域的高频分量。因此，边缘检测方法在本质上是检测信号的高频分量，但是信号的噪声也是高频信号，对边缘的检出有很大的干扰，使得边缘检测任务变得困难。

一般图像边缘检测的步骤为：

1) 平滑滤波

对图像进行平滑处理以降噪。在滤波器降低噪声的同时，边缘精度也会损失。

2) 图像增强

将邻域内灰度变化剧烈的点突出表示，可以通过梯度计算实现。

3) 边缘点检测

通过局部操作，提取出边缘的候选点。

4) 边缘定位

从候选点集合中选出组成边缘的点，以得到更精确的边缘位置。

1.3 本文工作

本文对常见的边缘检测算子 Prewitt 算子、Sobel 算子和 Canny 算子进行了介绍以及实现，并通过实验对这三种算子的性能进行了简单对比。

第2章 边缘检测算法

2.1 Prewitt 算子

Prewitt 算子是离散型一阶差分算子，用于计算图像灰度函数的梯度近似值。在图像的每一点处应用 Prewitt 算子，可以得到该点对应的梯度向量。Prewitt 算子包括方向差分运算和局部平均运算，根据像素点上下、左右相邻点的灰度差应该在边缘处达到极值的原理来进行边缘检测。

Prewitt 算子使用两个 3×3 的卷积核与图像进行卷积，分别计算垂直方向和水平方向的灰度梯度近似值。记 I_m 为原始图像， G_x 和 G_y 分别为水平方向和垂直方向检测后的图像，则计算公式为2.1:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * I_m \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * I_m \quad (2.1)$$

其中，* 为二维信号卷积运算。

图像在每一点处的梯度大小和梯度方向的计算公式为2.2:

$$G = \sqrt{G_x^2 + G_y^2} \quad \theta = \arctan \frac{G_y}{G_x} \quad (2.2)$$

特别地，在具体应用中，可以使用公式2.3近似计算梯度大小:

$$G = |G_x| + |G_y| \quad (2.3)$$

使用 Prewitt 算子进行边缘检测的算法流程如下:

- 1) 对源图像使用高斯滤波消除噪声;
- 2) 使用上述两个 3×3 的卷积核对图像进行卷积得到 G_x 和 G_y ;
- 3) 利用公式2.3计算图像每点的近似梯度大小 G ;
- 4) 选择合适的阈值对 G 进行二值化处理。

2.2 Sobel 算子

与 Prewitt 算子相似，Sobel 算子是离散型一阶差分算子，用于计算图像灰度函数的梯度近似值。在图像的每一点处应用 Sobel 算子，可以得到该点的梯度向量。Sobel 算子包括方向差分运算和局部加权平均运算，根据像素点上下、左右相邻点的灰度差应该在边缘处达到极值的原理来进行边缘检测。

Sobel 算子使用两个 3×3 的卷积核与图像进行卷积，分别计算垂直方向和水平方向的灰度梯度近似值。记 I_m 为原始图像， G_x 和 G_y 分别为水平方向和垂直方向检测后的图像，则计算公式为2.4:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I_m \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I_m \quad (2.4)$$

其中，* 为二维信号卷积运算。

图像在每一点处的梯度大小和梯度方向的计算公式同式 Gradient。特别地，在具体应用中，可以使用公式2.3近似计算梯度大小:

使用 Sobel 算子进行边缘检测的算法流程如下:

- 1) 对源图像使用高斯滤波消除噪声;
- 2) 使用上述两个 3×3 的卷积核对图像进行卷积得到 G_x 和 G_y ;
- 3) 利用公式2.3计算图像每点的近似梯度大小 G ;
- 4) 选择合适的阈值对 G 进行二值化处理。

2.3 Canny 算子

Canny 算子是基于最优化方法的边缘检测算子。Canny 提出边缘检测算子应该满足以下三个准则:

- 1) 良好的检测结果。即检测结果的漏检率和错检率都应该尽可能小。由于漏检率和错检率是输出信噪比的单调递减函数，因此该准则可以对应于最大信噪比准则。
- 2) 良好的定位性能。即检测出的边缘点应该尽可能在实际边缘的中心。
- 3) 单边缘响应。即单一边缘应该具有唯一的响应，抑制同一边缘产生多个响应。

Canny 算子是 Canny 从三个准则推导出的多级的最优边缘检测算子，是对信噪比和定位乘积的最优化逼近算子。算法流程如下:

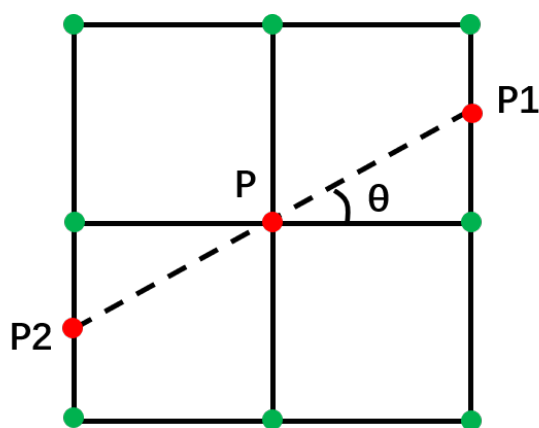


图 2.1: 非极大值抑制示意图

1) 高斯平滑

对源图像使用高斯滤波，平滑图像消除噪声。

2) 梯度计算

计算图像中每个像素点的梯度大小和方向。可以利用 Roberts、Sobel 或 Prewitt 算子计算像素点的梯度大小 G 和梯度方向 θ

3) 非极大值抑制

对梯度大小进行非极大值抑制操作。根据定义，边缘对应于图像梯度的局部极值处。因此需要寻找像素梯度大小的局部极大值点，并将非极大值点对应的像素记为非边缘点。如图所示2.1，为确定像素点 P 是否为其 8 邻域的梯度局部极大值点，我们需要计算 P 点的梯度方向（如图2.1中虚线所示），可以确定极大值分布于梯度方向上。我们通过线性插值得到图中交点 $P1, P2$ 的梯度大小，若 P 点的梯度大小比 $P1, P2$ 大，则保留 P 点为边缘点，否则 P 点为非边缘点。

4) 双阈值检测及边缘连接

为了解决杂散响应，消除假边缘，采用双阈值检测对步骤 3 中保留的边缘点进行过滤。对于梯度值大于高阈值的像素点，直接标记为强边缘点，对于梯度值小于低阈值的像素点，直接标记为非边缘点，对于梯度值大于低阈值小于高阈值的像素点，标记为弱边缘点。为了跟踪边缘连接，将 8 邻域内有强边缘点的弱边缘点保留为边缘，8 邻域内没有强边缘点的弱边缘点标记为非边缘点。

2.4 阈值自适应

从三种算子的算法流程描述中可以看出，Prewitt、Sobel 算子检测在进行二值化以及 Canny 算子在筛选强弱边缘点的时候，都需要人为设定阈值。该阈值对检测结果影响较大 (见图3.2, 图3.6, 图3.10), 并且对于不同的源图像, 三种算子合适的阈值取值也是不同的, 选择合适的阈值是一个比较困难的步骤。因此, 我们采用了最大类间方差法 (OTSU 算法)^[2]

自适应地确定阈值。最大类间方差法的基本思想是将图像像素分为背景和目标两类，选取的最佳阈值应该使得类间分离性最好。

对 Prewitt 算子和 Sobel 算子，在计算得到图像的梯度后，我们对梯度图使用最大类间方差法进行自适应的二值化得到边缘点。Canny 算子需要高低两个阈值，这里我们采用了一种简单的策略获得阈值。根据实验，我们观察到高阈值相比低阈值对整个图像边缘点检测影响更明显 (见图3.10)，因此我们用最大类间方差法去计算高阈值。根据 Canny 推荐的高低阈值比范围为 2:1 至 3:1^[3]，我们选择高低阈值比为 2.5:1，从而从高阈值计算得到低阈值。

由于寻找最大类间方差对应的阈值是一个比较耗时的操作，尤其是在区间跨度较大的时候。因此我们将计算得到的图像梯度变换到 $[0, 255]$ 的区间范围内，虽然可能会对检测精度产生影响，但是会提高检测速度。

第3章 实验结果及分析

3.1 Prewitt 算子检测结果

使用 Prewitt 算子检测边缘的效果如图3.1所示，其中图 (a) 为输入源图像，图 (b) 从左到右分别为 (1) 源图像灰度化后的图像，(2) 经过水平方向差分后得到的 G_x 图像，(3) 经过垂直方向差分后得到的 G_y 图像，(4) 二值化的边缘检测结果。从图中可以看出， G_x 图像主要提取出了垂直方向的边缘， G_y 图像主要提取出了水平方向的边缘。

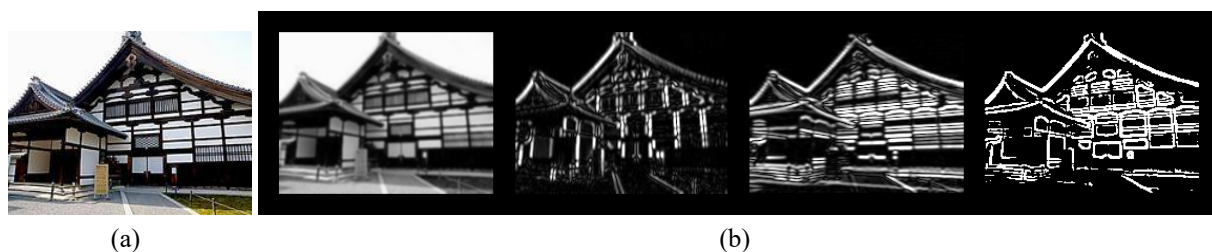


图 3.1: Prewitt 算子检测结果

图3.2展示了使用不同的阈值对梯度进行二值化的边缘检测结果。该组实验固定高斯滤波核的尺寸为 5×5 ，取二值化阈值为 100,150,200,250。从检测结果可以看出，当阈值较低时，保留了大量的假边缘，检测结果比较杂乱，检测的边缘比较粗；当阈值较高时，检测的边缘比较细，比较清晰，但是也存在边缘丢失的情况。

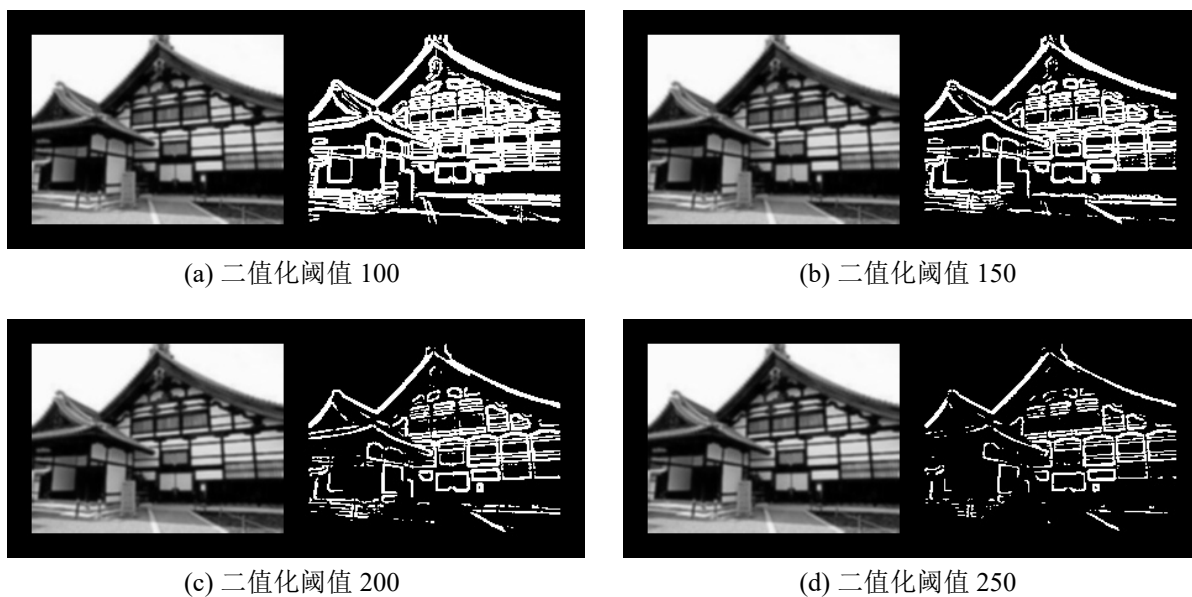


图 3.2: 不同阈值二值化后 Prewitt 算子检测结果

图3.3展示了使用不同大小的高斯核进行滤波时得到的边缘检测结果。该组实验使用阈值自适应的算法，对图像加噪，取高斯滤波核的尺寸为 3×3 , 5×5 , 7×7 , 9×9 。从检测结果可以看出，当高斯滤波核的尺寸增大时，图像变得模糊，对噪声的抑制越来越强，但边缘检测结果也相应变得模糊，只能检测出较模糊的边缘，一些边缘丢失，检测出的边缘比较粗。

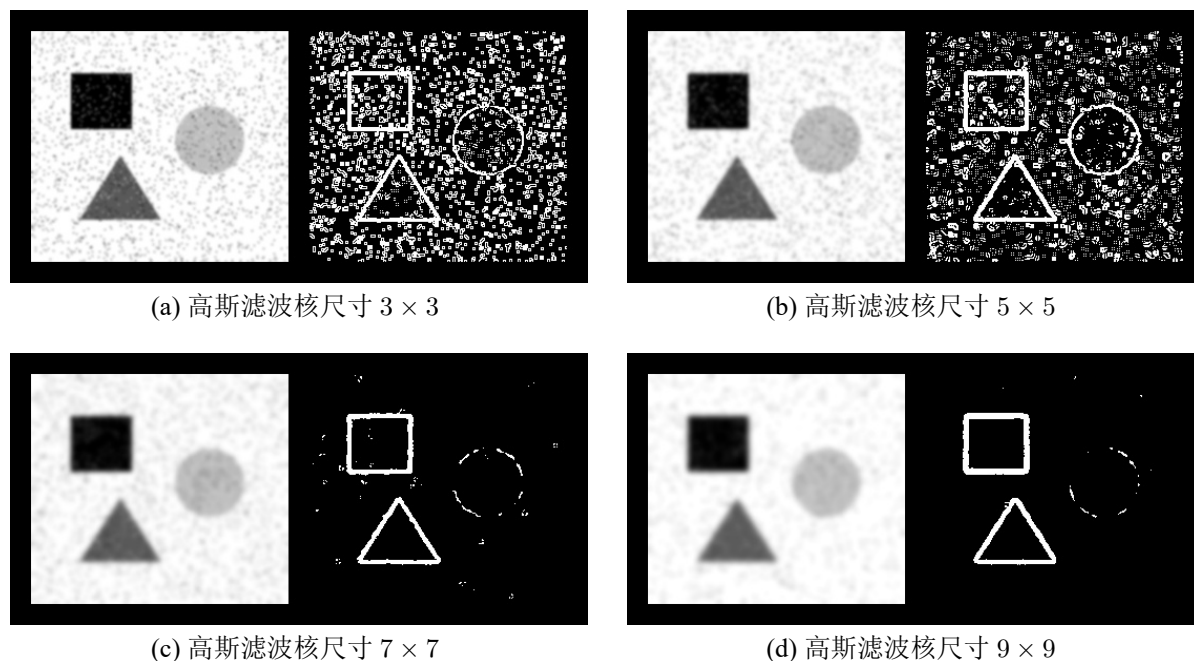
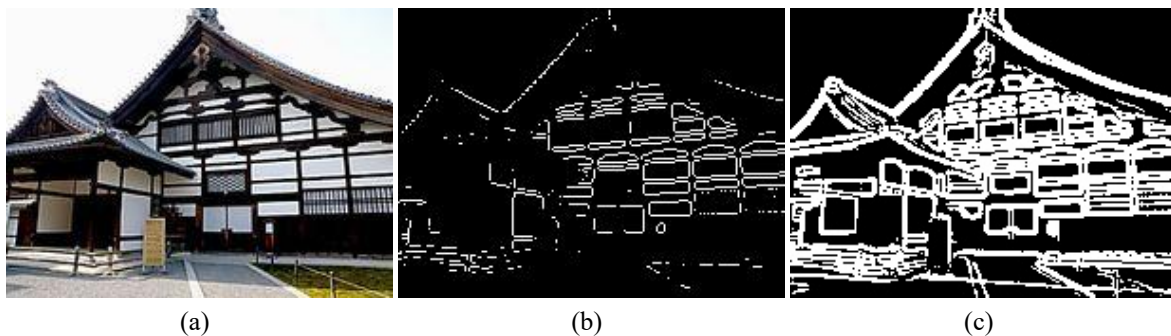


图 3.3: 不同高斯核滤波后 Prewitt 算子检测结果

图3.4为使用 Matlab 中 `edge` 函数以及本文实现 Prewitt 算子的检测结果。其中图 (a) 为输入源图像，图 (b) 为 Matlab 中 `edge` 函数使用自动选择阈值的 Prewitt 算子进行边缘检测的结果，图 (c) 为本文实现的阈值自适应的 Prewitt 算子的检测结果。从检测结果可以看出，Matlab 中 `edge` 函数提取的边缘较细，有一些边缘丢失，而本文实现算子提取的边缘较粗，边缘保留相对比较完整。我们认为可能是 Matlab 中自动选择的阈值比较高，导致保留的边缘较少，从而一些边缘被过滤。

图 3.4: Matlab 中 `edge` 函数和本文实现 Prewitt 算子的检测结果

3.2 Sobel 算子检测结果

使用 Sobel 算子检测边缘的效果如图3.5所示，其中图 (a) 为输入源图像，图 (b) 从左到右分别为 (1) 源图像灰度化后的图像，(2) 经过水平方向差分后得到的 G_x 图像，(3) 经过垂直方向差分后得到的 G_y 图像，(4) 二值化的边缘检测结果。从图中可以看出， G_x 图像主要提取出了垂直方向的边缘， G_y 图像主要提取出了水平方向的边缘。

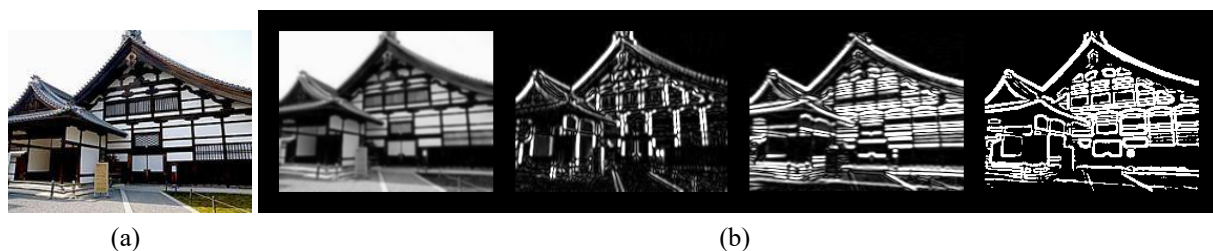


图 3.5: Sobel 算子检测结果

图3.6为使用不同的阈值对梯度进行二值化的边缘检测结果。该组实验固定高斯滤波核的尺寸为 5×5 ，取二值化阈值为 100,150,200,250，从检测结果可以看出，当阈值较低时，保留了大量的假边缘，检测结果比较杂乱，检测的边缘比较粗；当阈值较高时，检测的边缘比较细，比较清晰，但是也存在边缘丢失的情况。

图3.7展示了使用不同大小的高斯核进行滤波时得到的边缘检测结果。该组实验使用阈值自适应的算法，对图像加噪声，取高斯滤波核的尺寸为 $3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9$ 。检测结果与 Prewitt 算子检测结果相似，当高斯滤波核的尺寸增大时，图像变得模糊，对噪声的抑制越来越强，但边缘检测结果也相应变得模糊，只能够检测出较模糊的边缘，一些边缘丢失，检测出的边缘比较粗。Sobel 算子与 Prewitt 算子的检测结果相比，更好地抑制了噪声，但是检测出的边缘更粗，定位精确度更差。

图3.8为使用 Matlab 中 `edge` 函数以及本文实现 Sobel 算子的检测结果。其中图 (a) 为输入源图像，图 (b) 为 Matlab 中 `edge` 函数使用自动选择阈值的 Sobel 算子进行边缘检测的结

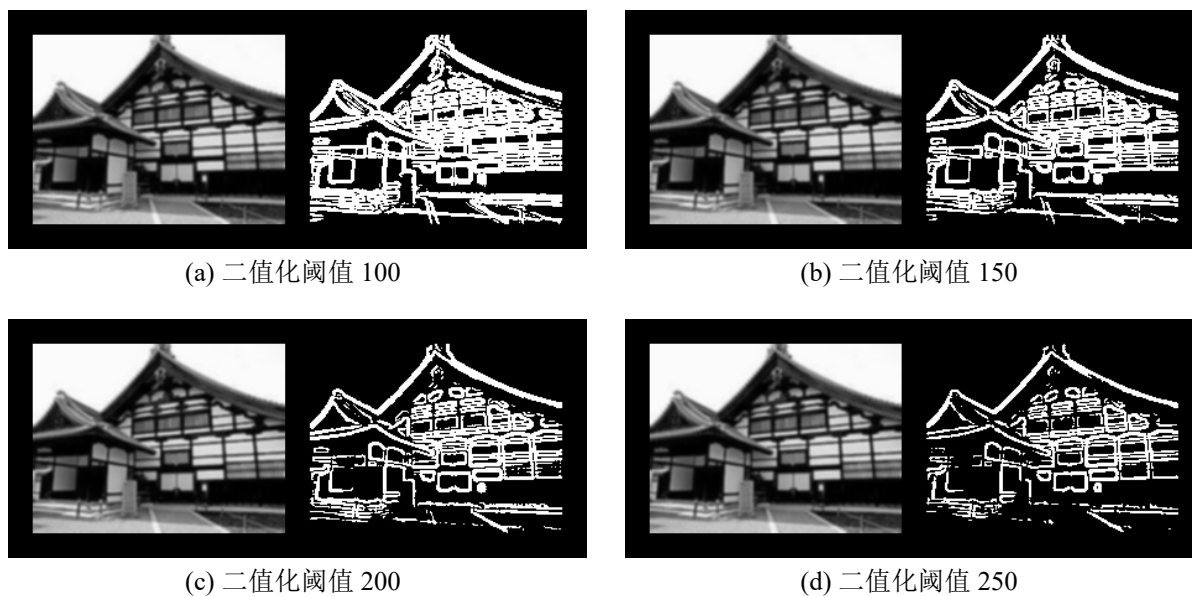


图 3.6: 不同阈值二值化后 **Prewitt** 算子检测结果

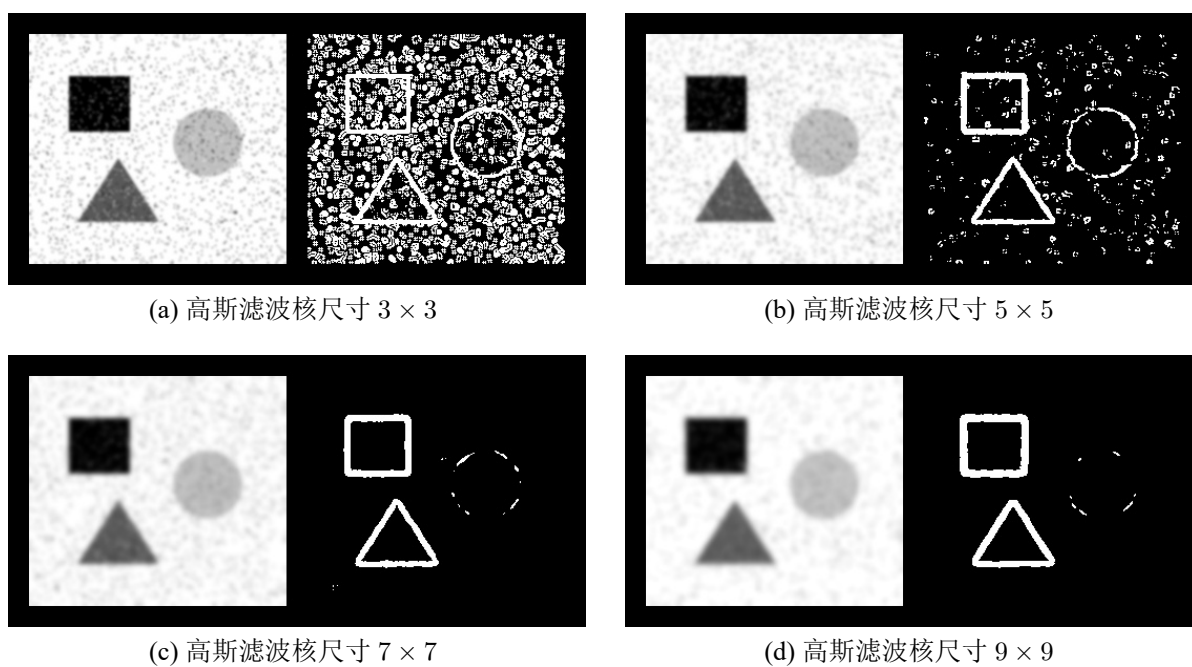


图 3.7: 不同高斯核滤波后 **Sobel** 算子检测结果

果,图(c)为本文实现的阈值自适应的 Sobel 算子的检测结果。从检测结果可以看出,Matlab 中 `edge` 函数提取的边缘较细,有一些边缘丢失,而本文实现算子提取的边缘较粗,边缘保留相对比较完整。同样,我们认为可能是 Matlab 中自动选择的阈值比较高,导致保留的边缘较少。

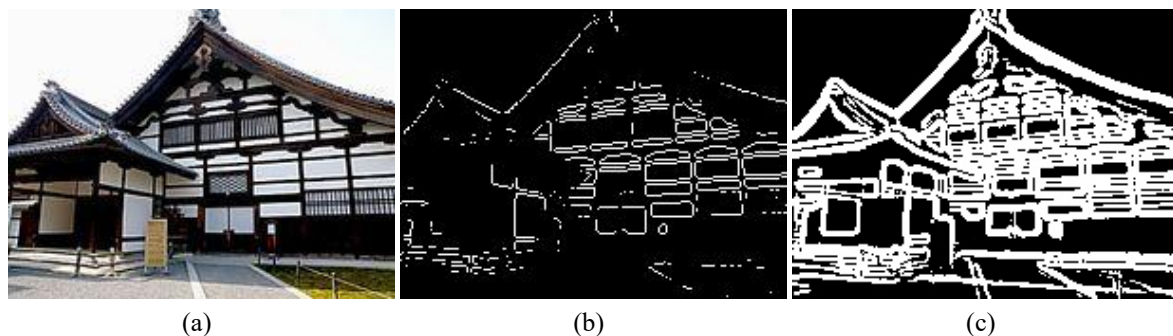


图 3.8: Matlab 中 `edge` 函数和本文实现 Sobel 算子的检测结果

3.3 Canny 算子检测结果

使用 Canny 算子检测边缘的效果如图3.9所示其中图(a)为输入源图像,图(b)从左到右分别为(1)源图像灰度化后的图像(2)二值化的边缘检测结果。从图中可以看出,Canny 算子的检测结果相比 Prewitt 和 Sobel 算子更为精细,对一些细节部分的边缘能更好地提取,Prewitt 算子和 Sobel 算子检测的边缘比较粗,而 Canny 算子检测的边缘比较细。

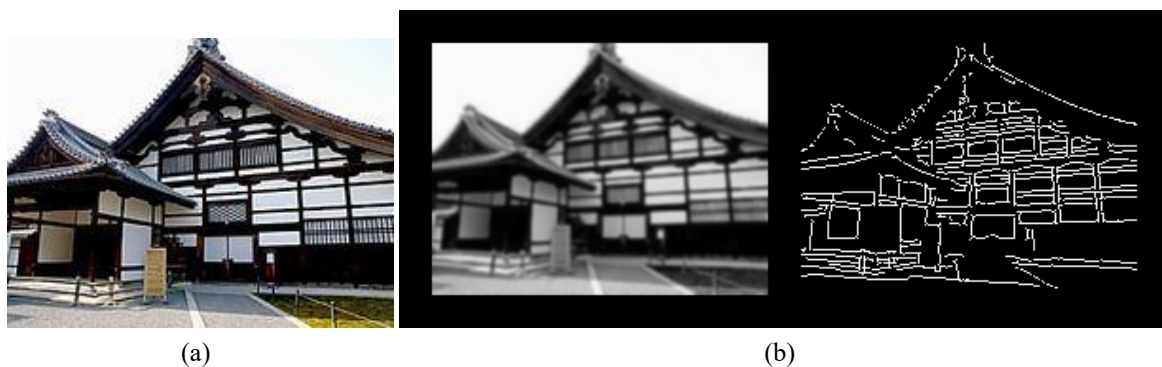


图 3.9: Canny 算子检测结果

图3.10为使用不同双阈值时得到的检测结果。该组实验固定高斯滤波核的尺寸为 5×5 , 分别取高、低阈值对 (T_H, T_L) 为 $(50, 300)$, $(50, 600)$, $(200, 300)$, $(200, 600)$ 。检测结果显示,高阈值对整个图像边缘的过滤效果比较明显。当高阈值取值较大时,被保留的边缘较少,边缘检测结果比较粗糙。低阈值相比高阈值对整个图像的边缘检测效果影响不是很明显,但是

对一些细节的边缘还有边缘连通情况有影响。当低阈值取值较小时，更多的边缘细节被保留下来，边缘的连通度也较高，但相应的可能存在假边缘被保留的情况。

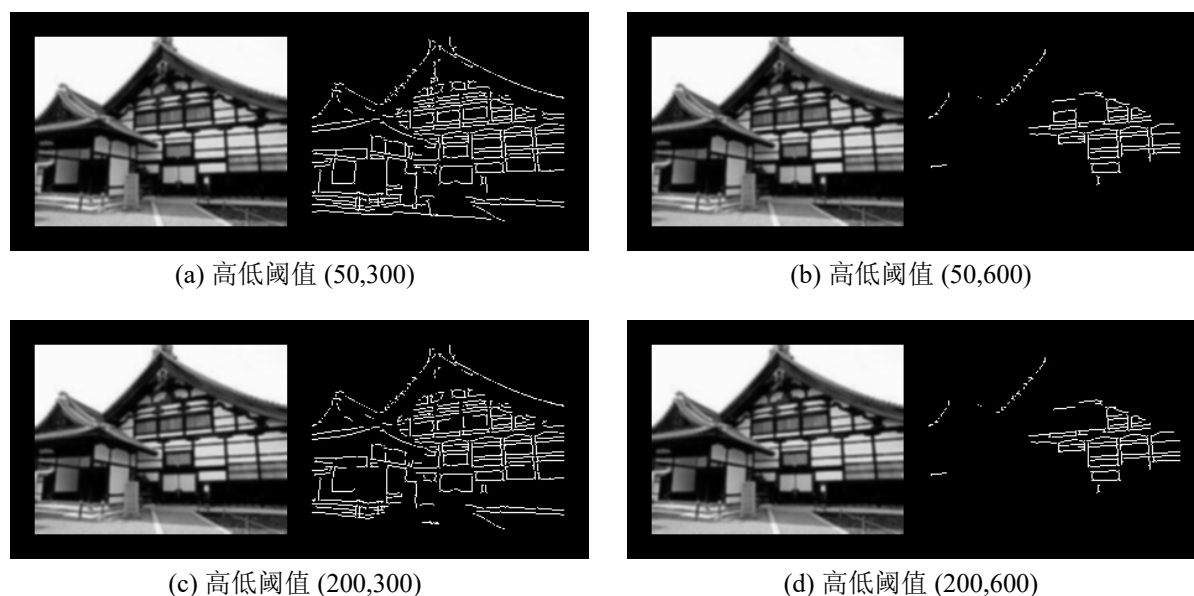


图 3.10: 不同高低阈值对应的 Canny 算子检测结果

图3.11展示了使用不同大小的高斯核进行滤波时得到的边缘检测结果。该组实验使用自适应的算子，对图像加噪，取高斯滤波核尺寸为 3×3 , 5×5 , 7×7 , 9×9 。从检测结果可以看出，当高斯滤波核的尺寸增大时，对噪声的抑制越来越强，但是边缘检测结果变得不准确，只能够检测出不完整不连续的边缘，一些边缘信息会丢失。与 Prewitt 和 Sobel 算子相比，高斯滤波核尺寸增加，检测出的边缘还是相对较细，并没有明显变粗，并且对噪声的抑制更有效。

图3.12为使用 Matlab 中 edge 函数以及本文实现 Canny 算子的检测结果。其中图 (a) 为输入源图像，图 (b) 为 Matlab 中 edge 函数使用自动选择阈值的 Canny 算子进行边缘检测的结果，图 (c) 为本文实现的阈值自适应的 Canny 算子的检测结果。从检测结果可以看出，Matlab 中 edge 函数提取的边缘较完整精细，连通性较好，而本文实现的 Canny 算子的检测结果连通性较差，有部分响应杂乱的情况，总体效果不如 Matlab 中的 edge 函数。对此我们认为本文所采用的阈值自适应策略还不是非常合适，可能需要更复杂有效的策略来自适应地确定阈值。

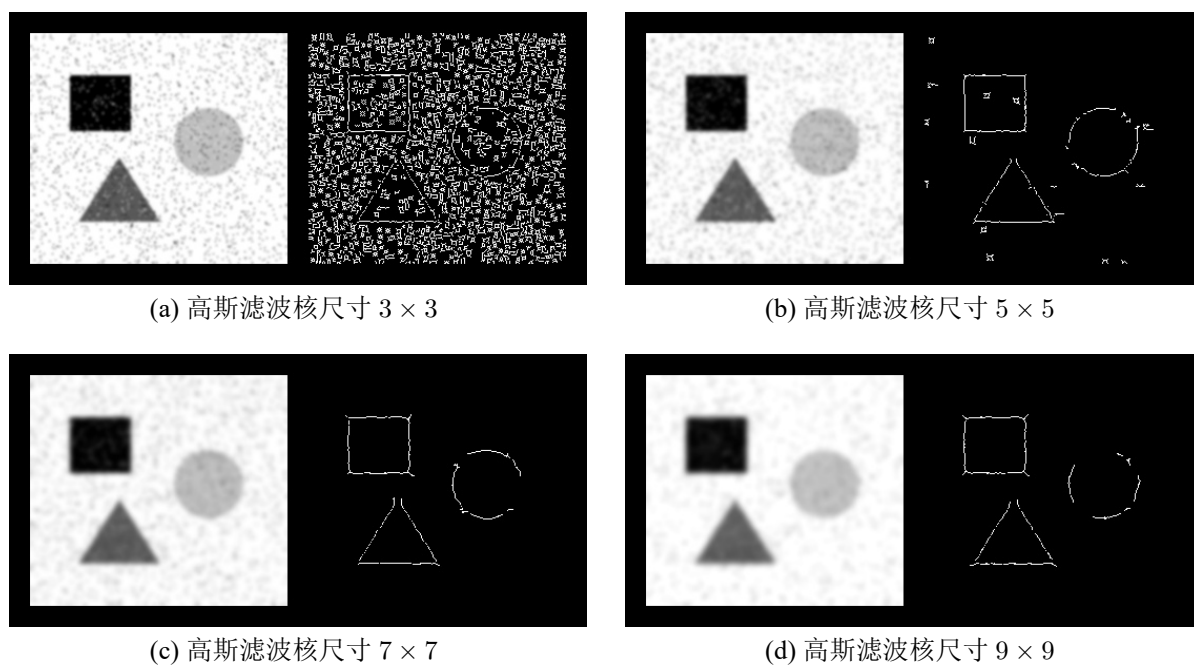


图 3.11: 不同高斯核滤波后 **Canny** 算子检测结果

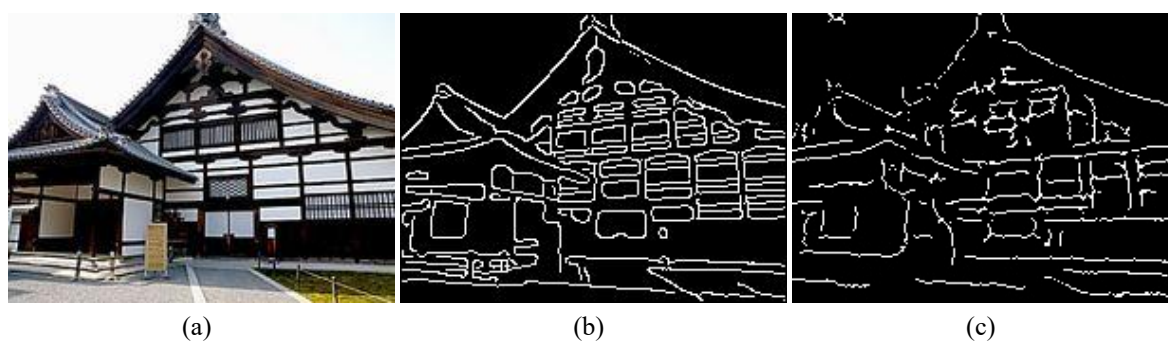


图 3.12: **Matlab** 中 **edge** 函数和本文实现 **canny** 算子的检测结果

3.4 性能对比分析

本节中，我们将从抗噪性能、计算复杂度这些方面对 Prewitt、Sobel 和 Canny 算子进行性能对比及分析。这里为了更好地观察检测结果，我们对 Prewitt、Sobel 和 Canny 算子的阈值不采用人工设定的方式，而是采用自适应的方式确定，具体见2.4节。

3.4.1 抗噪性能

在该组实验中，我们给图像添加了椒盐噪声以及高斯噪声，得到的检测结果如图3.13所示。其中图像的第一列为灰度图，第二列为 Prewitt 算子检测结果，第三列为 Sobel 算子检测结果，第四列为 Canny 算子检测结果。图 (a) 为源图像不添加噪声的检测结果，图 (b) 为添加了 2000 个点椒噪声和 2000 个点盐噪声图像的检测结果，图 (c) 为添加了均值为 0，方差为 0.8 的高斯噪声的图像的检测结果。

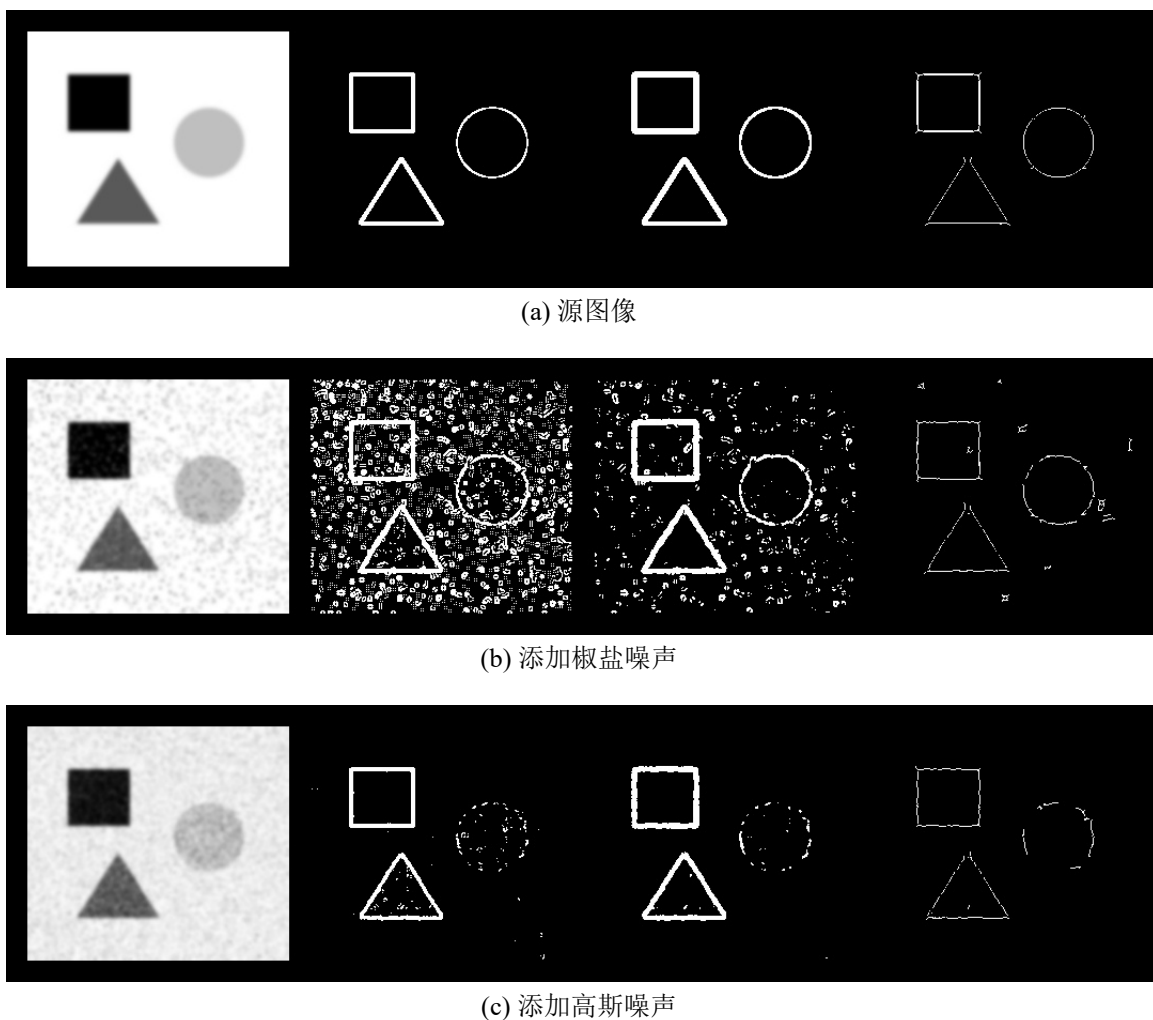


图 3.13: 添加噪声的边缘检测结果

从实验结果可以看出，在源图像的检测结果比较准确的情况下，

- 1) 添加了椒盐噪声的图像检测结果显示, 相比 Prewitt 和 Sobel 算子, Canny 算子受到噪声的影响较小, 边缘的检出和定位相对还是比较准确的, 且噪声保留较少, 但是边缘的连通性和平滑性受到了影响。而 Prewitt 算子和 Sobel 算子受到噪声的影响稍大, 虽然具有一定的抗噪能力, 对象的边缘能完整检出, 但是检测结果包含较多的噪声点。从结果来看, Sobel 算子比 Prewitt 算子的检测结果要稍清晰一些。
- 2) 添加了高斯噪声的图像检测结果显示, Canny 算子受到噪声的影响, 检测结果的连通性和平滑性变差, 对加噪图像的边缘定位相对准确, 且噪声点几乎都被过滤, 但是也有部分边缘信息丢失。Prewitt 和 Sobel 算子的检测结果保留了部分噪声点, 但是边缘的信息比较完整。同样 Sobel 算子相比 Prewitt 算子过滤了更多的噪声点。

因此我们认为 Prewitt 算子和 Sobel 算子具有一定的抗噪能力, 检出边缘较粗, 在噪声干扰下仍然能够保留较多边缘信息, 但是对噪声的过滤表现较差, 保留了较多的噪声点。Canny 算子在抗噪性能和定位精确度上表现较好, 能过滤较多的噪声, 但是在噪声的干扰下, 边缘的连通性和平滑性容易受到影响。

3.4.2 计算复杂度

我们对相同的一张 245×219 的图像进行测试, 其计算复杂度如表3.1所示。

- 1) 从时间复杂度上看, Prewitt 算子的运算速度最快, Sobel 算子的运算时间与 Prewitt 相近, 但是从计算梯度的卷积核可知, Sobel 算子与 Prewitt 算子相比每次卷积增加了乘法和加法运算, 因此运算时间更长。Canny 算子除了需要计算梯度, 还有非极大值抑制、双阈值检测和边缘连接的过程, 运算时间最长。
- 2) 从空间复杂度上看, Prewitt 算子和 Sobel 算子需要额外空间较少, 仅需要一个图像大小的矩阵用于记录梯度值。而 Canny 算子需要的额外空间较多, 需要两个图像大小的矩阵用于记录梯度值, 梯度方向以及在非极大值抑制、双阈值检测和边缘连接的过程中进行标记, Canny 算子需要的空间也最大。

| 算法 | 运行时间 | 需要的最少额外空间 |
|---------|-------------|---------------------------|
| Prewitt | 0.0112071 s | 245×219 |
| Sobel | 0.0120303 s | 245×219 |
| Canny | 0.0576211 s | $245 \times 219 \times 2$ |

表 3.1: 三种算子的计算复杂度

第4章 总结

通过实验，我们对三种边缘检测算子的特点总结如下：

- Prewitt 算子的边缘检测结果比较粗糙，检出的边缘一般比较宽也比较模糊，对边缘的定位精确度较弱。该算子包含差分运算和局部平均运算，因而对噪声有一定的抑制能力，但是相比 Canny 算子抗噪性能较弱。Prewitt 算子的运算比较简单，运算速度在三者中 fastest，且需要的空间也较少。
- Sobel 算子和 Prewitt 算子相似，边缘检测的结果也比较粗糙，检出的边缘比较宽且比较模糊，对边缘的定位精确度较弱。该算子包含差分运算和局部加权平均运算，由于卷积核在中心系数上使用了权值 2，使得 Sobel 算子和 Prewitt 算子相比能更好地抑制噪声，但其抗噪能力与 Canny 算子相比仍然比较弱。Sobel 算子的运算也比较简单，运算速度仅次于 Prewitt，需要的空间也较少。
- Canny 算子的检测结果比较精细，检出的边缘一般比较细，并且能够保留一些弱边缘，从而可以更好地检测到边缘细节信息，准确性较高。该算子具有较好的抗噪性能，并且对边缘的定位精确度较高，但是边缘连通性和平滑性容易受到噪声的干扰。与 Prewitt 算子和 Sobel 算子相比，Canny 算子计算的时间复杂度和空间复杂度都较大。

另外，我们通过实验总结出滤波核的大小、阈值的选取对三种算子的影响：

- 为了减少噪声干扰，在进行梯度计算前我们往往选择对图像进行平滑处理。当滤波核的尺寸较大时，虽然能够更好地抑制噪声干扰，但是边缘也会相应变得更加模糊，使得边缘检测的结果不准确不完整。
- Prewitt 算子和 Sobel 算子的二值化阈值以及 Canny 算子的高低阈值均对检测结果有比较大的影响。对 Prewitt 算子和 Sobel 算子，当阈值较低时，可能保留了大量的假边缘，检测结果杂乱，边缘比较粗；而阈值较高时，检出边缘比较细且清晰，但是可能有边缘丢失、不连续的情况。对 Canny 算子，高阈值对整个图像边缘的过滤效果更明显，当高阈值较小时，可能保留了大量的假边缘，检测结果细而杂乱；而高阈值较大时，检测结果比较粗糙，被保留的边缘较少，边缘检测不完整。低阈值的边缘过滤效果相对不明显，对一些细节以及边缘连通情况有一定影响，当低阈值较小时，更多的边缘细节保留，边缘的连通度也较高，但相应的可能存在假边缘被保留的情况。

由此可以看出三种边缘检测算子各有利弊。一阶差分算子计算简单，但抗噪性能和定位精确性较弱；基于最优化的 Canny 算子抗噪性能和定位精确度较好，但是计算复杂性高，且边缘连通性和平滑性容易受到噪声干扰。另外，我们发现抑制噪声和精确边缘的平衡问题以及对阈值的选择都是比较困难的。即使我们使用阈值自适应，依旧可能找不到最合适的阈值。同时，在实验过程中我们发现大多数对检测结果的评价都是人为的定性评价，针

对边缘检测并没有比较通用的定量评价标准。因而边缘检测还是一个尚存很多问题且比较复杂困难的任务。

参考文献

- [1] 马颂德张正友. 计算机视觉[M]. : 科学出版社, 1998.
- [2] OTSU N. A threshold selection method from gray-level histograms[J]. IEEE transactions on systems, man, and cybernetics, 1979, 9(1): 62-66.
- [3] CANNY J. A computational approach to edge detection[J]. IEEE Transactions on pattern analysis and machine intelligence, 1986(6): 679-698.