形态学边缘提取

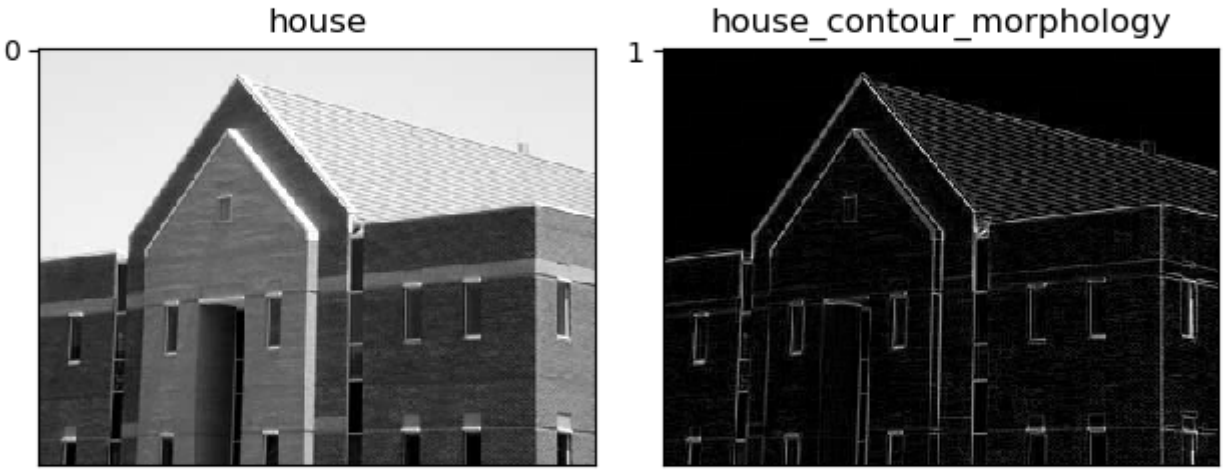
$\beta(A) = A - (A \Theta B)$

实现代码：

```
def contour_morphology(image):
    return image - cv2.erode(image, np.ones((3, 3), np.uint8))
```

实现很简单，原图减腐蚀的图

结果

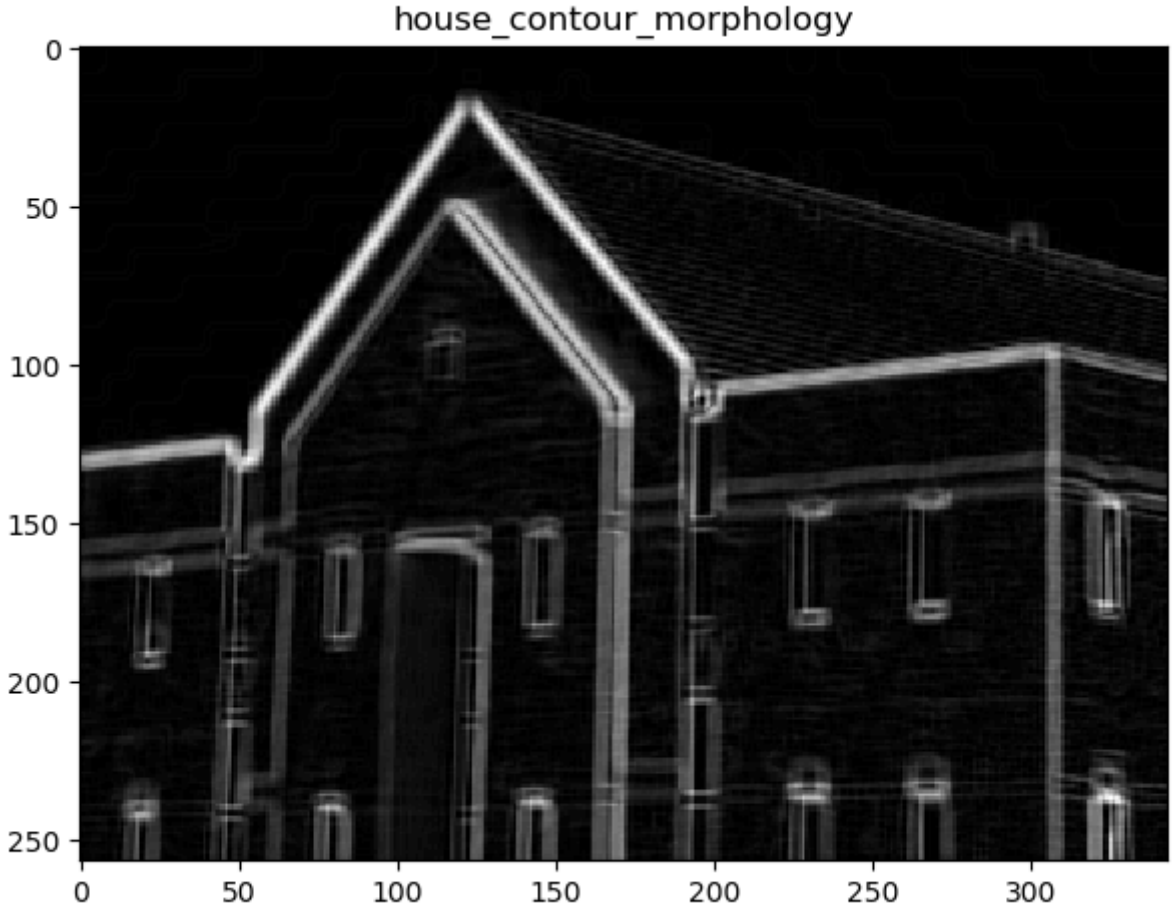


为了去除房顶瓦片，对其进行均值平滑处理后的结果

```
mask_averaging = np.array([
    [1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1],
```

```
    [1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1],
], np.float)/25
house_averaging = cv2.filter2D(house, -1, mask_averaging)
house_contour_morphology = contour_morphology(house_averaging )
```

结果



虽然瓦片被去除了，可是轮廓边粗了，房顶边缘变弱

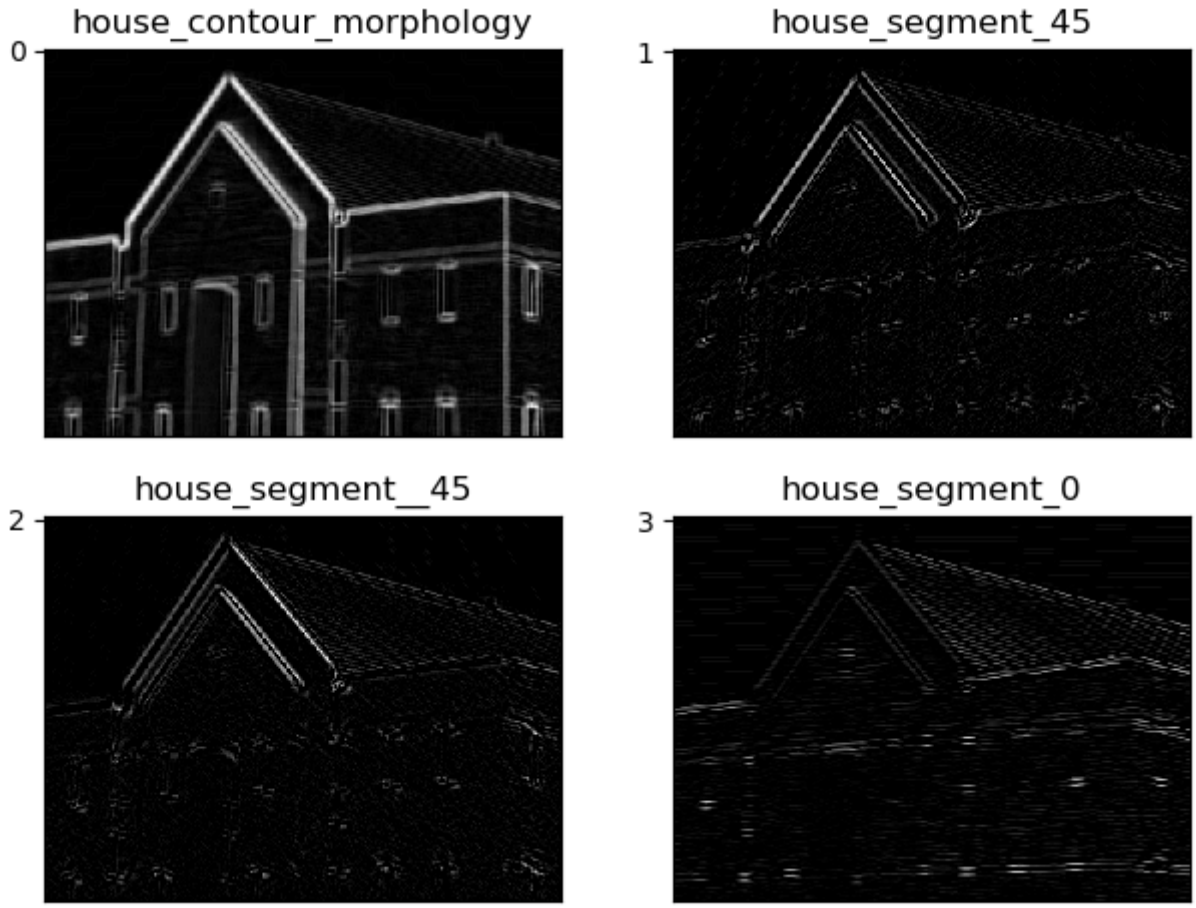
线的检测

```
mask_45 = np.array([
    [-1, -1,  2],
    [-1,  2, -1],
    [2,  -1, -1],
])

house_segment_45 = cv2.filter2D(house_contour_morphology, -1, mask_45)
mask__45 = np.array([
    [2, -1, -1],
    [-1, 2, -1],
    [-1, -1, 2],
])
house_segment__45 = cv2.filter2D(house_contour_morphology, -1, mask__45)
mask_0 = np.array([
    [-1, -1, -1],
    [2,   2,  2],
    [-1, -1, -1],
])
house_segment_0 = cv2.filter2D(house_contour_morphology, -1, mask_0)
```

用上一步的输出house_contour_morphology作为输入：

结果



方向固定，不能很好的检测直线

梯度算子

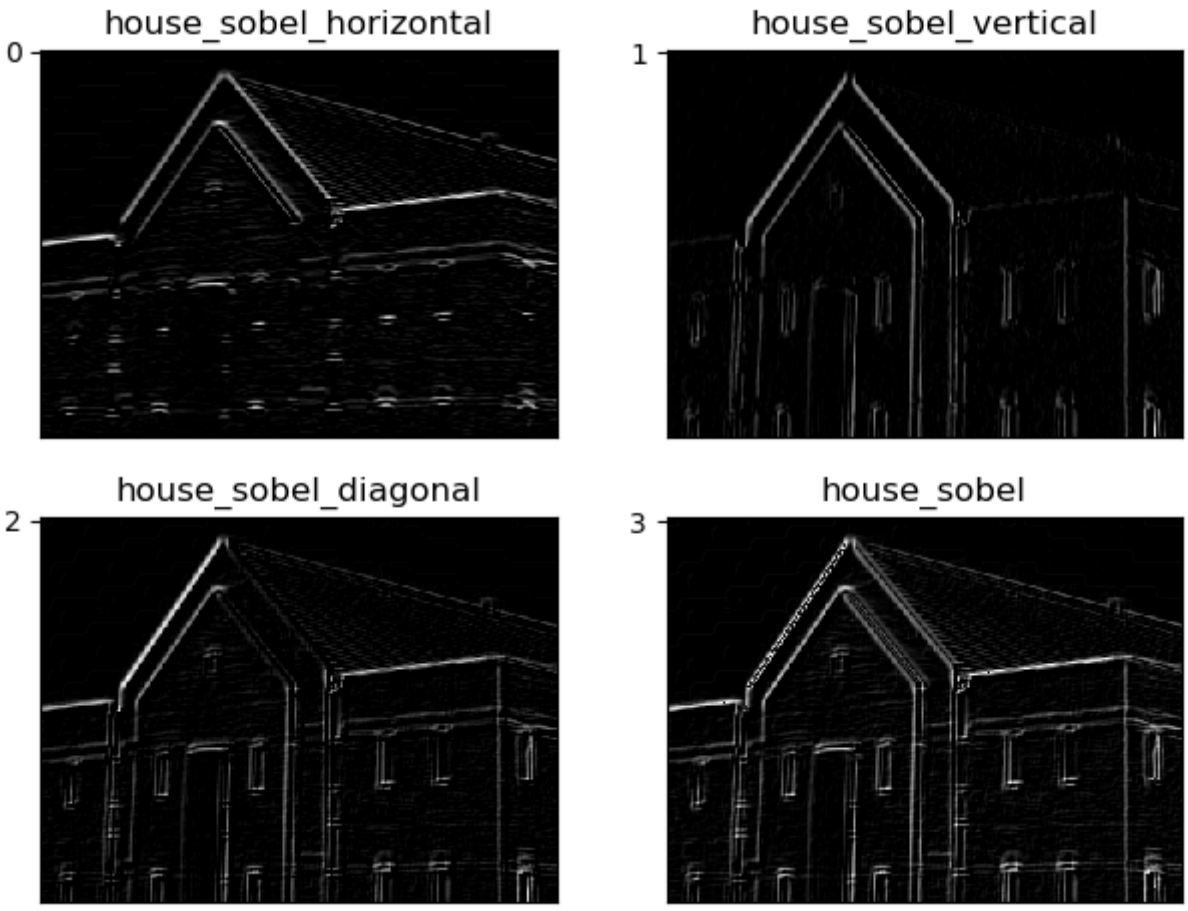
```
mask_sobel_horizontal = np.array([
    [-1, -2, -1],
    [0,   0,  0],
    [1,   2,  1],
])
mask_sobel_vertical = np.array([
    [-1, 0, 1],
    [-2, 0, 2],
    [-1, 0, 1],
])

mask_sobel_diagonal = np.array([
    [-2, -1, 0],
    [-1, 0,  1],
    [0,  1,  2],
])
```

```
house_sobel_horizontal = cv2.filter2D(house_contour_morphology, -1, mask_sobel_h
house_sobel_vertical = cv2.filter2D(house_contour_morphology, -1, mask_sobel_ver
house_sobel_diagonal = cv2.filter2D(house_contour_morphology, -1, mask_sobel_dia
plt_show_opcv("house_sobel", house_sobel)
```

用上一步的输出house_contour_morphology作为输入：

结果

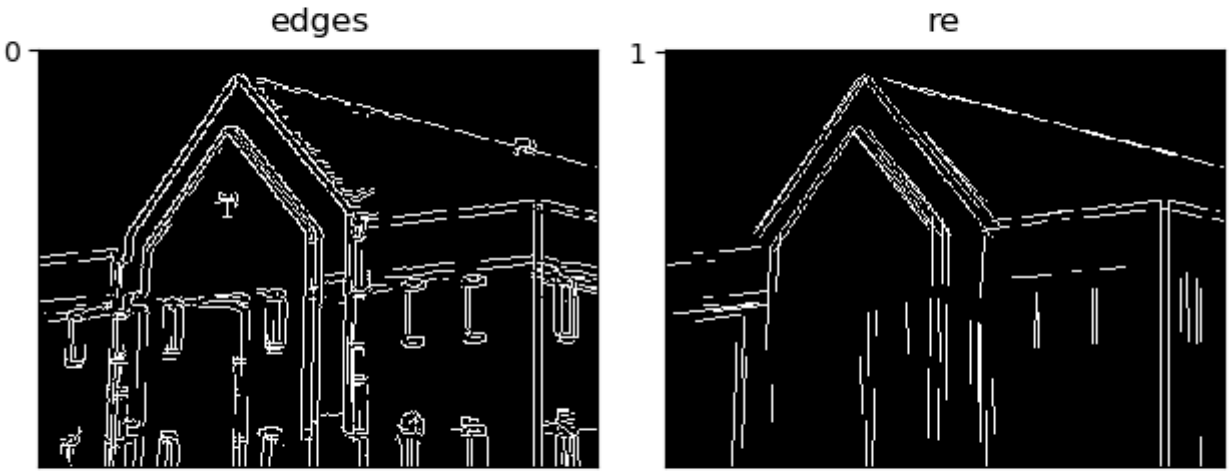


霍夫/Hough变换

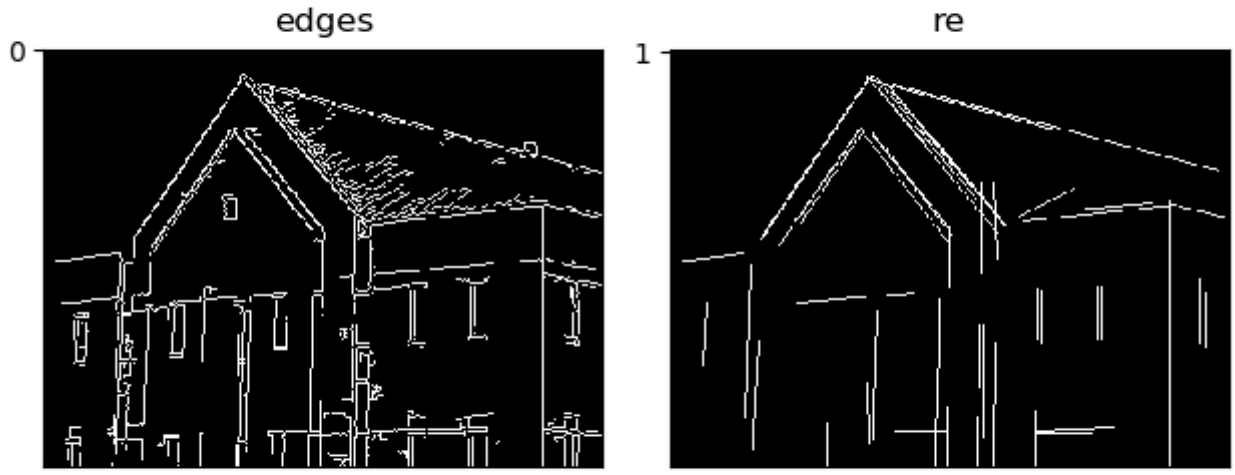
```
edges = cv2.Canny(house_contour_morphology, 50, 20)
re = np.zeros(house.shape, np.uint8)
lines = cv2.HoughLinesP(edges, 1, np.pi/180, 30, minLineLength=30, maxLineGap=5)
lines = lines[:, 0, :]
for x1, y1, x2, y2 in lines:
    cv2.line(re, (x1, y1), (x2, y2), (255, 255, 255), 1)
```

先用Canny算法进行边缘检测，再用霍夫变换检测直线轮廓。用上一步的输出house_contour_morphology作为输入：

结果



原图作为输入：


edges
0
re
1