MIT Lab.
Multimedia
Intelligent Technical
Laboratory

提取輪廓

目標任務

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

- **對此次課程提供的影像使用Sobel運算子、Laplacian濾波器、Hough直線檢測進行邊緣檢測(Edge Detection)**
- **使用函數:**
  - cv2.Sobel
  - cv2.Laplacian
  - cv2.HoughLines
- **使用以下參數對影像做Sobel操作**
  - 使用cv2.CV_32F型態
  - 使用尺寸為3的卷積核(Kernel)
  - 使用cv2.threshold對Sobel處理後的影像進行細線化步驟
- **使用以下參數對影像做Laplacian操作**
  - 使用cv2.CV_8U型態
  - 使用尺寸為3的卷積核(Kernel)
- **使用以下參數對影像做HoughLines操作**
  - 使用cv2.Canny處理後的輪廓影像作為輸入
  - 設定rho的數值為1
  - 設定theta的數值為np.pi/180
  - 設定threshold的數值為200

Sobel濾波器

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
cv2.Sobel(src, ddepth, dx, dy[, dst[, ksize[, scale[, delta[, borderType]]]]]) -> dst
```

Parameters	Description
src	input image.
dst	output image of the same size and the same number of channels as src .
ddepth	output image depth; in the case of 8-bit input images it will result in truncated derivatives.
dx	order of the derivative x.
dy	order of the derivative y.
ksize	size of the extended Sobel kernel; it must be 1, 3, 5, or 7.
scale	optional scale factor for the computed derivative values; by default, no scaling is applied.
delta	optional delta value that is added to the results prior to storing them in dst.
borderType	pixel extrapolation method. BORDER_WRAP is not supported.

Sobel Operator

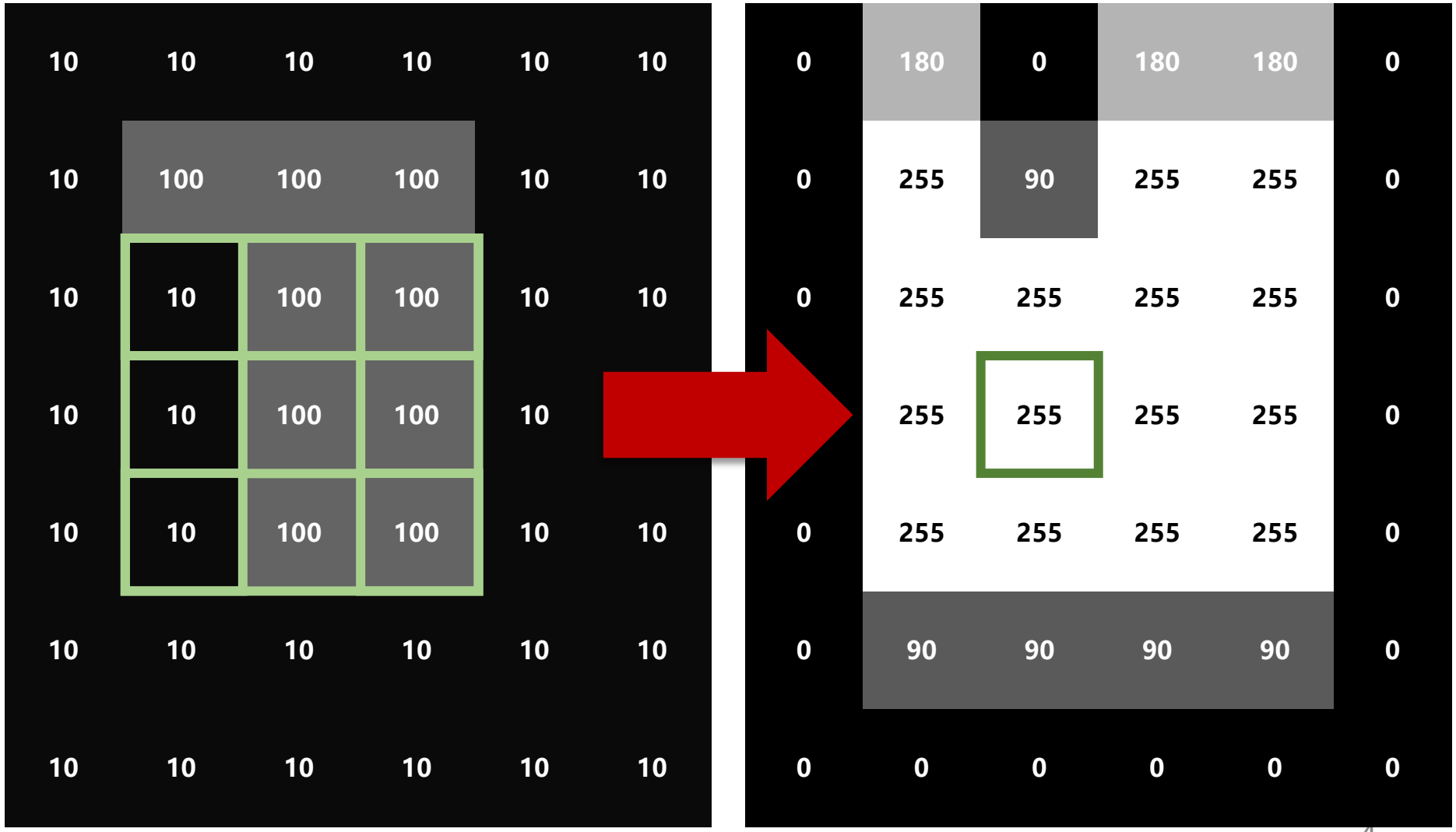
MIT Lab.
Multimedia
Intelligent Technical
Laboratory

Horizontal changes

-1	0	1
-2	0	2
-1	0	1

Vertical changes

-1	-2	-1
0	0	0
1	2	1



Sobel Operator

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
def Sobel_edge_detection(grayimage):
    grad_x = 
    grad_y = 
    absX = cv2.convertScaleAbs(grad_x)
    absY = cv2.convertScaleAbs(grad_y)
    sobel = cv2.addWeighted(absX, 0.5, absY, 0.5, 0)
    # 對Sobel影像進行細線化
    ret, th = cv2.threshold(sobel, 127, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return sobel, th
```

Laplacian Operator

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
cv2.Laplacian(src, ddepth[, dst[, ksize[, scale[, delta[, borderType]]]]]) -> dst
```

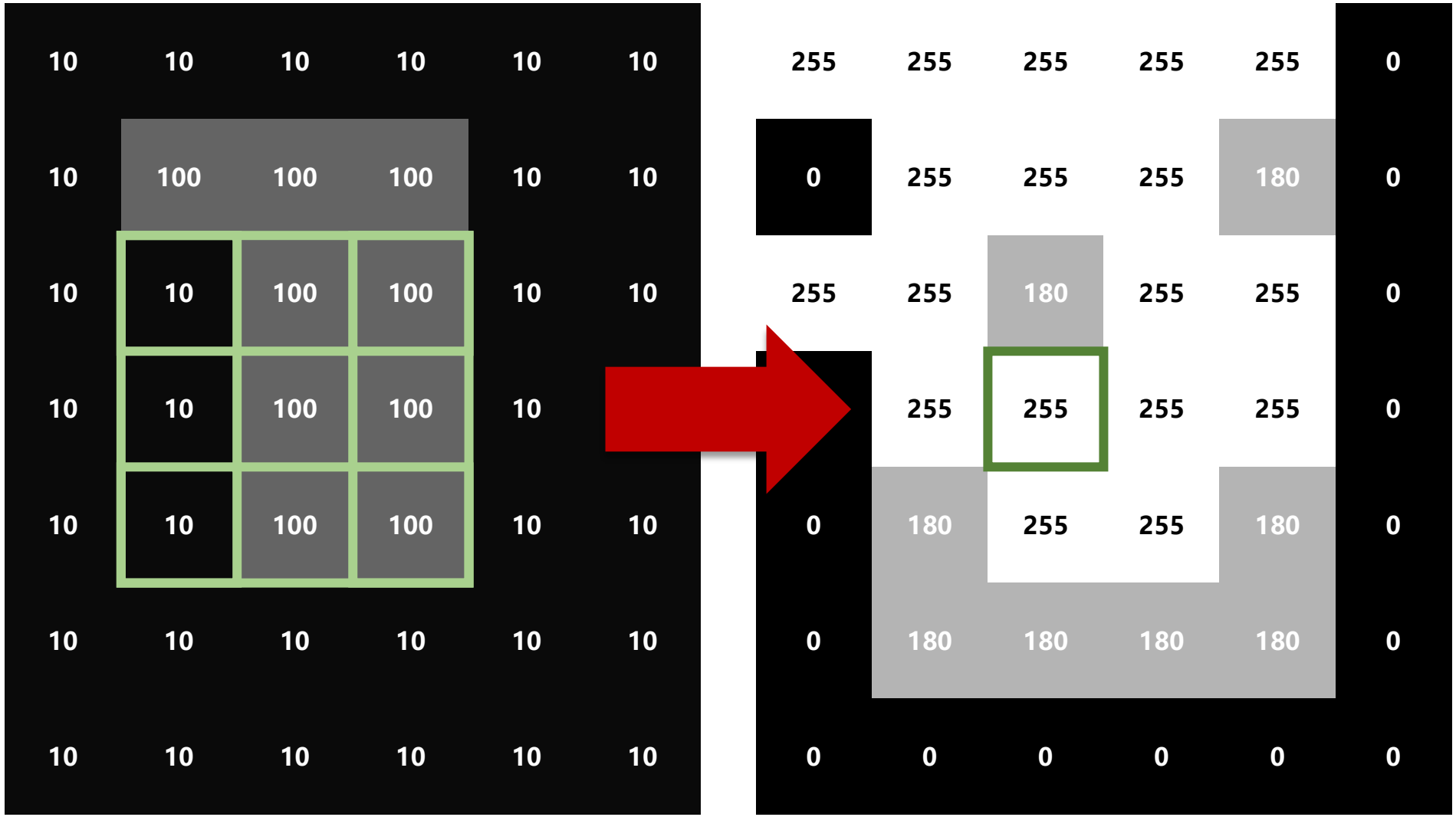
Parameters	Description
src	input image.
dst	Destination image of the same size and the same number of channels as src .
ddepth	Desired depth of the destination image, see combinations.
ksize	Aperture size used to compute the second-derivative filters. The size must be positive and odd.
scale	Optional scale factor for the computed Laplacian values. By default, no scaling is applied.
delta	optional delta value that is added to the results prior to storing them in dst.
borderType	pixel extrapolation method. BORDER_WRAP is not supported.

Laplacian Operator

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

Laplacian

0	1	0
1	-4	1
0	1	0



Laplacian Operator

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
def Laplacian_edge_detection(grayimage):
    lapimage = 
    lapimage = cv2.convertScaleAbs(lapimage)
    return lapimage
```

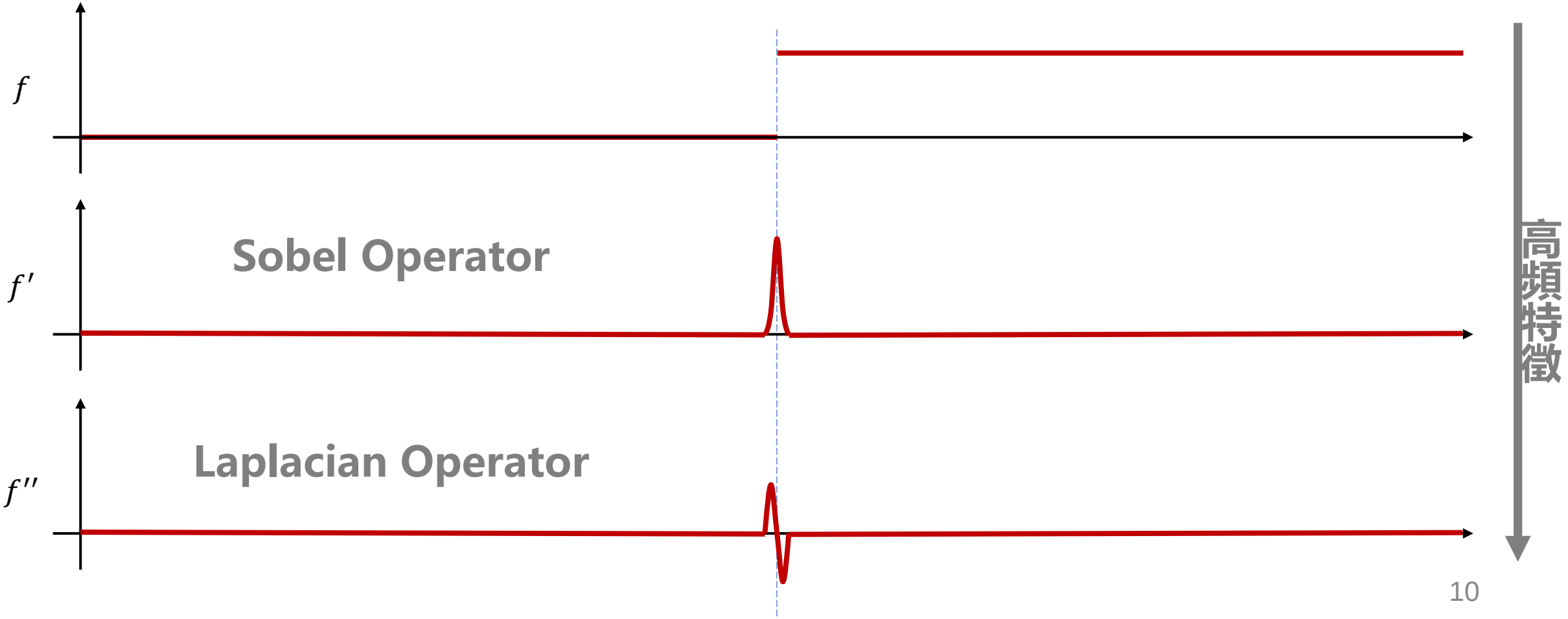
Operator Difference

ORIGIN

SOBEL

LAPLACIAN

Operator Difference



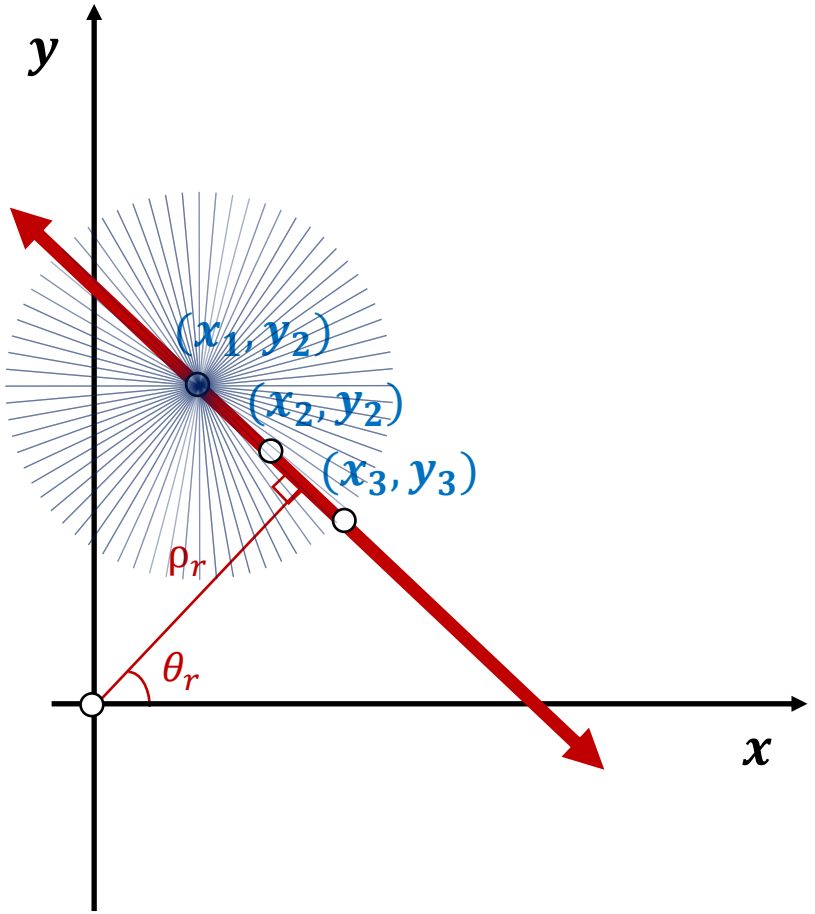
Hough Line Transform

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

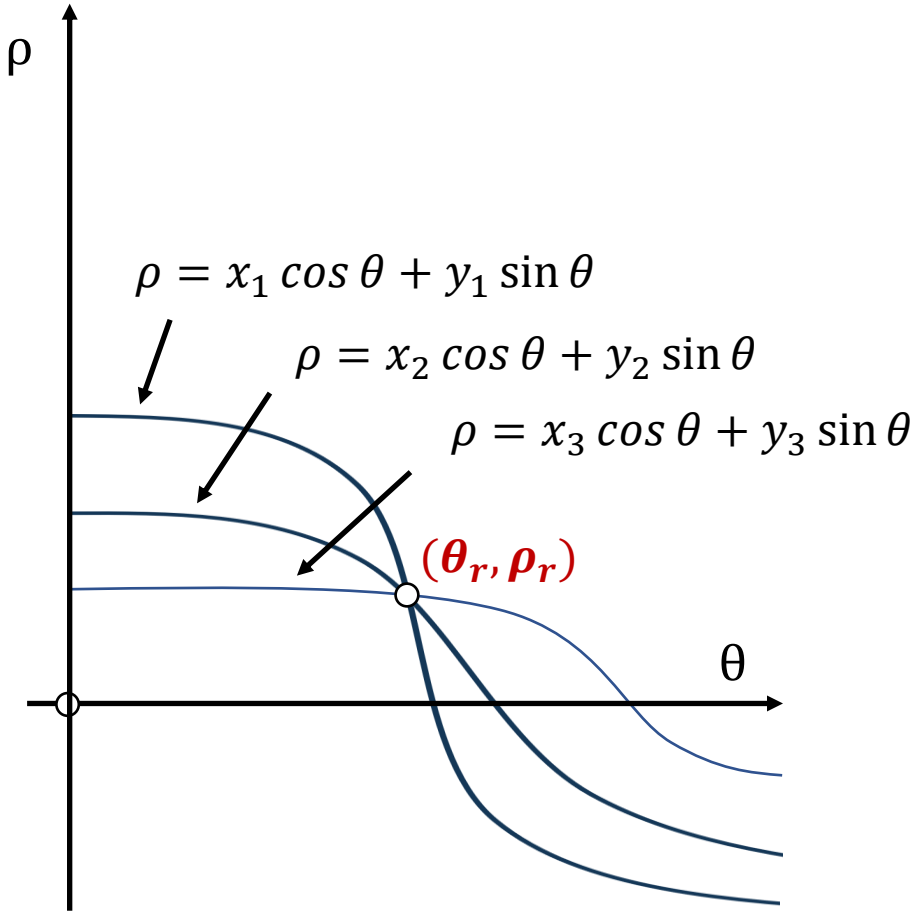
```
cv2.HoughLines(image, rho, theta, threshold[, lines[, srn[, stn[, min_theta[, max_theta]]]]]) -> lines
```

Parameters	Description
image	8-bit, single-channel binary source image. The image may be modified by the function.
lines	Output vector of lines.
rho	Distance resolution of the accumulator in pixels.
theta	Angle resolution of the accumulator in radians.
threshold	Accumulator threshold parameter. Only those lines are returned that get enough votes.
srn	For the multi-scale Hough transform, it is a divisor for the distance resolution rho. The coarse accumulator distance resolution is rho and the accurate accumulator resolution is rho/srn. If both srn=0 and stn=0, the classical Hough transform is used. Otherwise, both these parameters should be positive.
stn	For the multi-scale Hough transform, it is a divisor for the distance resolution theta.
min_theta	For standard and multi-scale Hough transform, minimum angle to check for lines. Must fall between 0 and max_theta.
max_theta	For standard and multi-scale Hough transform, an upper bound for the angle. Must fall between min_theta and CV_PI. The actual maximum angle in the accumulator may be slightly less than max_theta, depending on the parameters min_theta and theta.

Hough Line Transform



影像空間
(Image Space)



霍夫空間
(Hough Space)

Hough Line Transform

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
def HoughLinesDetection(image, grayimage):
    canny = cv2.Canny(grayimage, 50, 150, apertureSize=3)
    lines =
    lines = np.squeeze(lines)
    for rho, theta in lines:
        a, b = np.cos(theta), np.sin(theta)
        x0, y0 = a * rho, b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
        cv2.line(image, pt1, pt2, (0, 255, 0), 1)
    return canny, image
```

Main Function

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

image = cv2.imread('./sudoku.jpg')
grayimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
sobel, th = Sobel_edge_detection(grayimage)
lapimage = Laplacian_edge_detection(grayimage)
edges, hough = HoughLinesDetection(image, grayimage)

titles = ['GRAY', 'SOBEL', 'SOBEL THRESHOLD', 'LAPLACIAN', 'CANNY',
'HOUGH LINE']
images = [grayimage, sobel, th, lapimage, edges, hough]
for i in range(6):
    plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

實作結果

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

GRAY

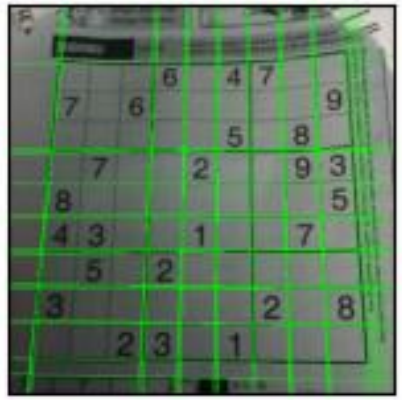
SOBEL

SOBEL THRESHOLD

LAPLACIAN

CANNY

HOUGH LINE



MIT Lab.
Multimedia
Intelligent Technical
Laboratory

Thanks for listening