MIT Lab.
Multimedia
Intelligent Technical
Laboratory

物體分割

目標任務

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

- **對此次課程提供的影像進行影像二值化(Image Thresholding)，以達到分割的效果**
- **使用函數：**
  - cv2.threshold
  - cv2.adaptiveThreshold
- **使用以下5種方法對影像做簡單二值化(門檻值：130)：**
  - cv2.THRESH_BINARY：將大於門檻值的灰階值設為最大灰階值，小於門檻值的值設為0。(Threshold Binary)
  - cv2.THRESH_BINARY_INV：將大於門檻值的灰階值設為0，其他值設為最大灰階值。(Threshold Binary, Inverted)
  - cv2.THRESH_TRUNC：將大於門檻值的灰階值設為門檻值，小於門檻值的值保持不變。(Truncate)
  - cv2.THRESH_TOZERO：將小於門檻值的灰階值設為0，大於門檻值的值保持不變。(Threshold to Zero)
  - cv2.THRESH_TOZERO_INV：將大於門檻值的灰階值設為0，小於門檻值的值保持不變。(Threshold to Zero, Inverted)
- **使用以下3種方法對影像做自適應二值化：**
  - cv2.THRESH_OTSU：透過窮舉法來從256個像素值內找出最佳的門檻值，該門檻值能使類間變異數最大(類內變異數最小)。
  - cv2.ADAPTIVE_THRESH_MEAN_C：將一個尺寸為blockSize的均值卷積核的計算結果減去常數C，以此來作為門檻值。
  - cv2.ADAPTIVE_THRESH_GAUSSIAN_C：將一個尺寸為blockSize的高斯卷積核的計算結果減去常數C，以此來作為門檻值。

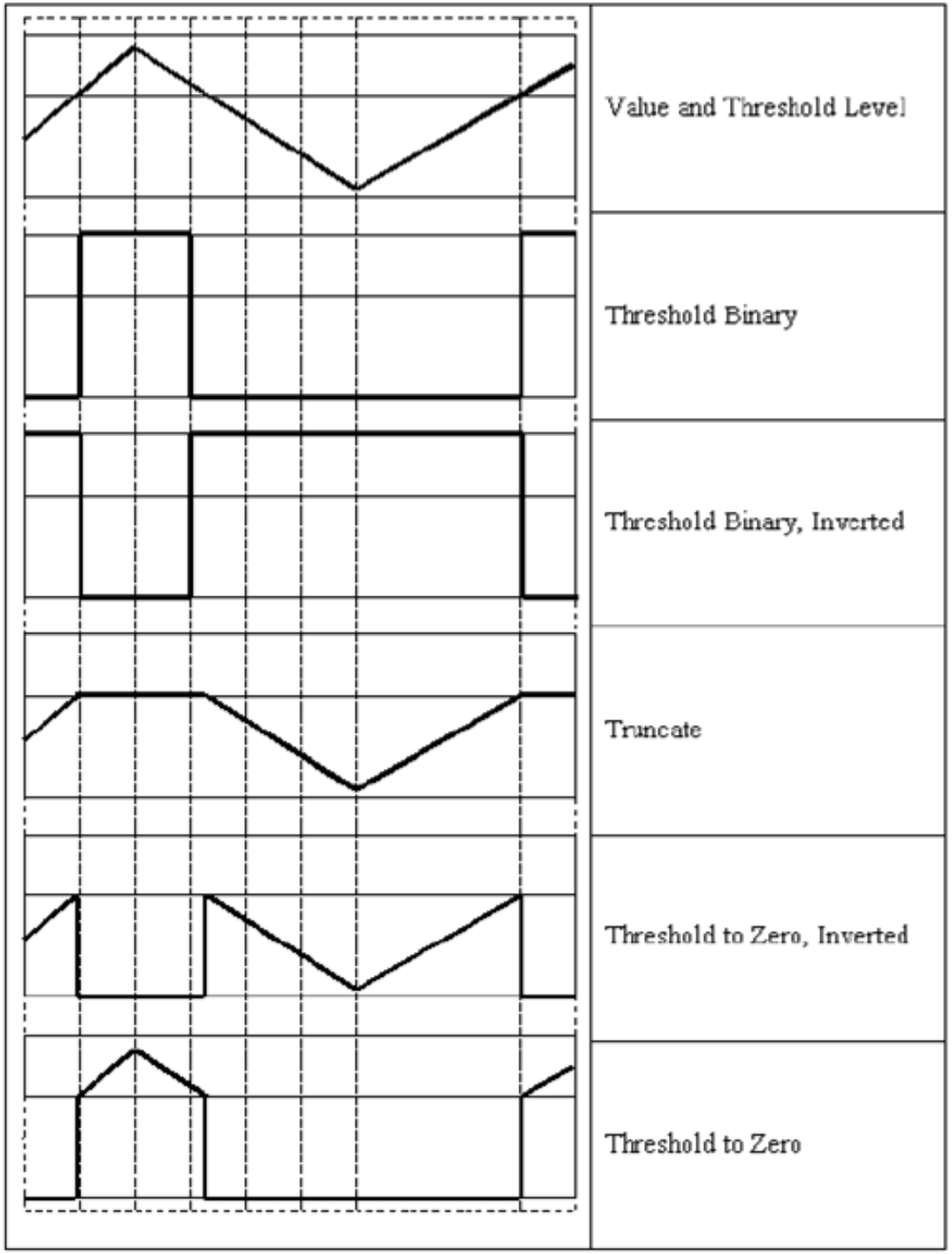
OTSU解說連結

簡單二值化

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

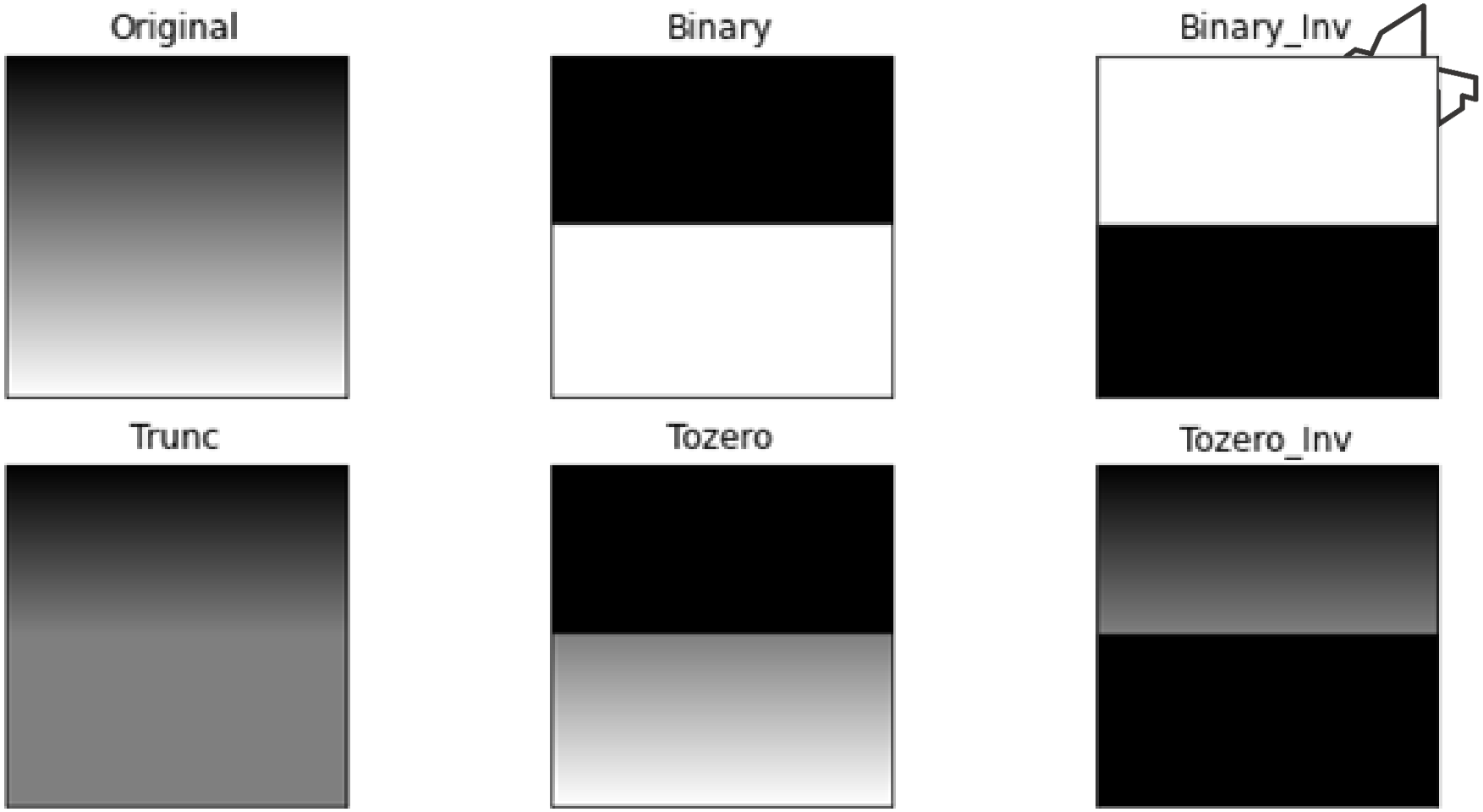
```
cv2.threshold(src, thresh, maxval, type[, dst]) -> retval, dst
```

Parameters	Description
src	input array (multiple-channel, 8-bit or 32-bit floating point).
dst	output array of the same size and type and the same number of channels as src.
thresh	threshold value.
maxval	maximum value to use with the THRESH_BINARY and THRESH_BINARY_INV thresholding types.
type	thresholding type (see ThresholdTypes).



MIT Lab.
Multimedia

Enumerator	
THRESH_BINARY Python: cv.THRESH_BINARY	$\texttt{dst}(x,y) = \begin{cases} \texttt{maxval} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_BINARY_INV Python: cv.THRESH_BINARY_INV	$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{maxval} & \text{otherwise} \end{cases}$
THRESH_TRUNC Python: cv.THRESH_TRUNC	$\texttt{dst}(x,y) = \begin{cases} \texttt{threshold} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$
THRESH_TOZERO Python: cv.THRESH_TOZERO	$\texttt{dst}(x,y) = \begin{cases} \texttt{src}(x,y) & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_TOZERO_INV Python: cv.THRESH_TOZERO_INV	$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$
THRESH_MASK Python: cv.THRESH_MASK	
THRESH_OTSU Python: cv.THRESH_OTSU	flag, use Otsu algorithm to choose the optimal threshold value
THRESH_TRIANGLE Python: cv.THRESH_TRIANGLE	flag, use Triangle algorithm to choose the optimal threshold value



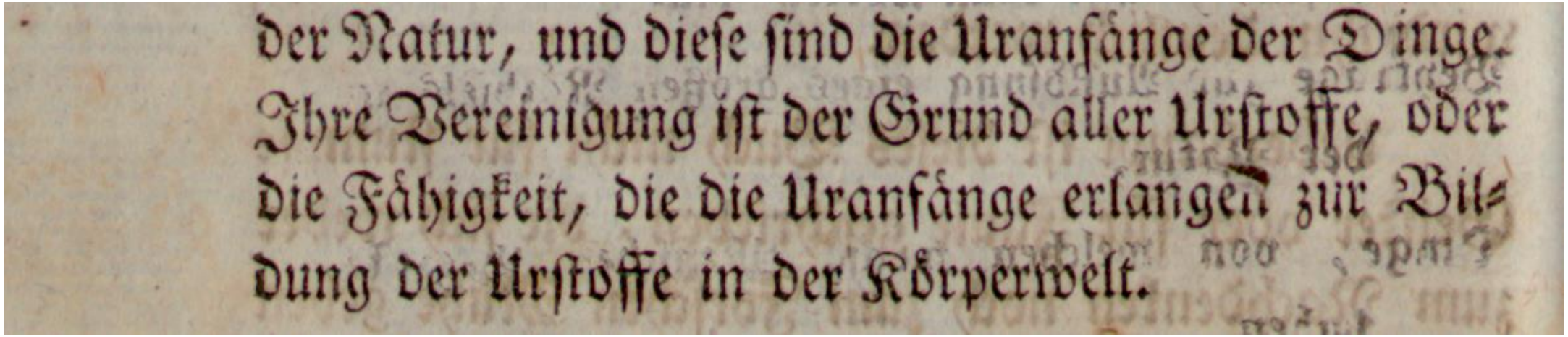
Threshold Type	Description
cv2.THRESH_BINARY	如果大於 127 就等於 255，反之等於 0
cv2.THRESH_BINARY_INV	如果大於 127 就等於 0，反之等於 255
cv2.THRESH_TRUNC	如果大於 127 就等於 127，反之數值不變
cv2.THRESH_TOZERO	如果大於 127 數值不變，反之數值等於 0
cv2.THRESH_TOZERO_INV	如果大於 127 等於 0，反之數值不變

CODE

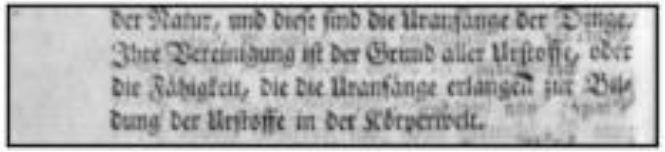
MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
import cv2
from matplotlib import pyplot as plt

image = cv2.imread('./test_P05.bmp')
grayimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

ret, th1 =
ret, th2 =
ret, th3 =
ret, th4 =
ret, th5 =

titles = ['Original', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
images = [grayimage, th1, th2, th3, th4, th5]

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

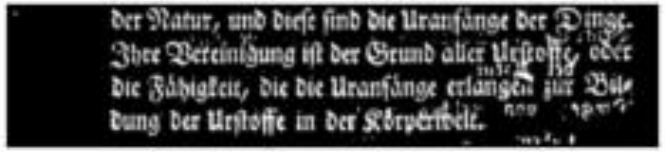


Original



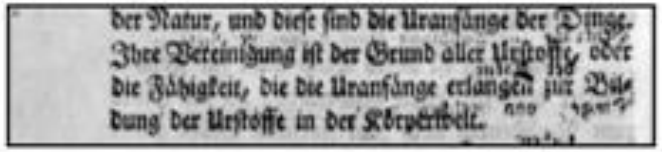
BINARY

BINARY_INV



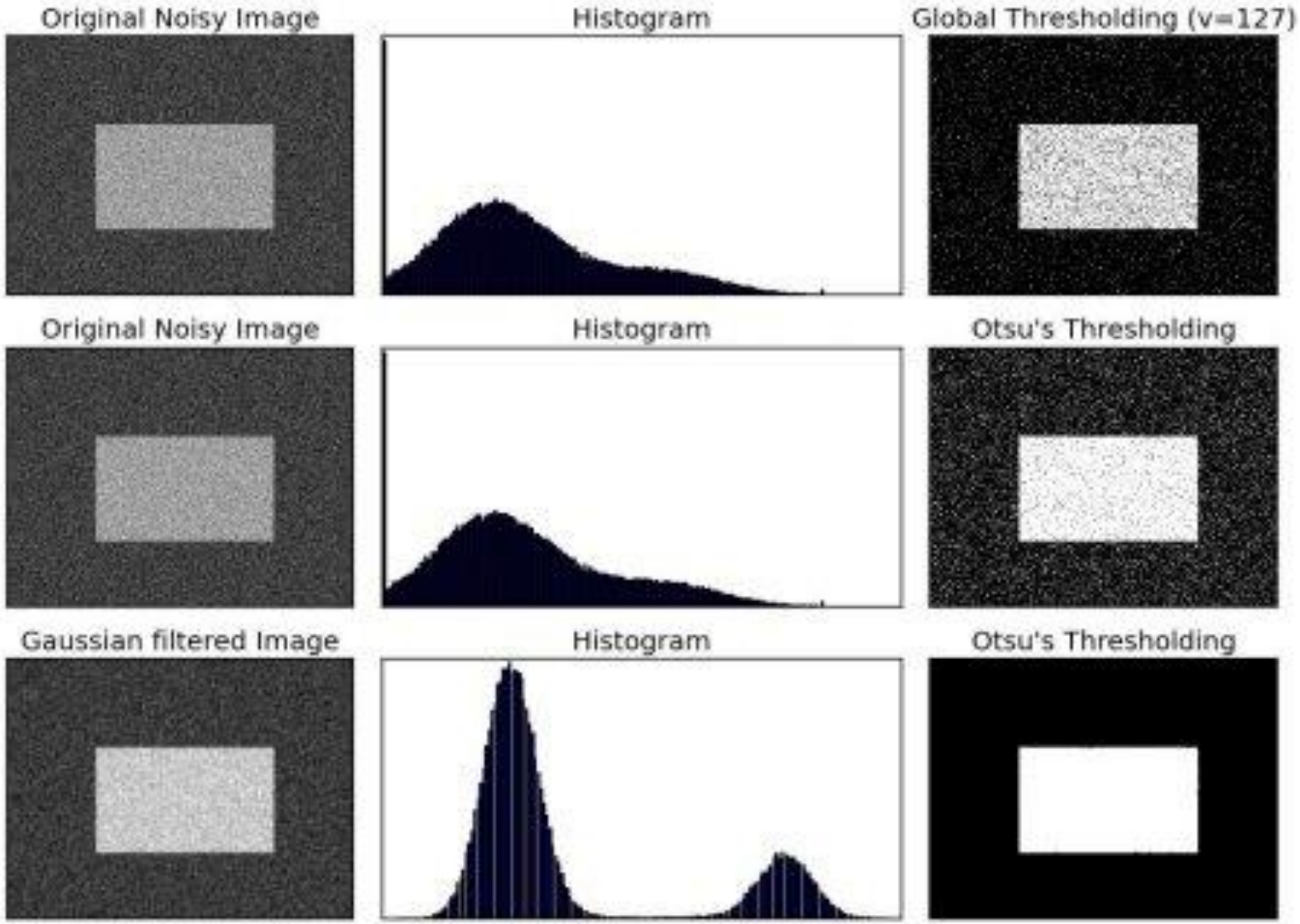
TRUNC

TOZERO



TOZERO_INV

OTSU 大津演算法


Original Noisy Image
Histogram
Global Thresholding (v=127)
Original Noisy Image
Histogram
Otsu's Thresholding
Gaussian filtered Image
Histogram
Otsu's Thresholding

自適應二值化

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C[, dst] ) -> dst
```

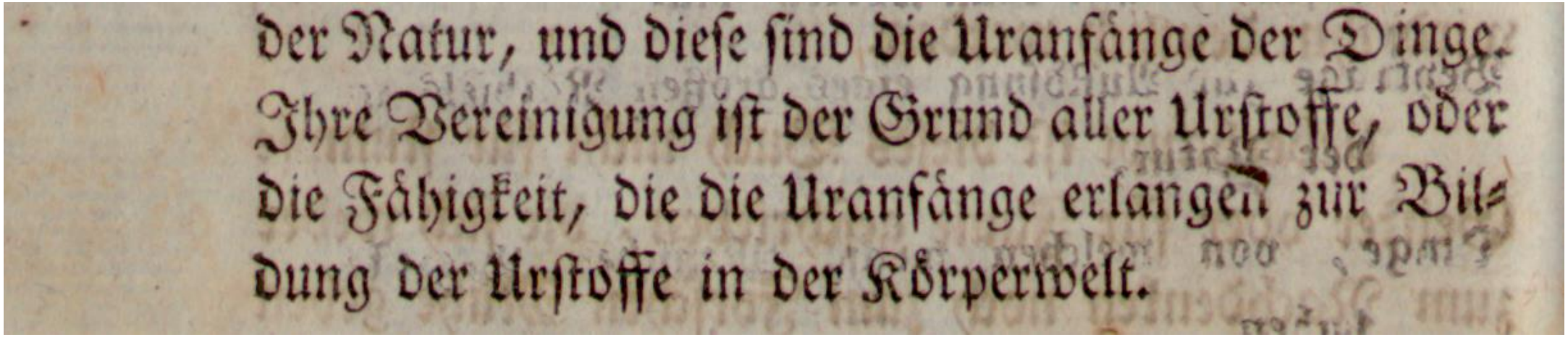
Parameters	Description
src	Source 8-bit single-channel image.
dst	Destination image of the same size and the same type as src.
maxValue	Non-zero value assigned to the pixels for which the condition is satisfied
adaptiveMethod	Adaptive thresholding algorithm to use. (cv2.ADAPTIVE_THRESH_MEAN_C or cv2.ADAPTIVE_THRESH_GAUSSIAN_C)
thresholdType	Thresholding type that must be either THRESH_BINARY or THRESH_BINARY_INV
blockSize	Size of a pixel neighborhood that is used to calculate a threshold value for the pixel: 3, 5, 7, and so on.
C	Constant subtracted from the mean or weighted mean. Normally, it is positive but may be zero or negative as well.

CODE

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

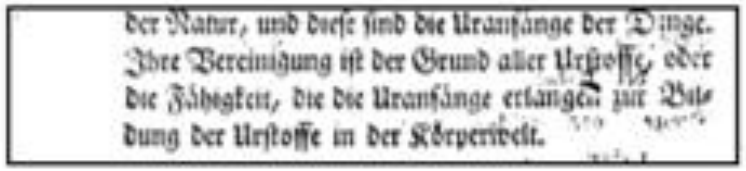
```
import cv2
from matplotlib import pyplot as plt

image = cv2.imread('./test_P05.bmp')
grayimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

ret, th1 =
th2 = cv2.
th3 = cv2.

titles = ['ORIGIN', 'OTSU THRESHOLD', 'ADAPTIVE MEAN THRESHOLD', 'ADAPTIVE GAUSSIAN THRESHOLD']
images = [grayimage, th1, th2, th3]

for i in range(4):
    plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

實作結果



ORIGIN

OTSU THRESHOLD



ADAPTIVE MEAN THRESHOLD

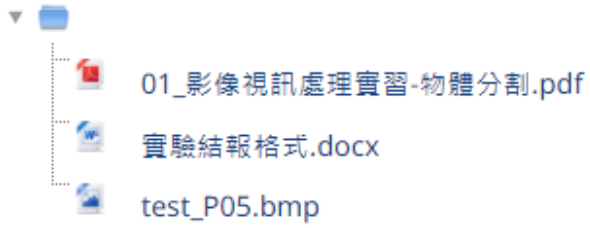
ADAPTIVE GAUSSIAN THRESHOLD

檔案命名規則

實驗一：影像中物體的分割 + （組別）

標記為完成

此處上傳實驗一需要的相關資料。



下載資料夾 編修

« PREVIOUS ACTIVITY NEXT ACTIVITY »

實驗一：影像中物體的分割 (組別一)

檔案內容

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

影像視訊處理實驗作業結報

班　　級	組長的	學　　號	組長的	姓　　名	組長的
實驗題目					

實驗內容　　*請勿抄襲，否則視為未交
一、　實驗簡介(至少 200 字)：
二、　實驗動機及其解決方法(至少 500 字)：
三、　程式碼(須附註解說明，截圖即可)：
四、　實驗結果(貼圖與簡述說明)：

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

Thanks for listening