计算机辅助手术讲座（4）
Image Guided Surgery (4)

卷积运算和图像滤波

Convolution and image filters

顾 力栩 (*Lixu Gu*)

上海交通大学 Med-X研究院

2009.11

Convolution

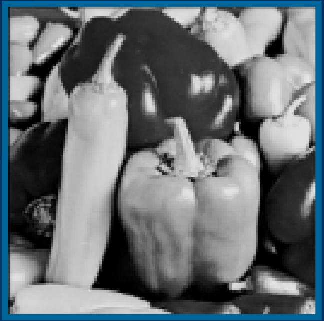
Convolution *vs.* Correlation

- Given an image *f(x,y)* and a kernel *w(a,b)*
  - Convolution: $$f * w = \sum_{\substack{(a,b)\in w \\ (x-a,y-b)\in f}} f(x-a, y-b)w(a,b)$$
  - Correlation: $$f \otimes w = \sum_{\substack{(a,b)\in w \\ (x+a,y+b)\in f}} f(x+a, y+b)w(a,b)$$
- Difference: Index is different.
  - Kernel asymmetric: convolution flips the kernel.

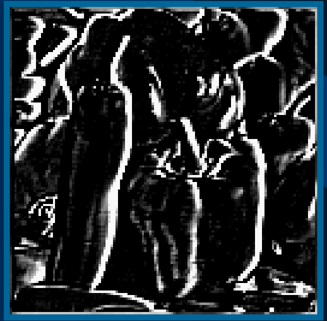
Convolution Application

- CONVOLUTION is used for image processing in all of these things...
  - Multimedia
  - Special effects
  - Photo enhancement
  - Computer image analysis

And more!



$$* \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow$$



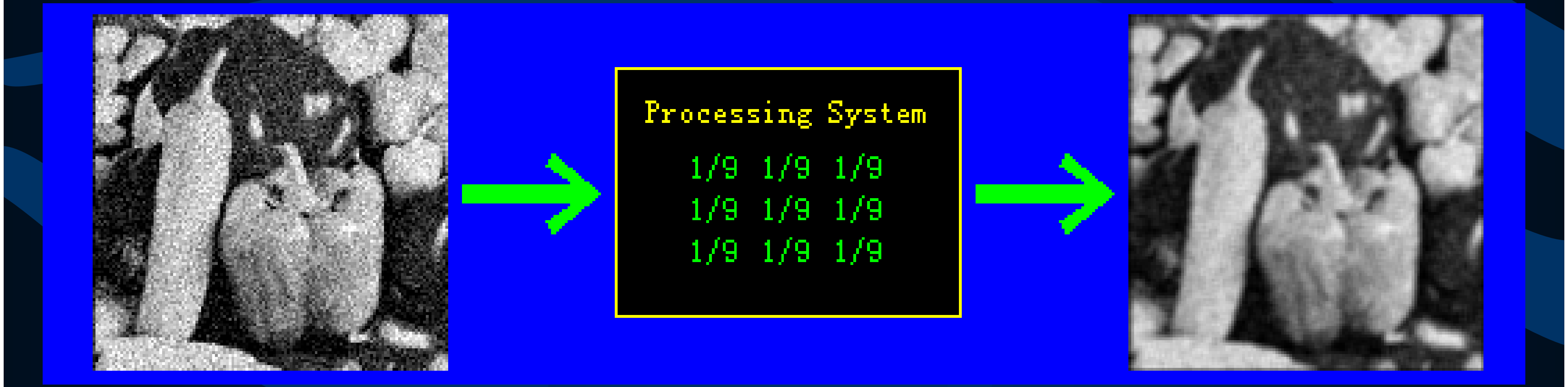
Sobel

Convolution Application

- What happened to the image?

  Noise removed Image blurred

  Edges sharpened Shades inverted



Processing System

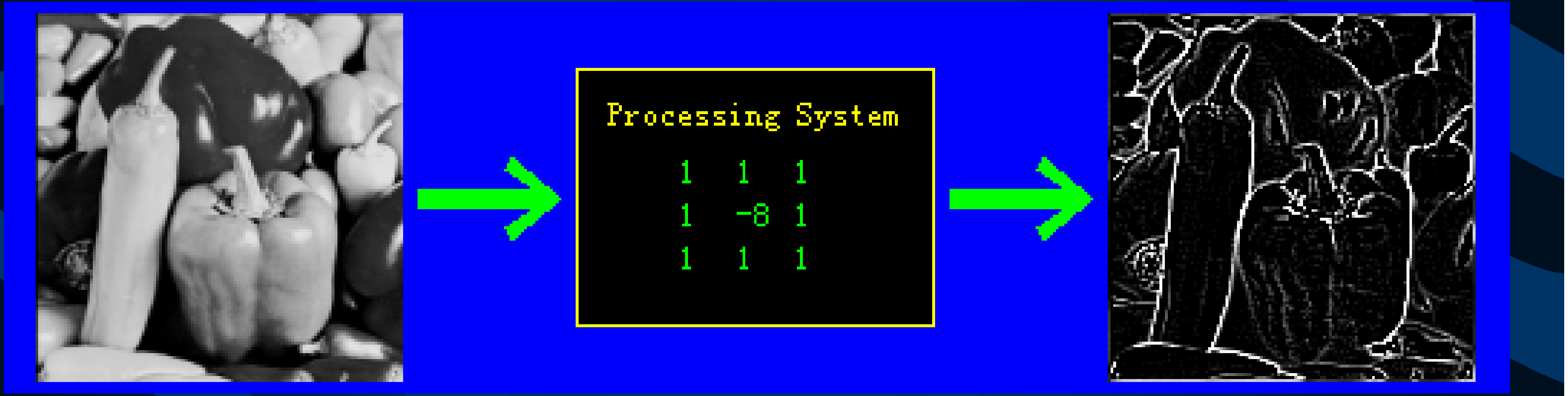
$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

Convolution Application

- What effect(s) did the system have on the image?

180 degree flip Image inverted

Edges enhanced Peppers colorized

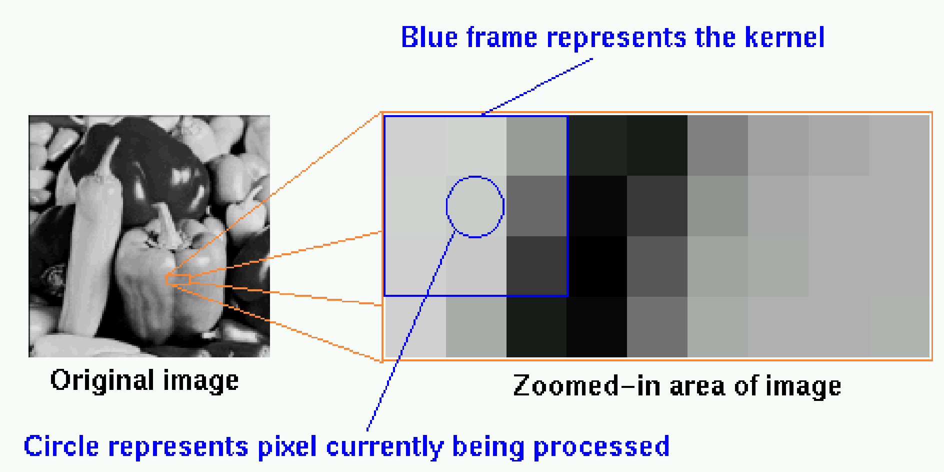


Processing System

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Convolution Kernel

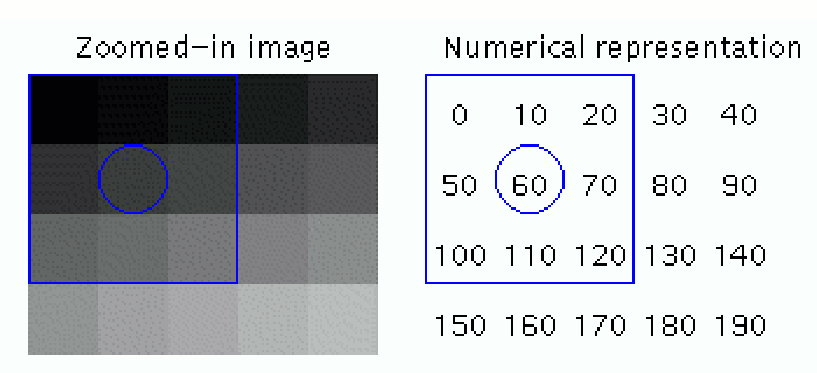
- **Kernel**: The nine "magic" numbers from the processing system are grouped together into a bundle.
  - First, place the center of the kernel at the first position to be processed. The kernel can operate on this part of the image using convolution.
  - Pixel can be seen when zoomed in.


Blue frame represents the kernel
Original image
Zoomed–in area of image
Circle represents pixel currently being processed

Convolution

- Convolution includes three steps:
  1. Position the center of the kernel at the first pixel of an image and flip the kernel.
  2. Multiply and sum.
  3. Slide to next pixel.

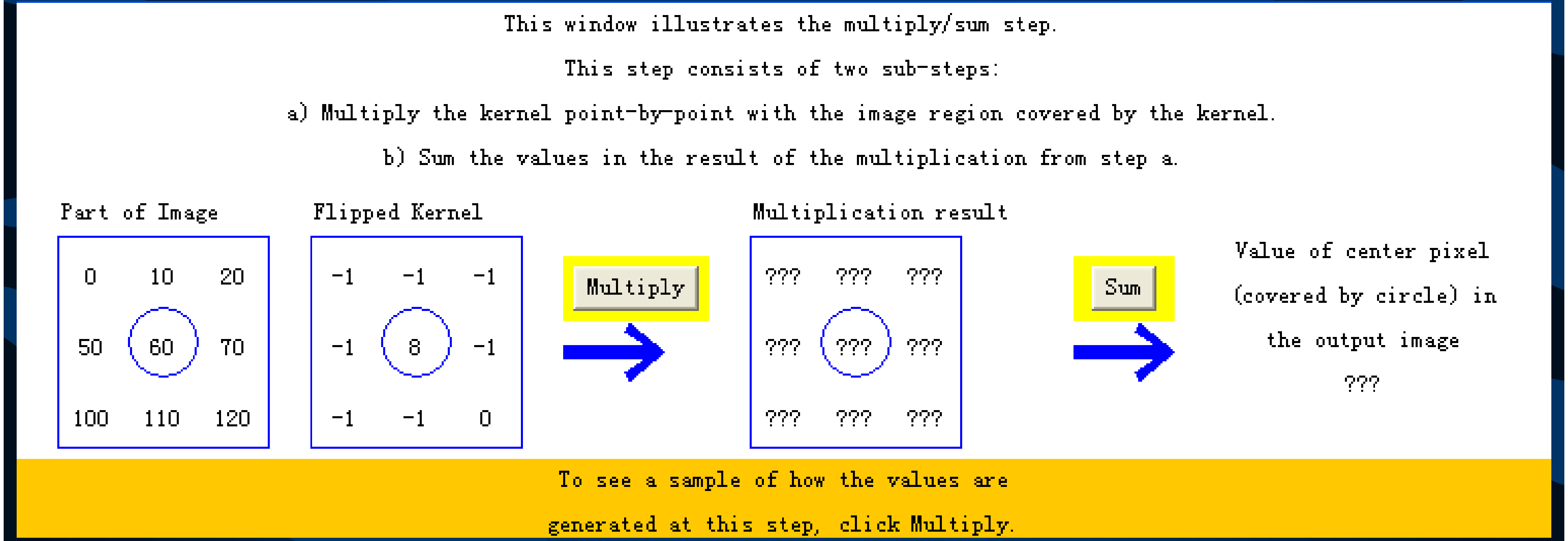
     Repeat steps 2 and 3 until the whole image is processed
- How kernel located in an image:



0	10	20	30	40
50	60	70	80	90
100	110	120	130	140
150	160	170	180	190

Convolution

1. Flip the kernel (turn 180 degree)
2. Multiply and sum



This window illustrates the multiply/sum step.

This step consists of two sub-steps:

a) Multiply the kernel point-by-point with the image region covered by the kernel.

b) Sum the values in the result of the multiplication from step a.

Part of Image

0	10	20
50	60	70
100	110	120

Flipped Kernel

-1	-1	-1
-1	8	-1
-1	-1	0

Multiplication result

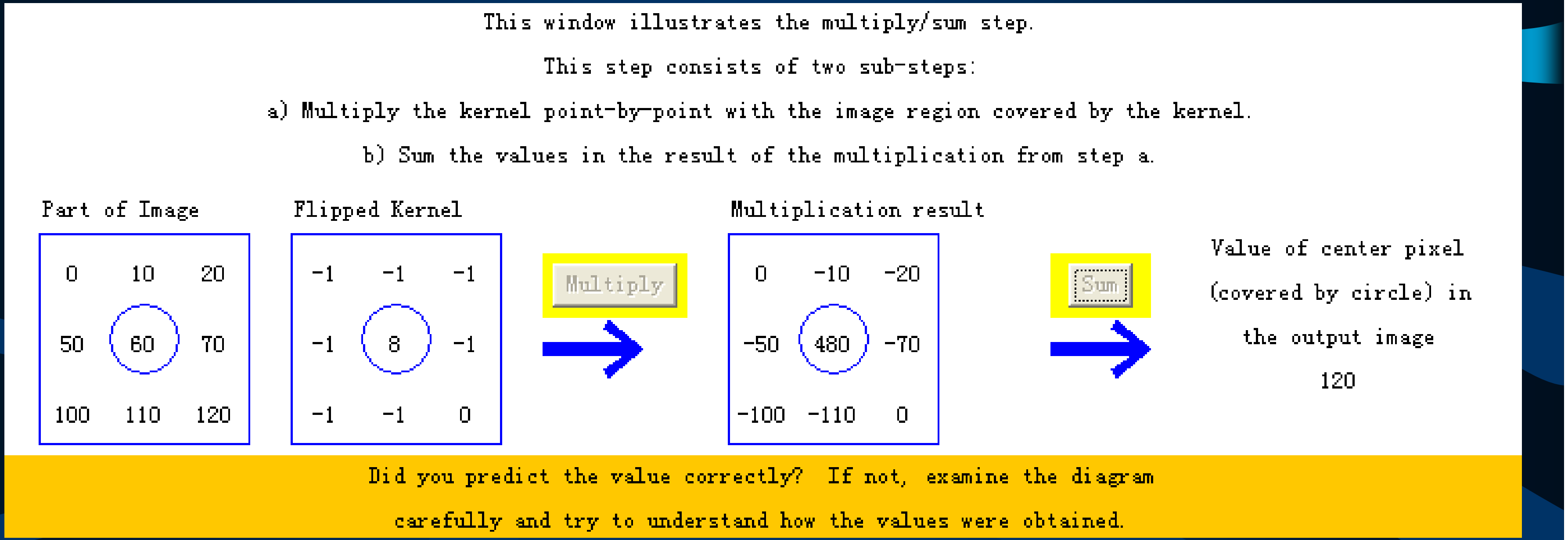
???	???	???
???	???	???
???	???	???

Value of center pixel (covered by circle) in the output image

???

To see a sample of how the values are generated at this step, click Multiply.

Convolution



This window illustrates the multiply/sum step.

This step consists of two sub-steps:

a) Multiply the kernel point-by-point with the image region covered by the kernel.

b) Sum the values in the result of the multiplication from step a.

Part of Image

0	10	20
50	60	70
100	110	120

Flipped Kernel

-1	-1	-1
-1	8	-1
-1	-1	0

Multiplication result

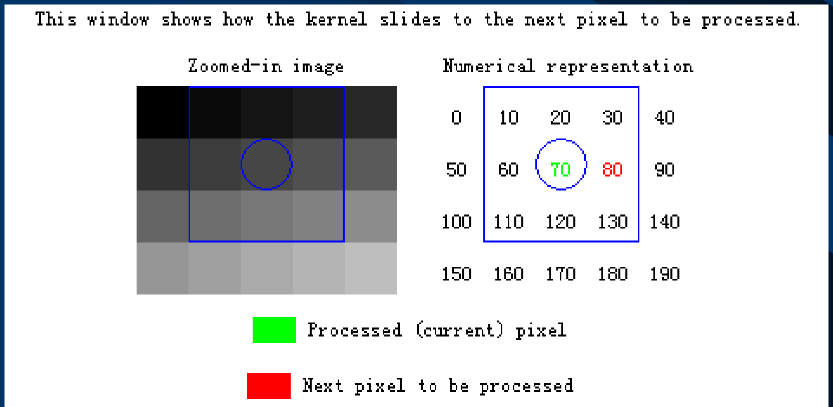
0	-10	-20
-50	480	-70
-100	-110	0

Value of center pixel (covered by circle) in the output image

120

Did you predict the value correctly? If not, examine the diagram carefully and try to understand how the values were obtained.

3. Slide to next pixel:



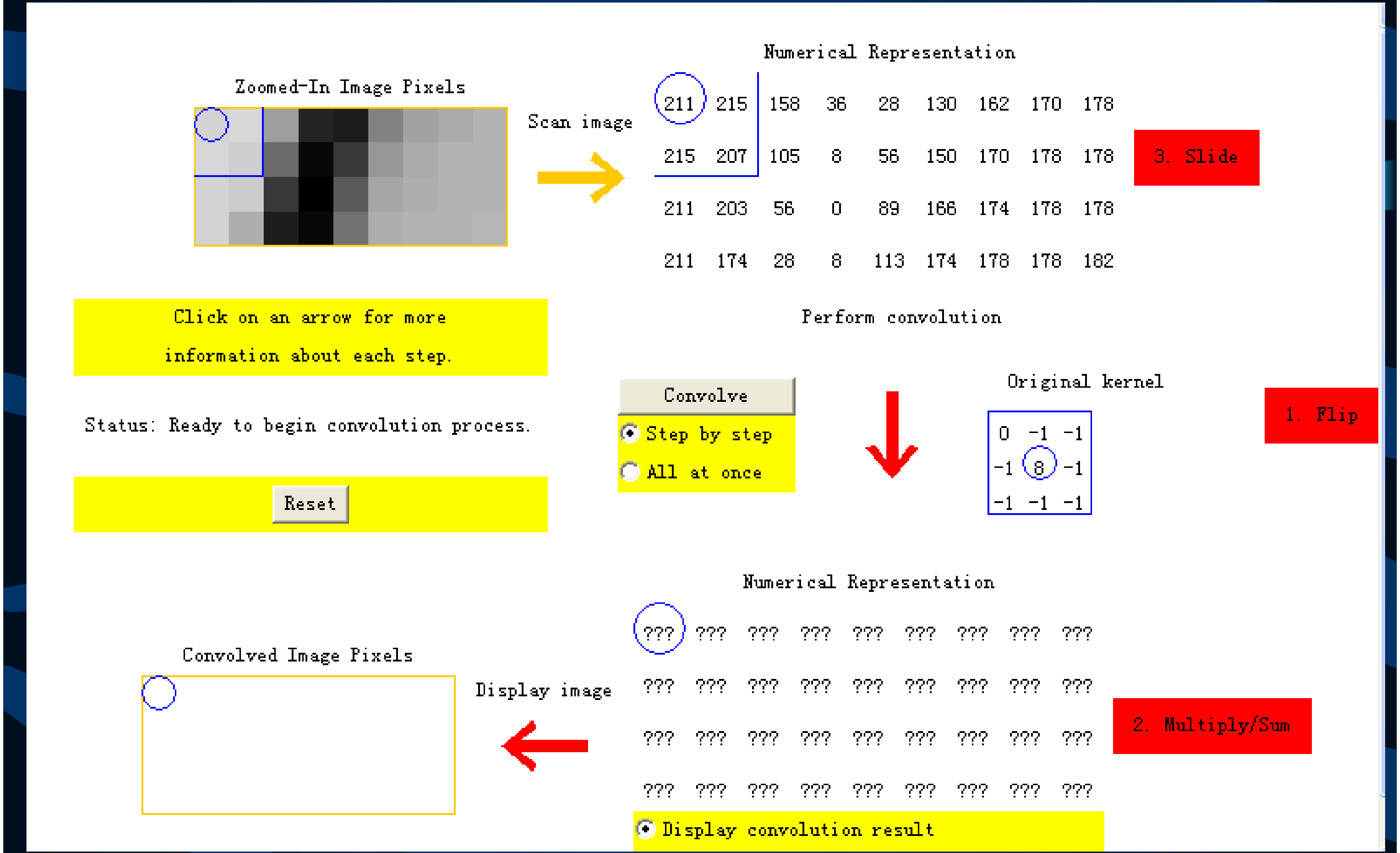
This window shows how the kernel slides to the next pixel to be processed.

Numerical representation

0	10	20	30	40
50	60	70	80	90
100	110	120	130	140
150	160	170	180	190

🟩 Processed (current) pixel

🟥 Next pixel to be processed



Numerical Representation

211	215	158	36	28	130	162	170	178
215	207	105	8	56	150	170	178	178
211	203	56	0	89	166	174	178	178
211	174	28	8	113	174	178	178	182

Click on an arrow for more information about each step.

Status: Ready to begin convolution process.

● Step by step

○ All at once

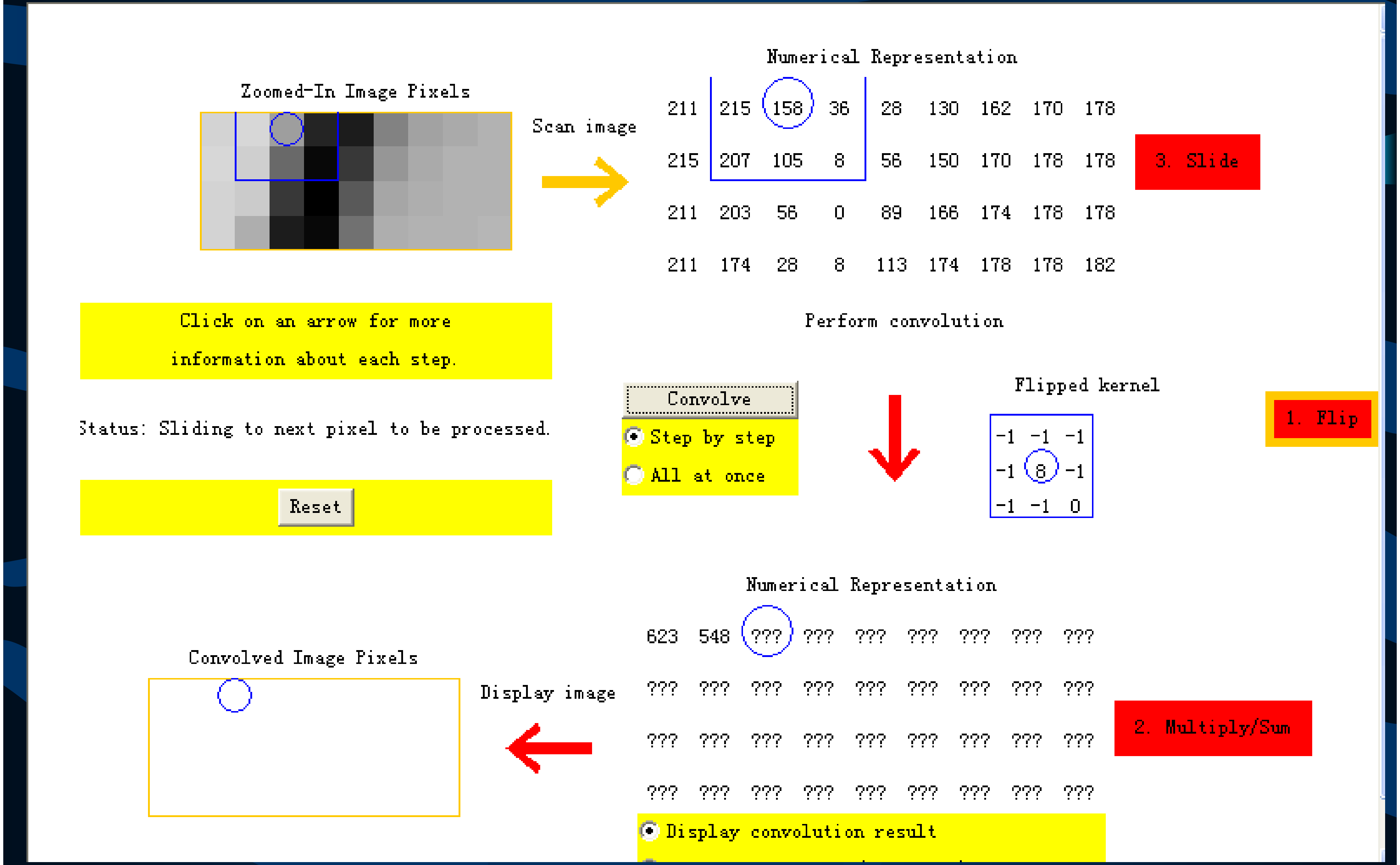
Original kernel

0	-1	-1
-1	8	-1
-1	-1	-1

Numerical Representation

???	???	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???

● Display convolution result



Zoomed-In Image Pixels

Scan image

Numerical Representation

211	215	158	36	28	130	162	170	178
215	207	105	8	56	150	170	178	178
211	203	56	0	89	166	174	178	178
211	174	28	8	113	174	178	178	182

3. Slide

Click on an arrow for more information about each step.

Status: Sliding to next pixel to be processed.

Reset

Perform convolution

Convolve

● Step by step

○ All at once

Flipped kernel

-1	-1	-1
-1	8	-1
-1	-1	0

1. Flip

Numerical Representation

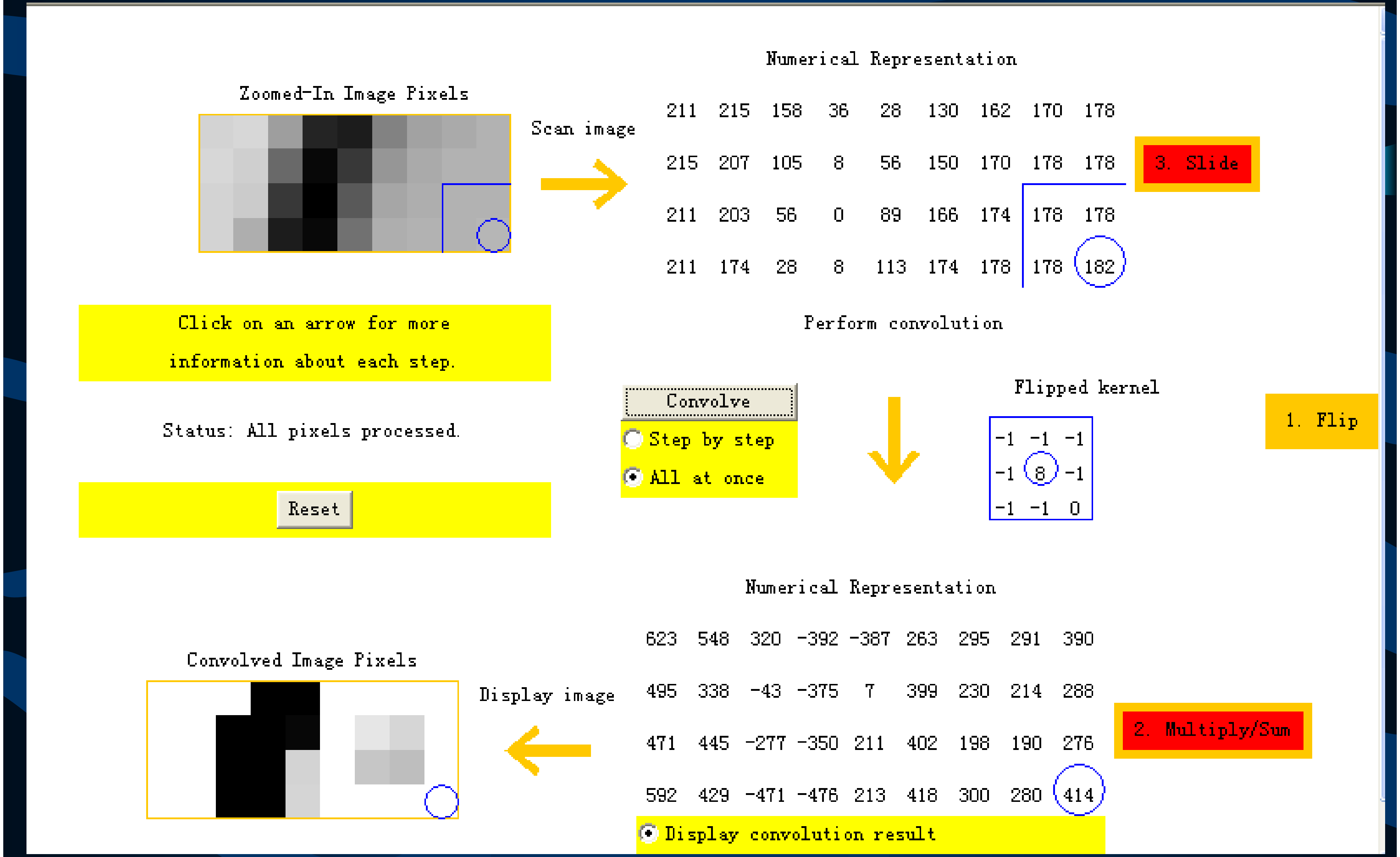
623	548	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???
???	???	???	???	???	???	???	???	???

2. Multiply/Sum

Convolved Image Pixels

Display image

● Display convolution result



Numerical Representation

211	215	158	36	28	130	162	170	178
215	207	105	8	56	150	170	178	178
211	203	56	0	89	166	174	178	178
211	174	28	8	113	174	178	178	182

Click on an arrow for more information about each step.

Status: All pixels processed.

○ Step by step

● All at once

Flipped kernel

-1	-1	-1
-1	8	-1
-1	-1	0

Numerical Representation

623	548	320	-392	-387	263	295	291	390
495	338	-43	-375	7	399	230	214	288
471	445	-277	-350	211	402	198	190	276
592	429	-471	-476	213	418	300	280	414

● Display convolution result

Edge Detection

What Is an Edge?

- An edge is a set of connected pixels that lie on the boundary between two regions
- The pixels on an edge are called edge points
- Position & orientation of edge
- Gray level discontinuity across an edge

How it comes?

- Different colors, brightness, textures, material, tissues, …
- Different normal directions of surfaces
- Different illuminance

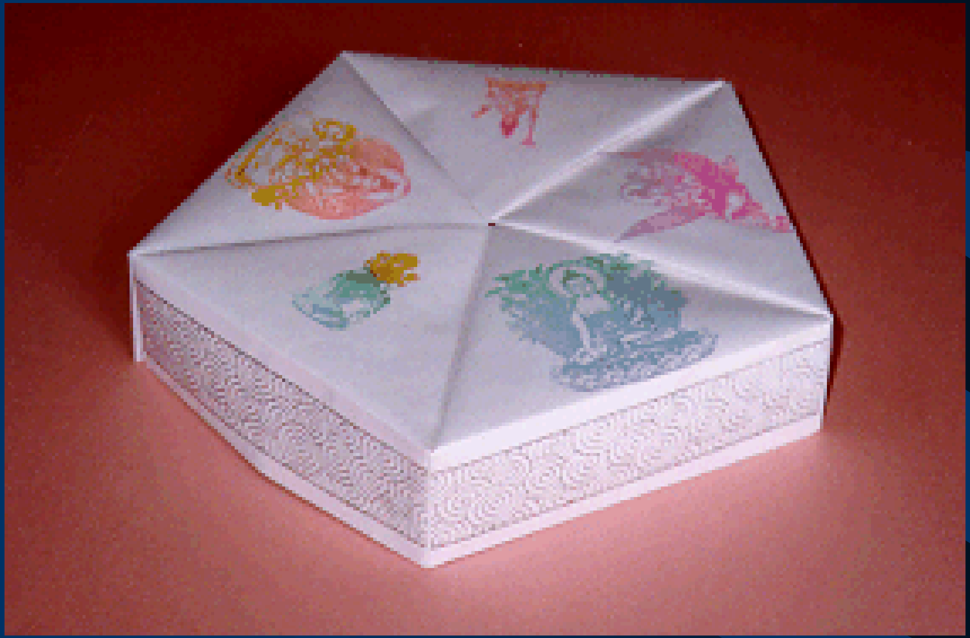
Different Edges

Different color

Different brightness

Different Edges

Different texture



Different surfaces

Unique

- Most of edges are unique in space, i.e., its position and orientation keep the same in the space when viewing from different points
- Non-unique edge-Limb edge

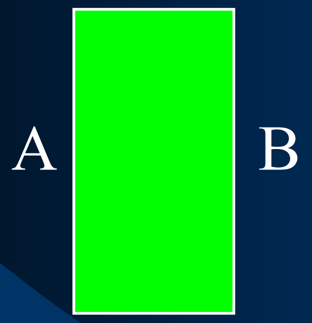


Image 1

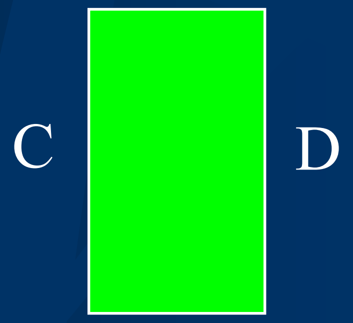
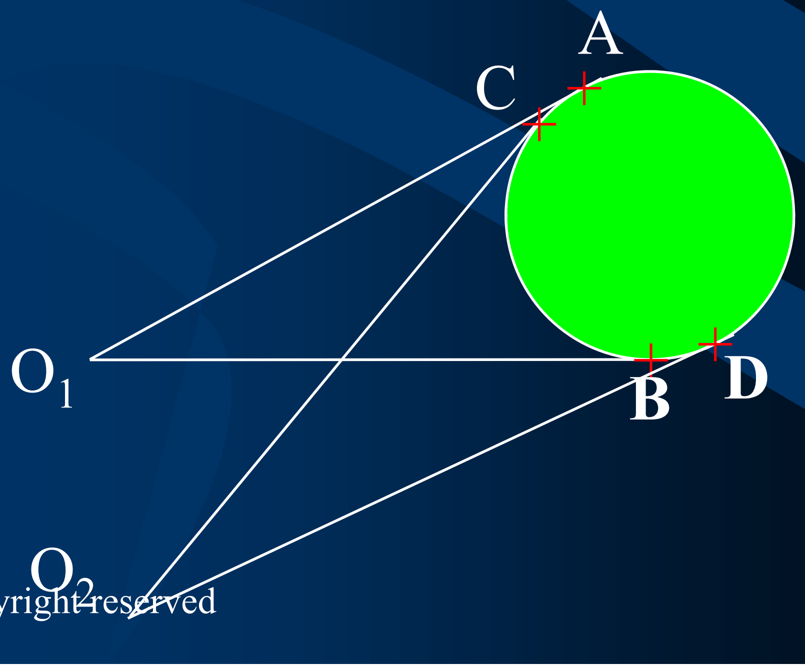


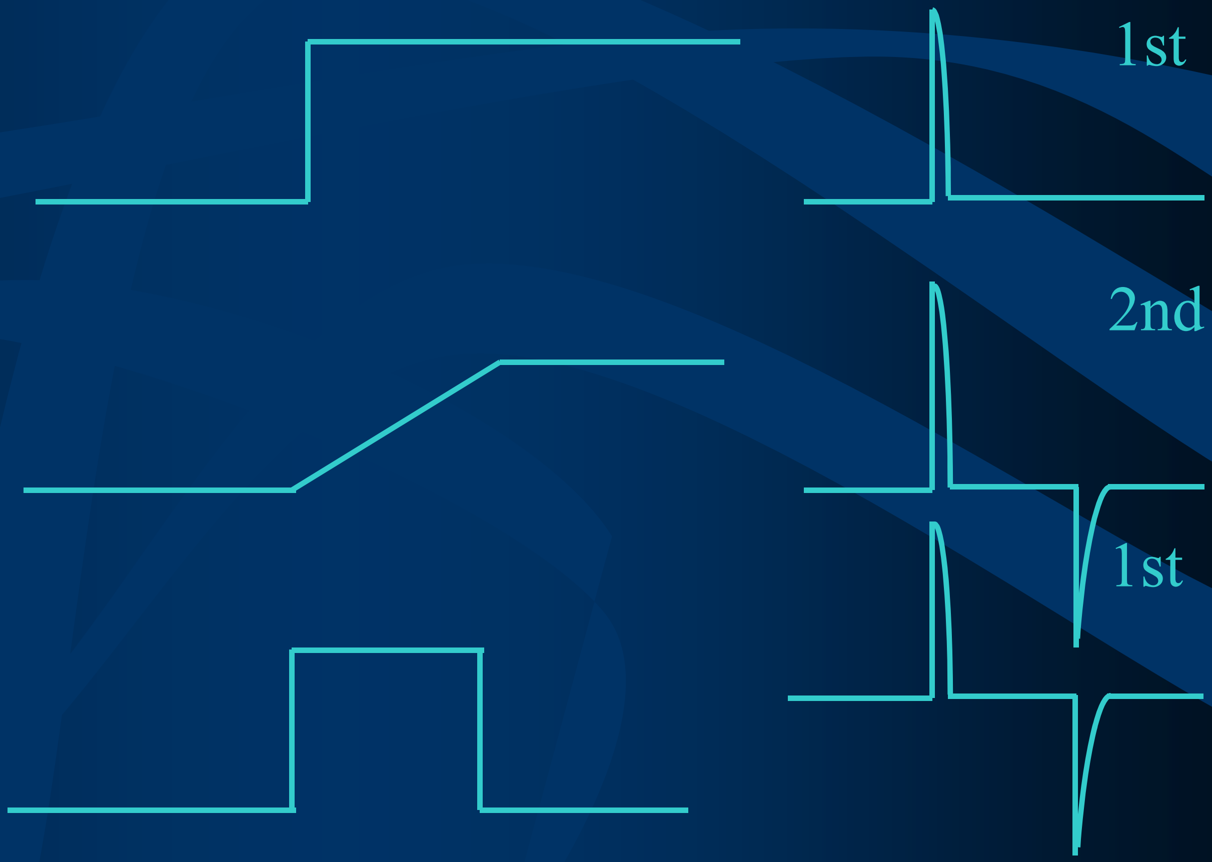
Image 2



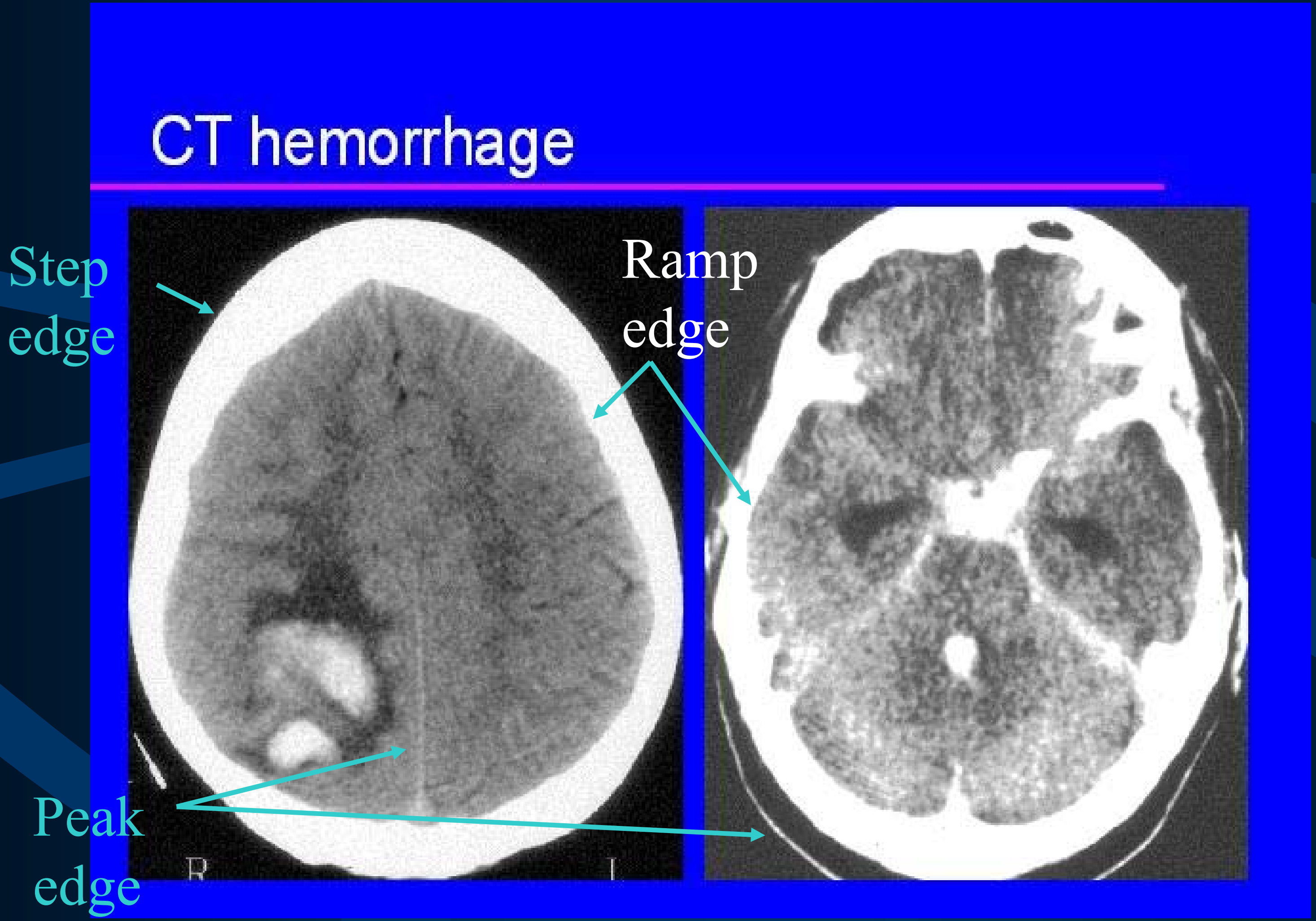
Edge Catagory

Gray level profile derivatives

- Step edge:
- Ramp edge:
- Peak edge:



Edge *vs.* CT Image



CT hemorrhage

Gradient

- A point is defined as an edge point if its 2-D first or second -order derivative is greater than a specified threshold.
- Gradient of digital image, $f(x, y)$, is defined by ***a vector:***

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient

- Gradient at an edge point, (x,y), can also be interpreted as a complex number with its magnitude determined by

$$\nabla f(x,y) = mag(\nabla \mathbf{f}(x,y)) = \left[G_x^2 + G_y^2 \right]^{1/2}$$

and the direction determined by

$$\alpha(x,y) = \tan^{-1}(\frac{G_y}{G_x})$$

Edge Operations

- The partial derivatives in x and y, G_x, G_y, can be estimated using different ways:

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

➢ Roberts operator:

$$G_x = (z_9 - z_5), \quad G_y = (z_8 - z_6)$$

➢ Prewitt operator:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

➢ Sobel operator:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Edge Operations

- All operators can be performed by the convolution using different masks.
- Roberts operator masks:

-1	0
0	1

G_x

0	-1
1	0

G_y

Edge Operations

- Prewitt operator masks:

-1	-1	-1
0	0	0
1	1	1

G_x

-1	0	1
-1	0	1
-1	0	1

G_y

- Sobel operator masks:

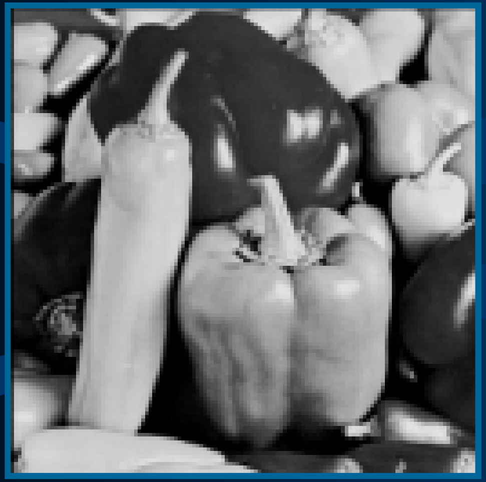
-1	-2	-1
0	0	0
1	2	1

G_x

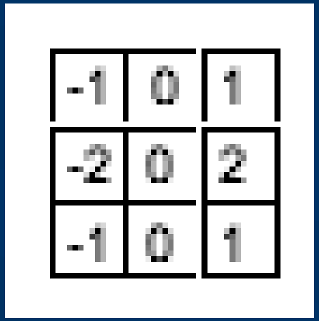
-1	0	1
-2	0	2
-1	0	1

G_y

Application

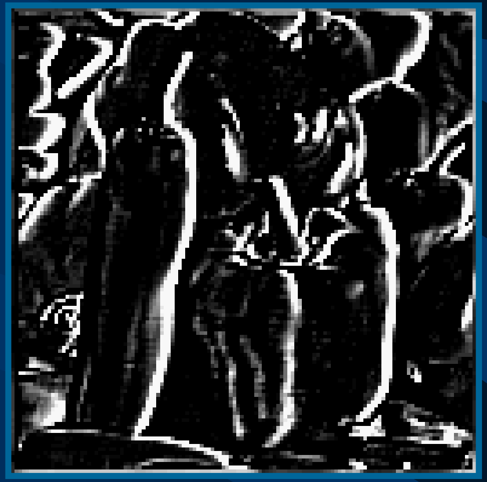


*



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

Sobel



Noise Reduction

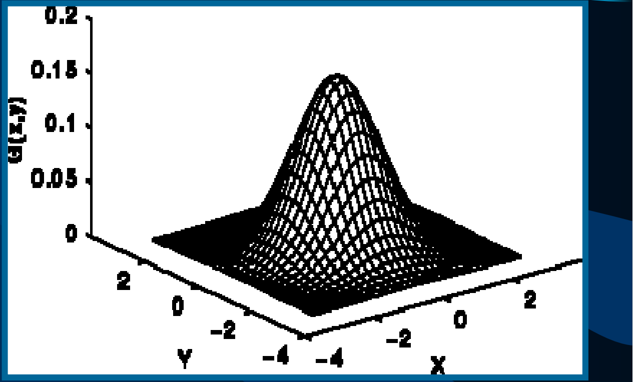
Noise Reduction

- Noise reduction is also called noise cleaning or smoothing.
- Coherence vs. incoherence:
  - Replace incoherent pixel values by values more specially coherent which are based on some or all the pixels in an appropriate neighbourhood.
- Two categories:
  - Convolution based: Gaussian smoothing, median filter
  - Morphology based: Opening, Closing (TBD)

Gaussian Reduction

- Gaussian kernel:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



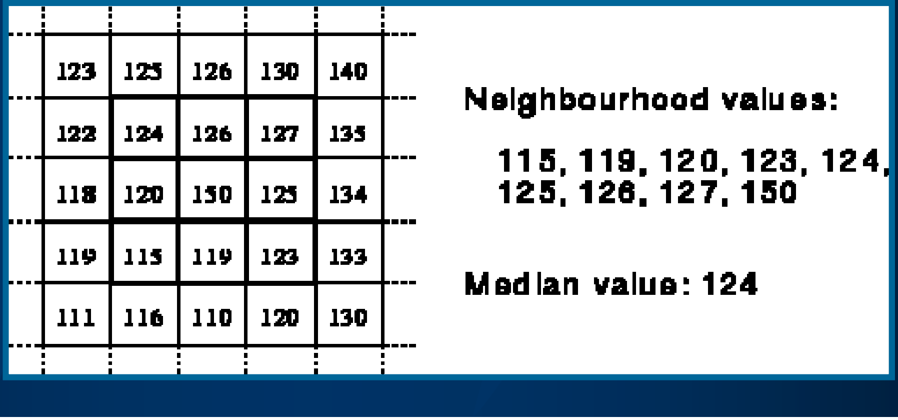
- The effect of Gaussian smoothing is to blur an image. The degree of smoothing is determined by the standard deviation σ of the Gaussian
- The Gaussian outputs a `weighted average' of each pixel's neighbourhood, with the average weighted more towards the value of the central pixels

Median Filter

- Mean Filter: The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbours, including itself.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- Median Filter: Instead of simply replacing the pixel value with the *mean* of neighbouring pixel values, replaces it with the *median* of those sorted values.



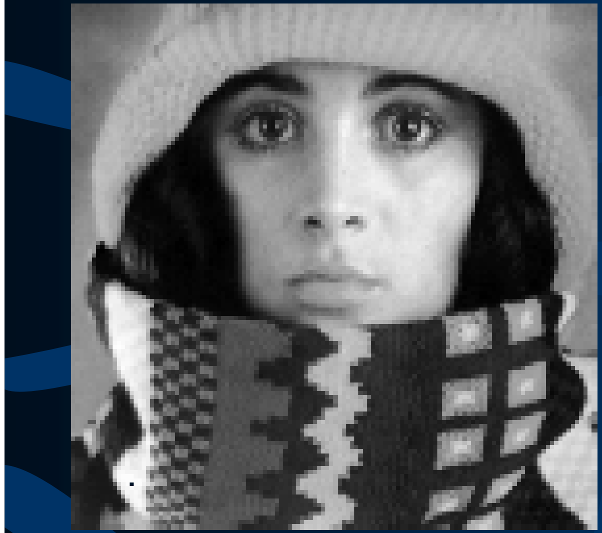
123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

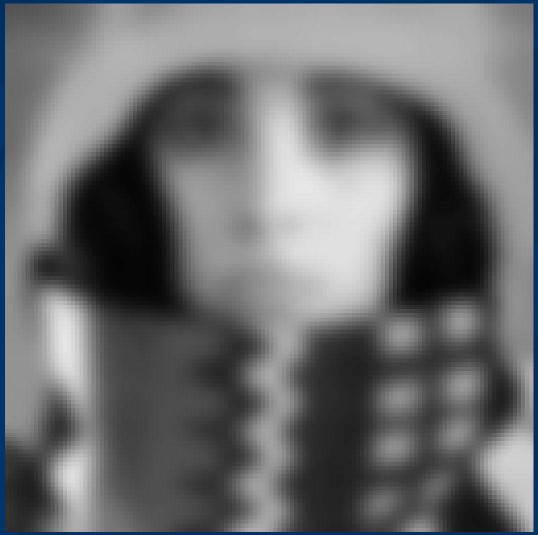
115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

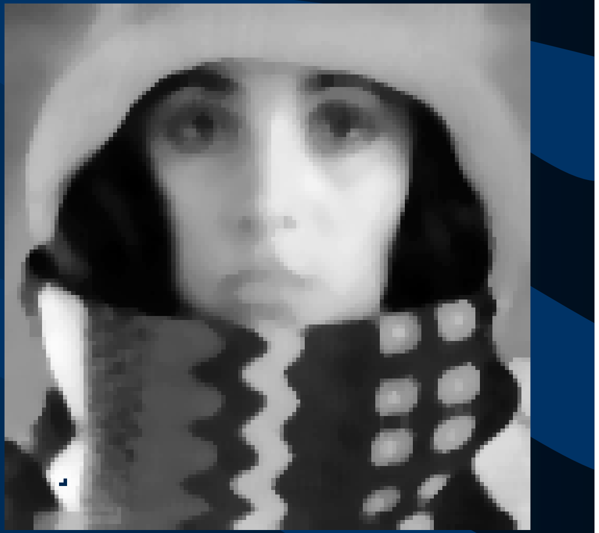
Gaussian *vs.* Median Filter



Source



Gaussian (σ = 2.5)



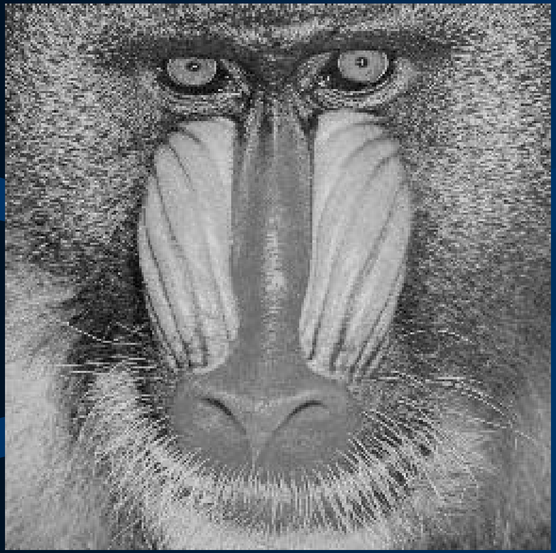
Median (5 X 5)

Projects

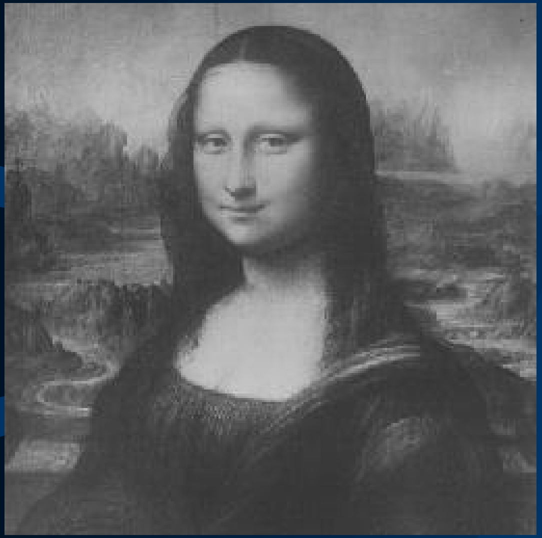
Project - 2

- Convolution and Image Filters
  - Requirement：
    - Program to realize the convolution operation and one of the next filters
      - ✓ Roberts operator; Prewitt operator; Sobel operator;
      - ✓ Gaussian filter and Median filter
    - Design proper UI and result display
    - The edge detection and noise reduction

Classic Image Samples



Classic Image Samples



Discussion