Medical Image Analysis

Angshuman Paul

Assistant Professor

Department of Computer Science & Engineering

Medical Image Enhancement

Image Quality Parameters

- Resolution
- Contrast
- Noise and artifacts

Resolution

- Level of detail contained in an image
- Spatial resolution
  - Number of pixels contained in a given area of image
    - Pixels per inch (PPI)
- Better resolution: more detailed information

Resolution

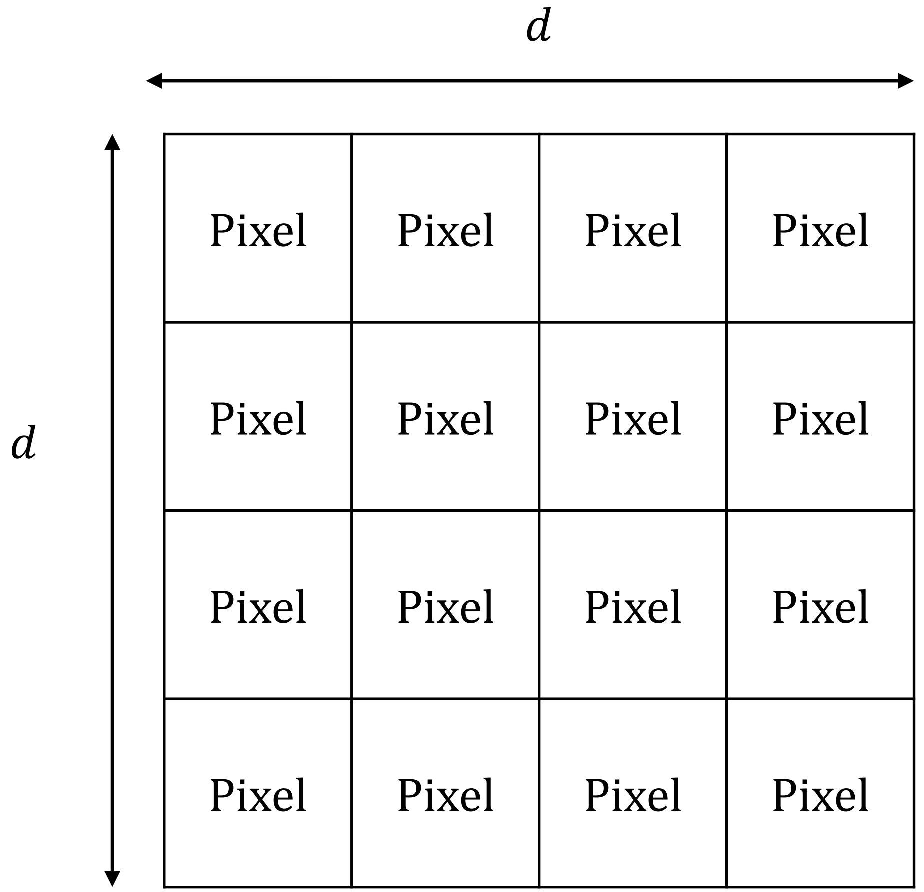
HR

Resolution

LR

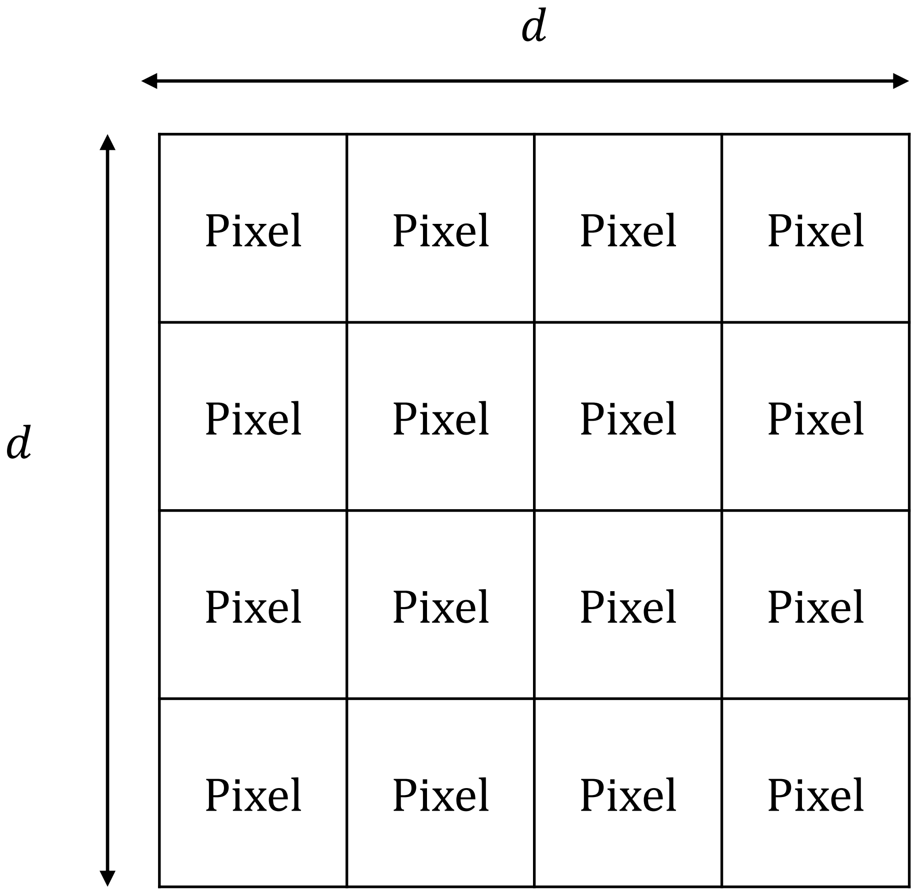
https://www.adobe.com/uk/creativecloud/photography/discover/image-resolution.html#:~:text=Resolution%3A%20FAQs.-,Image%20resolution%20definition.,and%20definition%20you%20will%20see.

Resolution



Suppose the display size is $d \times d$

Resolution



Each pixel has got some intensity

Resolution

d

d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

Each pixel has got some intensity

Resolution

d

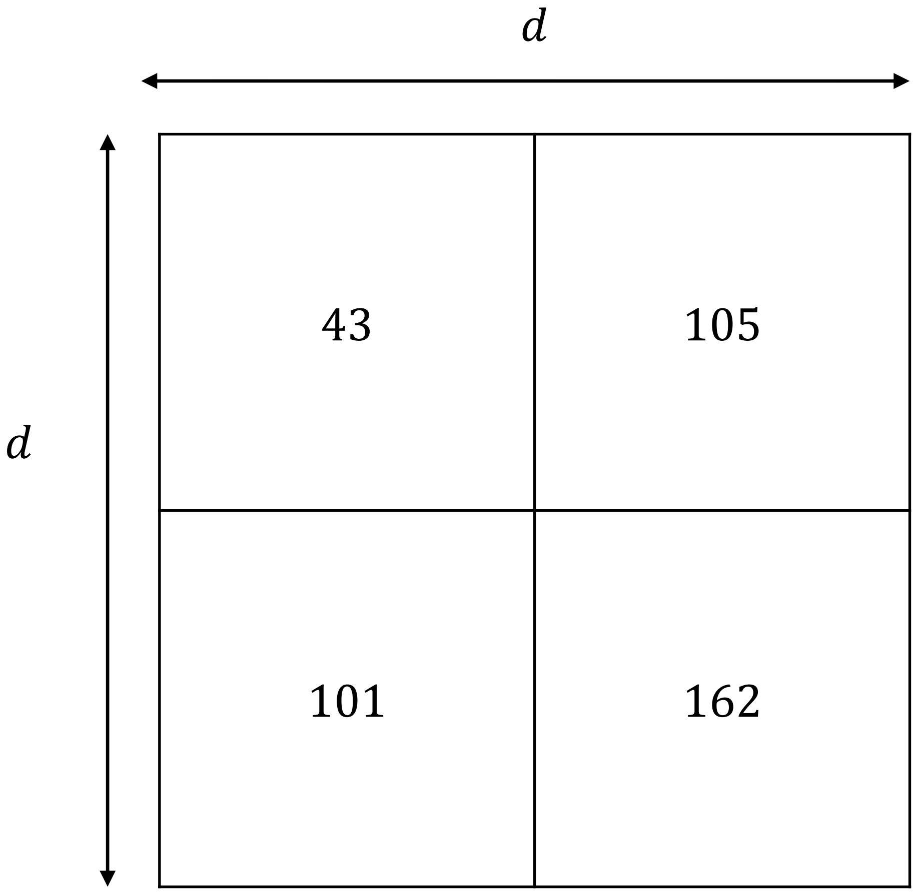
d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

Each pixel has got some intensity

You can see certain amount of details/ information (through the changes in intensity)

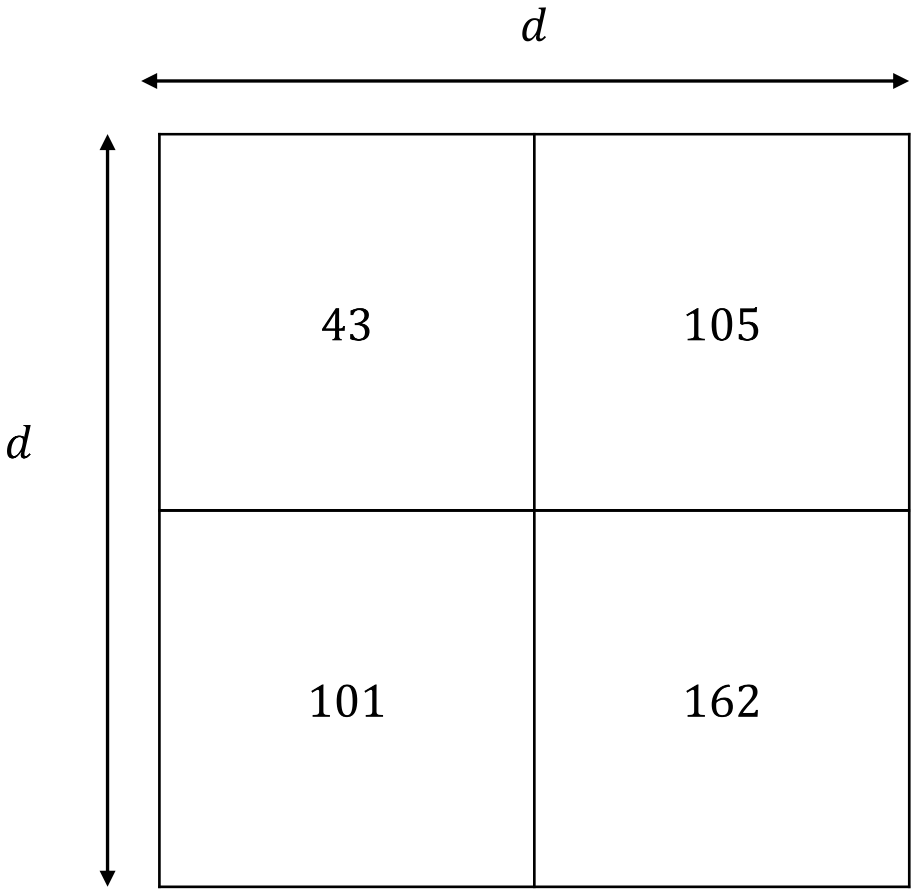
Resolution



Each pixel has got some intensity

You can see certain amount of details/ information (through the changes in intensity)

Resolution



Each pixel has got some intensity

You can see certain amount of details/ information (through the changes in intensity)

The variation of intensity is less here

Resolution

d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

More details

Same display size

d

43	105
101	162

Less details

Resolution

d

d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

More details – High resolution

Same display size

d

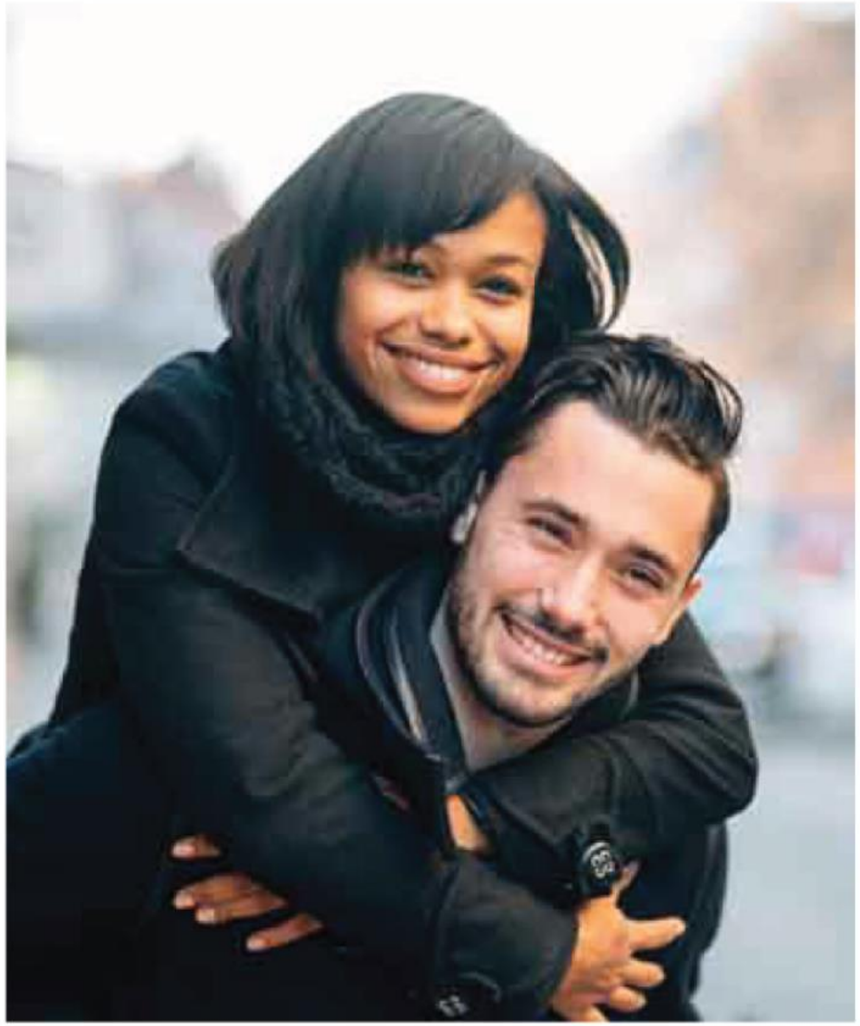
d

43	105
101	162

Less details – Low resolution

Resolution

Less details – Low resolution



More details – High resolution

https://www.susannewberrydesigns.com/susan-newberry-designs/high-vs-low-resolution-does-it-matter

Resolution

Less details – Low resolution

More details – High resolution

https://www.susannewberrydesigns.com/susan-newberry-designs/high-vs-low-resolution-does-it-matter

Resolution

d

d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

More details – High resolution

Same display size

d

d

43	105
101	162

Less details – Low resolution

Different Display Size

d

d

10	14	67	35
11	97	82	142
31	170	201	90
103	92	112	43

More details – High resolution

d'

d'

43	105
101	162

Less details – Low resolution

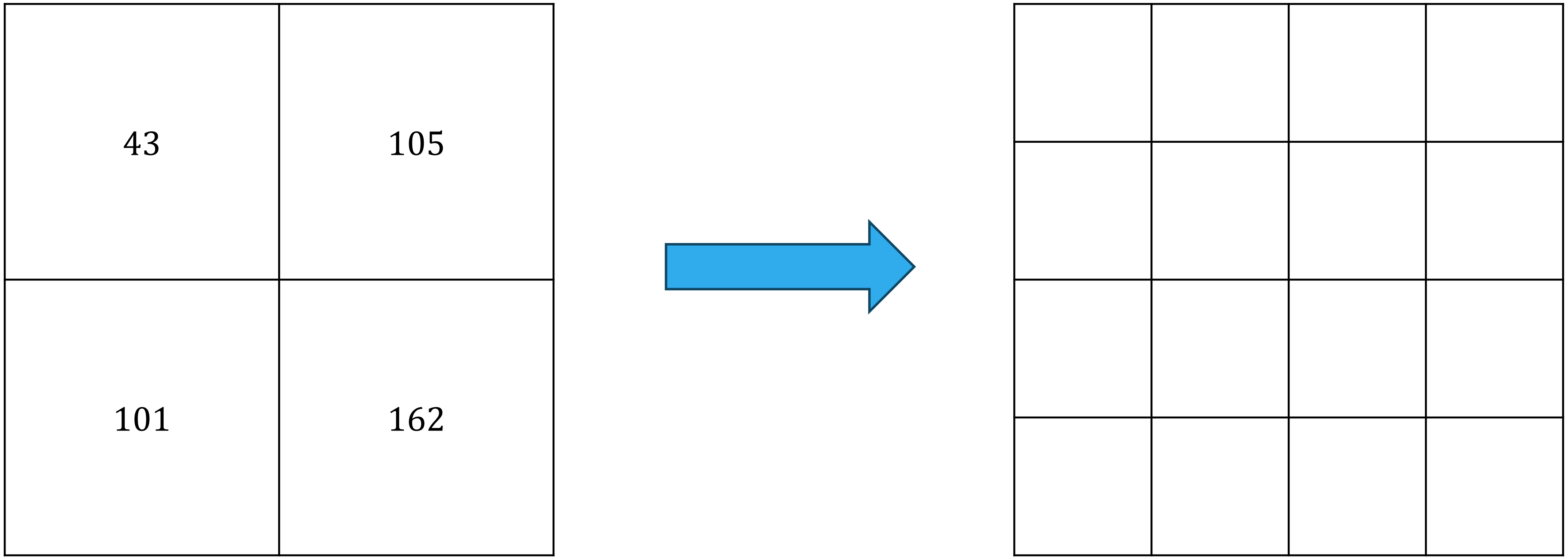
Different Display Size

d

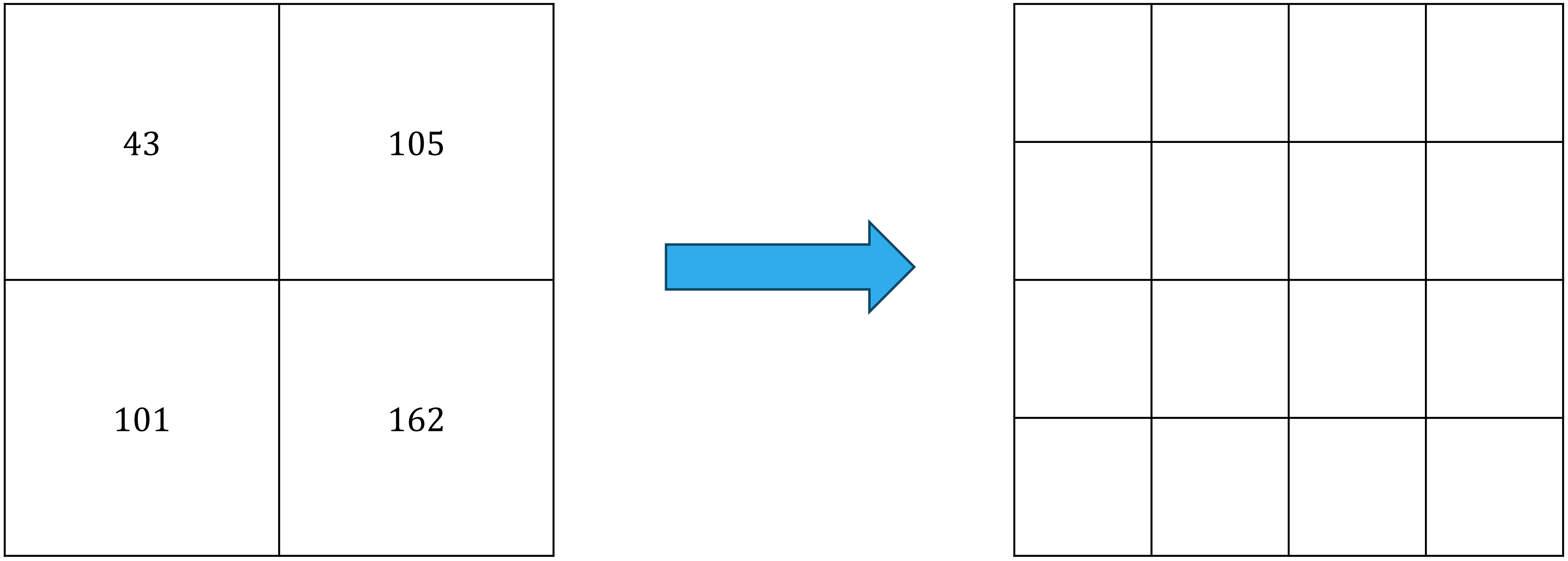
More details – High resolution

Less details – Low resolution

Super-resolution

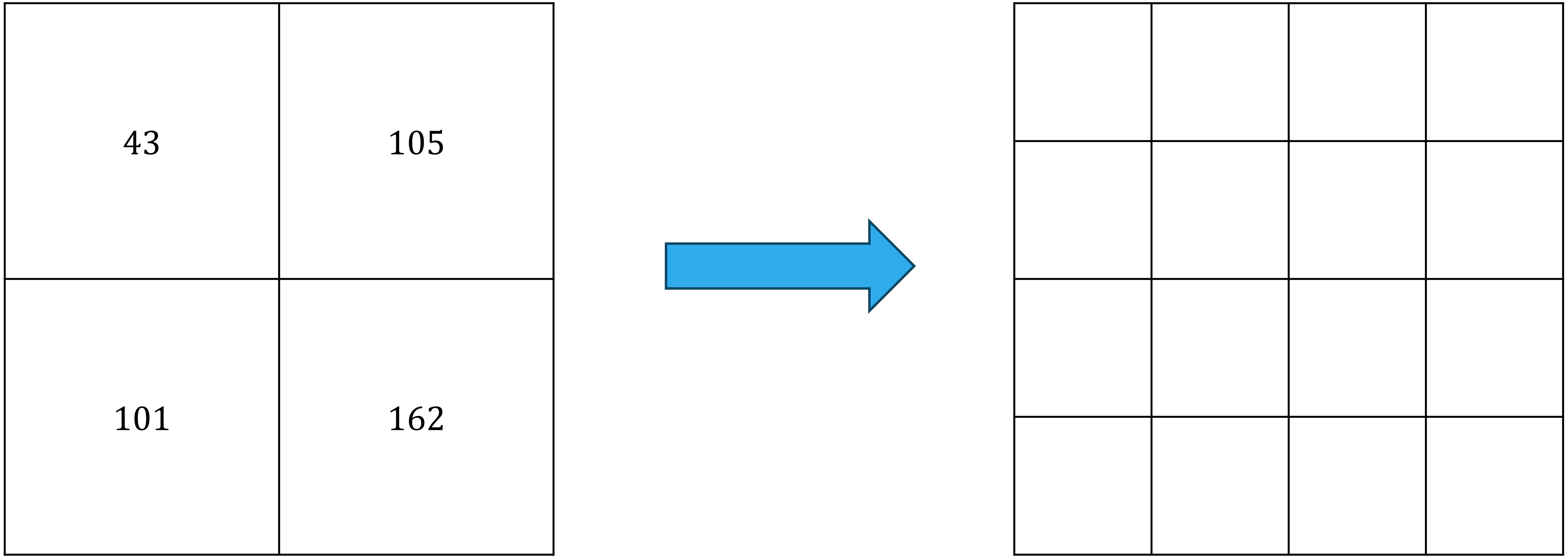


Super-resolution



How to fill in these pixel values?

Super-resolution



How to fill in these pixel values?

Ill-posed problem

Super-resolution using Interpolation

Nearest Neighbour Interpolation

43	105
101	162

43	43	105	105
43	43	105	105
101	101	162	162
101	101	162	162

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0				
1				
2				
3				

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43			105
1				
2				
3	101			162

Bilinear Interpolation

43	105
101	162

	0	**1**	**2**	**3**
0	43			105
1		##		
2				
3	101			162

Suppose, I want to calculate the pixel value for the position ##

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43			105
1		##		
2				
3	101			162

Suppose, I want to calculate the pixel value for the position ##

I will have to do two linear interpolations

Bilinear Interpolation

43	105
101	162

	0	**1**	**2**	**3**
0	43	*		105
1				
2				
3	101			162

First interpolation

$$43 \times \frac{3-1}{3-0} + 105 \times \frac{1-0}{3-0} \approx 64$$

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43	64		105
1				
2				
3	101	*		162

First interpolation

$$101 \times \frac{3-1}{3-0} + 162 \times \frac{1-0}{3-0} \approx 121$$

Bilinear Interpolation

43	105
101	162

	0	**1**	**2**	**3**
0	43	64		105
1		##		
2				
3	101	121		162

$$101 \times \frac{3-1}{3-0} + 162 \times \frac{1-0}{3-0} \approx 121$$

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43	64		105
1		##		
2				
3	101	121		162

Second interpolation

$$64 \times \frac{3-1}{3-0} + 121 \times \frac{1-0}{3-0} \approx 83$$

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43	64		105
1		83		
2				
3	101	121		162

Second interpolation

$$64 \times \frac{3-1}{3-0} + 121 \times \frac{1-0}{3-0} \approx 83$$

Bilinear Interpolation

43	105
101	162

	0	1	2	3
0	43	64		105
1		83		
2				
3	101	121		162

Creates smoother images compared to nearest neighbour interpolation

Bicubic Interpolation

SISR Using Deep Learning

SISR Using Deep Learning

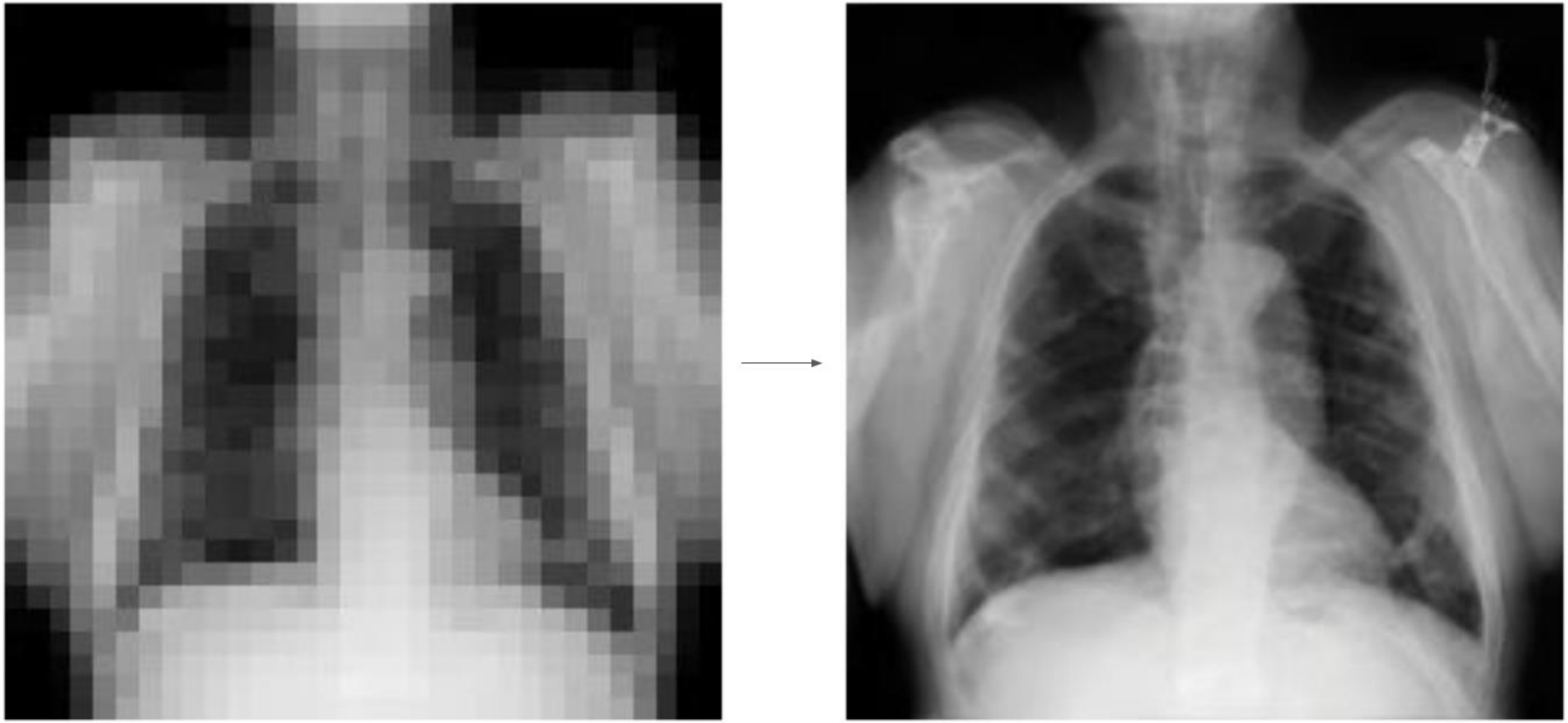


Image Quality Parameters

- Resolution
- Contrast
- Noise and artifacts

Contrast

- Differences between the intensity of an object and surrounding/ background

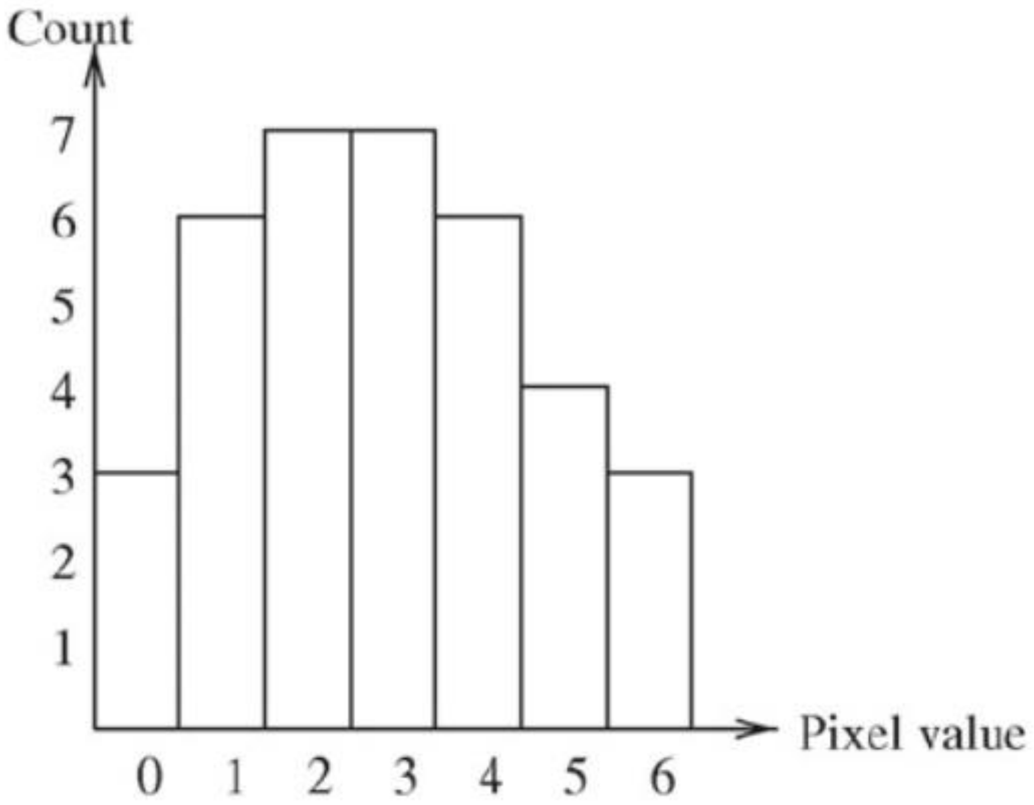
- Due to
  - Poor illumination
  - Lack of dynamic range in imaging sensor

- Better details can be revealed as a result of contrast enhancement

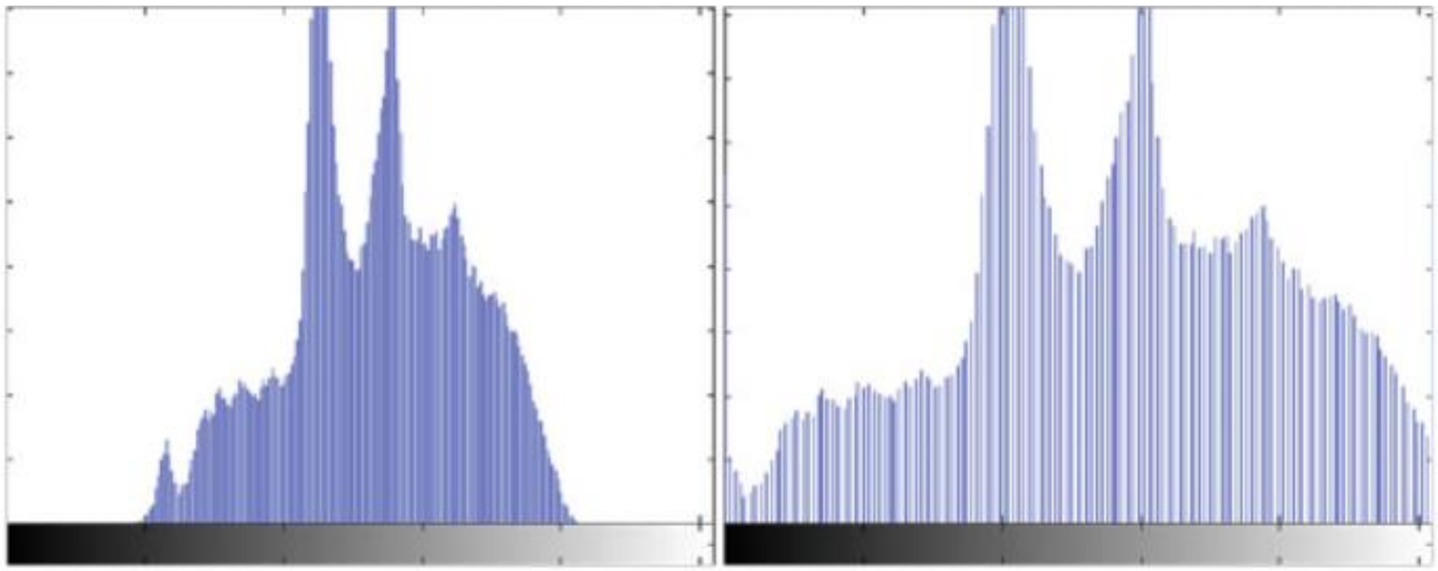
Contrast Enhancement

Image Histogram

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1



Histogram Stretching

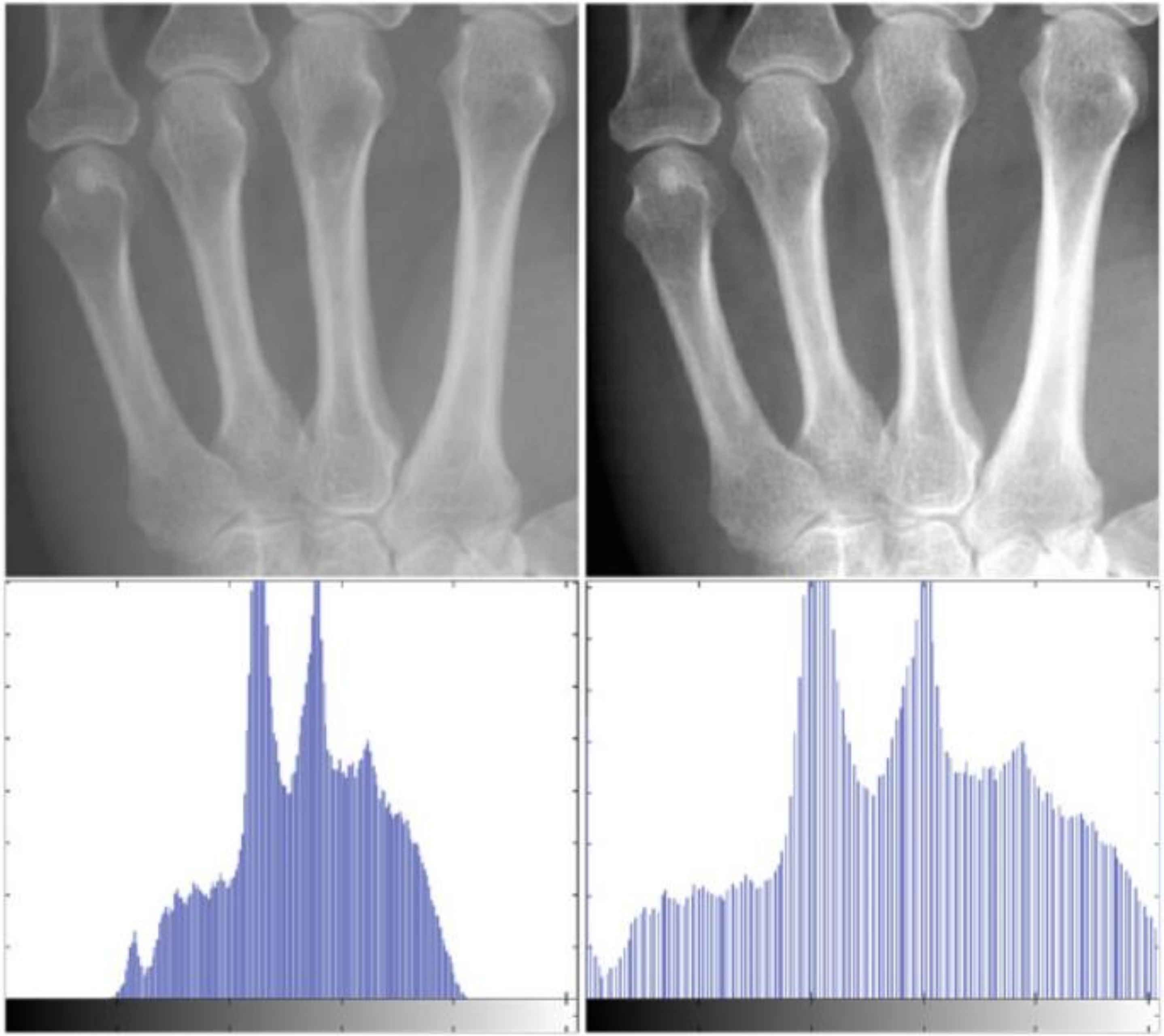


Histogram Stretching

- $f(x, y)$: Intensity of original image at x, y
  - f_{min}, f_{max}: min and max intensities of original image

- After histogram stretching
  - $g(x, y) = \frac{(f(x,y) - f_{min})}{(f_{max} - f_{min})} \times \max \mathit{possible\ intensity}$
  - $g(x, y) = \frac{(f(x,y) - f_{min})}{(f_{max} - f_{min})} \times 255$ (for 8-bit image)

Histogram Stretching

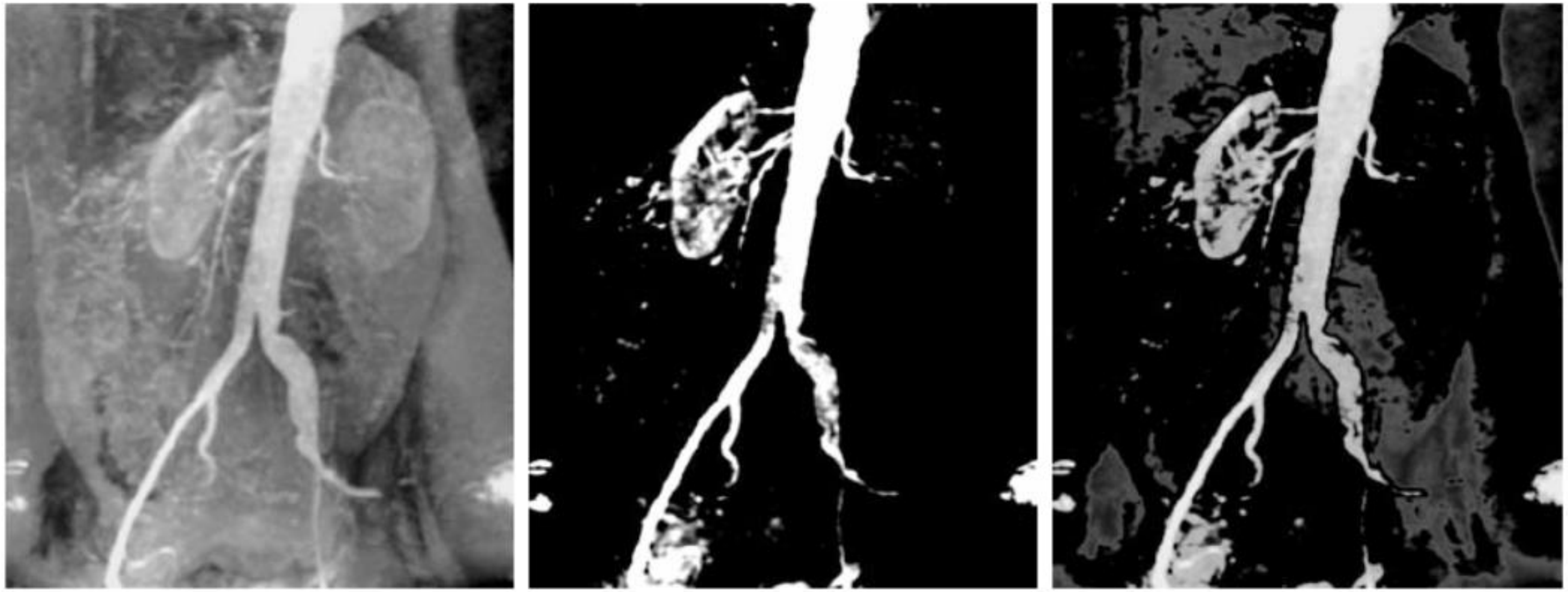


Before After

Intensity-level Slicing

- We may be interested to highlight a specific set of intensities
  - For certain applications
  - e.g: anomalies in x-ray, water bodies in satellite image, etc.
- Brightens desired range of intensities
- Darkens everything else

Intensity-level Slicing

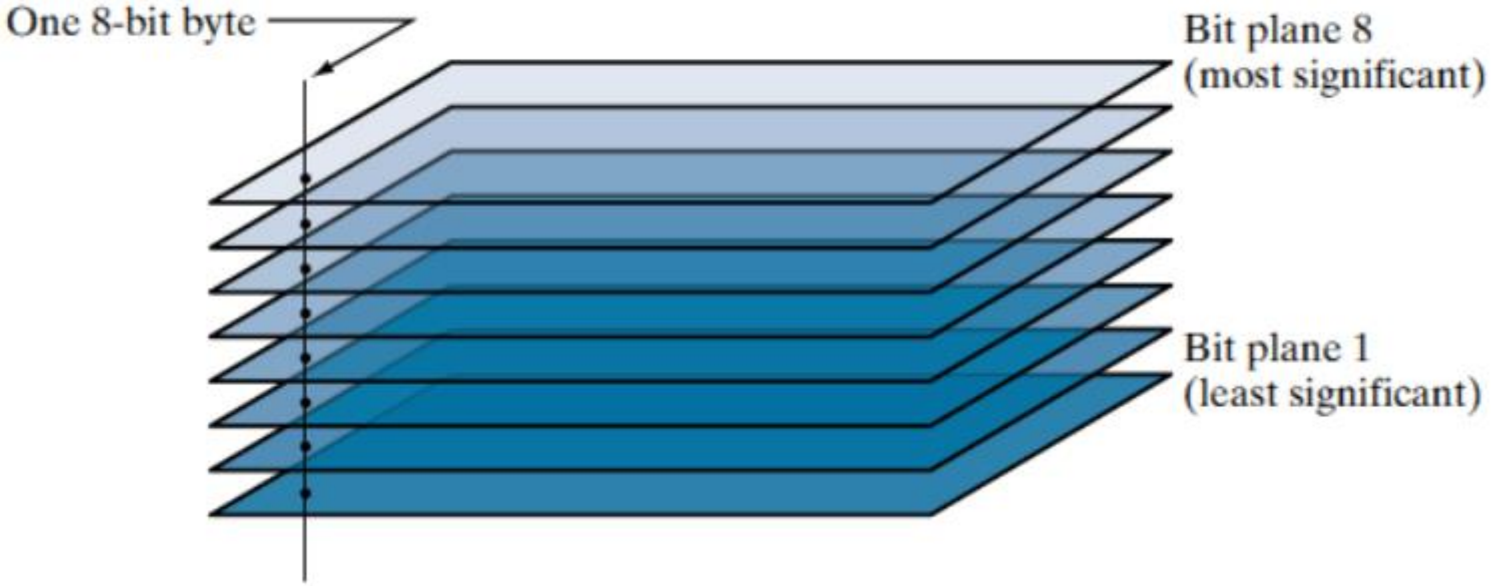


a b c

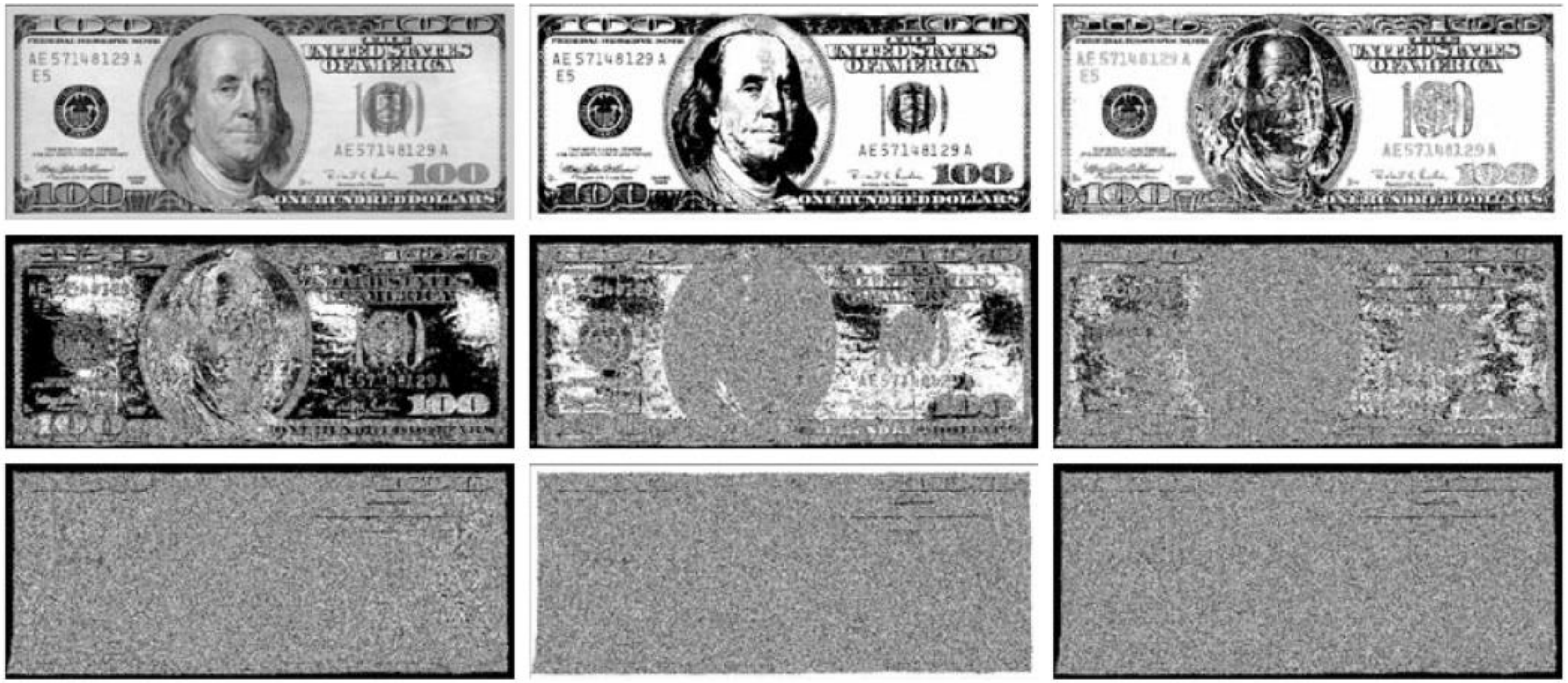
FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected range set near black, so that the grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Bit-plane Slicing

FIGURE 3.13
Bit-planes of an 8-bit image.



Bit-plane Slicing



a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 550×1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

Histogram Equalization

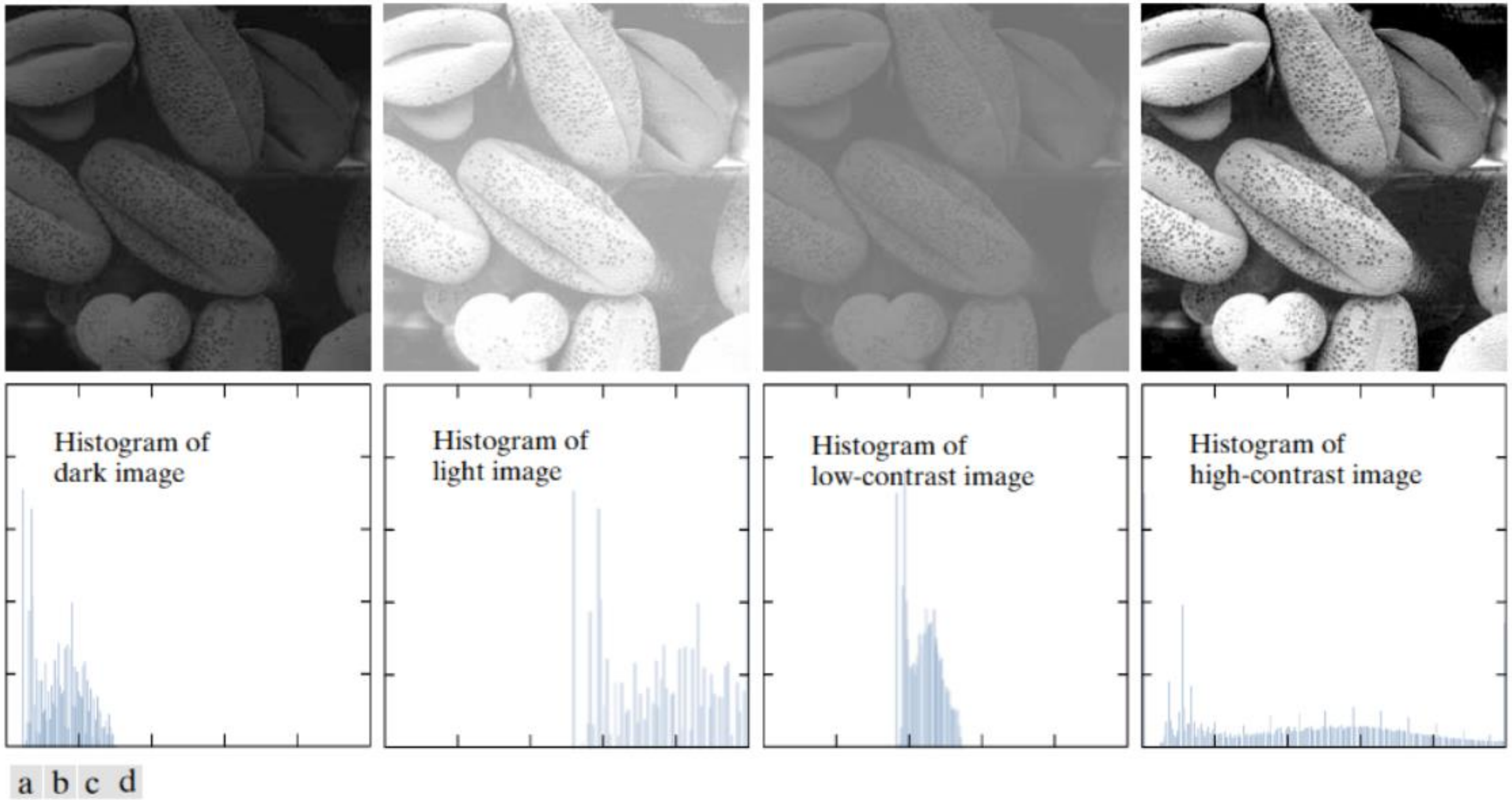


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Histogram Equalization

- Goal: Make the histogram uniform

- Use a suitable transformation in the original image
  - Histogram of the resultant image is nearly uniform

Histogram Equalization

- Use a suitable transformation in the original image
  - Histogram of the resultant image is nearly uniform

- For continuous intensity value, let $s = T(r)$ be such a transformation
  - r, s: intensities in original and target images, respectively
  - $T(r)$: monotonically increasing function in $0 \le r \le L - 1$
    - L: Max possible number of intensity VALUE
  - $0 \le T(r) \le L - 1$ for $0 \le r \le L - 1$

Histogram Equalization

- For continuous intensity value, let $s = T(r)$ be such a transformation
  - r, s: intensities in original and target images, respectively
  - $T(r)$: monotonically increasing function in $0 \leq r \leq L - 1$
    - L: Max possible number of intensity VALUE
  - $0 \leq T(r) \leq L - 1$ for $0 \leq r \leq L - 1$

- One possible discrete version
  - $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p(r_j)$; $\quad k = 0,1,2, \ldots, (L-1)$
    - $p(r_j) = \frac{N_j}{N}$
    - N_j: Total number of pixels in intensity j in the original image
    - N: Total number of pixels in the original image

Histogram Equalization

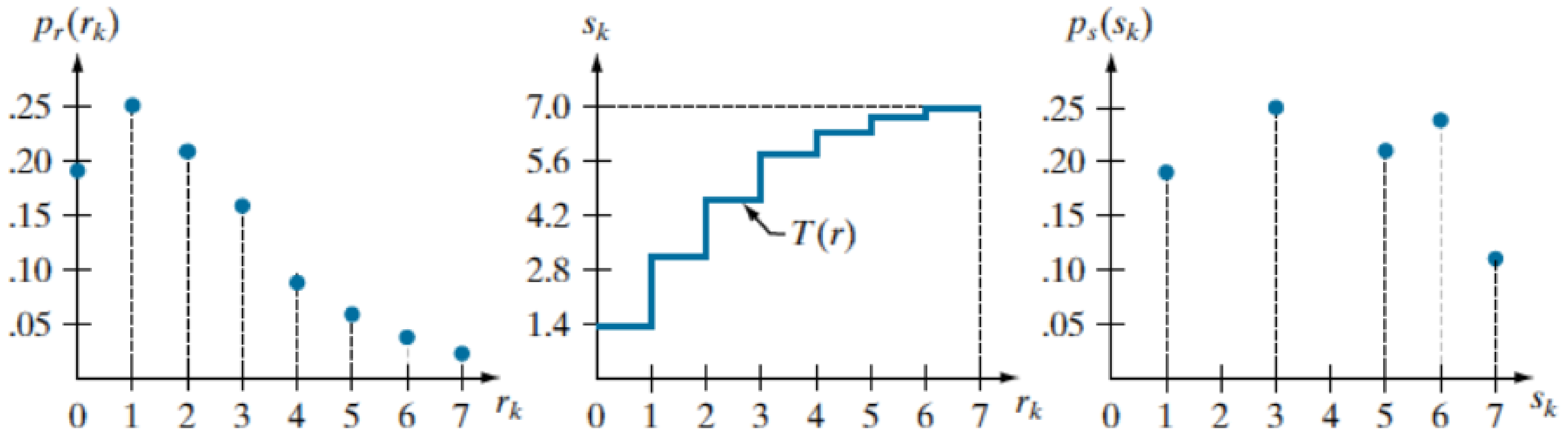
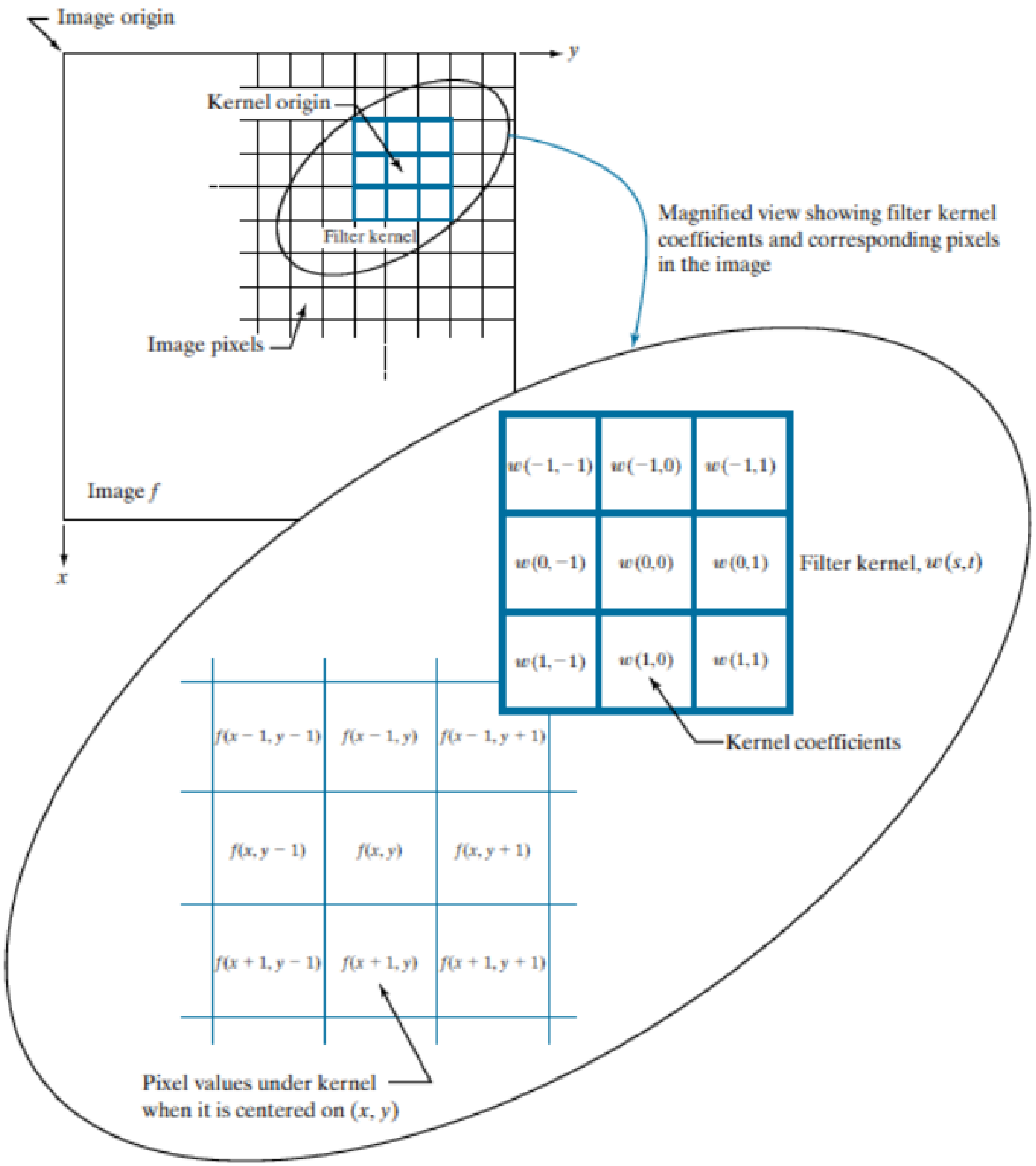


FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Spatial Filtering

Spatial Correlation

FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.



$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Spatial Correlation

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Spatial Convolution

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x - s, y - t)$$

Spatial Correlation and Convolution

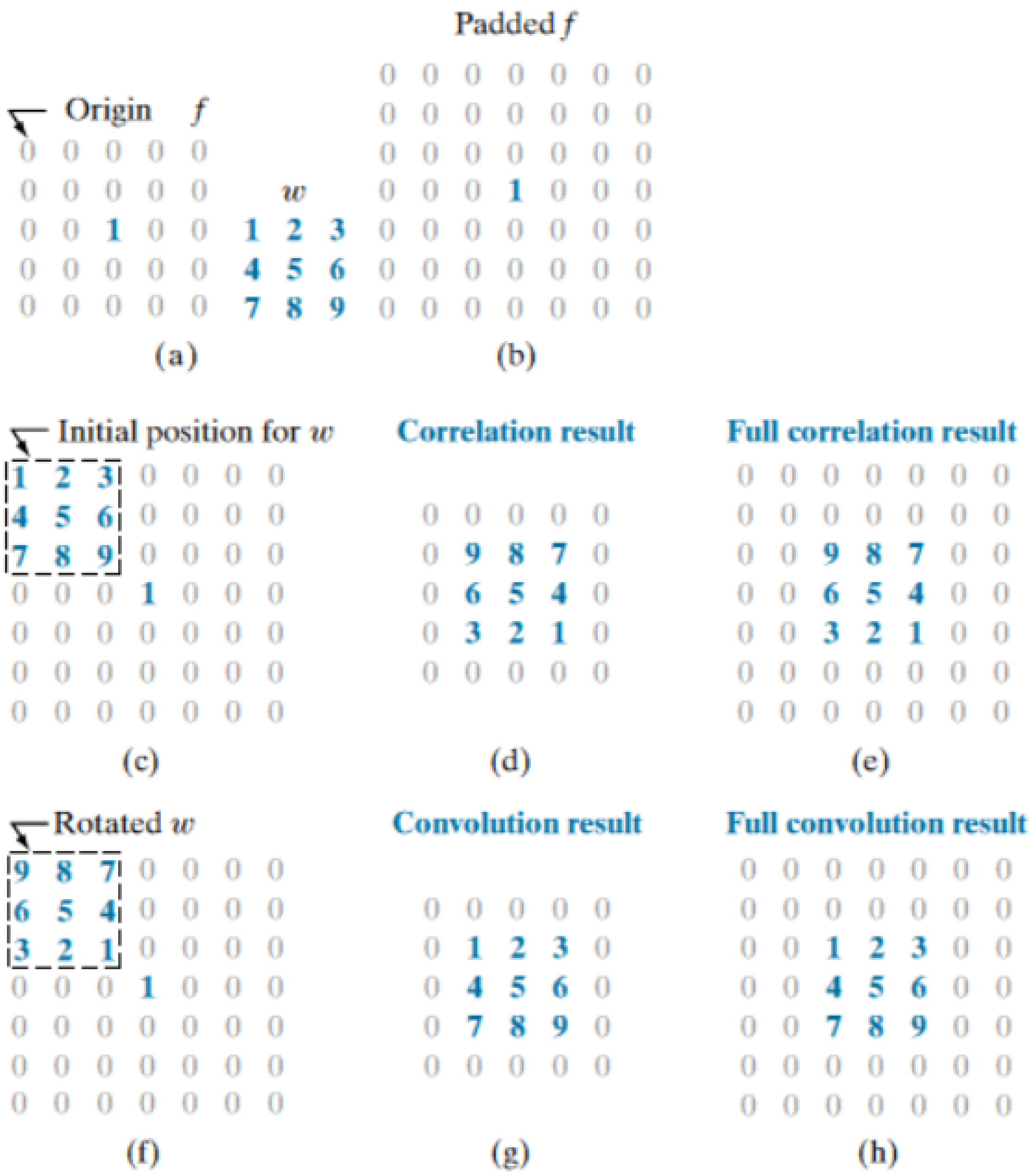


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0's are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of x and y. As these variable change, they *displace* one function with respect to the other. See the discussion of Eqs. (3-36) and (3-37) regarding full correlation and convolution.

Spatial Correlation

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Kernel: array with which sum of product of pixel values are performed

Separable Filter Kernel

- Kernel: aka Maks, template, window
  - Defines the size of neighbourhood
  - Coefficients determine the nature of the filter

- Separable function
  - A 2-D function $w(x, y)$ is separable if it can be written as the product of two 1-D functions, $w_1(x), w_2(y)$
  - $w(x, y) = w_1(x) * w_2(y)$ [$*$ indicates matrix multiplication]

Separable Filter Kernel: Example

- $w(x, y) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

- $c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ r = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

- $w(x, y) = cr^T$

Separable Filter Kernel

- Computational advantage

- $M \times N$: Image size
- $m \times n$: Kernel size

$$(w \star f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- Without separability
  - Computation: $MNmn$ multiplications and additions
- With separability
  - Computation: $MN(m+n)$ multiplications and additions

Smoothing Spatial Filters

- Low pass filters
  - Removes noise (high-frequency)
  - Reduces sharp transitions in intensity

- Box filter
- Gaussian Filter
- Mean Filter
- Rank filters

Box Filter

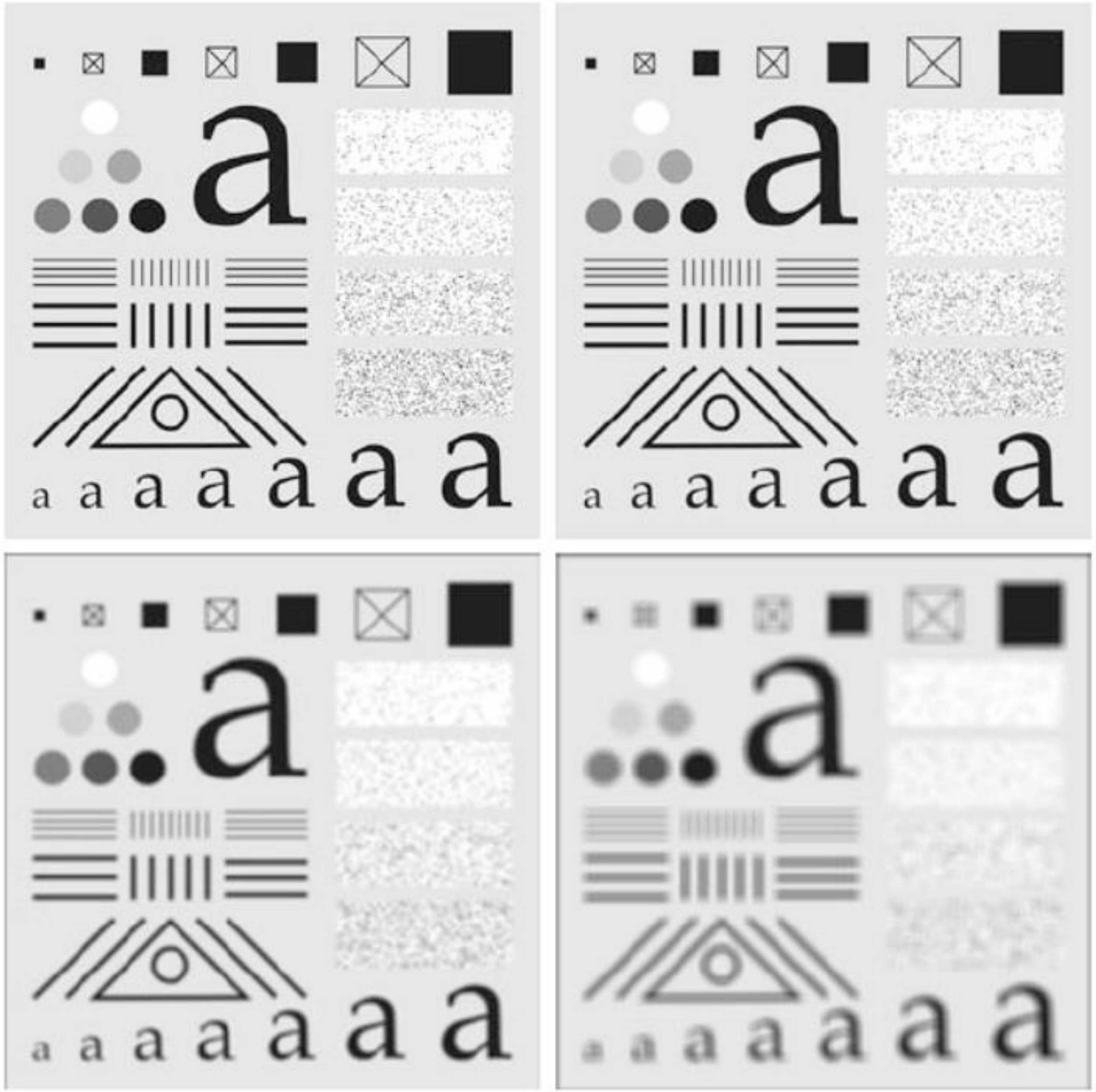
$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3×3 Kernel

Box Filter

a	b
c	d

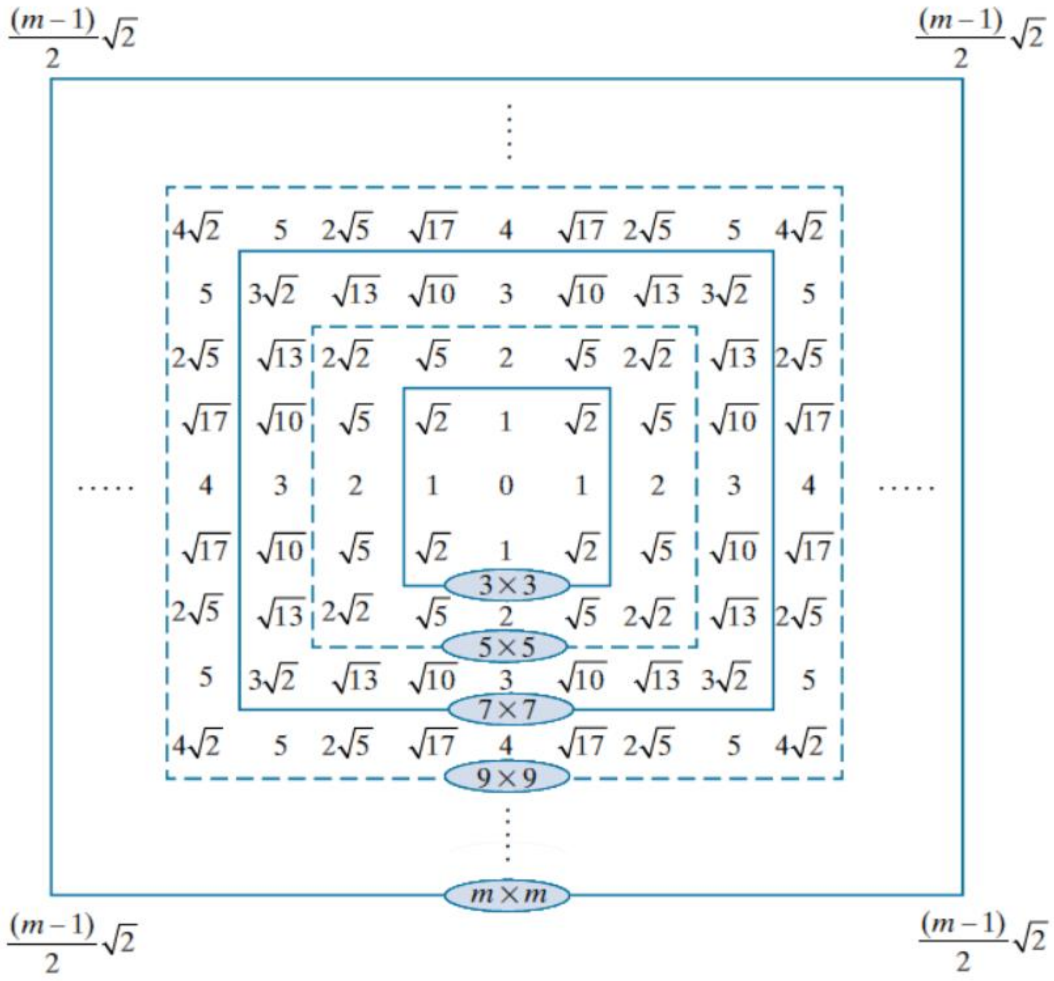
FIGURE 3.33
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.



Gaussian Filter

FIGURE 3.34
Distances from the center for various sizes of square kernels.

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

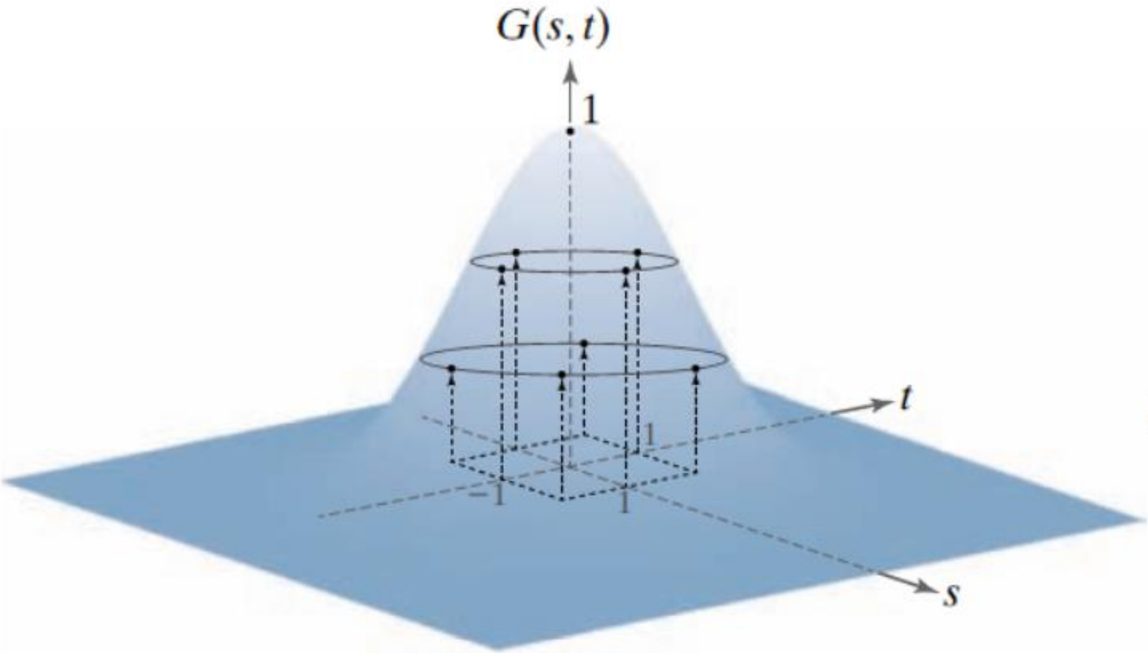


Gaussian Filter

3×3 Kernel

a b

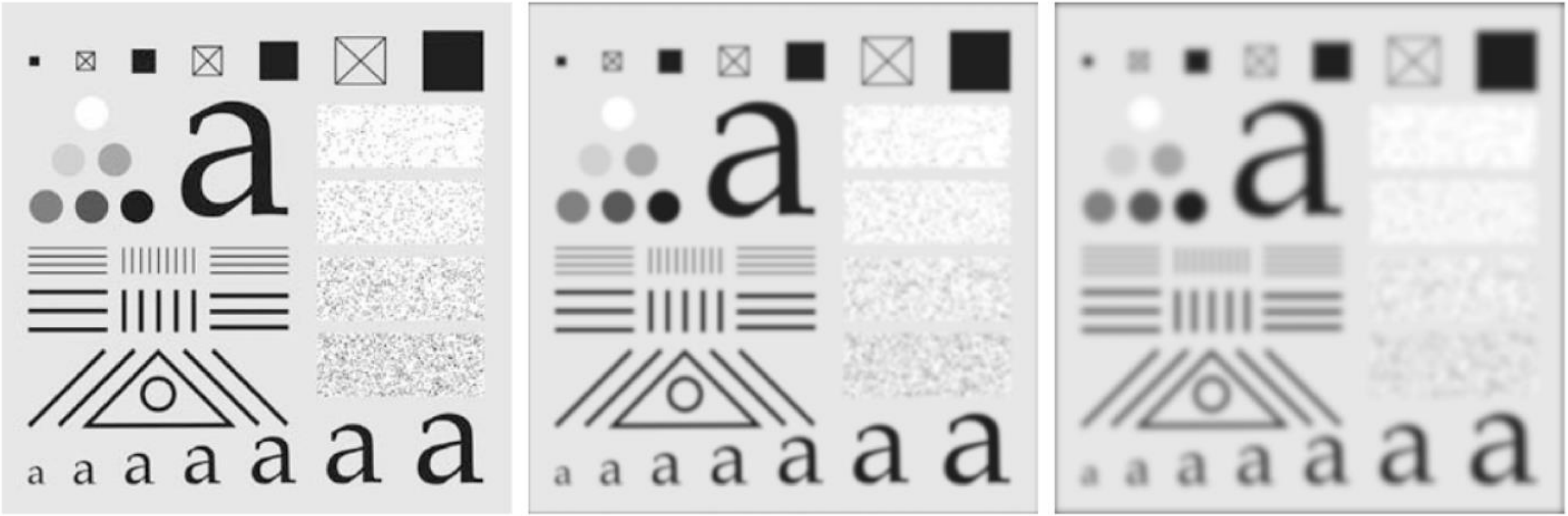
FIGURE 3.35
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.31(b)].



$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

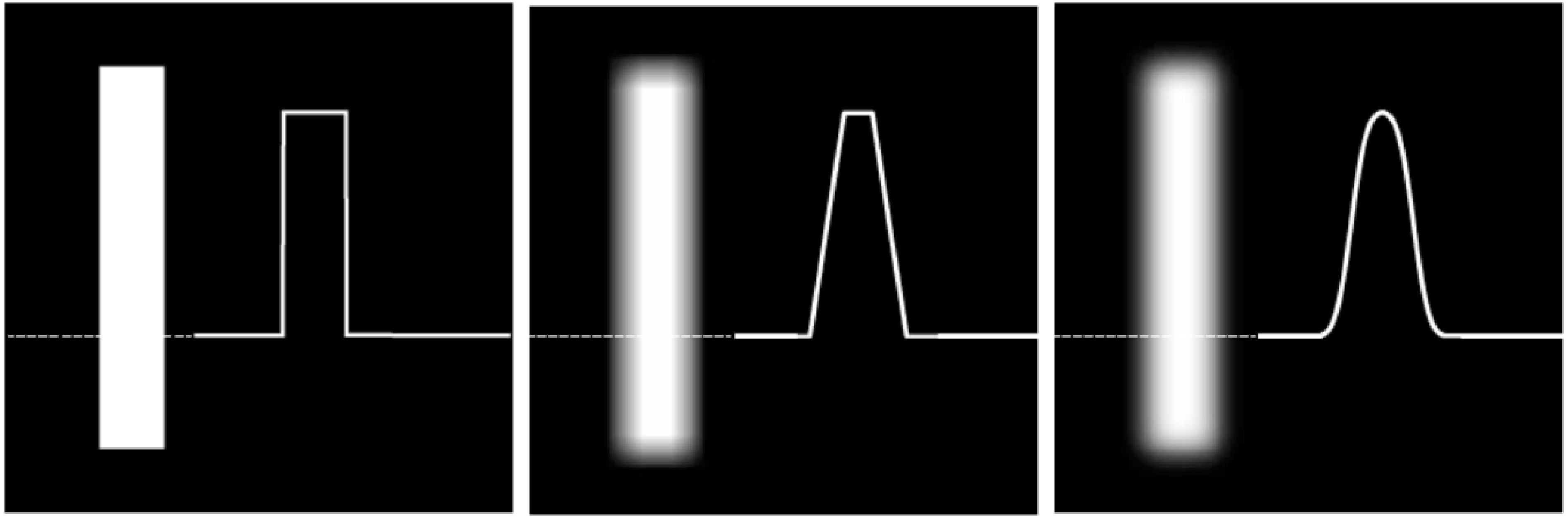
Gaussian Filter



a b c

FIGURE 3.36 (a)A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.33(d). We used $K = 1$ in all cases.

Gaussian and Box Filters



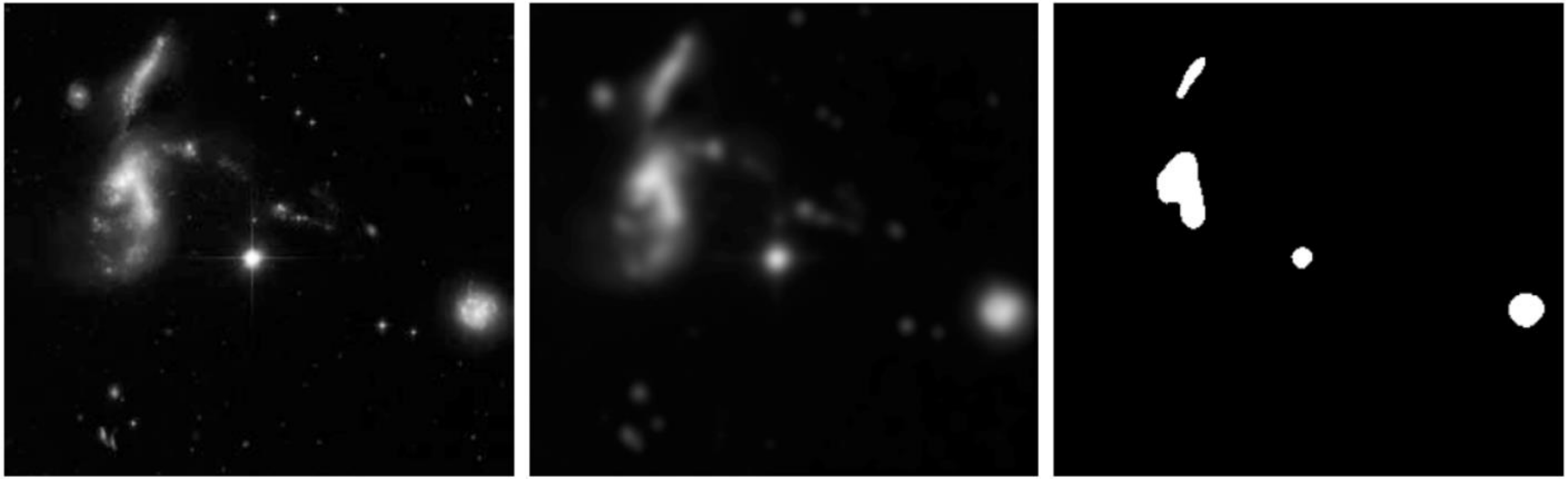
a b c

FIGURE 3.38 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.

Mean Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Lowpass Filtering and Thresholding for Region Extraction



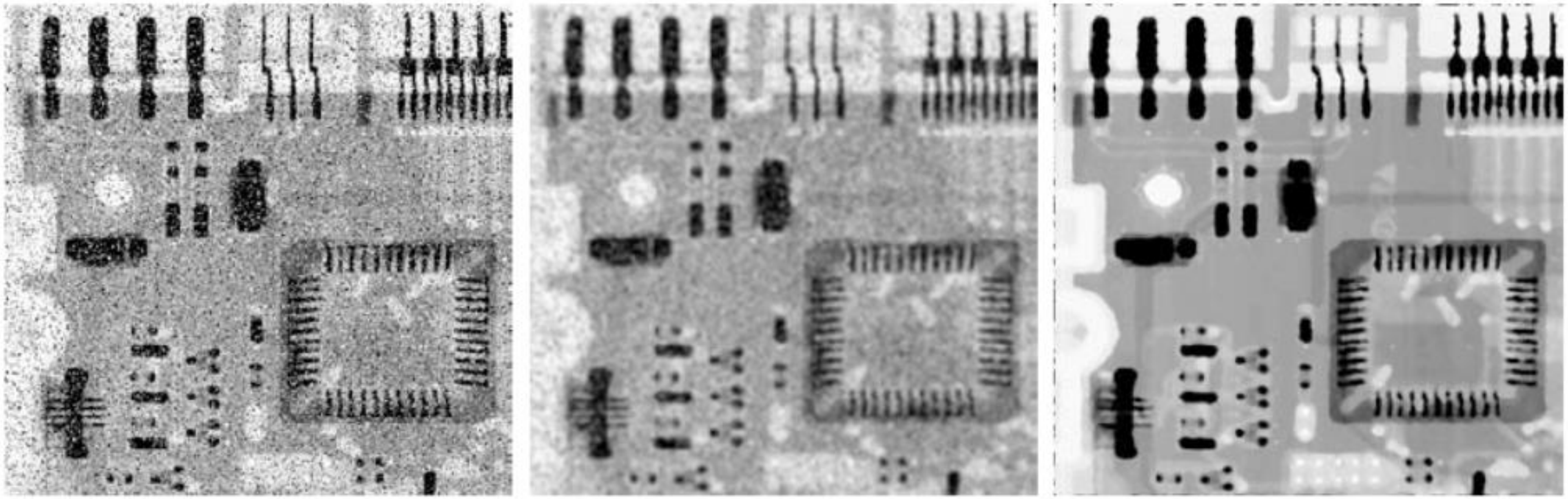
a b c

FIGURE 3.41 (a) A 2566×2758 Hubble Telescope image of the *Hickson Compact Group*. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range [0, 1]). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)

Order-statistic (Nonlinear) Filters

- Filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter
- Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result
- Example: median filter

Low pass vs Median Filter



a b c

FIGURE 3.43 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening Spatial Filters

- High pass filters
- Highlights transitions in intensity
- Use of derivative operators
  - Image differentiation
    - Enhances edges and other discontinuities (such as noise)
    - De-emphasizes areas with slowly varying intensities

Laplacian Filter

- Use of second derivative

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplacian Filter

- Use of second derivative

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

Laplacian Filter

- Use of second derivative

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Laplacian Filter

0	1	0
1	−4	1
0	1	0

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Laplacian Filter

- Sharpened image

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

Laplacian Filter

0	1	0
1	−4	1
0	1	0

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Try filtering with non-zero value of diagonal elements

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Laplacian Filter

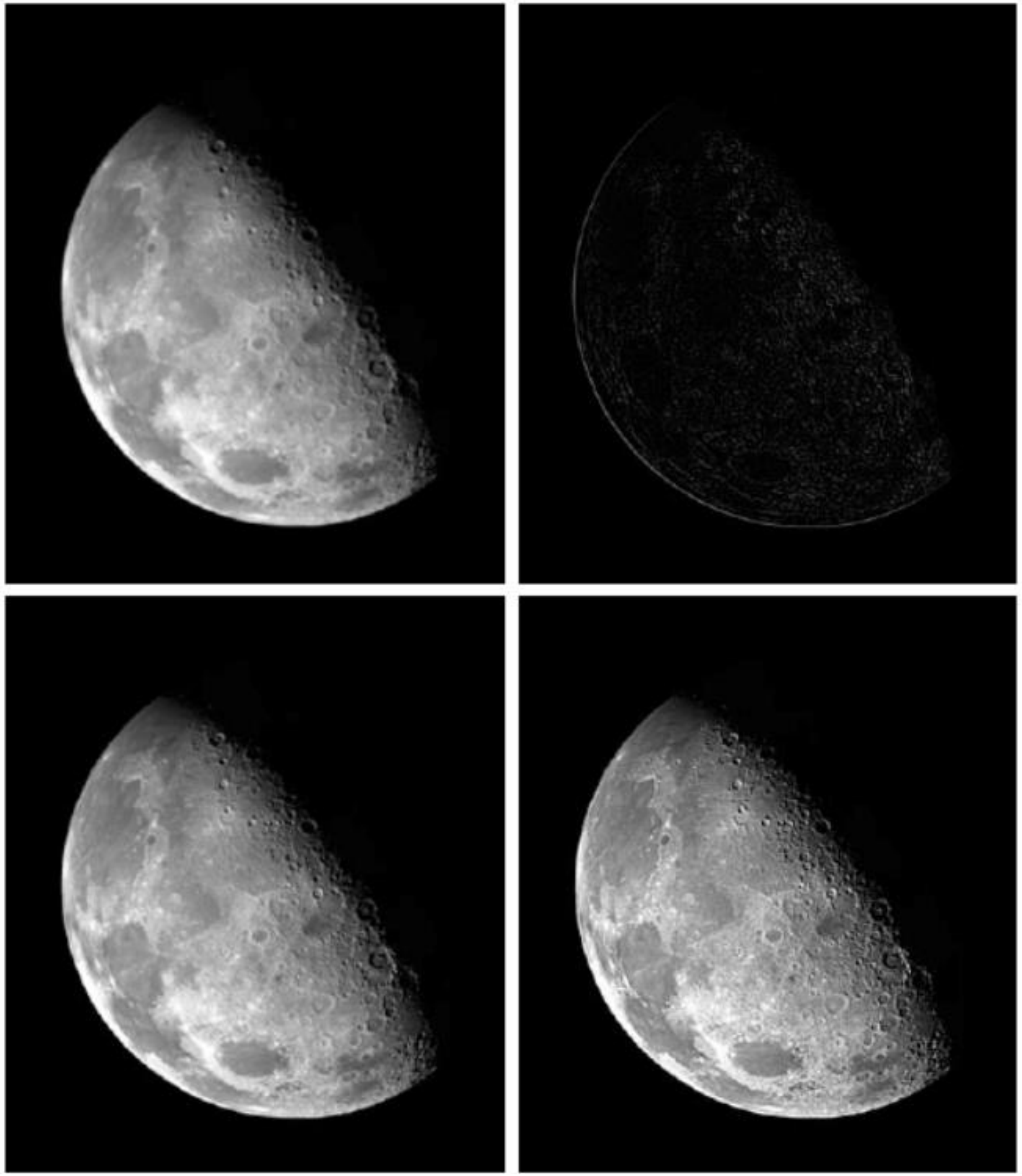
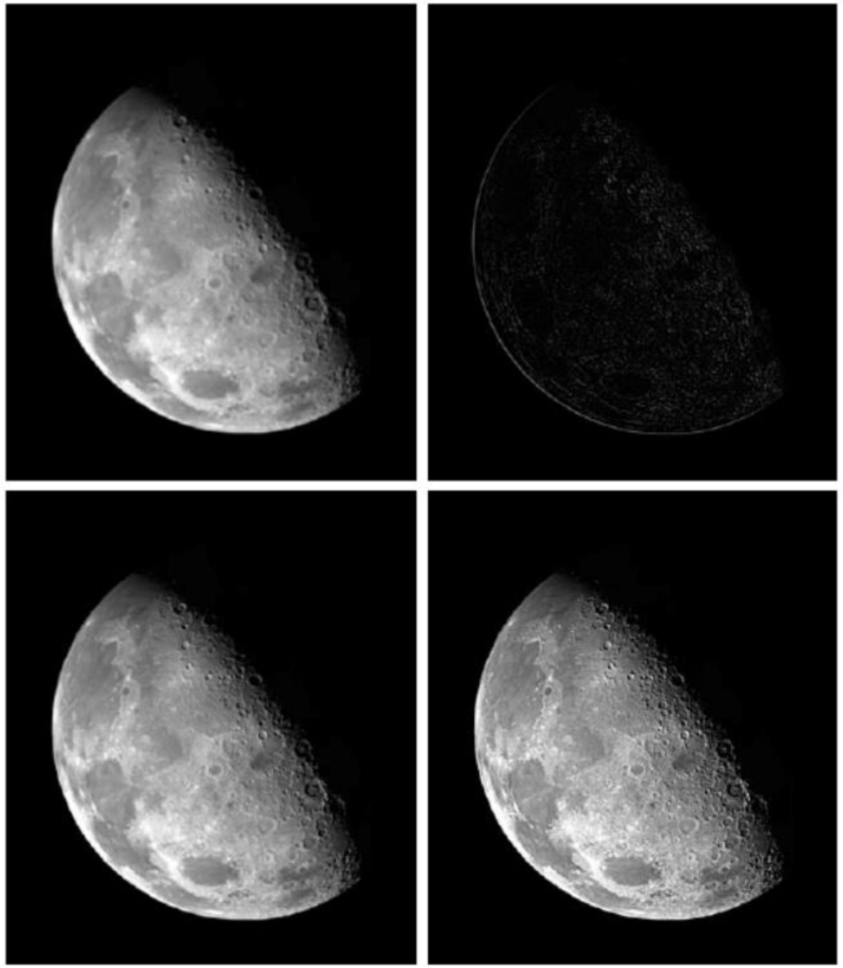


FIGURE 3.46 (a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 3.45(a). (c) Image sharpened using Eq. (3-54) with $c = -1$. (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b). (Original image courtesy of NASA.)

Laplacian Filter

a b
c d

FIGURE 3.46
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.45(a).
(c) Image sharpened using Eq. (3-54) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b).
(Original image courtesy of NASA.)



Can be used for edge detection

Unsharp Masking and Highboost Filtering

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the mask.)
3. Add the mask to the original.

$$M(x, y) = f(x, y) - \bar{f}(x, y)$$
$$g(x, y) = f(x, y) + kM(x, y)$$

$g(x, y)$ is the sharpened image

Unsharp Masking and Highboost Filtering

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the mask.)
3. Add the mask to the original.

$$M(x,y) = f(x,y) - \bar{f}(x,y)$$
$$g(x,y) = f(x,y) + kM(x,y)$$

$g(x,y)$ is the sharpened image

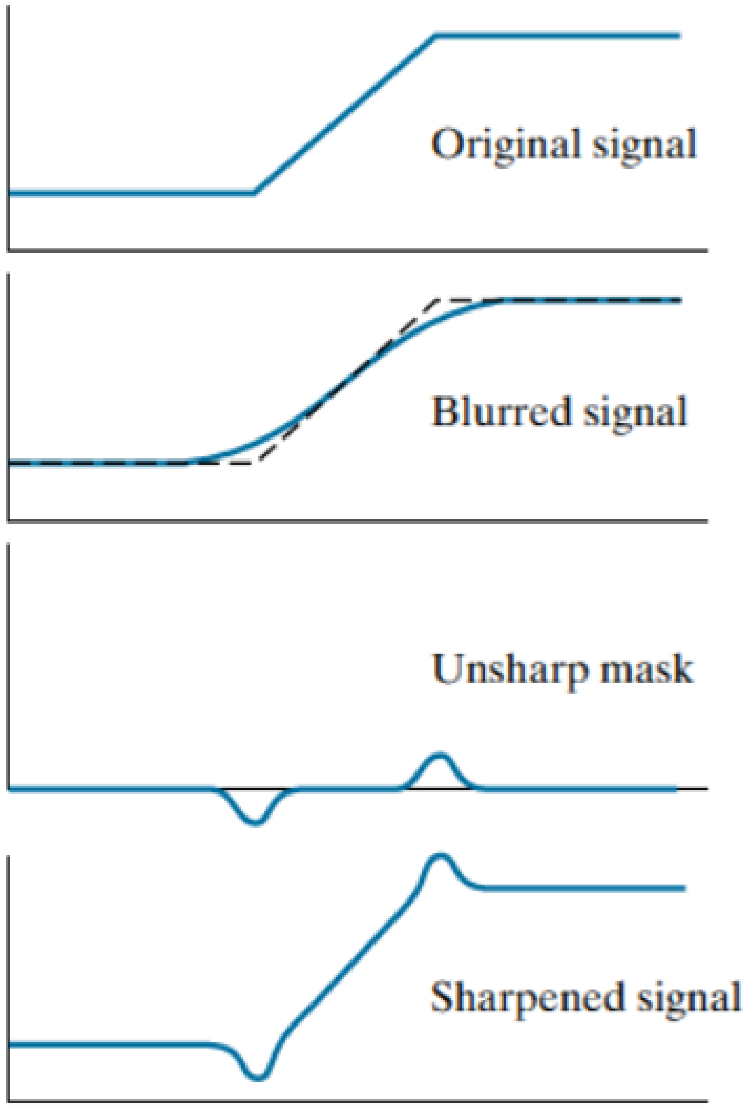
$\boldsymbol{k = 1}$**: unsharp masking**

$\boldsymbol{k > 1}$**: highboost filtering**

Unsharp Masking and Highboost Filtering

a
b
c
d

FIGURE 3.48
1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

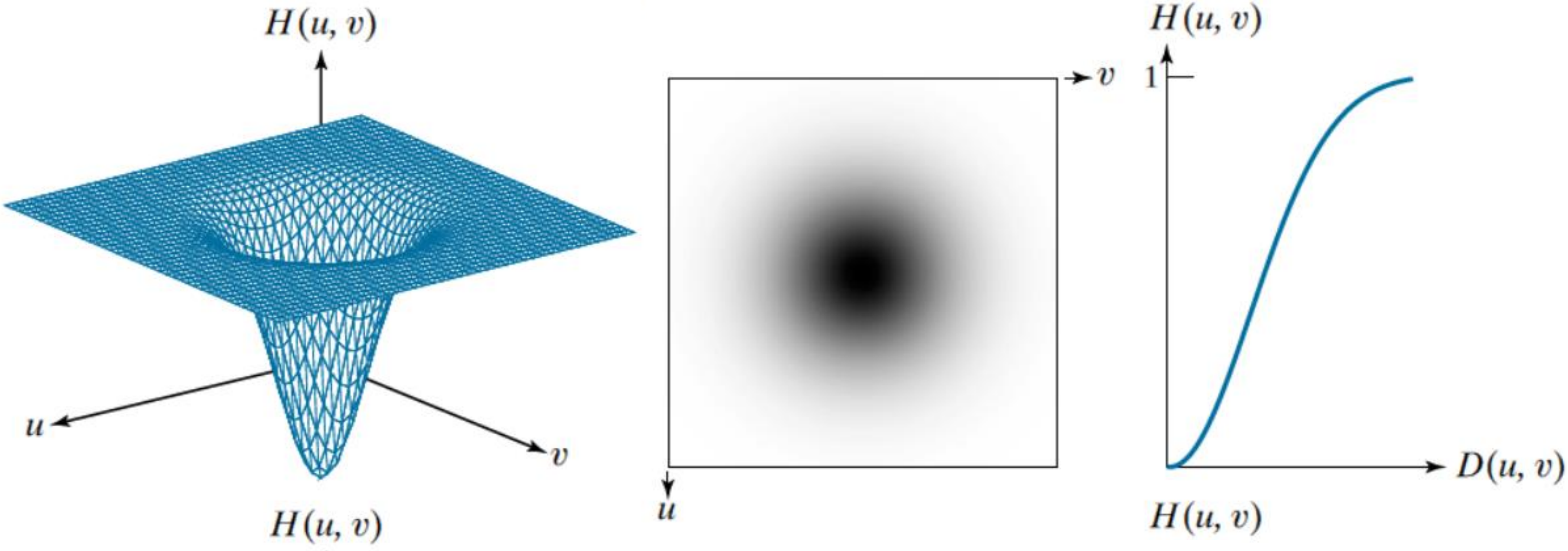


Unsharp Masking and Highboost Filtering

a b c
d e

FIGURE 3.49 (a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) Result of highboost filtering with $k = 4.5$.

Gaussian High Pass Filter



High-frequency Emphasis

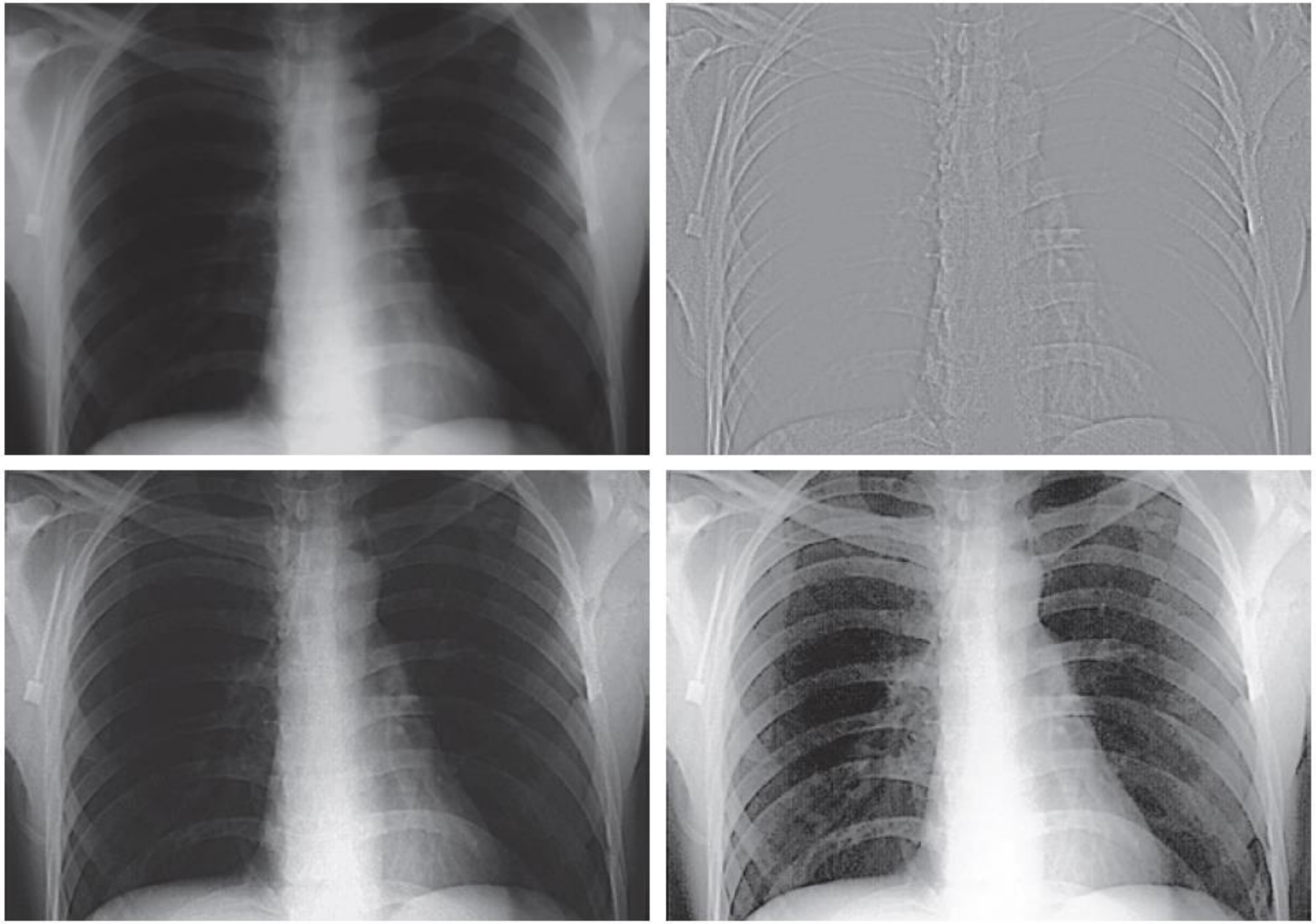
$$g(x, y) = \Im^{-1}\{[1 + kH_{\mathrm{HP}}(u, v)]F(u, v)\}$$

$$\downarrow$$

$$g(x, y) = f(x, y) + kM_{HP}(x, y)$$

$M_{HP}(x, y)$: *output of highpass filter when* $f(x, y)$ *is applied*

An Example



a	b
c	d

FIGURE 4.57
(a) A chest X-ray. (b) Result of filtering with a GHPF function. (c) Result of high-frequency-emphasis filtering using the same GHPF. (d) Result of performing histogram equalization on (c). (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

Another Type of Filtering

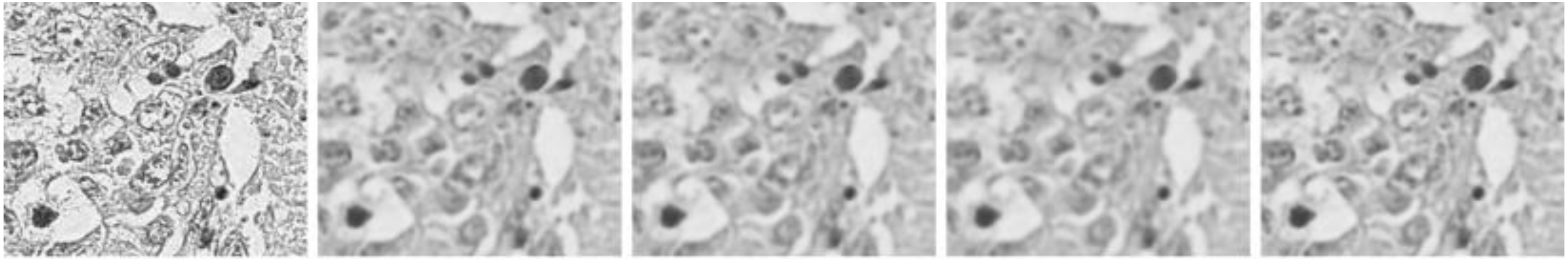
Another Type of Filtering

Another Type of Filtering

Another Type of Filtering

- What am I doing?
- How to do it? (Assignment)

Another Type of Filtering



(a) (b) (c) (d) (e)

- What am I doing?
- How to do it? (Assignment)

How to Measure the Quality of Noise Removal

- Mean-squared error, PSNR
  - Between denoised image and noise free image

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right)$$

How to Measure the Quality of Noise Removal

- SSIM: Structural similarity (SSIM) index for measuring image quality

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}$$

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Medical Image Segmentation

Image Segmentation

- R: Entire spatial region occupied by an image
- Image segmentation: partitions R into n subregions, $R_1, \ldots, R_n$ such that

(a) $\bigcup_{i=1}^{n} R_i = R.$

(b) R_i is a connected set, for $i = 0, 1, 2, \ldots, n.$

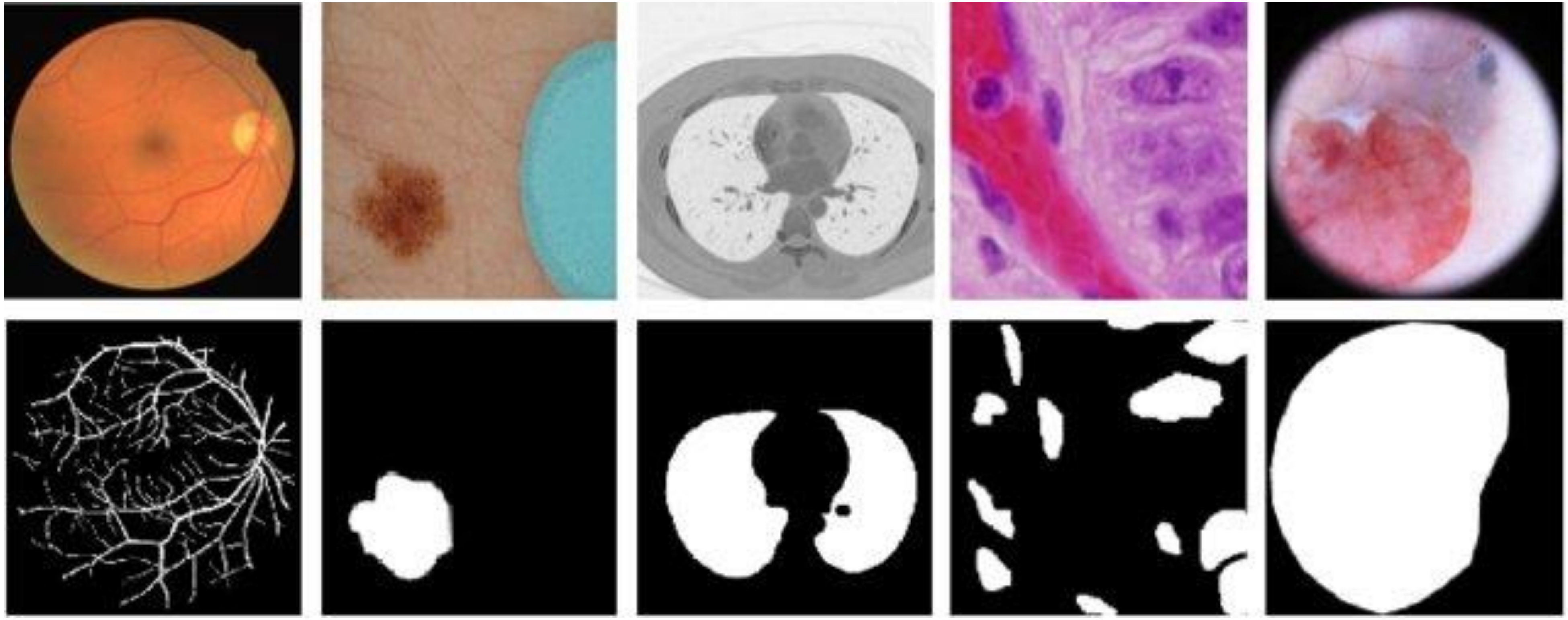
(c) $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j.$

(d) $Q(R_i) = \text{TRUE}$ for $i = 0, 1, 2, \ldots, n.$

(e) $Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j.

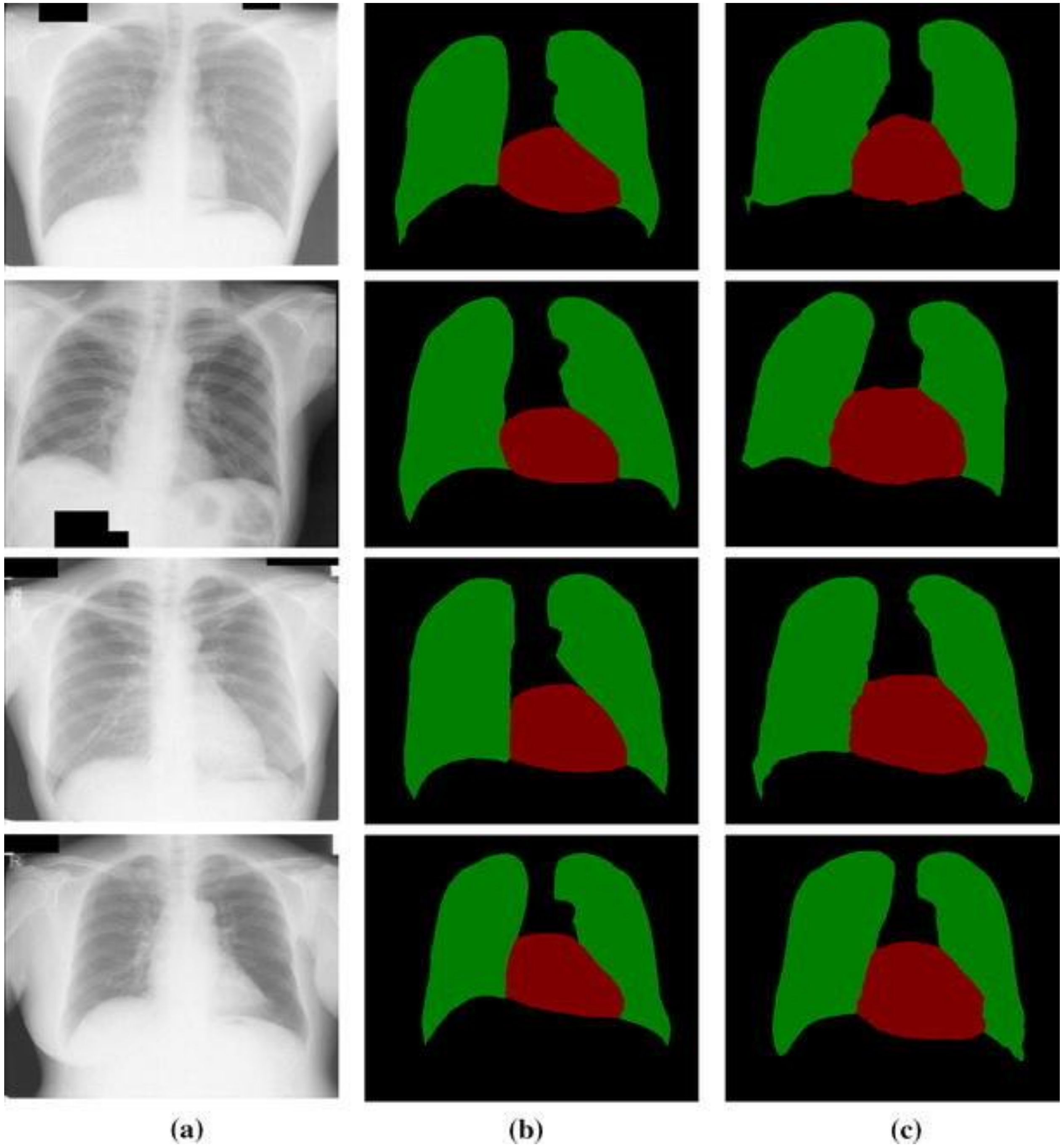
$Q(R_k)$: A logical predicate

Image Segmentation



https://www.researchgate.net/publication/339873650_Multi-level_Context_Gating_of_Embedded_Collective_Knowledge_for_Medical_Image_Segmentation/figures?lo=1

Semantic Segmentation



Thresholding for Segmentation

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \le T \end{cases}$$

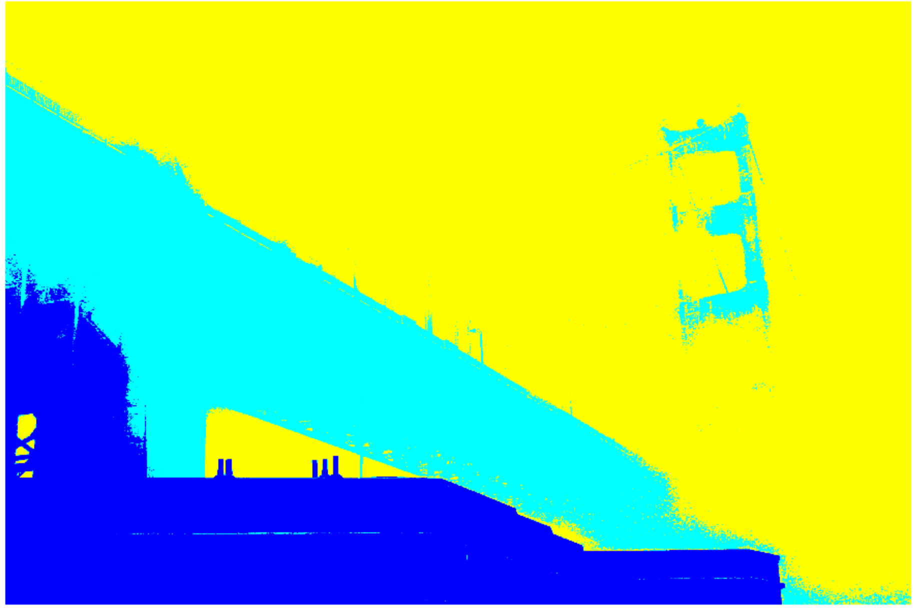
- Binary segmentation
- T: Threshold
- Global thresholding
- Local/ regional/ dynamic/ adaptive thresholding

Multiple/ Multilevel Thresholding

$$g(x,y)=\begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 < f(x,y) \le T_2 \\ c & \text{if } f(x,y) \le T_1 \end{cases}$$

Grayscale Image

Segmented Image



How to Choose the Threshold

- Possible options
  - Mean intensity
  - $(Max + Min)/2$
- Otsu's method
  - PDF of background and foreground class
  - Assumption: PDFs are Gaussian

How to Choose the Threshold

- Otsu's method
  - Properly thresholded classes should be distinct w.r.t. the intensity values of their pixels
- Normalized histogram has components
  - $p_i = \frac{n_i}{N}$
- $\sum_{i=0}^{L-1} p_i = 1$

How to Choose the Threshold

- Otsu's method
- Select a threhold $T(k) = k, 0 < k < L - 1$
- Threshold the image and create two classes
  - Class $c1$: pixels with intensity $0 - k$
  - Class $c2$: pixels with intensity $k + 1 - L - 1$
- Probability that a pixel is assigned to c_1
  - $Q_1(k) = \sum_{i=o}^{k} p_i$
- Probability that a pixel is assigned to c_2
  - $Q_2(k) = 1 - Q_1(k) = \sum_{i=k}^{L-1} p_i$

How to Choose the Threshold

- Otsu's method
- Probability that a pixel is assigned to c_1
  - $Q_1(k) = \sum_{i=o}^{k} p_i$
- Probability that a pixel is assigned to c_2
  - $Q_2(k) = 1 - Q_1(k) = \sum_{i=k}^{L-1} p_i$

- Effectiveness of threshold is measured using

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

How to Choose the Threshold

- Otsu's method
- Effectiveness of threshold is measured using

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

$$\boldsymbol{\sigma_B^2 = Q_1(m_1 - m_G)^2 + Q_2(m_2 - m_G)^2}$$

How to Choose the Threshold

- Otsu's method

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

- Optimum threshold

$$\sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

How to Choose the Threshold

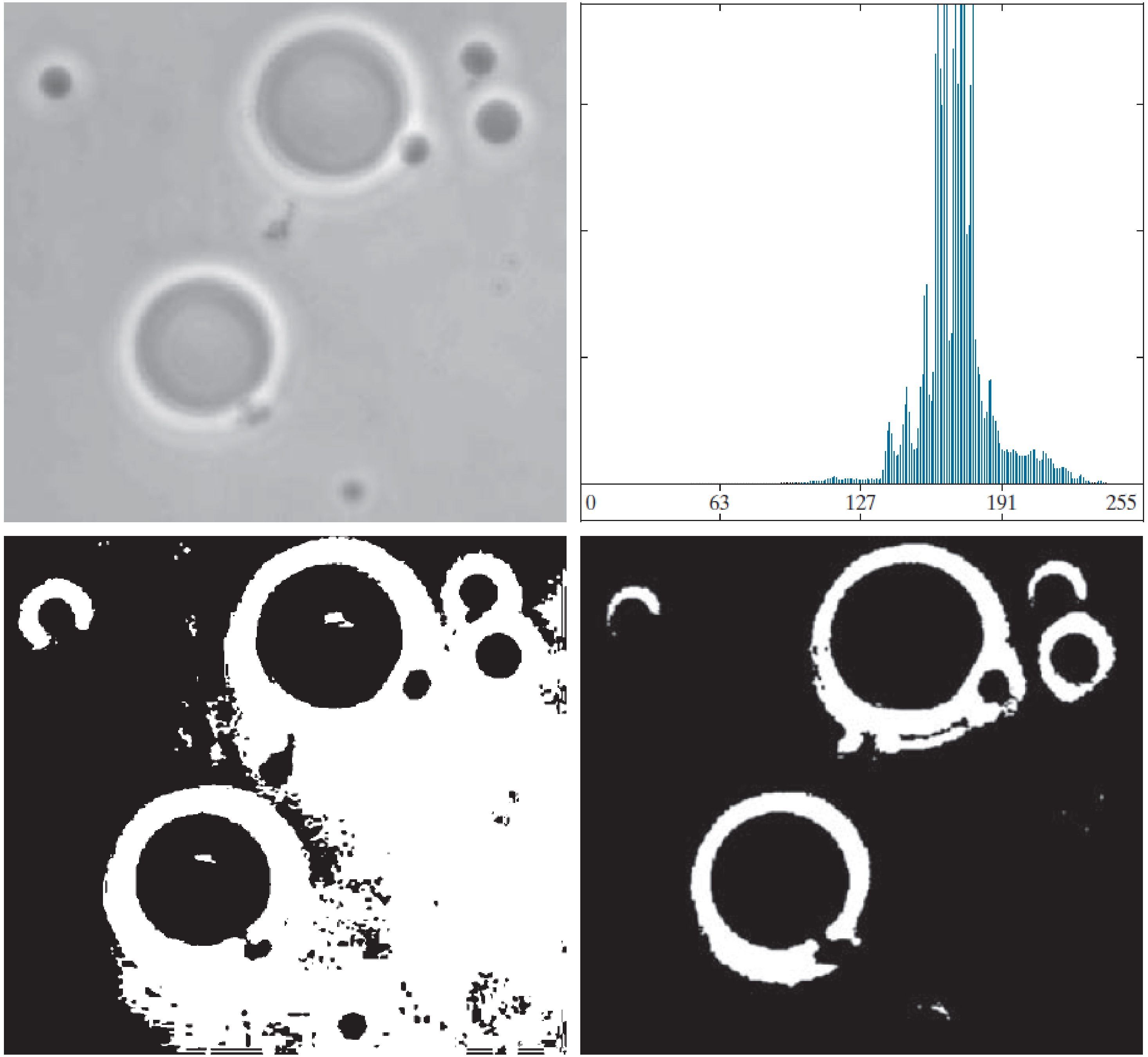
- Otsu's method

- Optimum threshold

$$\sigma_B^2\left(k^*\right) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) \le k^* \end{cases}$$

Example



a b
c d

FIGURE 10.36 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3. (d) Result using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Smoothing for Thresholding

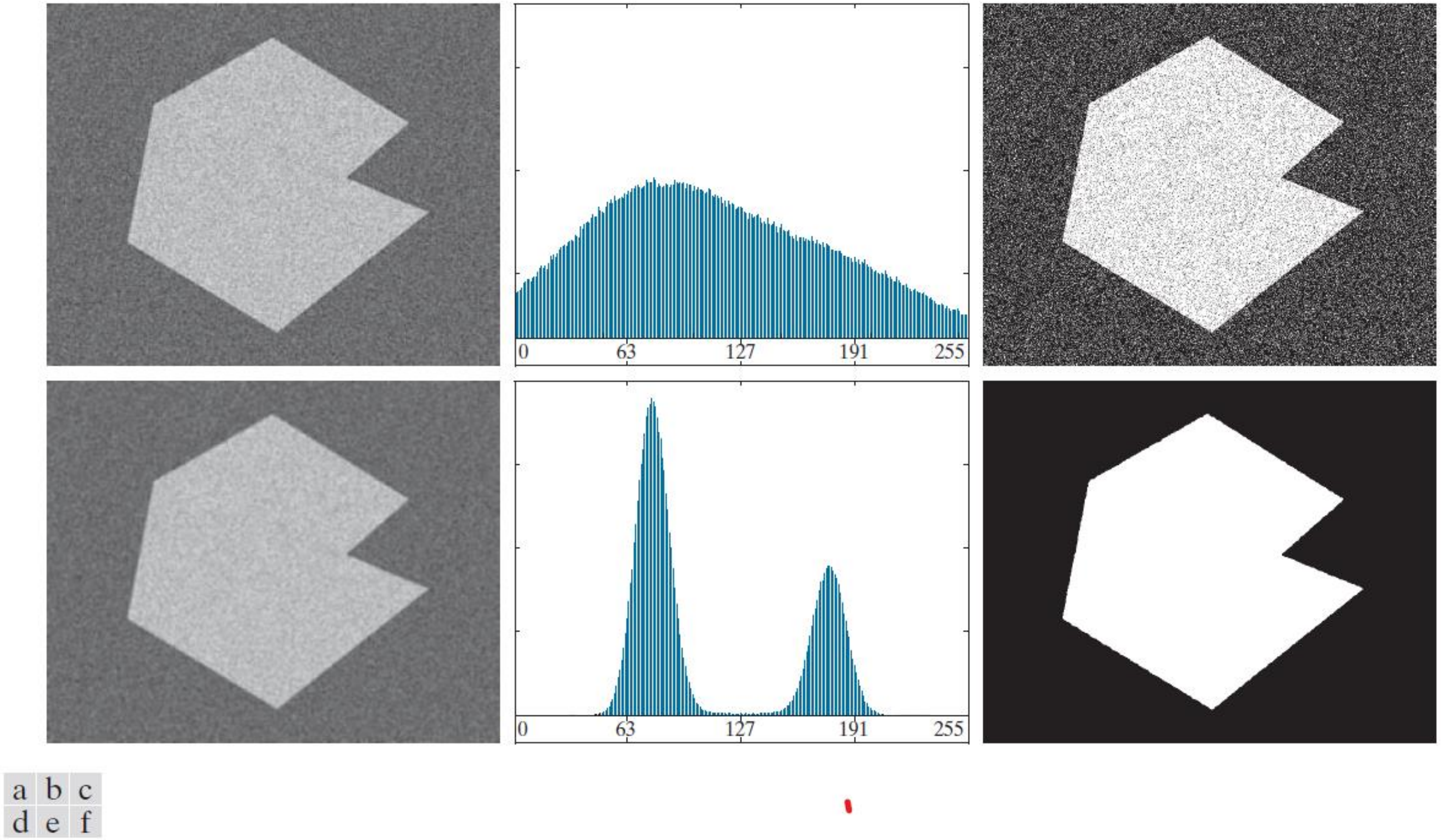


FIGURE 10.37 (a) Noisy image from Fig. 10.33(c) and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method.

Maximum Entropy Thresholding

$$\text{Entropy} = \sum_{i=0}^{L-1} p_i \log \frac{1}{P_i}$$

Choose a gray level t

$$\text{Total Entropy } E(t) = \sum_{i=0}^{t} p1_i \log \frac{1}{P1_i} + \sum_{i=t+1}^{L-1} p2_i \log \frac{1}{P2_i}$$

Find out t, such that $E(t)$ is maximum

Maximum Entropy Thresholding

$$\text{Entropy} = \sum_{i=0}^{L-1} p_i \log \frac{1}{P_i}$$

Choose a gray level t

$$\text{Total Entropy } \; E(t) = \sum_{i=0}^{t} p1_i \log \frac{1}{P1_i} + \sum_{i=t+1}^{L-1} p2_i \log \frac{1}{P2_i}$$

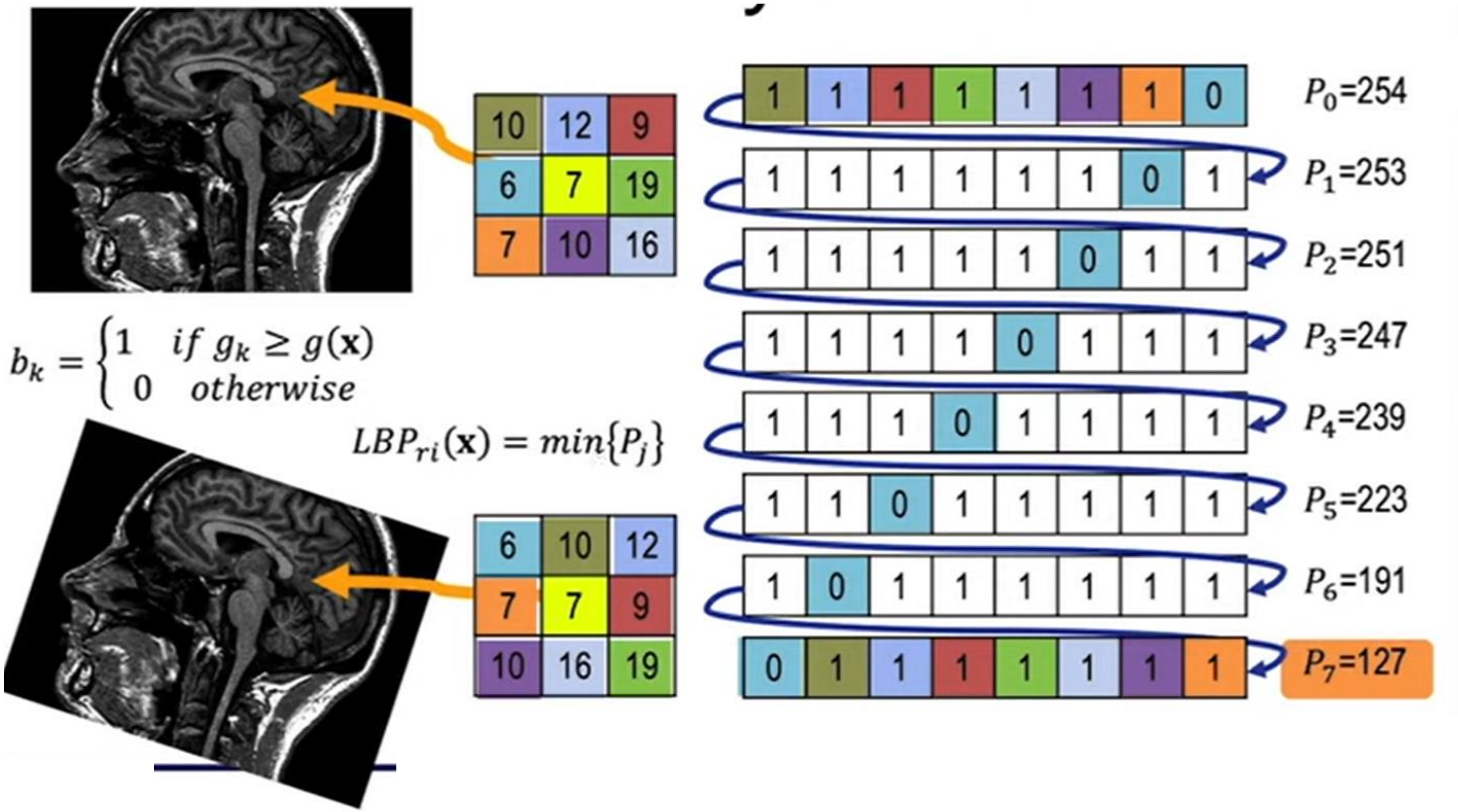
Find out t, such that $E(t)$ is maximum

Maximization of information between object and background

Assignment: Presentation

- Region growing and merging
- Show results on images of any medical image segmentation problem

Texture through Local Binary Pattern



https://www.youtube.com/watch?v=Y_zQ9B4rWMY&list=PLi7vCu7jEp8_nFoyZ-8exq5UYW_CAZ6zM&index=8

Clustering for Segmentation

k-means clustering

$$\arg\min_{C}\left(\sum_{i=1}^{k}\sum_{\mathbf{z}\in C_i}\left\|\mathbf{z}-\mathbf{m}_i\right\|^2\right)$$

Clustering for Segmentation

1. ***Initialize the algorithm:*** Specify an initial set of means, $\mathbf{m}_i(1)$, $i = 1, 2, \ldots, k$.
2. ***Assign samples to clusters:*** Assign each sample to the cluster set whose mean is the closest (ties are resolved arbitrarily, but samples are assigned to only *one* cluster):

$$\mathbf{z}_q \rightarrow C_i \text{ if } \|\mathbf{z}_q - \mathbf{m}_i\|^2 < \|\mathbf{z}_q - \mathbf{m}_j\|^2 \quad j = 1, 2, \ldots, k \ (j \neq i); \ q = 1, 2, \ldots, Q$$

3. ***Update the cluster centers (means):***

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \qquad i = 1, 2, \ldots, k$$

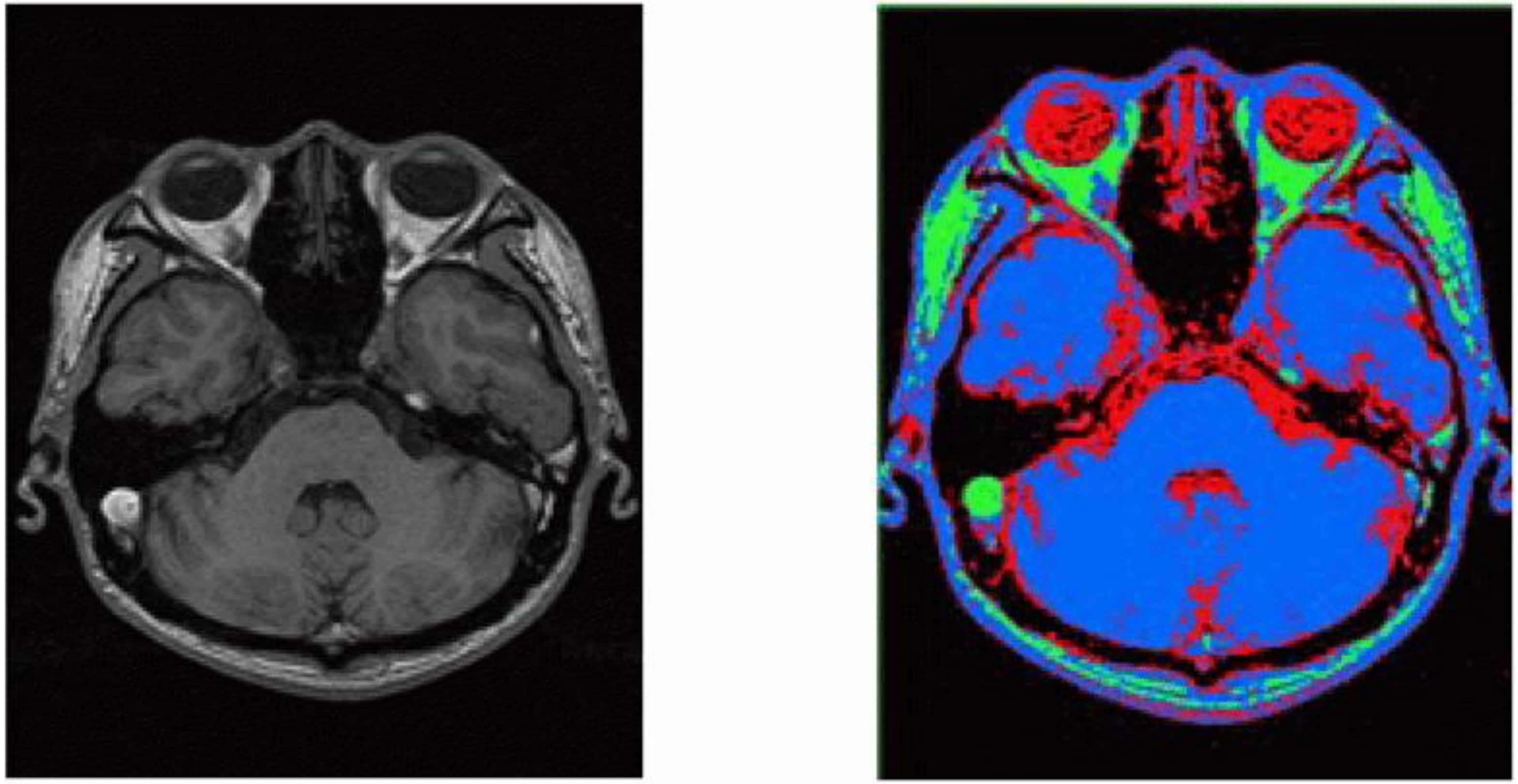
   where $|C_i|$ is the number of samples in cluster set C_i.
4. ***Test for completion:*** Compute the Euclidean norms of the differences between the mean vectors in the current and previous steps. Compute the residual error, E, as the sum of the k norms. Stop if $E \leq T$, where T a specified, nonnegative threshold. Else, go back to Step 2.

Clustering for Segmentation

a b

FIGURE 10.49
(a) Image of size 688×688 pixels.
(b) Image segmented using the k-means algorithm with $k = 3$.

K-means Clustering for Segmentation



(a) (b)

https://ieeexplore.ieee.org/document/1633722

Active Contour

- Kass et al. (1988): Snake
  - A curve that is fitted to the data while retaining certain smoothness properties

- Find a contour that separates (part of) an object from the background

- Utilizes the gradient

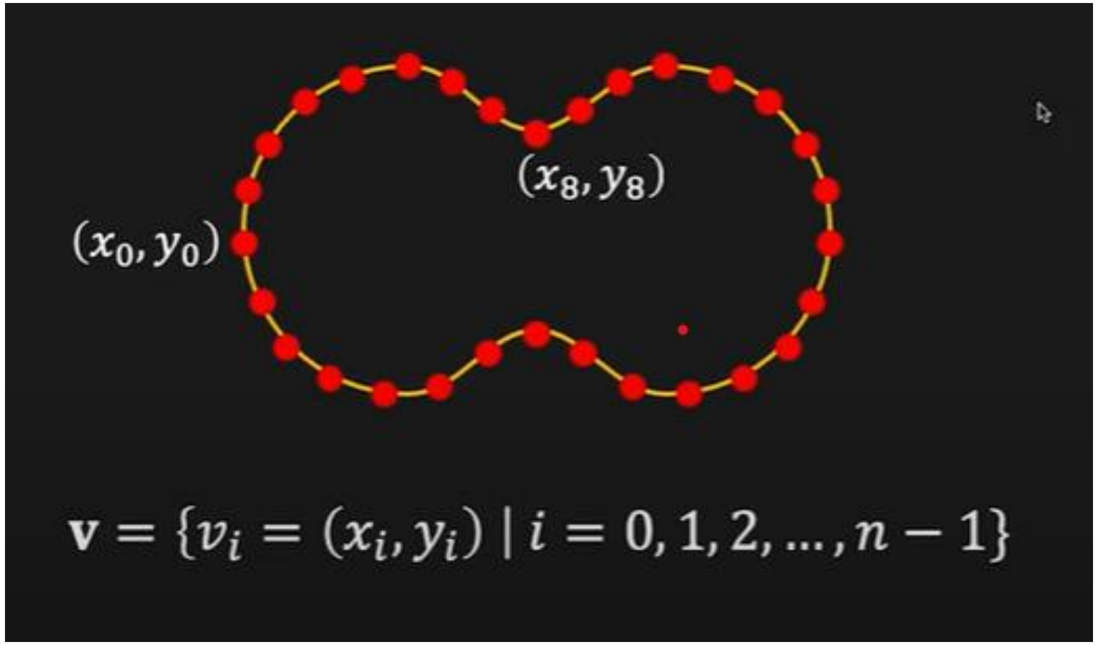
Active Contour

- Given: an approximate boundary of an object
- Task: Evolve the contour to fit the exact boundary

- Active contour
  - Iteratively deform the initial contour so that
    - The contour moves near pixels with high gradient (edges)
    - It is smooth

Active Contour

- Contour v: an ordered list of 2D vertices (control/ contour points) connected through straight lines



https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

- Apply forces

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

- Apply forces

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

- We try to bring the contour to a position so that gradient magnitude squared is maximized

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

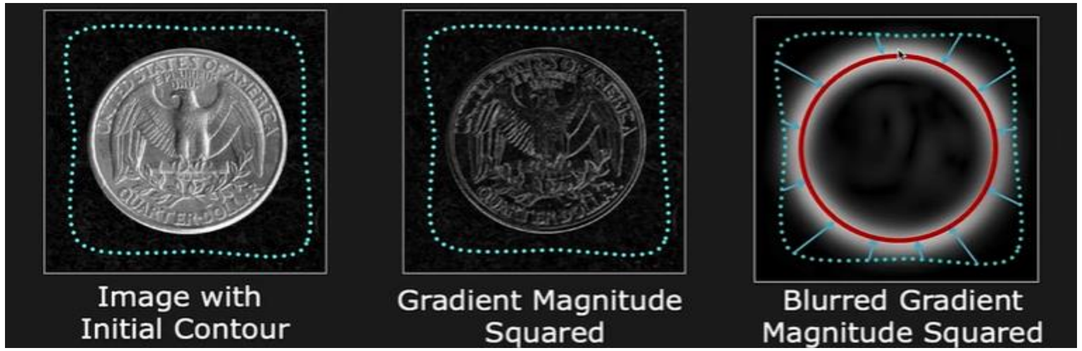
Attracting Contour to Edges

- We try to bring the contour to a position so that sum of gradient magnitude squared is maximized

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

- We try to bring the contour to a position so that sum of gradient magnitude squared is maximized
- Minimize –ve of sum of gradient magnitude squared
- Internal energy or image energy

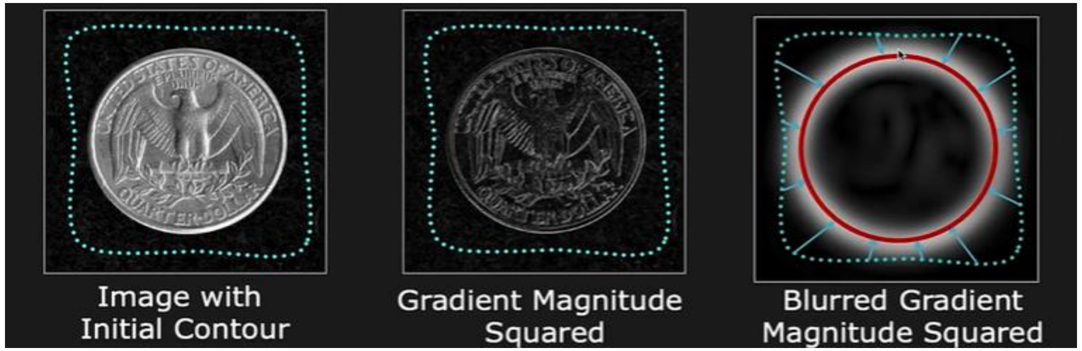


https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

- Internal energy or image energy
- Option 1
  - $E_{img} = -\sum_{i=0}^{n-1} \left|\left| w(v_i) * \nabla I(v_i) \right|\right|^2$
- Option 2
  - $E_{img} = -\sum_{i=0}^{n-1} \left|\left| \nabla I(v_i) \right|\right|^2$
- Other options may also be possible

$w(v_i)$: Smoothing kernel



https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Contour Deformation: Greedy Algorithm

- For each contour point v_i, move v_i withing a window A where E_{img} is minimum

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Contour Deformation: Greedy Algorithm

- For each contour point v_i, move v_i withing a window A where E_{img} is minimum

- Do it for each contour point

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Contour Deformation: Greedy Algorithm

- For each contour point v_i, move v_i withing a window A where E_{img} is minimum

- Do it for each contour point

- Repeat the above steps until the sum of movement of all contour points is below a threshold

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

Contour fitted to gradient magnitude

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

Contour fitted to gradient magnitude

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

Contour fitted to gradient magnitude

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Attracting Contour to Edges

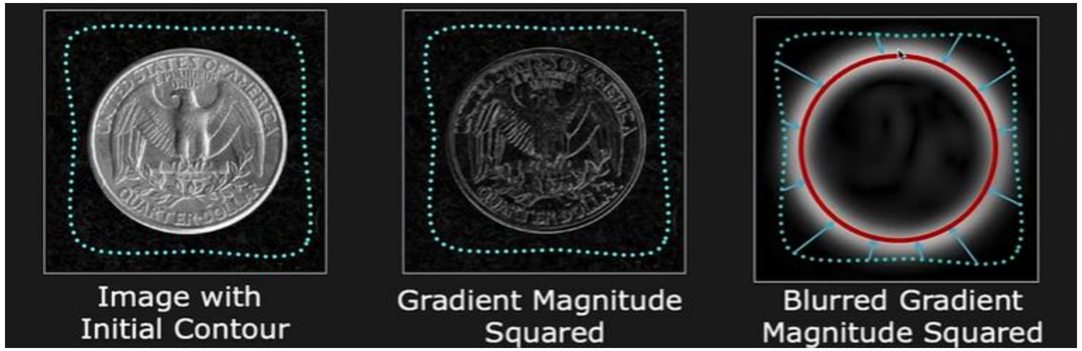
Contour fitted to gradient magnitude

Some non idealities towards the end because of noise

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

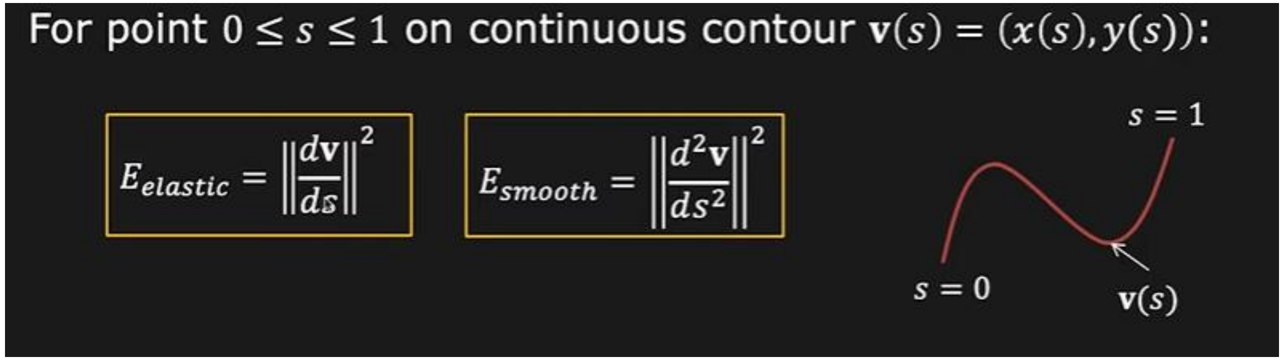
Solution

- Add more constraints on the shape of the contour
- Contour should evolve in such a way that it remains smooth
  - Smoothness
- The contour behaves like a rubber band
  - Elasticity
- Internal energy of the contour



https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Internal Energy of Contour



- Contour should evolve in such a way that it remains smooth
  - Smoothness

- The contour behaves like a rubber band
  - Elasticity

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Internal Energy of Contour

Discrete approximations at control point $\mathbf{v}_i$:

$$E_{elastic}(\mathbf{v}_i) = \left\|\frac{d\mathbf{v}}{ds}\right\|^2 \approx \|\mathbf{v}_{i+1} - \mathbf{v}_i\|^2 = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2$$

$$E_{smooth}(\mathbf{v}_i) = \left\|\frac{d^2\mathbf{v}}{ds^2}\right\|^2 \approx \|(\mathbf{v}_{i+1} - \mathbf{v}_i) - (\mathbf{v}_i - \mathbf{v}_{i-1})\|^2$$

- Contour should evolve in such a way that it remains smooth
  - Smoothness

- The contour behaves like a rubber band
  - Elasticity

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Internal Energy of Contour

$$E_{elastic} = \sum_{i=0}^{n-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]$$

$$E_{smooth} = \sum_{i=0}^{n-1} [(x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2]$$

- Contour should evolve in such a way that it remains smooth
  - Smoothness

- The contour behaves like a rubber band
  - Elasticity

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Internal Energy of Contour

$$E_{elastic} = \sum_{i=0}^{n-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]$$

$$E_{smooth} = \sum_{i=0}^{n-1} [(x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2]$$

- Contour/ internal energy
- $E_{contour} = w_1 E_{elastic} + w_2 E_{smooth}$

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Total Energy of Contour

- $E_{total} = E_{image} + E_{contour}$
- Minimize the total energy

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

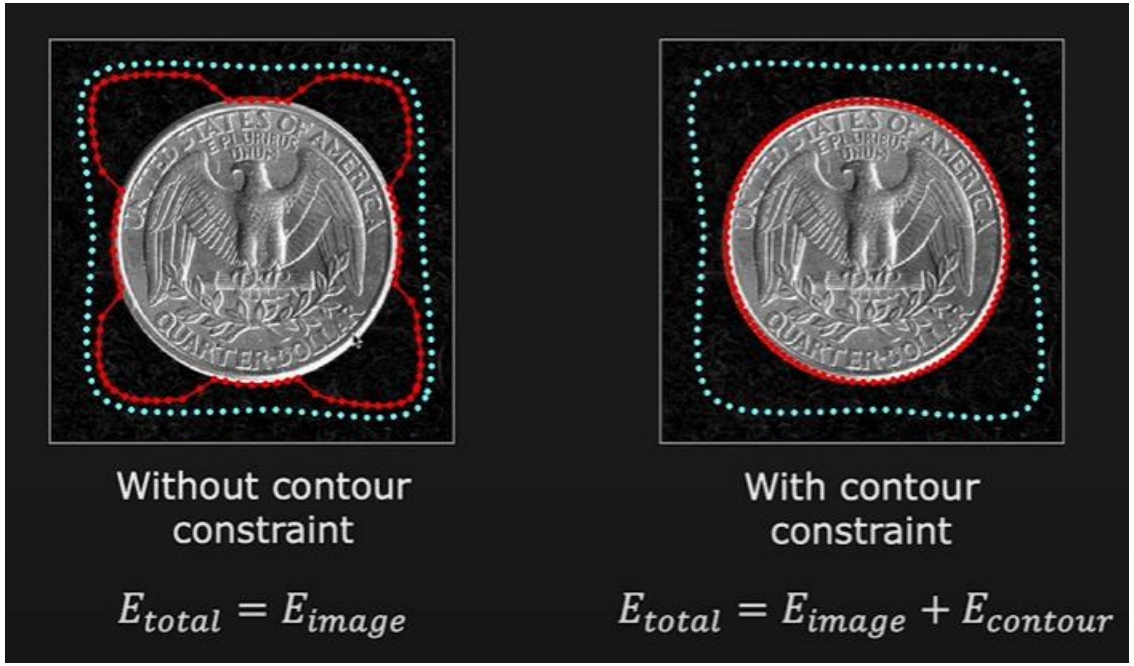
Greedy Algorithm

- Uniformly sample the contour to get n contour points
- For each contour point v_i, move v_i withing a window A where E_{total} is minimum
- Do it for each contour point
- Repeat the above steps until the sum of movement of all contour points is below a threshold

https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Greedy Algorithm

- Uniformly sample the contour to get n contour points
- For each contour point v_i, move v_i withing a window A where E_{total} is minimum
- Do it for each contour point
- Repeat the above steps until the sum of movement of all contour points is below a threshold



https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Iterative Solver

$$E_{contour} = w_1 E_{elastic} + w_2 E_{smooth}$$

$$\mathbf{A} = \begin{pmatrix} -2w_1 + 6w_2 & w_1 - 4w_2 & w_2 & \dots & 0 \\ w_1 - 4w_2 & -2w_1 + 6w_2 & w_1 - 4w_2 & & \dots \\ w_2 & w_1 - 4w_2 & -2w_1 + 6w_2 & & w_2 \\ \dots & \dots & \dots & \dots & w_1 - 4w_2 \\ 0 & 0 & 0 & \dots & -2w_1 + 6w_2 \end{pmatrix}$$

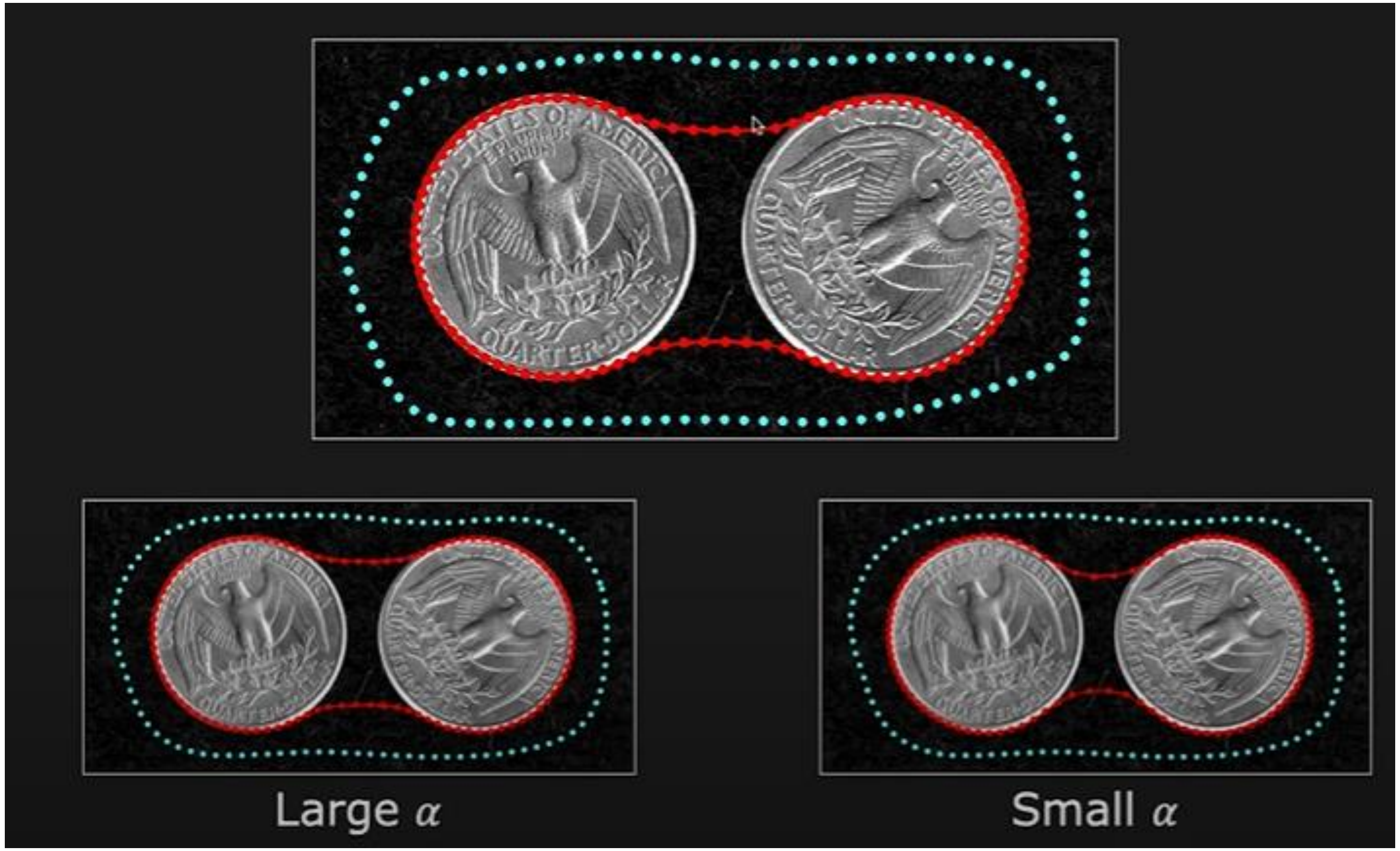
Iterative Solver

$$\mathbf{x}^{(t+1)} = (\mathbf{A} - \gamma\mathbf{I})^{-1}\left(\mathbf{x}^{(t)} - \frac{\partial E(\mathbf{s}^{(t)})}{\partial x}\right)$$

$$\mathbf{y}^{(t+1)} = (\mathbf{A} - \gamma\mathbf{I})^{-1}\left(\mathbf{y}^{(t)} - \frac{\partial E(\mathbf{s}^{(t)})}{\partial y}\right)$$

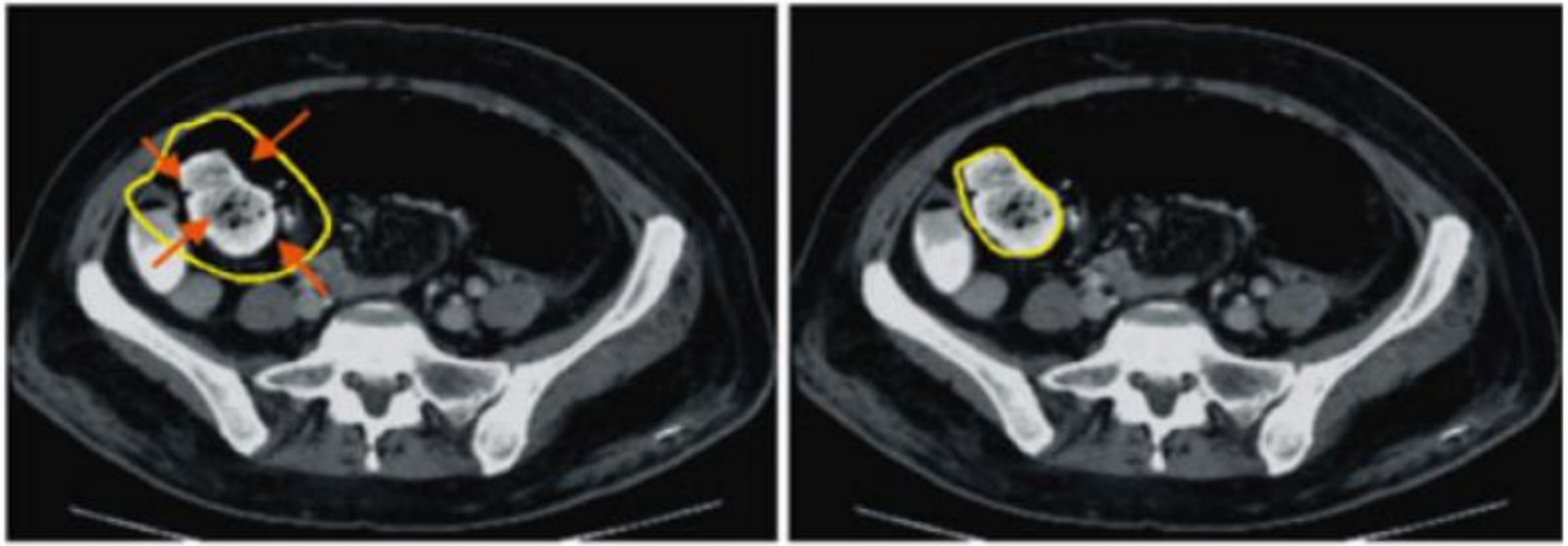
γ: Model parameter

Greedy Algorithm



https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Example



Alternate Options

- Add more constraints
  - Shape constraint
  - Penalizes deviation from a pre-defined shape
- Changes in initialization
- Replace contracting force (elasticity) with ballooning force to expand
  - Initialization within the object

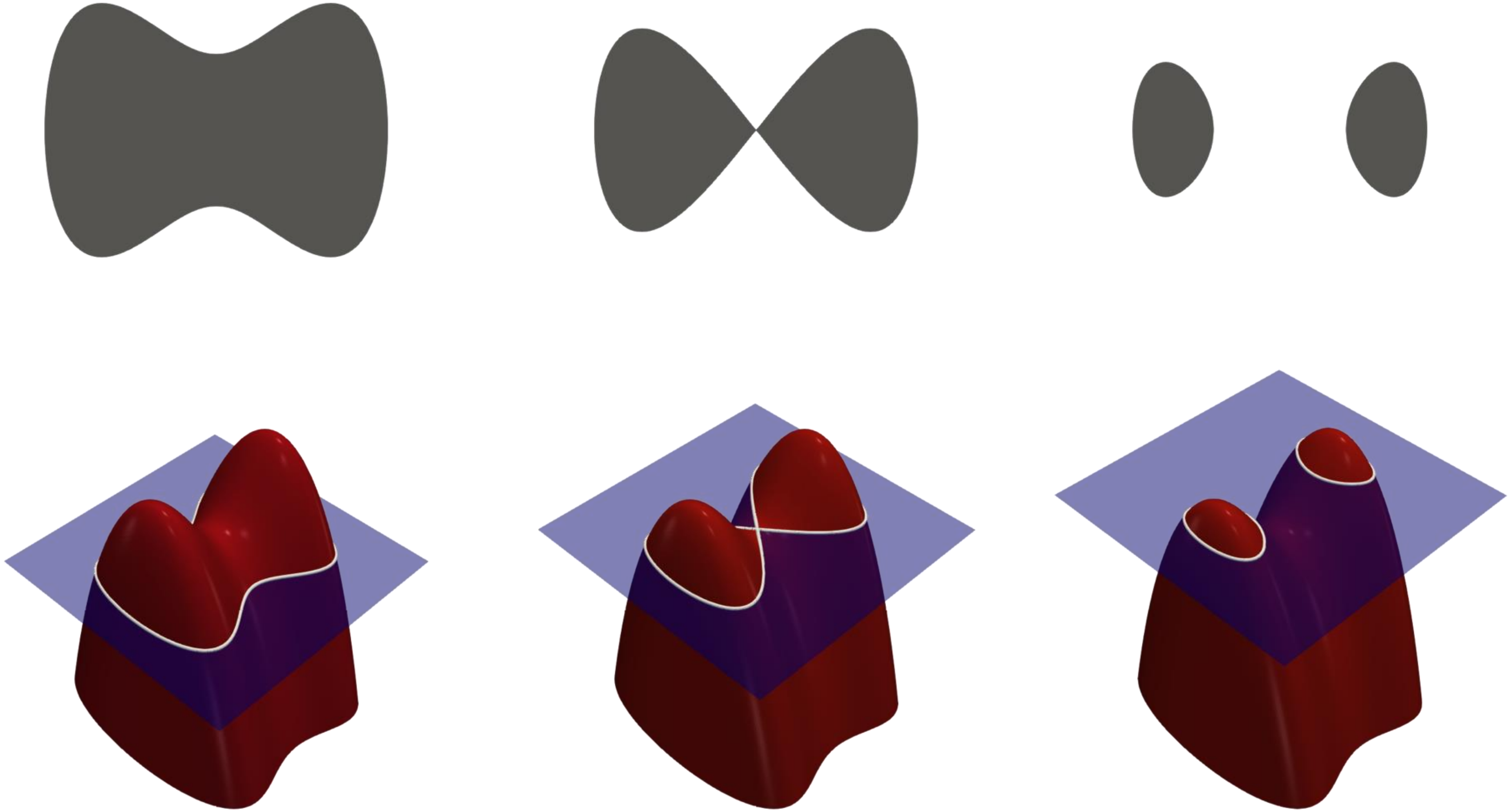
https://www.youtube.com/watch?v=FROJUMk9P3Y&t=412s

Drawbacks of Active Contour

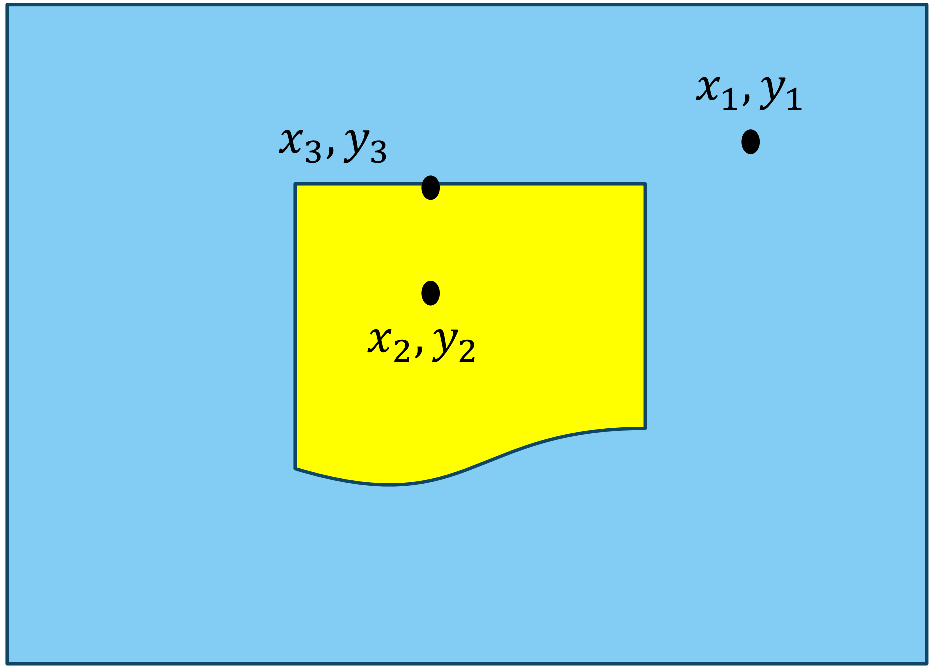
- Difficult to keep track of control points
- Can’t wrap around multiple objects at once
- Can’t deal with objects with holes

Level Sets

- Does not require parameterization

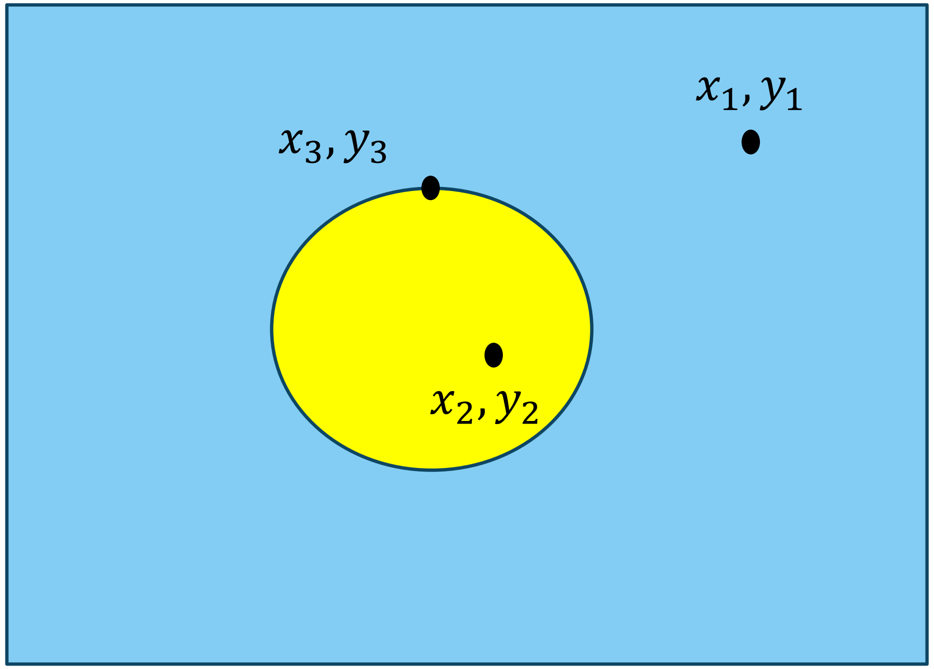


Level Sets



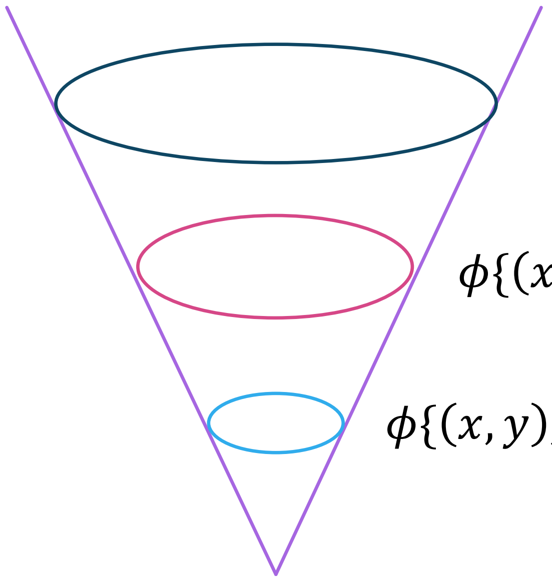
- Suppose, I define a function $\phi(x, y)$ such that
  - $\phi(x, y) > 0$ if x, y lies in the blue region
  - $\phi(x, y) < 0$ if x, y lies in the yellow region
  - $\phi(x, y) = 0$ if x, y lies in the edge

Level Sets



- Suppose, $\phi(x, y) = x^2 + y^2 - r^2$ (r is the radius of the circle)
  - $\phi(x, y) > 0$ if x, y lies in the blue region
  - $\phi(x, y) < 0$ if x, y lies in the yellow region
  - $\phi(x, y) = 0$ if x, y lies in the edge

Level Sets



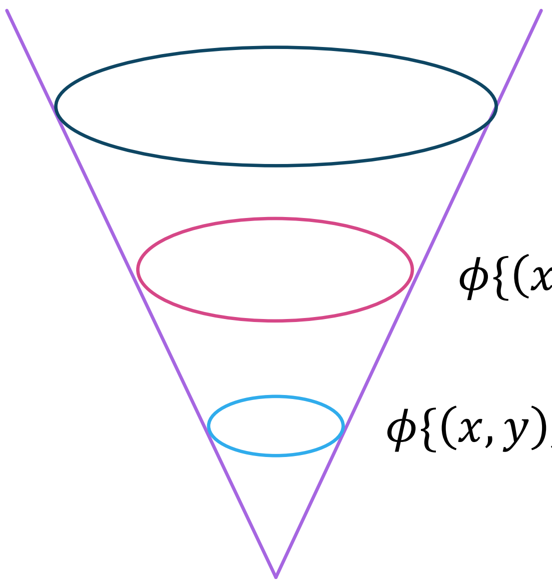
$$\phi\{(x,y),1\} = x^2 + y^2 - r_1^2$$

$$\phi\{(x,y),2\} = x^2 + y^2 - r_2^2$$

$$\phi\{(x,y),3\} = x^2 + y^2 - r_3^2$$

- Suppose, $\phi(x,y) = x^2 + y^2 - r^2$ (r is the radius of the circle)
  - $\phi(x,y) > 0$ if x, y lies in the blue region
  - $\phi(x,y) < 0$ if x, y lies in the yellow region
  - $\phi(x,y) = 0$ if x, y lies in the edge

Level Sets



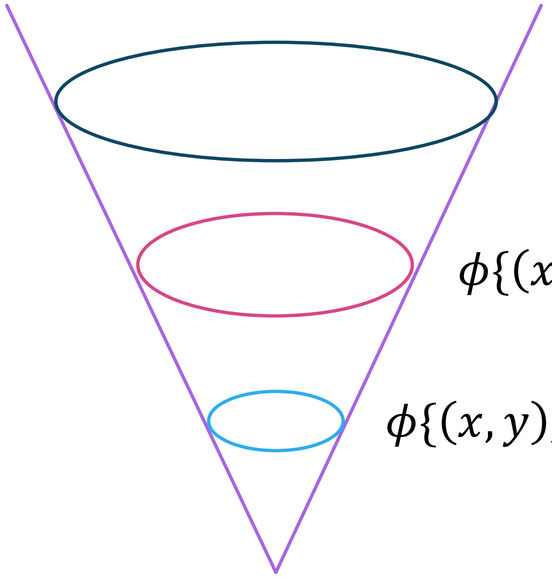
$$\phi\{(x,y),1\} = x^2 + y^2 - r_1^2$$

$$\phi\{(x,y),2\} = x^2 + y^2 - r_2^2$$

$$\phi\{(x,y),3\} = x^2 + y^2 - r_3^2$$

$$\boldsymbol{\phi\{(x,y),t\} = x^2 + y^2 - r_t^2}$$

Level Sets



$\phi\{(x,y),1\} = x^2 + y^2 - r_1^2$

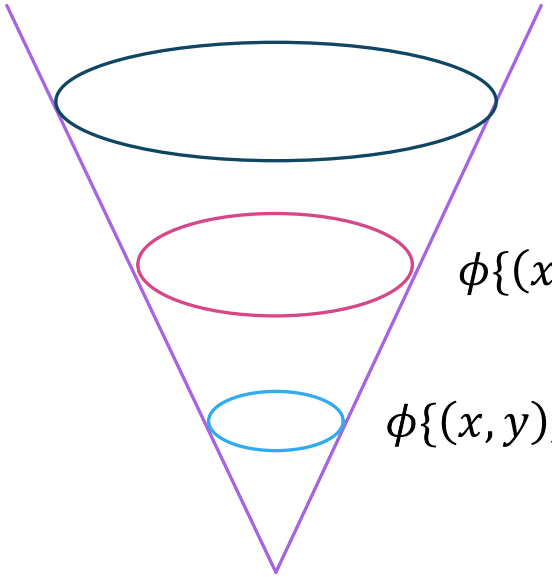
$\phi\{(x,y),2\} = x^2 + y^2 - r_2^2$

$\phi\{(x,y),3\} = x^2 + y^2 - r_3^2$

$$\boldsymbol{\phi\{(x,y),t\} = x^2 + y^2 - r_t^2}$$

Moving up or down over time results in evolution of $\boldsymbol{\phi(x,y)}$ over time

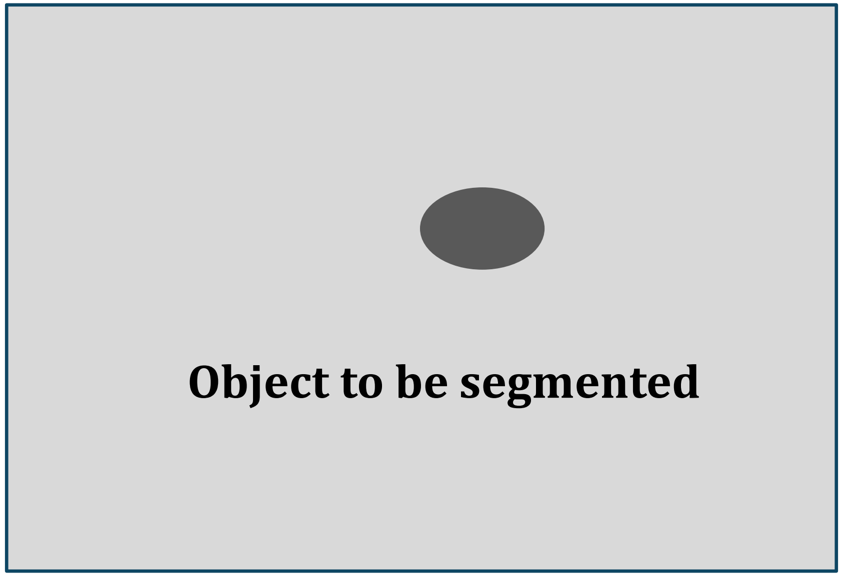
Level Sets for Image Segmentation



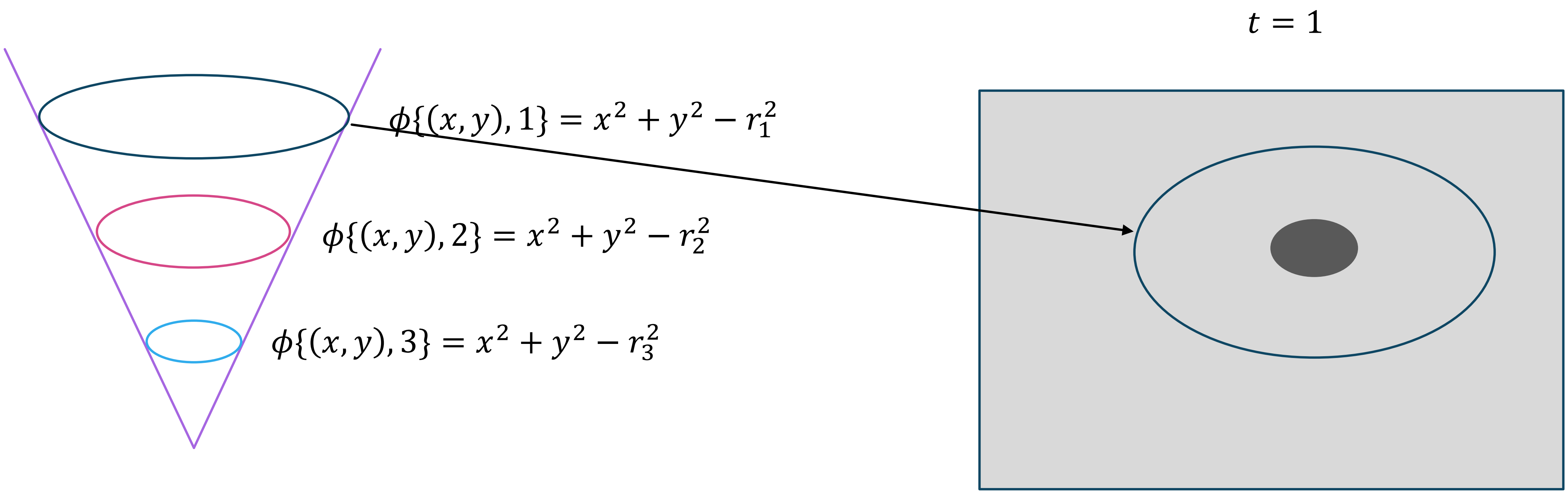
$$\phi\{(x,y),1\} = x^2 + y^2 - r_1^2$$

$$\phi\{(x,y),2\} = x^2 + y^2 - r_2^2$$

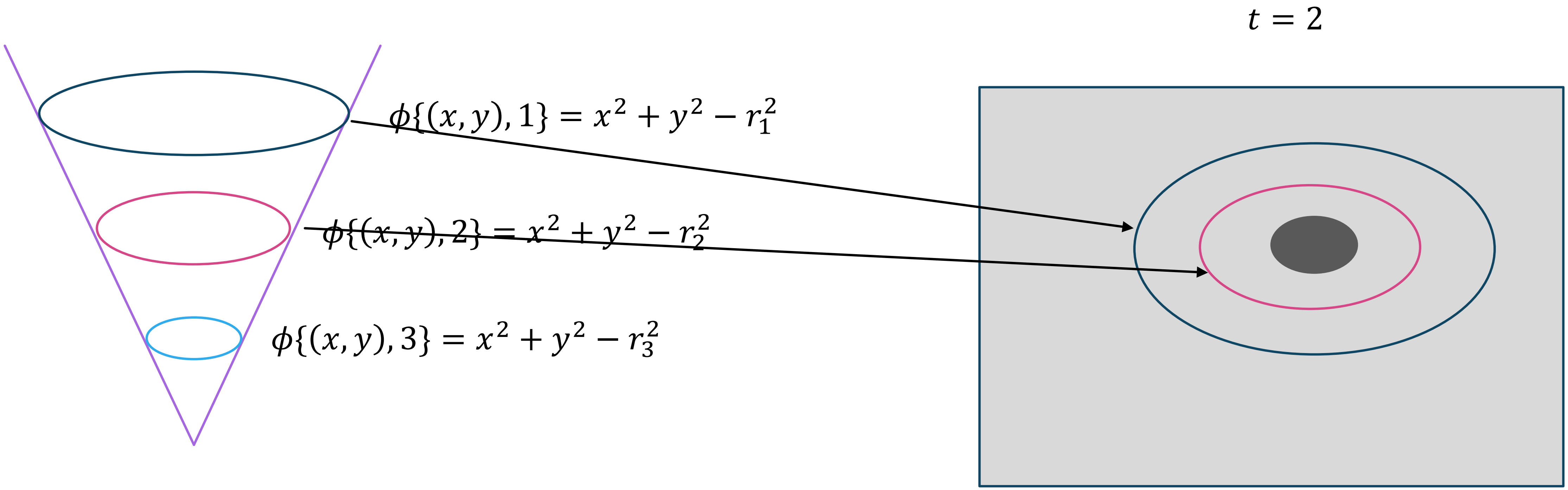
$$\phi\{(x,y),3\} = x^2 + y^2 - r_3^2$$



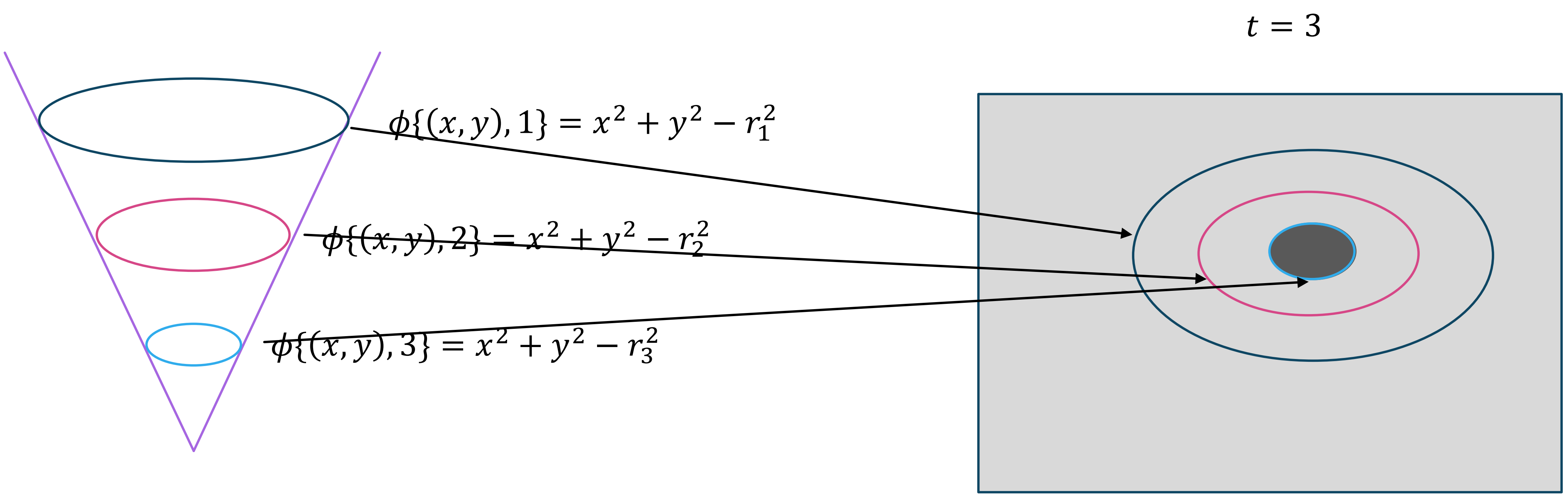
Level Sets for Image Segmentation



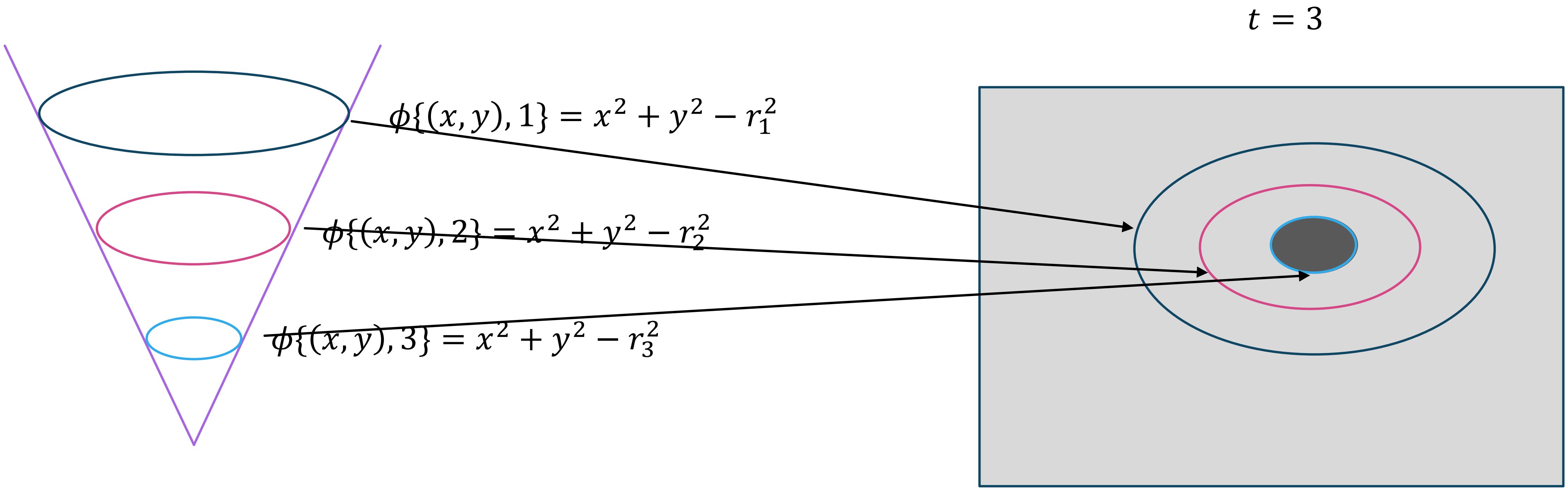
Level Sets for Image Segmentation



Level Sets for Image Segmentation

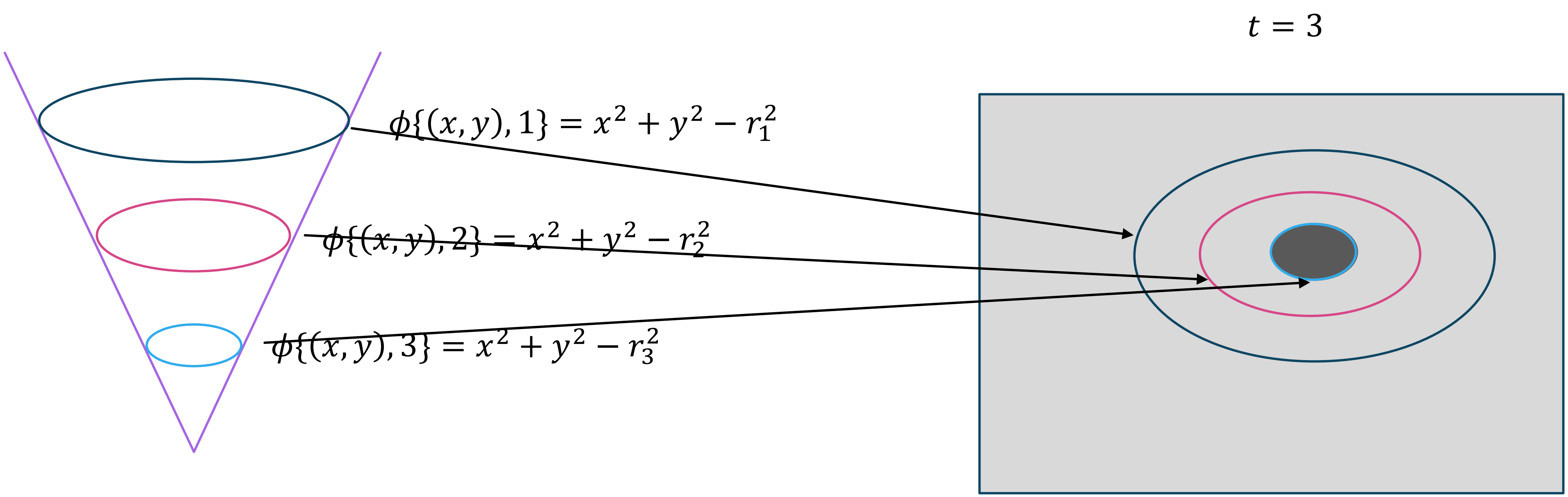


Level Sets for Image Segmentation



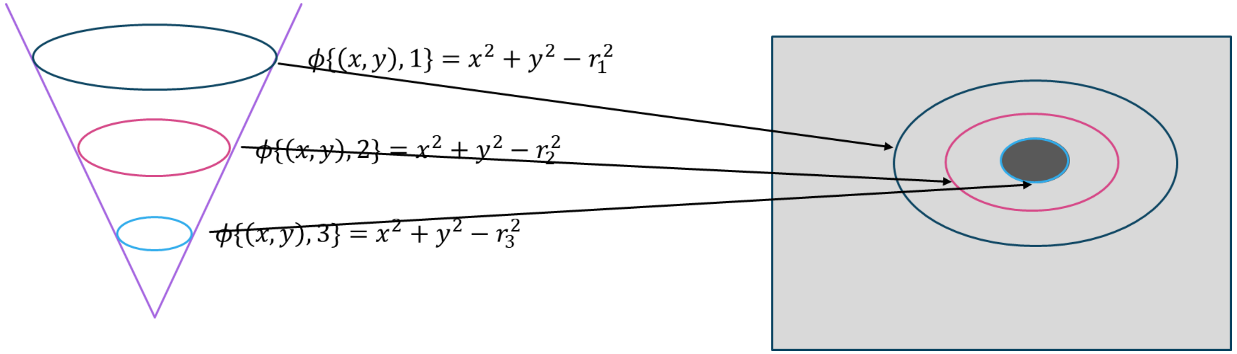
At time t, the points where $\phi\{(x, y), t\} = 0$ defines a contour $C(t)$

Level Sets for Image Segmentation



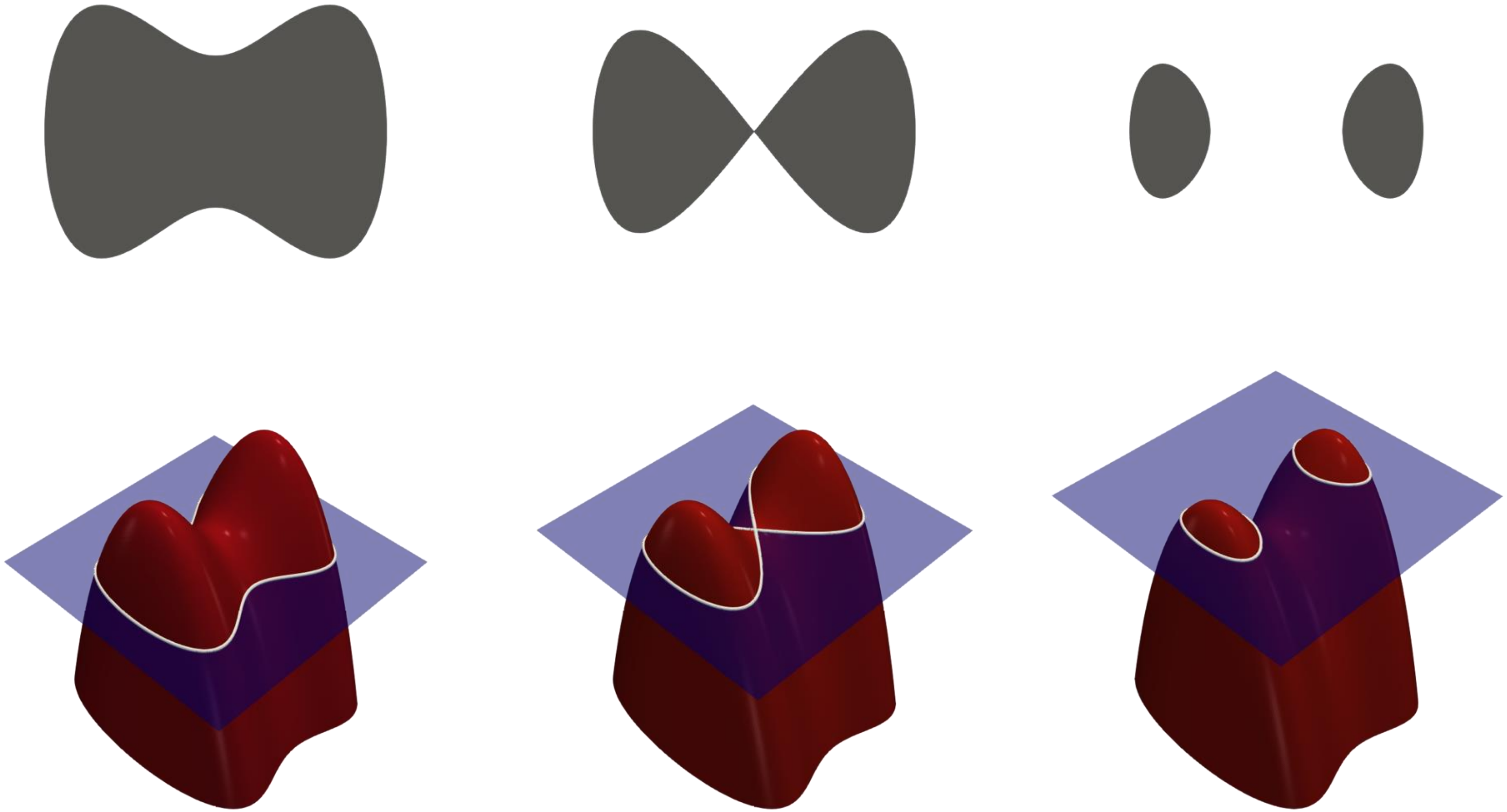
As $\phi\{(x,y),t\}$ evolves, $C(t)$ also evolves; at certain point of evolution the points inside and on $C(t)$ may define the segmented object

Level Sets for Image Segmentation

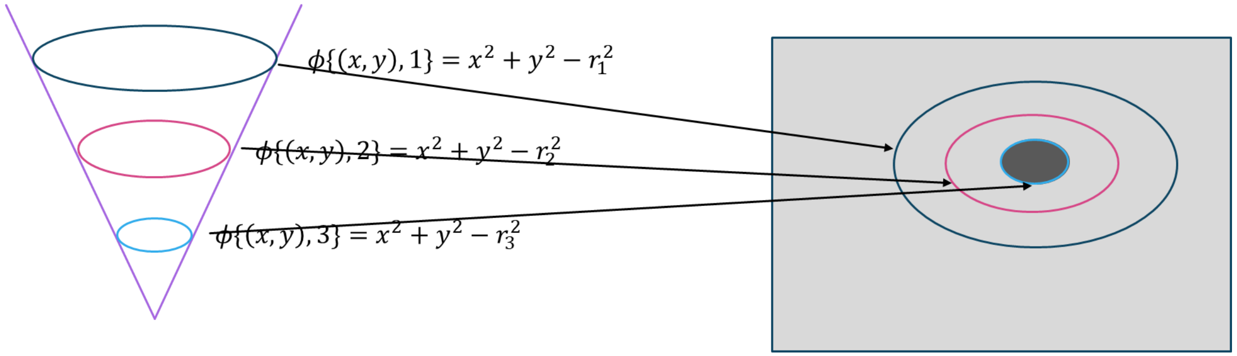


- Define a suitable $\phi\{(x,y),t\}$
- Define a suitable energy $E(t)$ in terms of $\phi\{(x,y),t\}$
- Minimize $E(t)$ through evolution of $\phi\{(x,y),t\}$ over time
- Let's say $E(t)$ is minimized at $t = t^*$
- Find out contour $C(t^*)$ using the points for which $\phi\{(x,y),t^*\} = 0$
- Points on and inside $C(t^*)$ represents the points of segmented objects

Level Sets for Image Segmentation



Level Set Function



- $\phi\{(x,y),t\}$: Level set function
- $\phi\{(x,y),t\} = \begin{cases} d\{(x,y),C(t)\} & if\ (x,y)\ is\ inside\ C(t) \\ 0 & if\ (x,y)\ is\ on\ C(t) \\ -d\{(x,y),C(t)\} & if\ (x,y)\ is\ outside\ C(t) \end{cases}$
- Signed distance function

Chan-Vese Model

- $\phi\{(x, y), t\}$: Level set function

- Trick: A new way to define $E(t)$ based on $\phi\{(x, y), t\}$

Chan-Vese Model

- $\phi\{(x, y), t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x, y), t\}$

- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) +$

  $\lambda_1 \int_{inside\ C(t)} |I(x, y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x, y) - m_2(t)|^2 dx\ dy$
  - $I(x, y)$: Intensity of pixel (x, y)
  - $m_1(t)$: Mean intensity of region inside $C(t)$ [mean computed at time t]
  - $m_2(t)$: Mean intensity of region outside $C(t)$ [mean computed at time t]

- Goal: minimize$E\big(m_1(t), m_2(t), C(t)\big)$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x,y),t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x,y),t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x,y) - m_2(t)|^2 dx\ dy$
- $Length\big(C(t)\big) = \int_\Omega |\nabla H(\phi\{(x,y),t\})|\ dx\ dy = \int_\Omega \delta_0(\phi\{(x,y),t\})|\nabla(\phi\{(x,y),t\})|\ dx\ dy$
  - $H(z) = \begin{cases} 1, & if\ z \geq 0 \\ 0, & otherwise \end{cases}$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x, y), t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x, y), t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x, y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x, y) - m_2(t)|^2 dx\ dy$
- $Area\big(inside\ C(t)\big) = \int_\Omega H(\phi\{(x, y), t\})\ dx\ dy$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x,y),t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x,y),t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x,y) - m_2(t)|^2 dx\ dy$
- $\int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy = \int_{(x,y):\phi\{(x,y),t\}>0} |I(x,y) - m_1(t)|^2 H(\phi\{(x,y),t\}) dx\ dy$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x,y),t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x,y),t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x,y) - m_2(t)|^2 dx\ dy$
- $\int_{outside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy = \int_{(x,y):\phi\{(x,y),t\}<0} |I(x,y) - m_1(t)|^2 H(1 - \phi\{(x,y),t\}) dx\ dy$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x,y),t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x,y),t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x,y) - m_2(t)|^2 dx\ dy$
- $m_1(t) = \dfrac{\int_{(x,y):\phi\{(x,y),t\}>0} I(x,y) H(\phi\{(x,y),t\}) dx\ dy}{\int_{(x,y):\phi\{(x,y),t\}>0} H(\phi\{(x,y),t\}) dx\ dy}$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x,y),t\}$: Level set function
- Trick: A new way to define $E(t)$ based on $\phi\{(x,y),t\}$
- $E\big(m_1(t), m_2(t), C(t)\big) = \alpha\ Length\big(C(t)\big) + \beta\ Area\big(inside\ C(t)\big) + \lambda_1 \int_{inside\ C(t)} |I(x,y) - m_1(t)|^2 dx\ dy + \lambda_2 \int_{outside\ C(t)} |I(x,y) - m_2(t)|^2 dx\ dy$
- $m_2(t) = \frac{\int_{(x,y):\phi\{(x,y),t\}<0} I(x,y) H(1-\phi\{(x,y),t\}) dx\ dy}{\int_{(x,y):\phi\{(x,y),t\}<0} H(1-\phi\{(x,y),t\}) dx\ dy}$

Chan-Vese Model in Terms of Level Set

- $\phi\{(x, y), t\}$: Level set function
- $E\big(m_1(t), m_2(t), C(t)\big) = f(\phi\{(x, y), t\})$
- $\phi\{C(t), t\} = 0 \;\; \forall t$
  - $0 = \frac{d}{dt}\phi\{C(t), t\} = \nabla\phi\{C(t), t\}.\frac{dC(t)}{dt} + \frac{\partial\phi\{C(t),t\}}{\partial t}$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = -\nabla\phi\{C(t), t\}.\frac{dC(t)}{dt}$

Chan-Vese Model in Terms of Level Set

- $\phi\{C(t), t\} = 0 \;\; \forall t$
  - $0 = \frac{d}{dt}\phi\{C(t), t\} = \nabla\phi\{C(t), t\}.\frac{dC(t)}{dt} + \frac{\partial\phi\{C(t),t\}}{\partial t}$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = -\nabla\phi\{C(t), t\}.\frac{dC(t)}{dt}$
  - $\frac{dC(t)}{dt} = V\hat{n} = -V\frac{\nabla\phi\{C(t),t\}}{|\nabla\phi\{C(t),t\}|}$ $\quad V: speed\ towards\ normal$

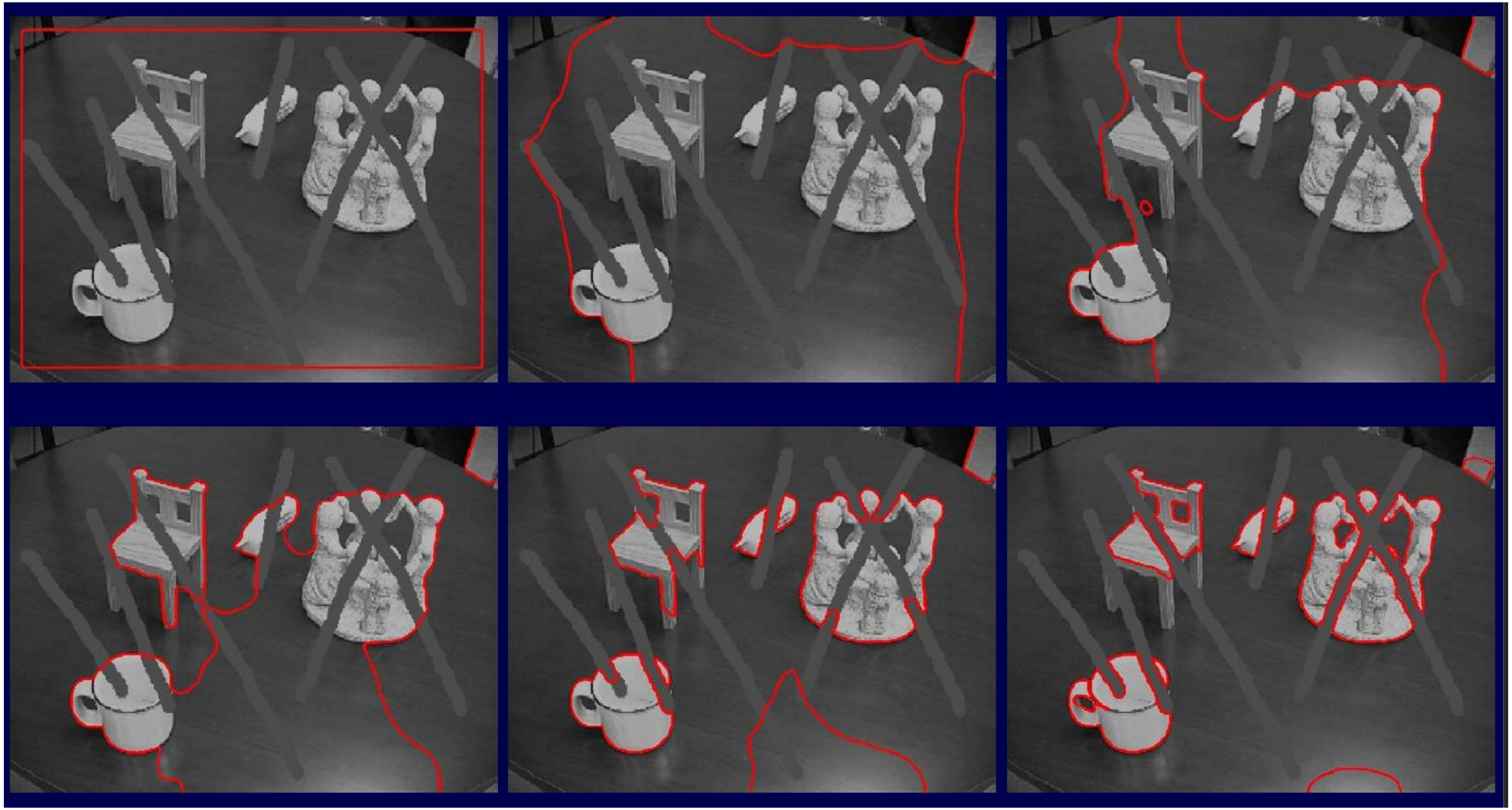
Chan-Vese Model in Terms of Level Set

- $\phi\{C(t), t\} = 0 \;\; \forall t$
  - $0 = \frac{d}{dt}\phi\{C(t), t\} = \nabla\phi\{C(t), t\}.\frac{dC(t)}{dt} + \frac{\partial\phi\{C(t),t\}}{\partial t}$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = -\nabla\phi\{C(t), t\}.\frac{dC(t)}{dt}$
  - $\frac{dC(t)}{dt} = V\hat{n} = -V\frac{\nabla\phi\{C(t),t\}}{|\nabla\phi\{C(t),t\}|}$ $\quad V: speed\ towards\ normal$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = V\frac{|\nabla\phi\{C(t),t\}|^2}{|\nabla\phi\{C(t),t\}|} = V|\nabla\phi\{C(t), t\}|$

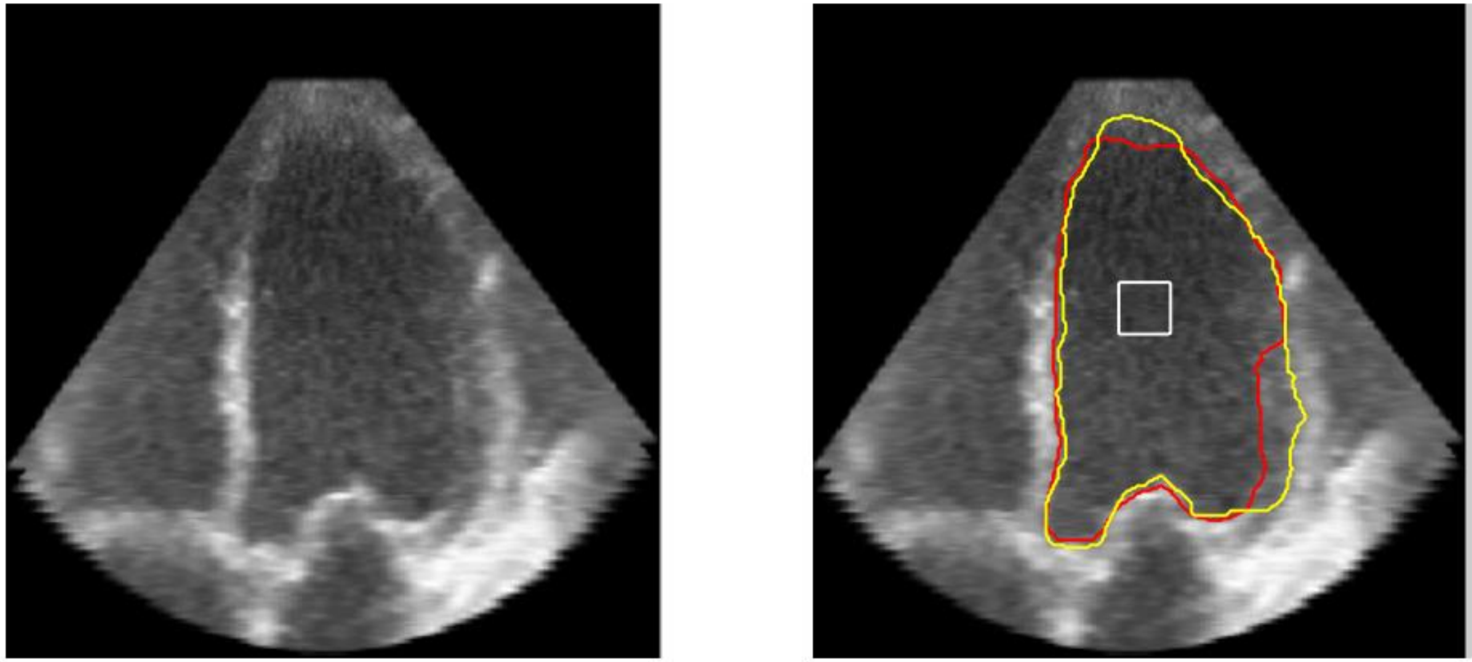
Chan-Vese Model in Terms of Level Set

- $\phi\{C(t), t\} = 0 \ \ \forall t$
  - $0 = \frac{d}{dt}\phi\{C(t), t\} = \nabla\phi\{C(t), t\}.\frac{dC(t)}{dt} + \frac{\partial\phi\{C(t),t\}}{\partial t}$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = -\nabla\phi\{C(t), t\}.\frac{dC(t)}{dt}$
  - $\frac{dC(t)}{dt} = V\hat{n} = -V\frac{\nabla\phi\{C(t),t\}}{|\nabla\phi\{C(t),t\}|}$ $\quad V: speed\ towards\ normal$
  - $\frac{\partial\phi\{C(t),t\}}{\partial t} = V\frac{|\nabla\phi\{C(t),t\}|^2}{|\nabla\phi\{C(t),t\}|} = V|\nabla\phi\{C(t), t\}|$ $\quad$ Level set equation

Chan-Vese Model



Level Set for Segmentation



(a) (b)

Fig. 10. (a) Ultrasound image of the left ventricle of the heart. (b) Contours obtained by PBLS (yellow) and our proposed method (red). The white curve denotes the initial contour.

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7847297

How to Measure the Quality of Segmentation

- Pixel accuracy

- Dice Coefficient $= 2\frac{|A \cap B|}{|A|+|B|}$

- Jaccard Index $= \frac{|A \cap B|}{|A \cup B|}$

Presentation

- Random walk for image segmentation

Image Registration

Image Registration

- Aligning two or more image so that they fit in the best possible way

- How to define fitting?
  - Based on position of landmark points
  - Other options

Steps for Image Registration

- Feature detection/ land mark identification/ landmark annotation
  - The landmarks/ features to be matched are identified/ detected
- Feature/ landmark matching
  - Establishing correspondences between landmarks/ features
- Estimation of transformation
- Transformation of template image

Steps for Image Registration

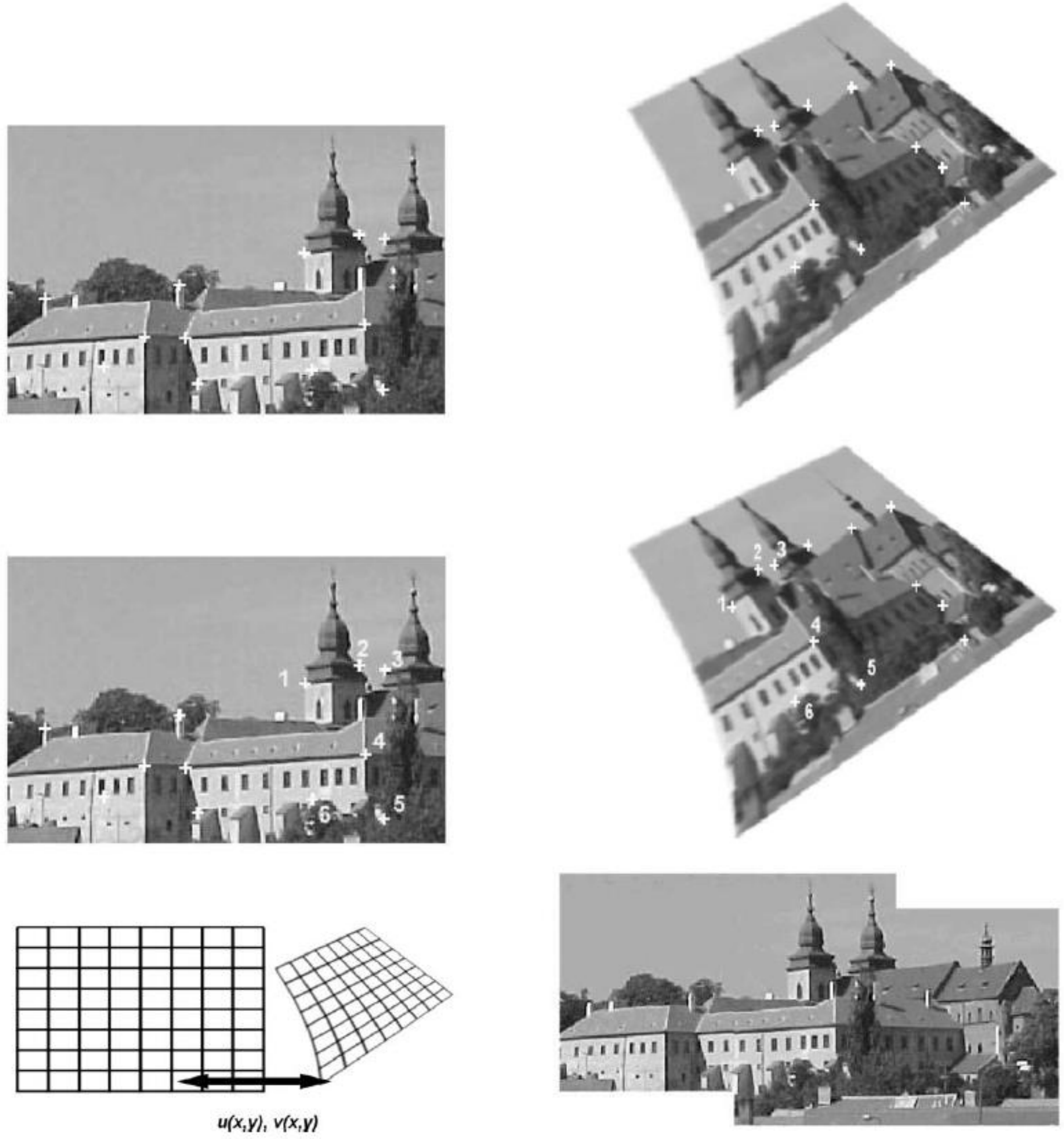


Fig. 1. Four steps of image registration: top row—feature detection (corners were used as the features in this case). Middle row—feature matching by invariant descriptors (the corresponding pairs are marked by numbers). Bottom left—transform model estimation exploiting the established correspondence. Bottom right—image resampling and transformation using appropriate interpolation technique.

Landmark-based Registration

- A set of landmark points
  - Manually or automatically chosen
- Anatomical landmark
  - A landmark assigned by an expert that corresponds between objects in a biologically meaningful way
- Mathematical landmark
  - A landmark that is located on a curve according to some mathematical or geometrical property (e.g., a point of maximum curvature).
- Pseudo landmark
  - A landmark that is constructed on a curve based on anatomical or mathematical landmarks (e.g. sampled equidistantly along an outline).

Landmark-based Registration

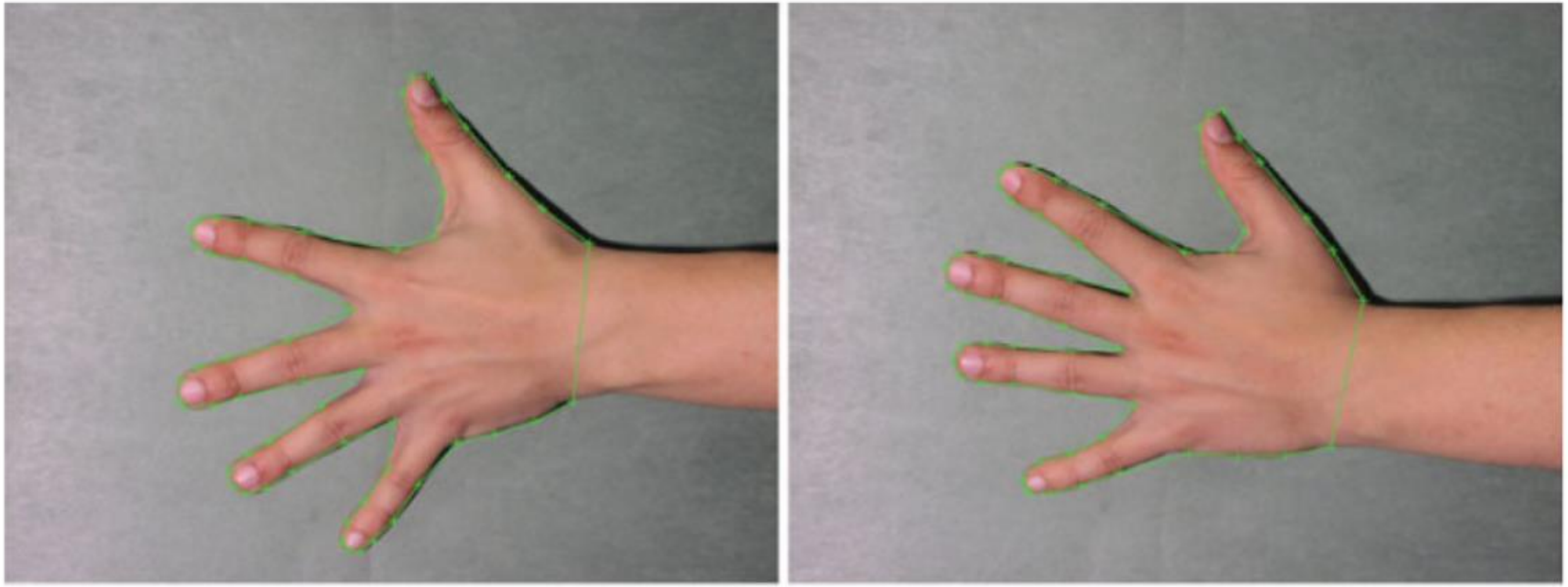


Fig. 11.2 Two images of the same hand. Each hand is annotated with 56 landmarks. Some landmarks are anatomical landmarks placed at the finger joints, others are mathematical landmarks placed at points of maximum surface curvatures, and a few of the landmarks are pseudo landmarks placed by sampling part of the curves equidistantly

Landmark-based Registration

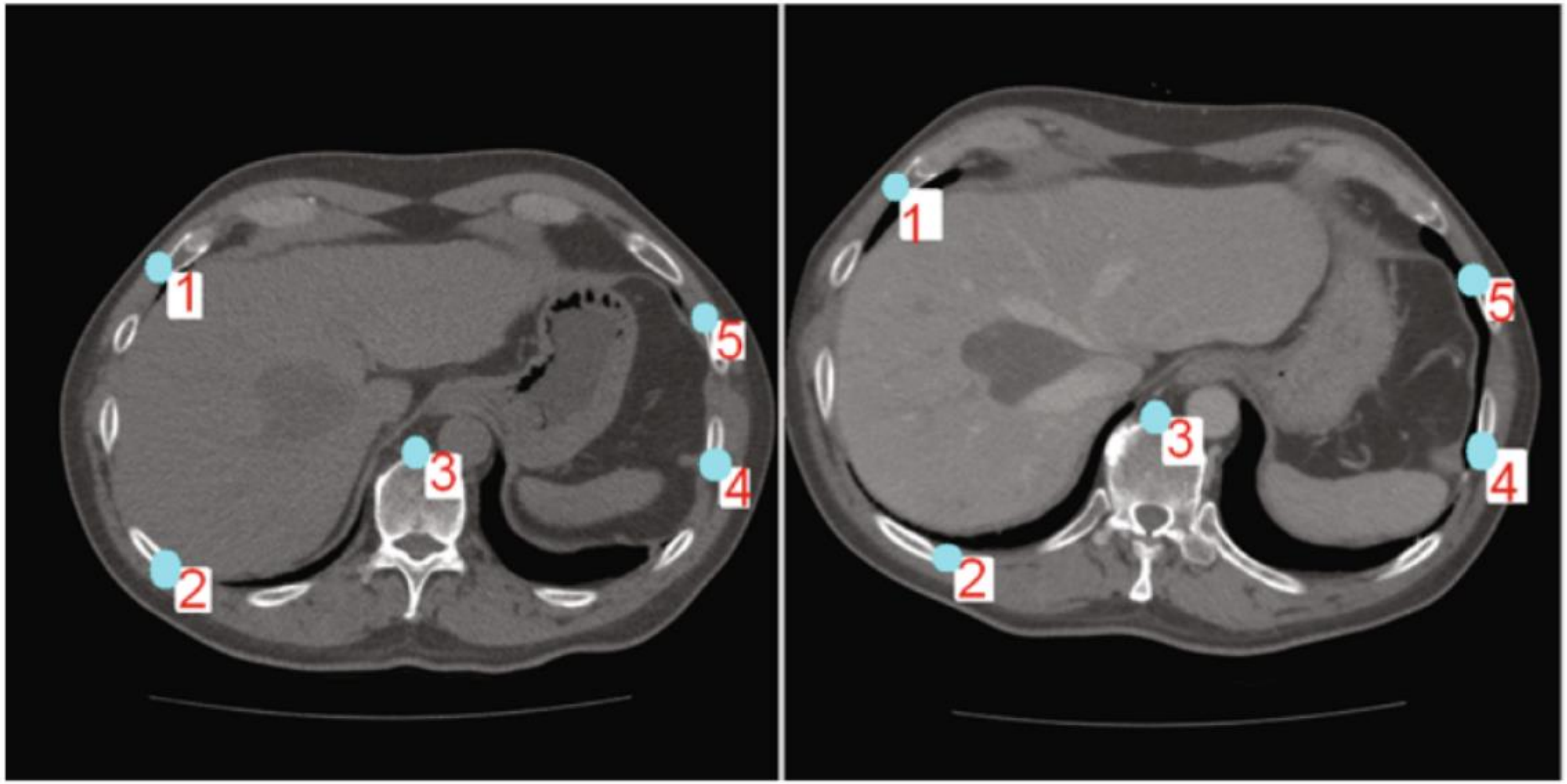
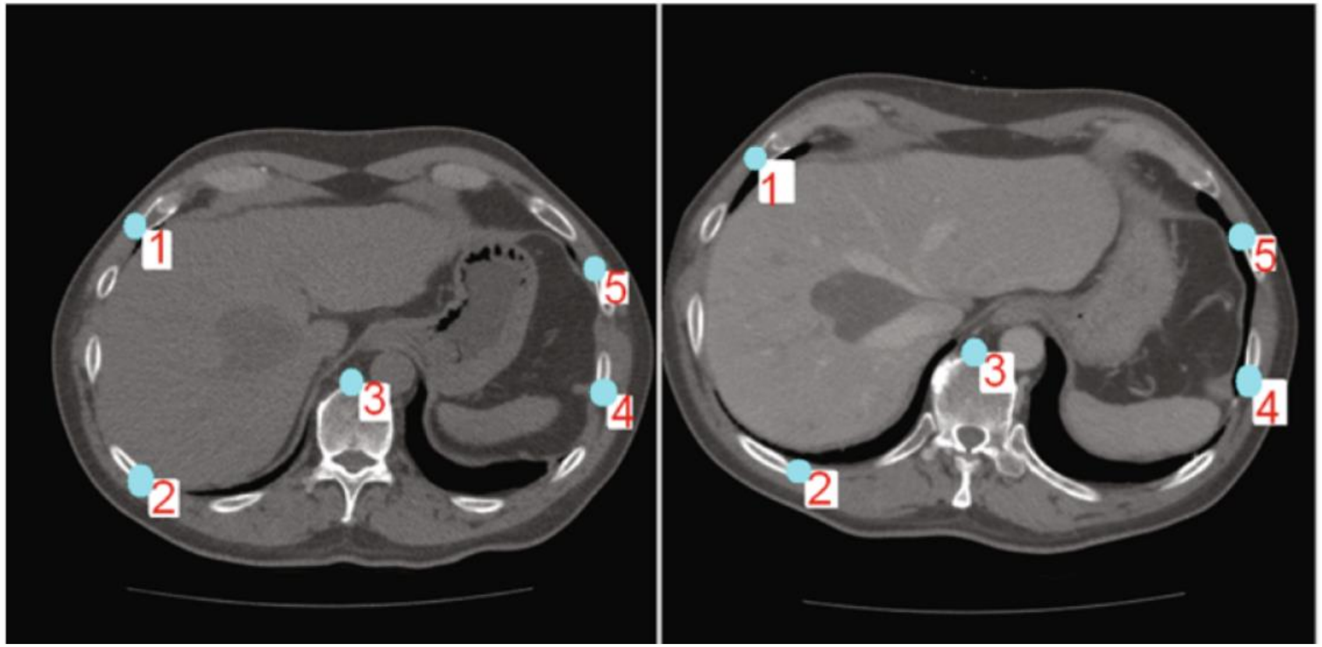


Fig. 11.3 A preoperative CT scan (the reference) is seen to the left and a postoperative CT scan (the template) is seen to the right. Five corresponding landmarks have been manually selected

Landmark-based Registration



i	a_i	b_i
1	(106.75, 178.75)	(74.125, 125.125)
2	(110.25, 374.25)	(108.875, 366.875)
3	(273.75, 299.75)	(246.125,275.125)
4	(469.25, 306.75)	(459.125, 297.625)
5	(465.75, 212.75)	(456.625, 182.875)

Landmark-based Registration

- Use transformations to map the landmark of the Reference image (fixed) to the landmark of the template image (that is changed)

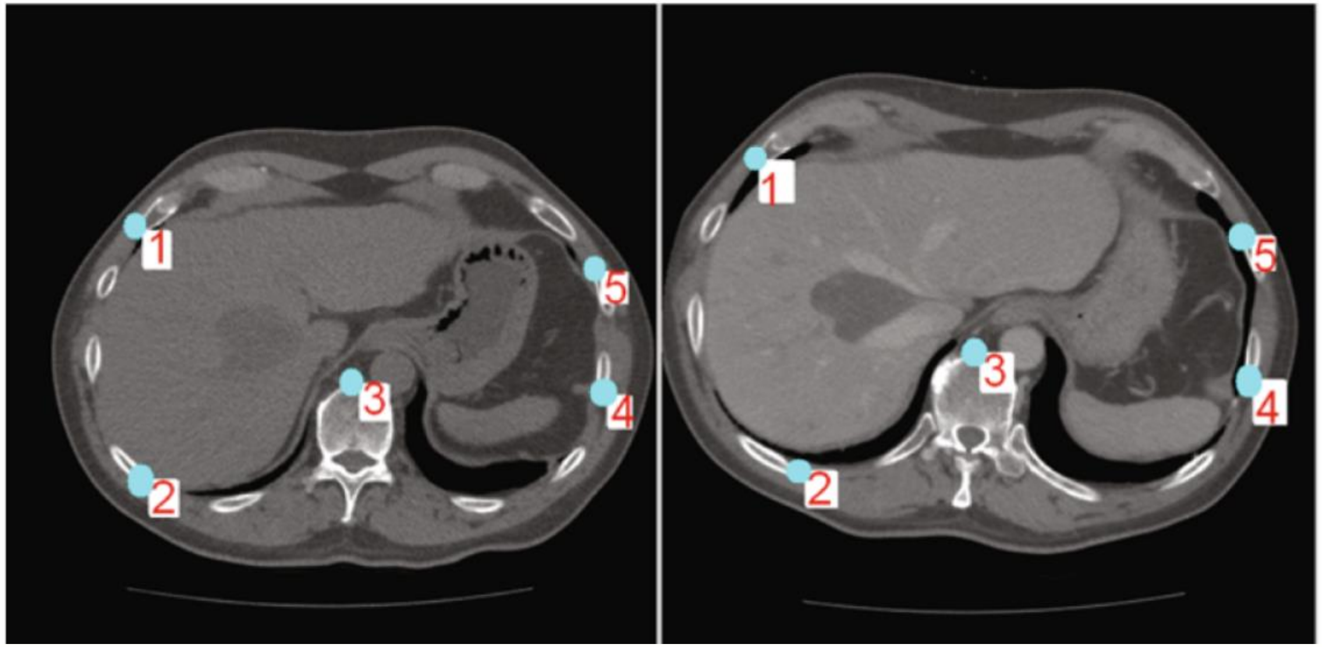
- $A: (x, y) \rightarrow A'(x', y')$
  - $A' = T(A)$
  - T can be translation, rotation, scaling or any combination of these

Landmark-based Registration

- Use transformations to map the landmark of the Reference image (fixed) to the landmark of the template image (that is changed)

- Landmark point i in template image $A_i: (x, y)$
  - $A_i' = T(A_i)$
- Landmark point i in reference image B_i
- Goal is to minimize $\mathrm{F} = \sum_{i=1}^{N} D(T(A_i), B_i)$

Landmark-based Registration



i	a_i	b_i
1	(106.75, 178.75)	(74.125, 125.125)
2	(110.25, 374.25)	(108.875, 366.875)
3	(273.75, 299.75)	(246.125,275.125)
4	(469.25, 306.75)	(459.125, 297.625)
5	(465.75, 212.75)	(456.625, 182.875)

The Optimization Problem

$$\hat{w} = \arg\min_{w} F,$$

w is the parameter that we want to optimize

w is Δx, Δy for translation

Translation

- Translation of landmark points

- $A: (x, y) \rightarrow A'(x', y')$
  - $A' = T(A)$
  - T is translation

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$F = \sum_{i=1}^{N} \|(a_i + t) - b_i\|^2$$

Translation

- Estimated optimal translation

$$\hat{t} = \bar{b} - \bar{a},$$

$$\bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i \qquad \text{and} \qquad \bar{b} = \frac{1}{N}\sum_{i=1}^{N} b_i .$$

Translation + Rotation: Rigid Transformation

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Translation + Rotation: Rigid Transformation

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

$$\tilde{a}_i = a_i - \bar{a} \text{ and } \tilde{b}_i = b_i - \bar{b}.$$

$$T\begin{pmatrix} x \\ y \end{pmatrix} = R\begin{bmatrix} x \\ y \end{bmatrix} + t.$$

$$F = \sum_{i=1} \|(Ra_i + t) - b_i\|^2.$$

Translation + Rotation: Rigid Transformation

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

$$T\begin{pmatrix} x \\ y \end{pmatrix} = R\begin{bmatrix} x \\ y \end{bmatrix} + t.$$

$$F = \sum_{i=1}^{N} \|(Ra_i + t) - b_i\|^2.$$

Centered landmark

$$\tilde{a}_i = a_i - \bar{a} \text{ and } \tilde{b}_i = b_i - \bar{b}.$$

Translation + Rotation: Rigid Transformation

$$T\begin{pmatrix} x \\ y \end{pmatrix} = R\begin{bmatrix} x \\ y \end{bmatrix} + t.$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

$$F = \sum_{i=1}^{N} \|(Ra_i + t) - b_i\|^2.$$

Centered landmark

$$\tilde{a}_i = a_i - \bar{a} \text{ and } \tilde{b}_i = b_i - \bar{b}.$$

$$F = \sum_{i=1}^{N} \|R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a})\|^2.$$

Optimization

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a}) \|^2.$$

Fiirst, differentiate, w.r.t. *t*

Optimization

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a}) \|^2.$$

Fiirst, differentiate, w.r.t. t

We get $t = \bar{b} - R\bar{a}$

Optimization

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a}) \|^2.$$

Fiirst, differentiate, w.r.t. t

We get $t = \bar{b} - R\bar{a}$

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i \|^2$$

$$= \sum_{i=1}^{N} \left[\tilde{a}_i^T R^T R \tilde{a}_i + \tilde{b}_i^T \tilde{b}_i - 2\tilde{a}_i^T R^T \tilde{b}_i \right].$$

Optimization

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a}) \|^2.$$

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i \|^2$$

$$= \sum_{i=1}^{N} \left[\tilde{a}_i^T R^T R \tilde{a}_i + \tilde{b}_i^T \tilde{b}_i - 2\tilde{a}_i^T R^T \tilde{b}_i \right].$$

$$R^T R = I$$

In order to to find R that minimizes F, do singular value decomposition of

$$H = \sum_{i=1}^{N} \tilde{a}_i \tilde{b}_i^T = UDV^T$$

Optimization

$$F = \sum_{i=1}^{N} \| R\tilde{a}_i - \tilde{b}_i + t - (\bar{b} - R\bar{a}) \|^2.$$

In order to find R that minimizes F, do singular value decomposition of

$$H = \sum_{i=1}^{N} \tilde{a}_i \tilde{b}_i^T = UDV^T$$

We get

$$\hat{R} = V \operatorname{diag}(1, 1, \det(VU)) U^T$$
$$\hat{t} = \bar{b} - \hat{R}\bar{a}$$

Translation + Rotation + Isotropic Scaling = Similarity Transformation

$$T\begin{pmatrix} x \\ y \end{pmatrix} = sR\begin{bmatrix} x \\ y \end{bmatrix} + t$$

parameter vector $w = (\Delta x, \Delta y, \theta, s)$

$$\hat{R} = V \operatorname{diag}(1, 1, \det(VU))U^T$$

$$\hat{s} = \frac{\sum_{i=1}^{N} \tilde{a}_i^T \hat{R}^T \tilde{b}_i}{\sum_{i=1}^{N} \tilde{a}_i^T \tilde{a}_i}$$

$$\hat{t} = \bar{b} - \hat{s}\hat{R}\bar{a}$$

Spline Transformation

- Template image acts like a rubber membrane
- Can be stretched or compressed locally

Steps for Image Registration

- Feature detection/ land mark identification/ landmark annotation
  - The landmarks/ features to be matched are identified/ detected
- Feature/ landmark matching
  - Establishing correspondences between landmarks/ features
- Estimation of transformation
- Transformation of template image

Steps for Image Registration

- Feature detection/ land mark identification/ landmark annotation
  - The landmarks/ features to be matched are identified/ detected
- Feature/ landmark matching
  - Establishing correspondences between landmarks/ features
- Estimation of transformation $F = S(\mathcal{R}, T(\mathcal{T}))$ $\hat{w} = \arg\min_{w} F,$
- Transformation of template image

Steps for Image Registration

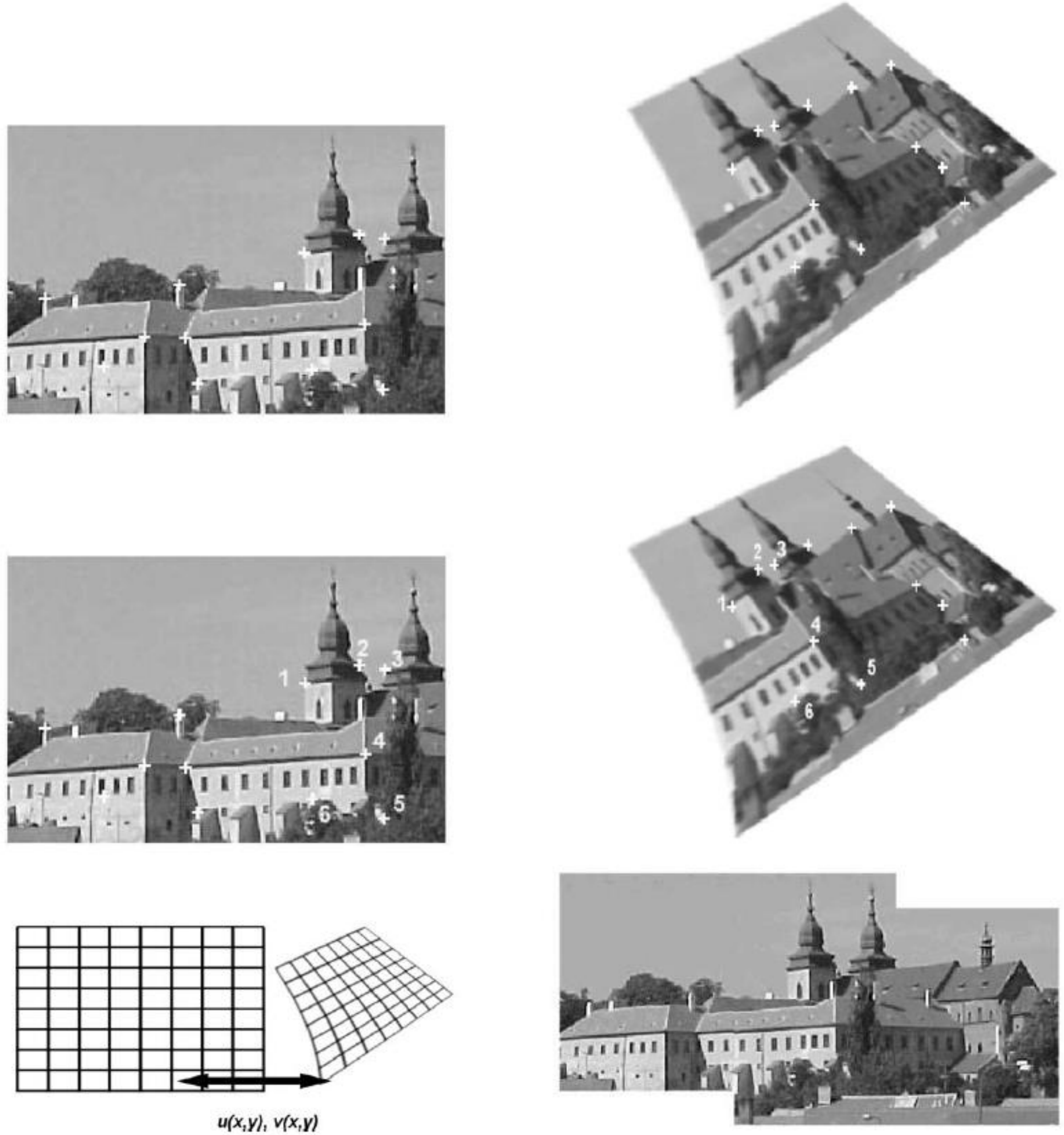


Fig. 1. Four steps of image registration: top row—feature detection (corners were used as the features in this case). Middle row—feature matching by invariant descriptors (the corresponding pairs are marked by numbers). Bottom left—transform model estimation exploiting the established correspondence. Bottom right—image resampling and transformation using appropriate interpolation technique.

Automated Feature Detection

- Huge volume of research work has been done

- Point features
  - Building, lake, tower, etc.
  - Can be detected by segmentation
  - Correspondences: center of mass, corners, etc.
- Line features
  - Boundary of organ, object contour, coast line, etc.
  - Can be detected using edge detector
  - Correspondences: end points of line, middle point, etc.

Automated Feature Detection

- Huge volume of research work has been done
- Region features
  - Local curvature, discontinuities, centroid, high variance points, etc.
- Other features
  - SIFT

Feature/ Landmark Matching: Area-based Methods

- Correlation
  - Window based
  - Salient objects, points are not utilized
  - Not useful if transformation other than translation is the cause of deformation
  - High possibility of incorrect matching especially for smooth regions

- Use of Fourier transform
  - Cross power spectrum

$$\frac{\mathscr{F}(f)\mathscr{F}(g)^*}{|\mathscr{F}(f)\mathscr{F}(g)^*|} = e^{2\pi i(ux_0+vy_0)}$$

Feature/ Landmark Matching

- Mutual information based methods
  - Maximize MI

$$MI(X,Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y),$$

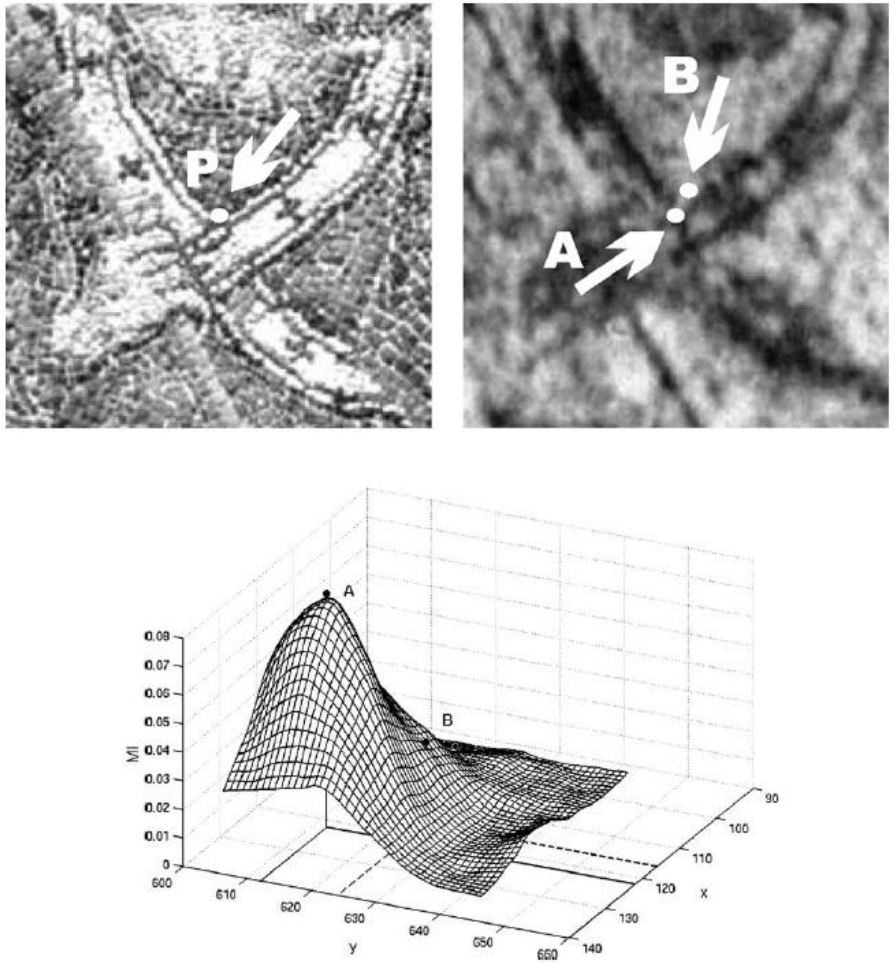


Fig. 3. Mutual information: MI criterion (bottom row) computed in the neighborhood of point P between new and old photographs of the mosaic (top row). Maximum of MI shows the correct matching position (point A). Point B indicates the false matching position selected previously by the human operator. The mistake was caused by poor image quality and by complex nature of the image degradations.

Feature/ Landmark Matching: Feature-based Methods

- Assume that two sets of features in the reference and sensed images represented by the points, end points or centers of line features,, etc. have been detected
  - Control points

- The aim is to find the pairwise correspondence between them using their spatial relations or various descriptors of features
  - Graph matching
  - Clustering
  - Invariant descriptors
  - Wavelets

Estimation of Transformation: Global

- Similarity transform: shape preserving

$$u = s(x\cos(\varphi) - y\sin(\varphi)) + t_x$$
$$v = s(x\sin(\varphi) + y\cos(\varphi)) + t_y$$

- Affine transform
  - Can map parallelogram onto a square
  - This model is defined by three non-collinear CPs
  - preserves straight lines and straight line parallelism

$$u = a_0 + a_1 x + a_2 y$$
$$v = b_0 + b_1 x + b_2 y,$$

Estimation of Transformation: Global

- Exactly describes a deformation of a flat scene photographed by a pin-hole camera the optical axis of which is not perpendicular to the scene

- It can map a general quadrangle onto a square while preserving straight lines and is determined by four independent CPs.

$$u = \frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}$$

$$v = \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y}$$

Estimation of Transformation: Local

- Can capture local deformation

- Radial basis function
  - Interpolation of irregular surfaces
  - 'Radial' reflects an important property of the function value at each point-it depends just on the distance of the point from the CPs, not on its particular position.

$$u = a_0 + a_1 x + a_2 y + \sum_{i=1}^{N} c_i g(\mathbf{x}, \mathbf{x}_i)$$

and similarly for v.

Estimation of Transformation: Local

- Elastic registration
  - The images are viewed as pieces of a rubber sheet, on which external forces stretching the image and internal forces defined by stiffness or smoothness constraints are applied to bring them into alignment with the minimal amount of bending and stretching
  - The feature matching and mapping function design steps of the registration are done simultaneously
  - This is one of the advantages of elastic methods, because feature descriptors invariant to complicated deformations are not known and the feature correspondence is difficult to establish in the traditional way
  - The registration is achieved by locating the minimum energy state in an iterative fashion

Estimation of Transformation

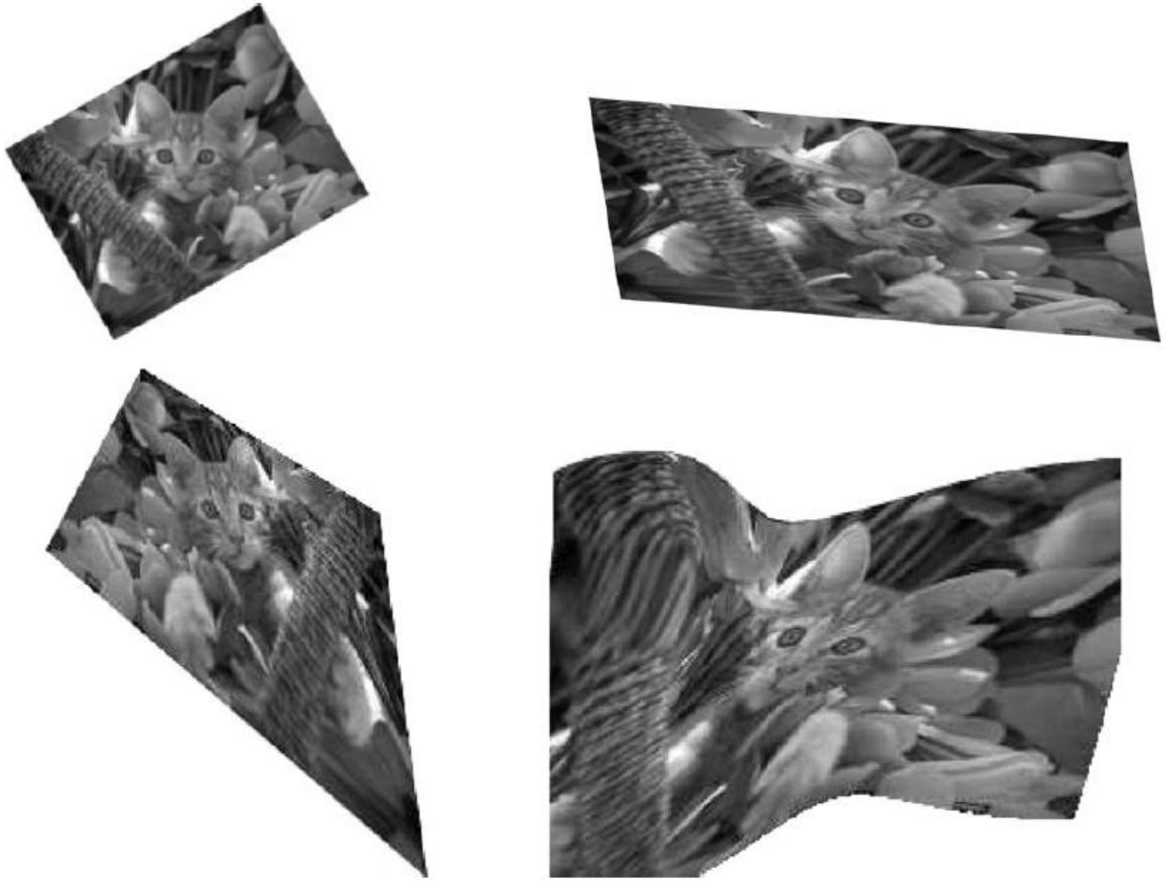


Fig. 5. Examples of various mapping functions: similarity transform (top left), affine transform (top right), perspective projection (bottom left), and elastic transform (bottom right).