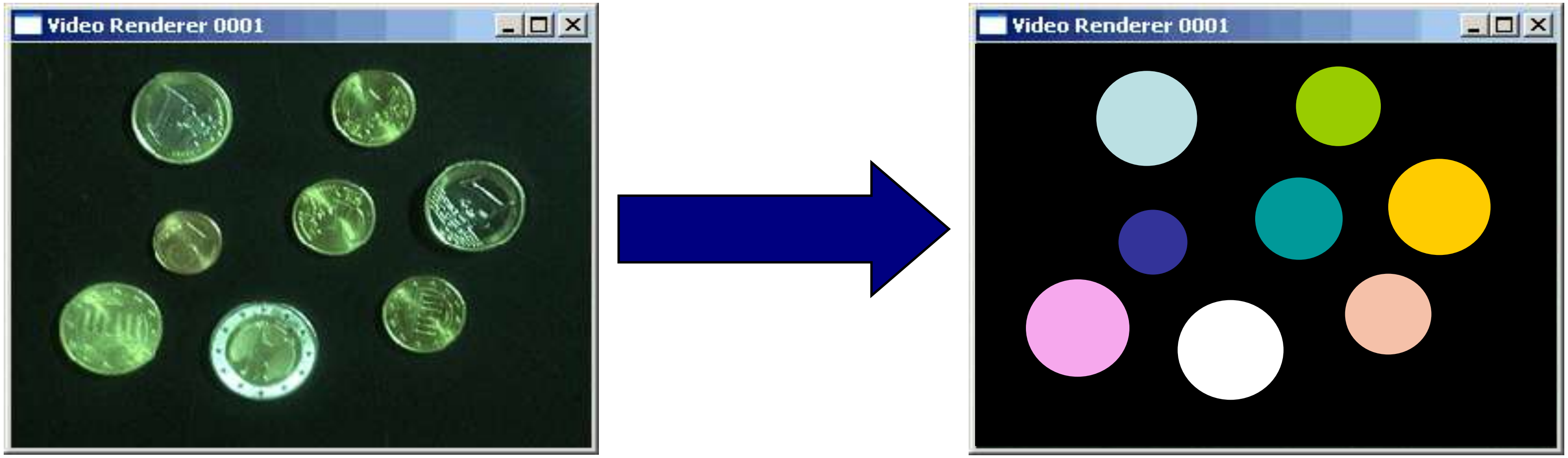
Image Segmentation:

Edge Detection and Thresholding

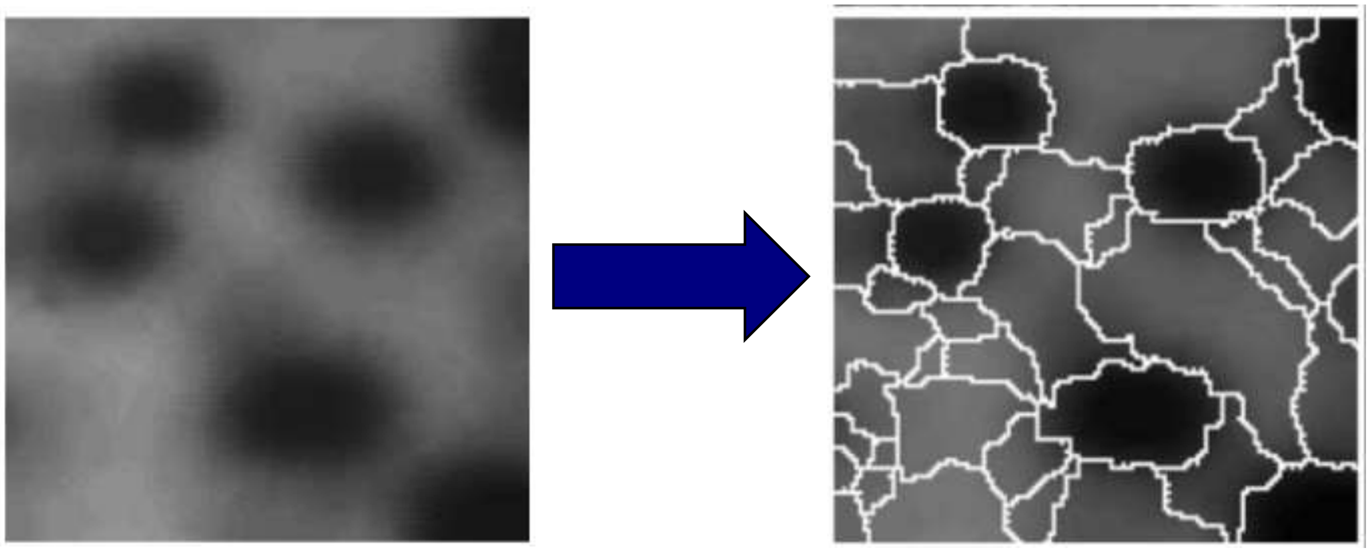
Image Segmentation

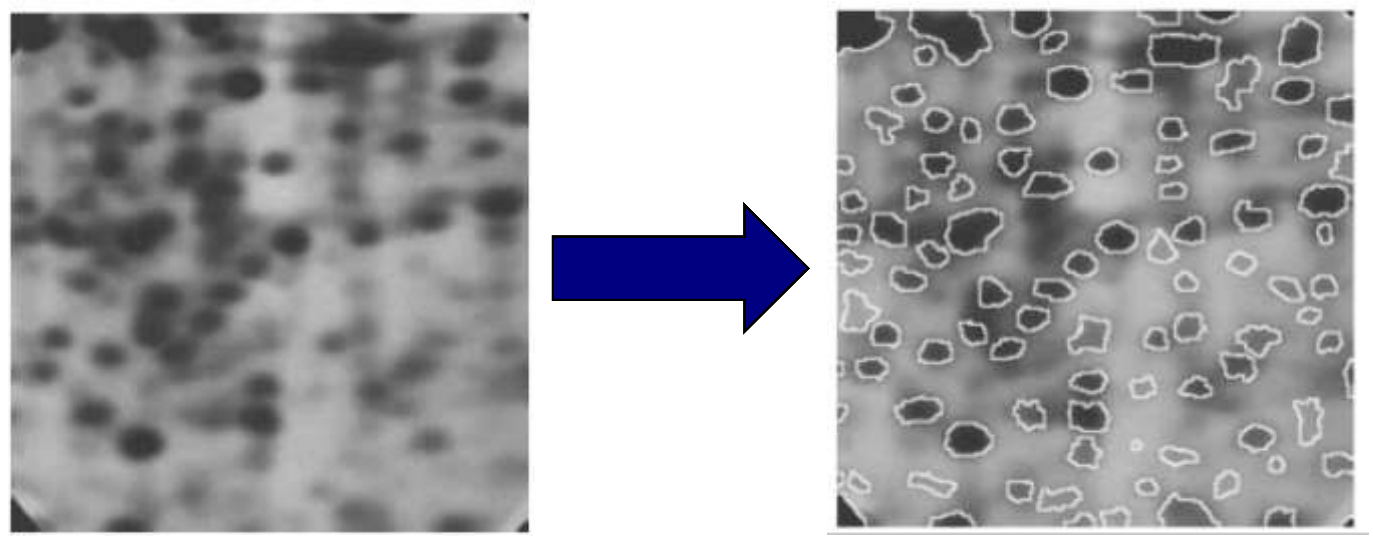
Segmentation process attempts to partition an image into its constituent parts or objects.

The target of segmentation is to isolate the objects or parts of interest from an image. Typically, segmentation is the first step in any automated computer vision application.



Segmentation Examples





Segmentation Examples

Image Segmentation

Segmentation in monochrome images generally are based on one of two basic properties of gray-level values:

- Discontinuity
- Similarity

In the first category, the approach is to partition an image based on abrupt changes in gray level.

In the second category, the principal approaches are based on thresholding, region growing, region splitting and merging.

Detection Of Discontinuities

There are three basic types of grey level discontinuities that we tend to look for in digital images:

- Points
- Lines
- Edges

We generally find discontinuities by running a mask through the image and using correlation.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

A 3X3 mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

An image

The output response of the mask at a point z_5 in the image is given by,

$$R_{z5} = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$

$$= \sum_{i=1}^{9} w_i z_i$$

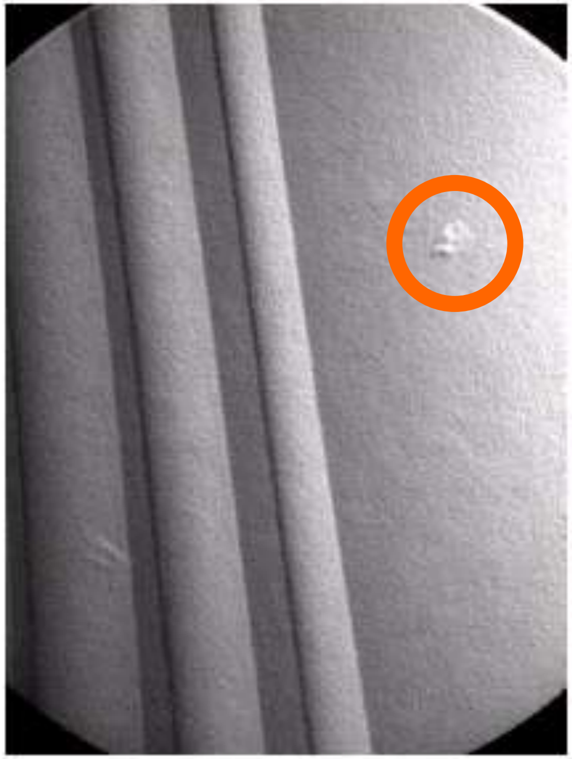
Point Detection

Point detection can be achieved simply using the mask below:

−1	−1	−1
−1	8	−1
−1	−1	−1

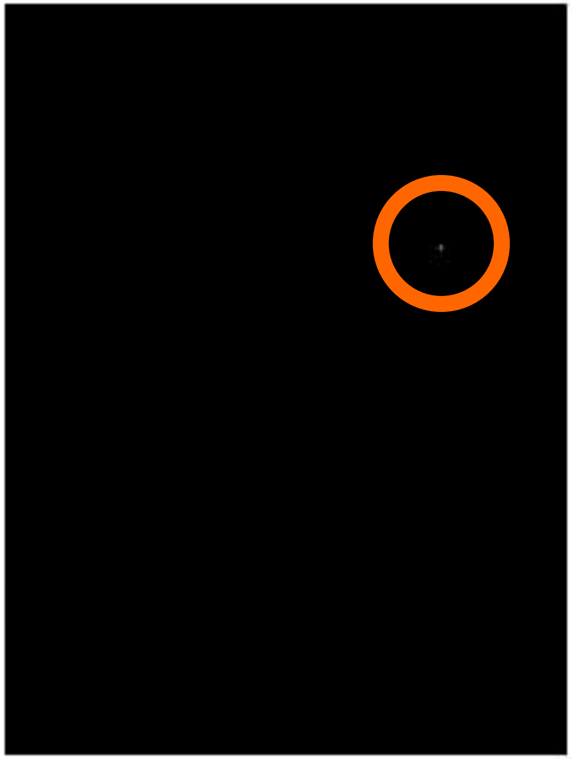
Points are detected at those pixels in the subsequent filtered image that are above a threshold value.

Point Detection (cont…)



X-ray image of
a turbine blade

Result of point
detection



Result of
thresholding

Line Detection

The next level of complexity is to try to detect lines.

The masks below will extract lines that are one pixel thick and running in a particular direction.

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

$+45°$

−1	2	−1
−1	2	−1
−1	2	−1

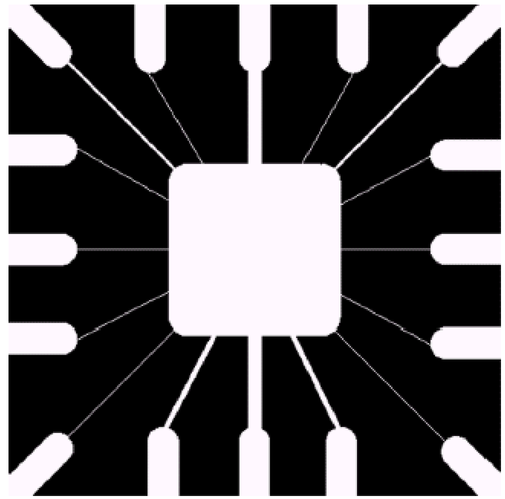
Vertical

2	−1	−1
−1	2	−1
−1	−1	2

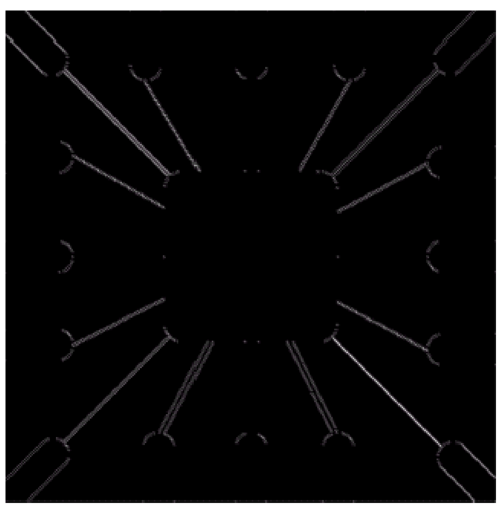
$-45°$

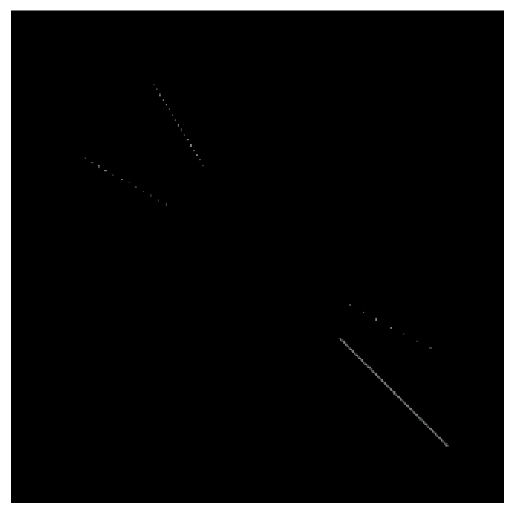
Line Detection (cont…)

Binary image of a wire bond mask



After processing with -45° line detector



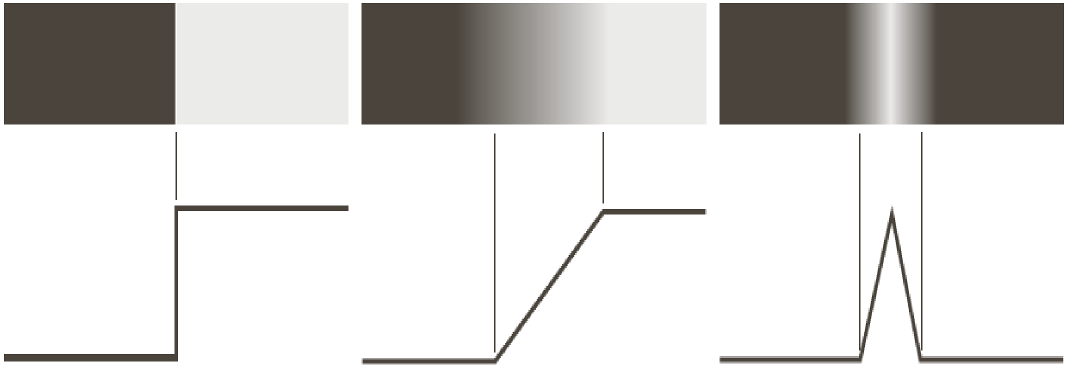


Result of thresholding filtering result

Edge Detection

An edge is a jump in intensity. Edges in the image can be defined as the discontinuities or abrupt changes in intensity. In typical images, edges characterize object boundaries and therefore useful for segmentation, registration and identification of objects in a scene.

Edge detection considers the intensity change that occurs in the pixel at the boundary (or edges) of an image.

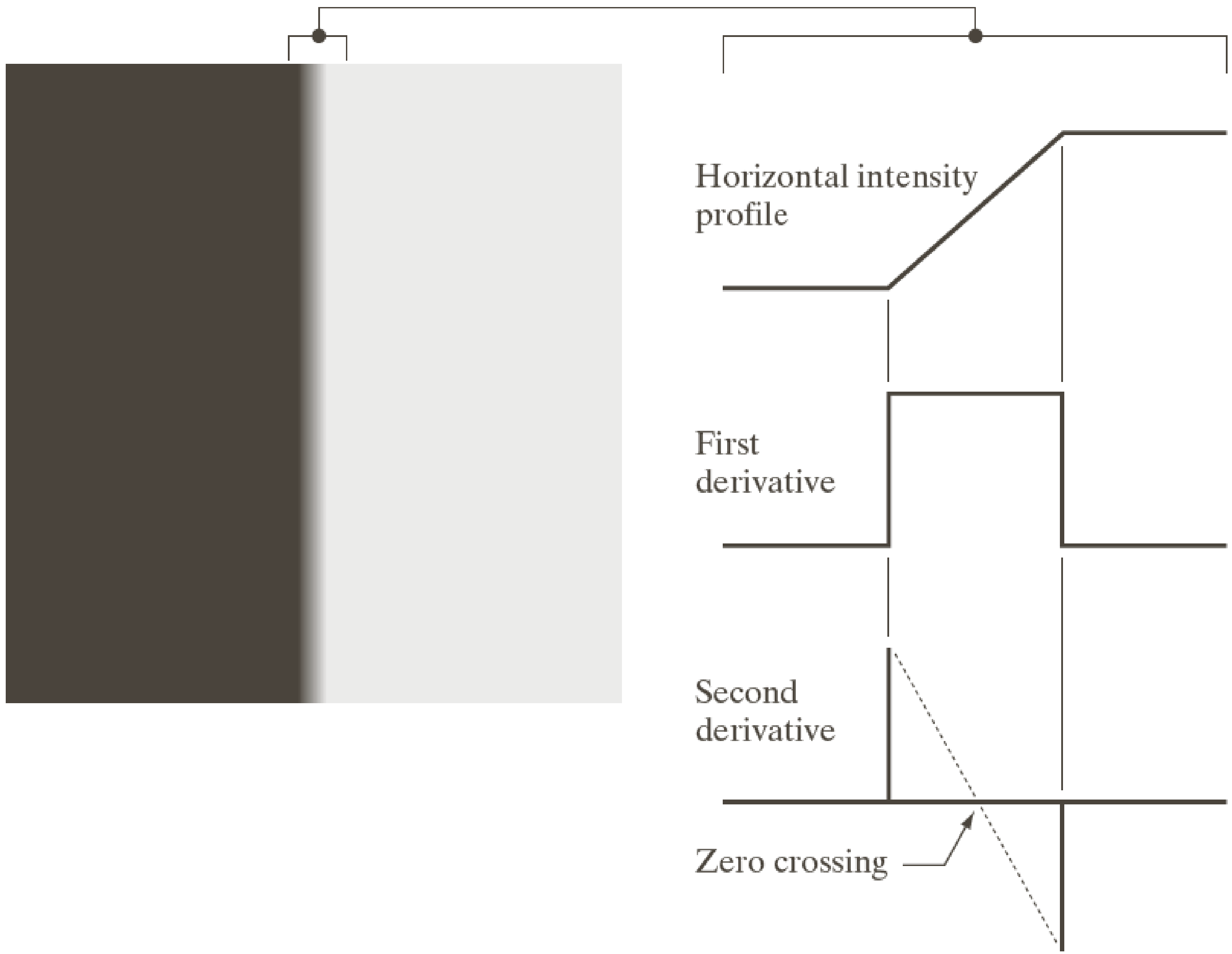


a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

An edge is a set of connected pixels that lie on the boundary between two regions.

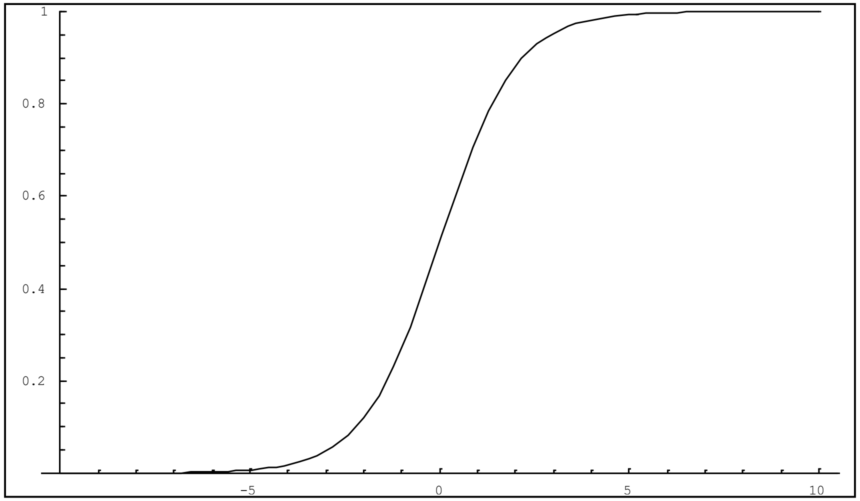
Edge Detection …



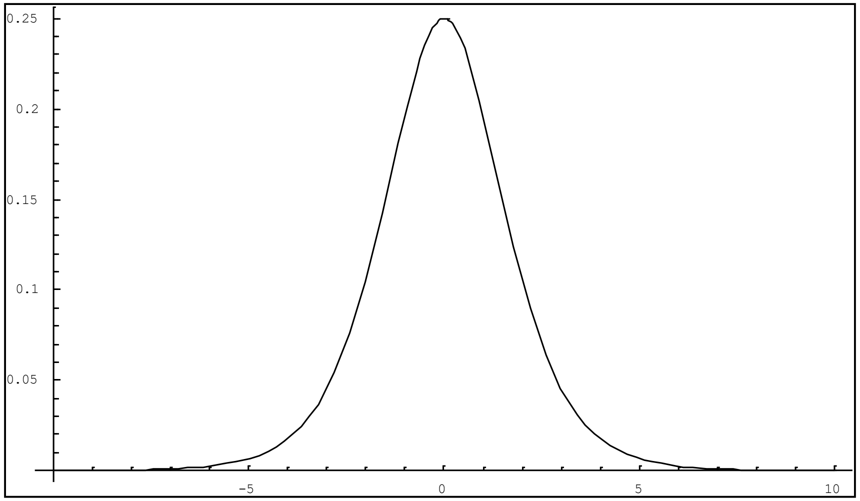
a b

FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Edge Detection …



An "edge" viewed as a one dimensional function.



The derivative of the edge shows a spike at the edge boundary.

Derivative filters produce images with high intensity at edges and low intensity at homogenous regions.

Edge Detection …

The basic idea behind the edge detection technique is the computation of a local derivative operator.

The 1st derivative assumes a local maximum at an edge. The 1st derivative tells where an edge is and 2nd derivative can be used to show edge direction.

The most used operators are:

- Gradient operators (1st Derivative-Sobel, Prewitt, Robert)
- Laplacian operator (2nd Derivative)
- LOG operator (Laplacian of Gaussian)

1^{st} Derivative

The formula for the 1^{st} derivative of a function $f(x,y)$ is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

It's just the difference between subsequent values and measures the rate of change of the function.

2^{nd} Derivative

The formula for the 2^{nd} derivative of the above function *f*(*x*,*y*) is as follows:

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

Simply takes into account the values both before and after the current value

Gradient operator

The gradient method detects the edges by looking for the maximum and minimum in the first derivatives of the image. The first derivative assumes a local maximum at an edge. This method includes the operators like: Sobel, Prewitt & Roberts operators.

The gradient operator of an image *f(x,y)* at location *x, y* can be expressed by it's magnitude as:

$$G[f(x,y)] = [G_x^2 + G_y^2]^{1/2}$$

$$= \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2}$$

For practical reasons this can be simplified as:

$$\nabla f \approx |G_x| + |G_y|$$

Robert Operator

- Consider the 3x3 region shown below and approximate the gradient at Z_5.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$df/dx \approx Z_5 - Z_6$

$df/dy \approx Z_5 - Z_8$

$$\left|\nabla f\right| \cong \left|z5 - z6\right| + \left|z5 - z8\right|$$

- We can thus define a kernel to implement the gradient approximation as follows, which is known as **Robert operator** .

1	-1
0	0

1	0
-1	0

Sobel Operators

We can again approximate the magnitude of the gradient at the center of a 3x3 region and obtain the following. Based on these equations we can derive the ***Sobel Operators***.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

The weight value 2 is to achieve smoothing by giving more important to the center point

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

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

To detect edges of an image it is filtered using both operators the results of which are added together.

Prewitt Operator

We can again approximate the magnitude of the gradient at the center of a 3x3 region as shown by the following eq^{ns}:

Based on these equations we can derive the ***Prewitt Operators*.**

$$|\nabla f| \cong \begin{array}{l} |(z7+z8+z9)-(z1+z2+z3)| + \\ |(z3+z6+z9)-(z1+z4+z7)| \end{array}$$

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

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

Edge Detection …

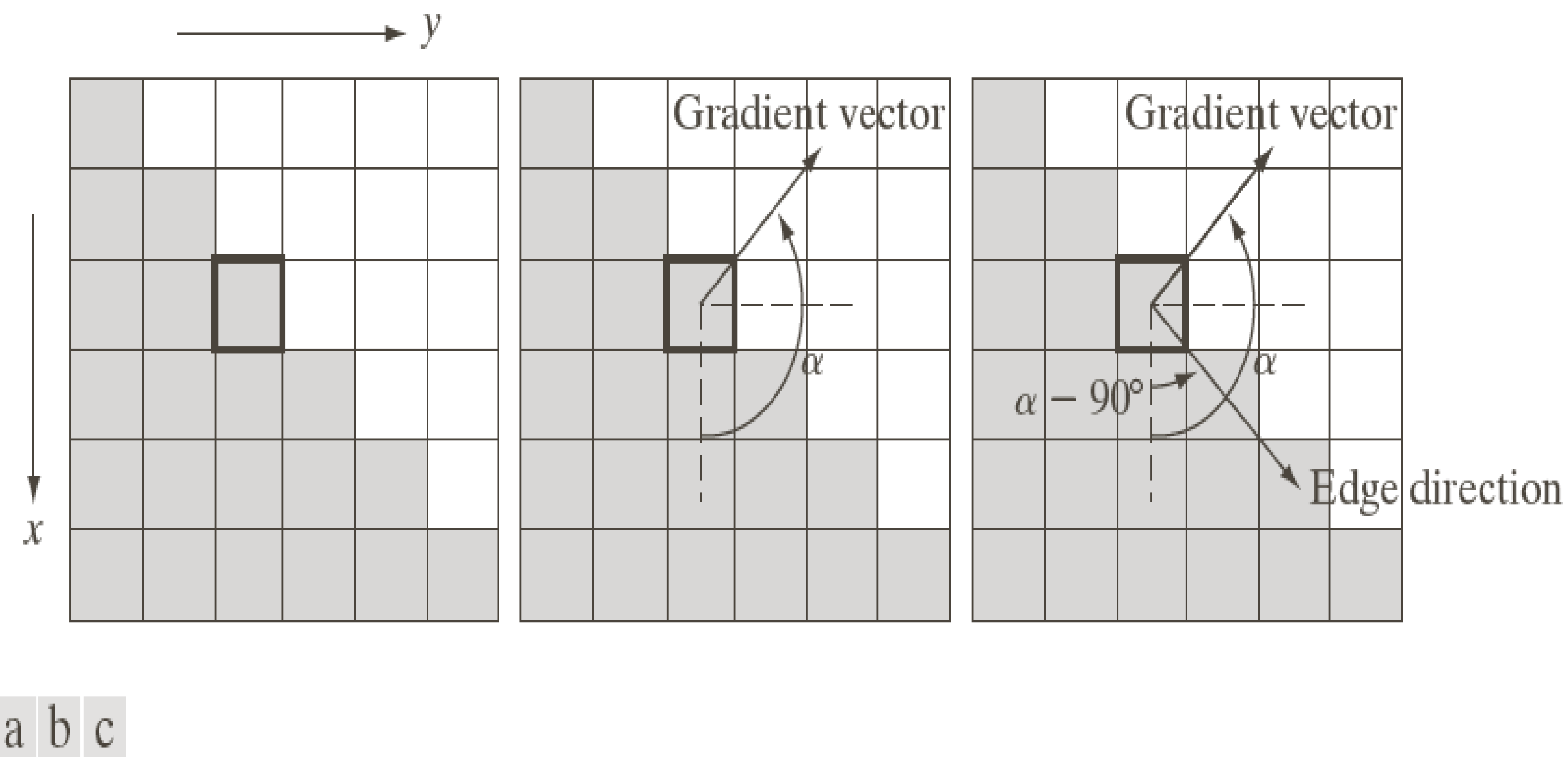


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Edge Detection Example

Original Image — Horizontal Gradient Component

Vertical Gradient Component — Combined Edge Image

The Laplacian Edge Detector

The Laplacian operator is a second order derivative which has a strong zero crossing. For this reason, an alternative edge-detection strategy, called Laplacian edge detector, locates zeros of the second derivatives of *f(x, y)*.

The Laplacian is defined as follows:

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

where the partial 2nd order derivative in the x direction is defined as follows:

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

and in the y direction as follows:

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

The Laplacian (cont…)

So, the Laplacian can be given as follows:

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

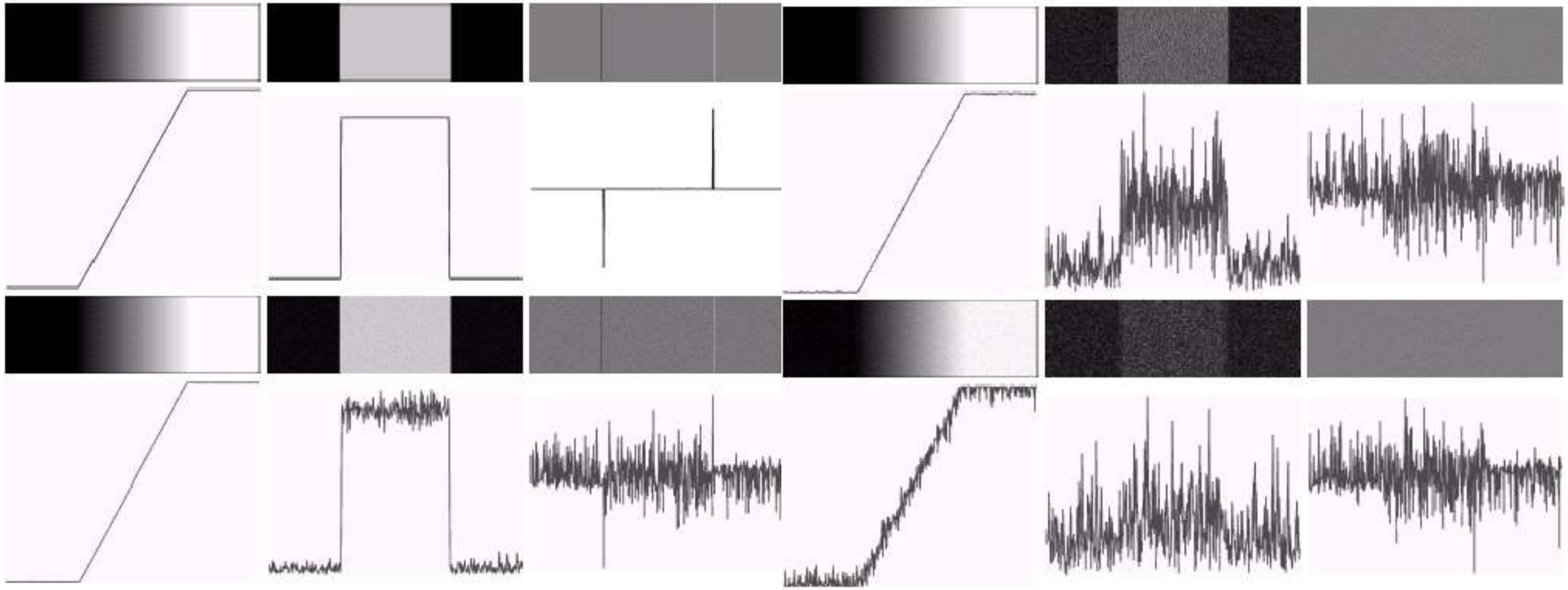
We can easily build a mask/operator/filter based on this as follows:

0	1	0
1	-4	1
0	1	0

Derivatives & Noise

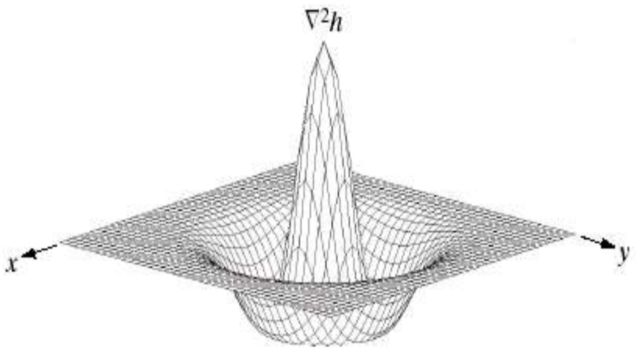
Derivative based edge detectors are extremely sensitive to noise.

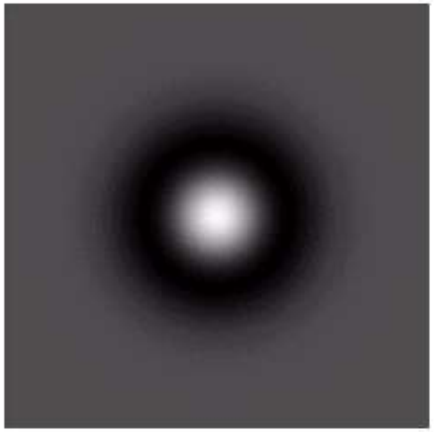
One way to overcome this is to smooth images prior to edge detection.

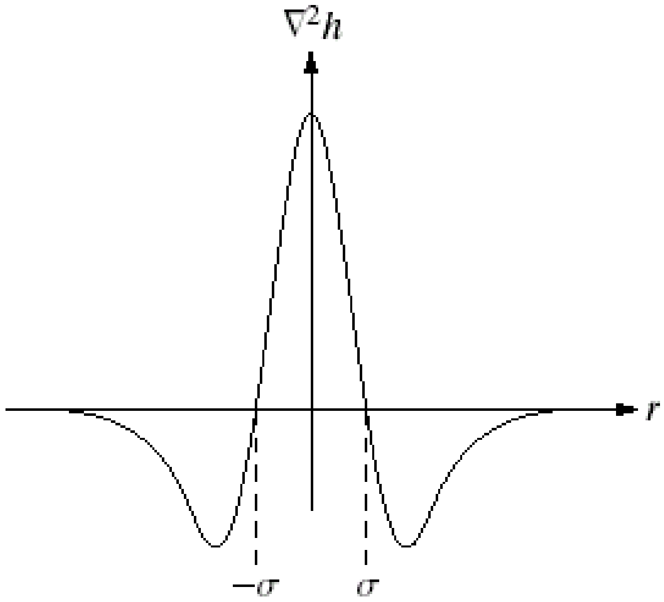


Laplacian Of Gaussian

The Laplacian of Gaussian (LoG or Mexican hat) filter uses the Gaussian for noise removal and the Laplacian for edge detection.

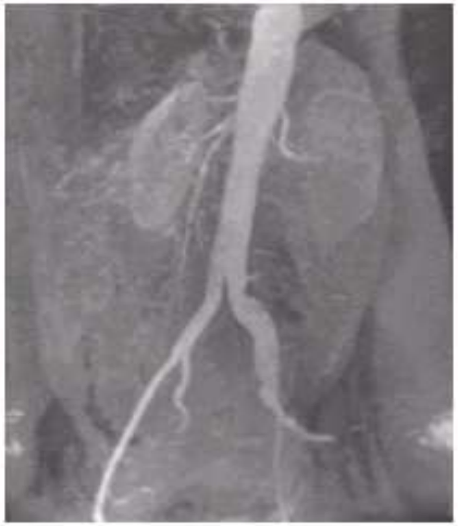


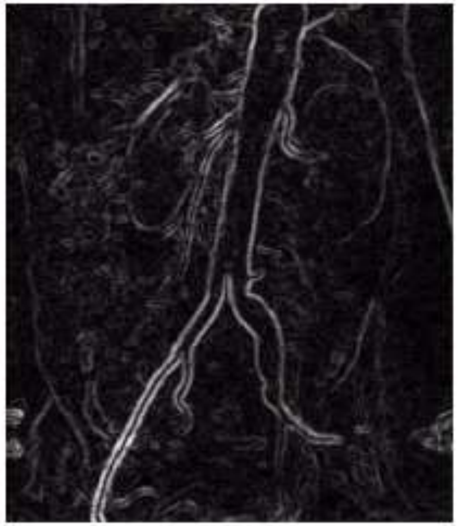


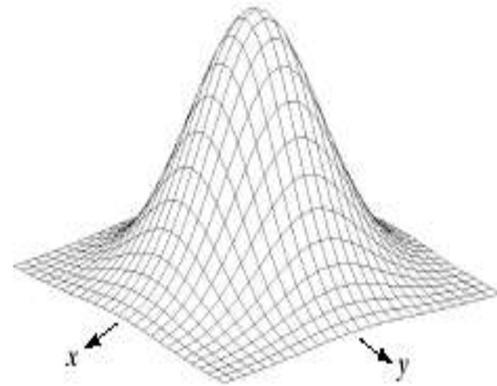


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

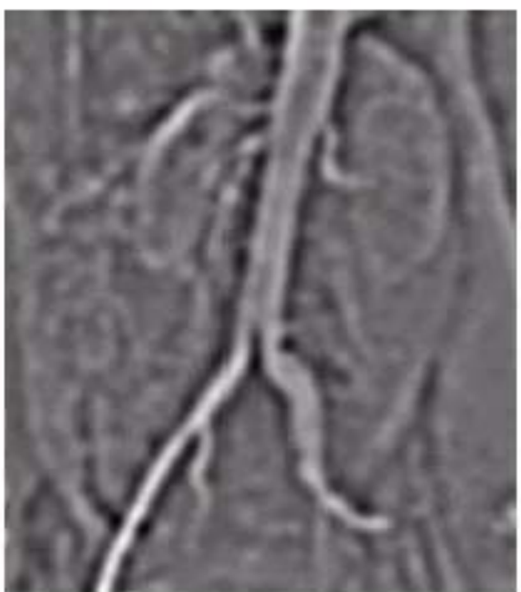
Laplacian Of Gaussian Example

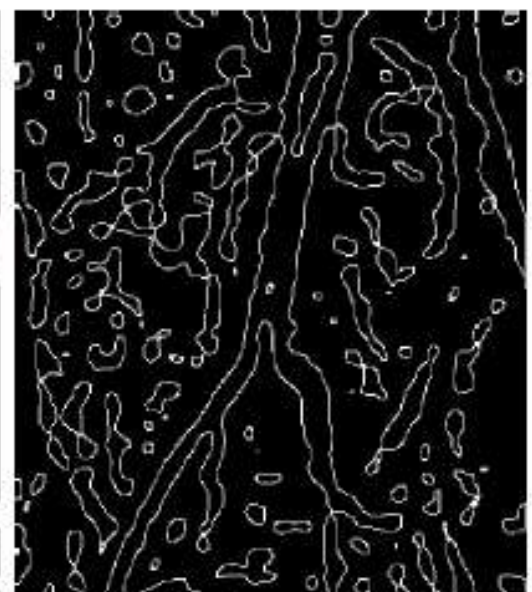






−1	−1	−1
−1	8	−1
−1	−1	−1





Some Applications of Edge Detection

- Industrial inspection for 3-D
- 3-D measurement of objects,
- Autonomous vehicles, robotics
- Medical, biomedical and bioengineering scanning
- Transport (traffic scene analysis)
- 3-D database for urban and town planning

Thresholding

We have talked about simple single value thresholding already.

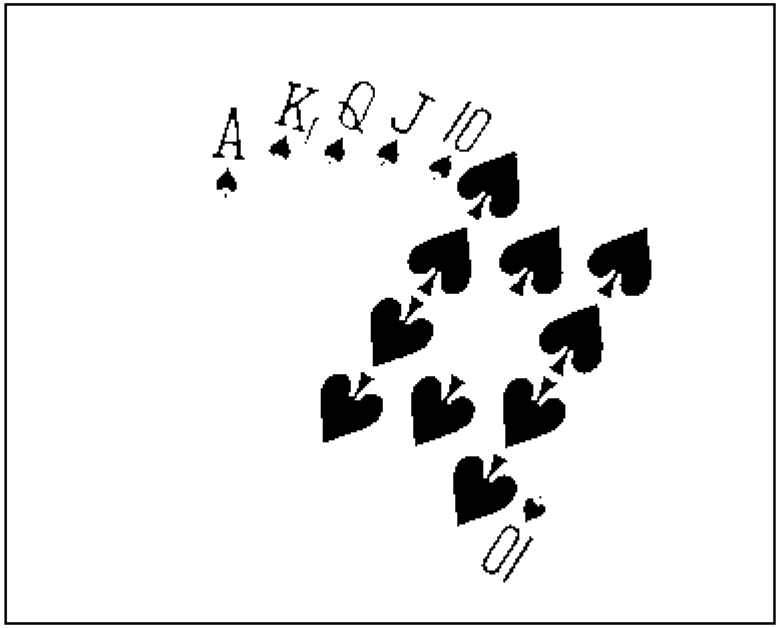
Single value thresholding can be given mathematically as follows:

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

Thresholding Example

Imagine a poker playing robot that needs to visually interpret the cards in its hand

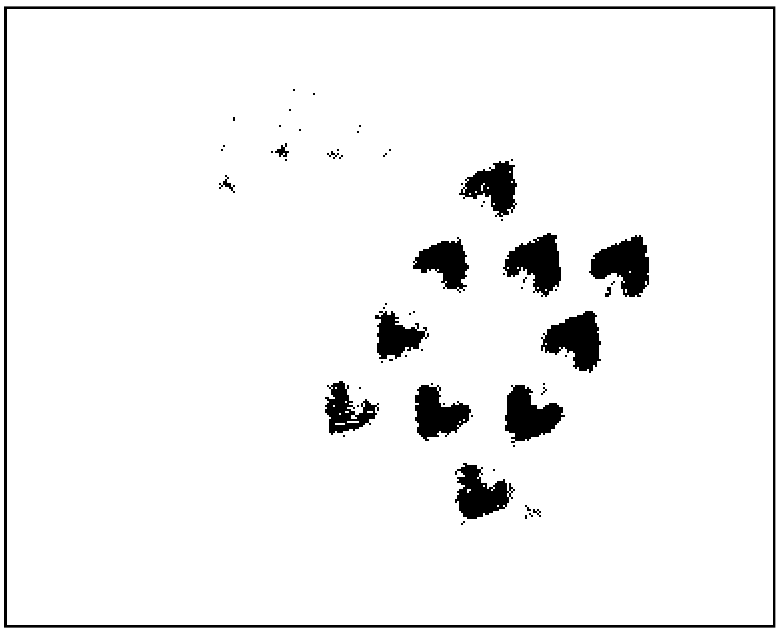
Original Image



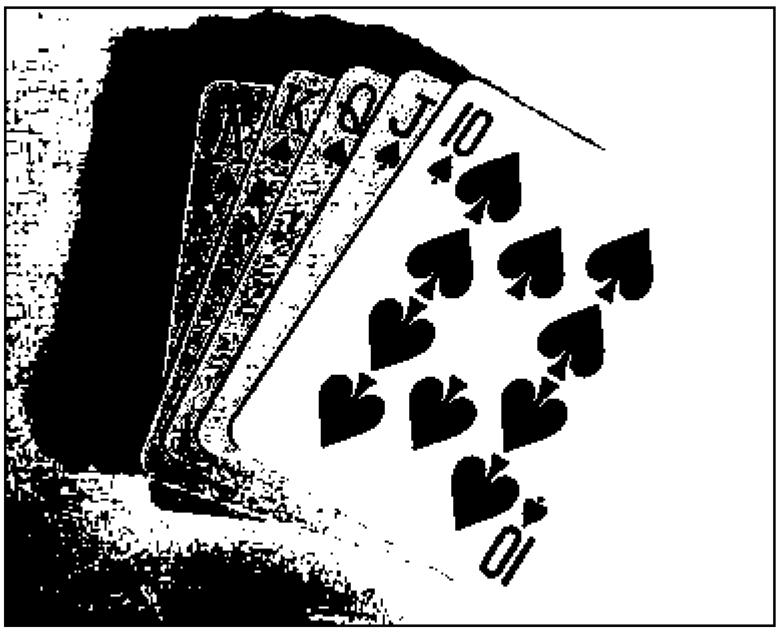
Thresholded Image

But Be Careful

If you get the threshold wrong the results can be disastrous



Threshold Too Low



Threshold Too High

Basic Global Thresholding

Based on the histogram of an image Partition the image histogram using a single global threshold.

The success of this technique very strongly depends on how well the histogram can be partitioned.

Basic Global Thresholding Algorithm

The basic global threshold, T, is calculated as follows:

1. Select an initial estimate for T (typically the average grey level in the image)
2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T
3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Basic Global Thresholding Algorithm

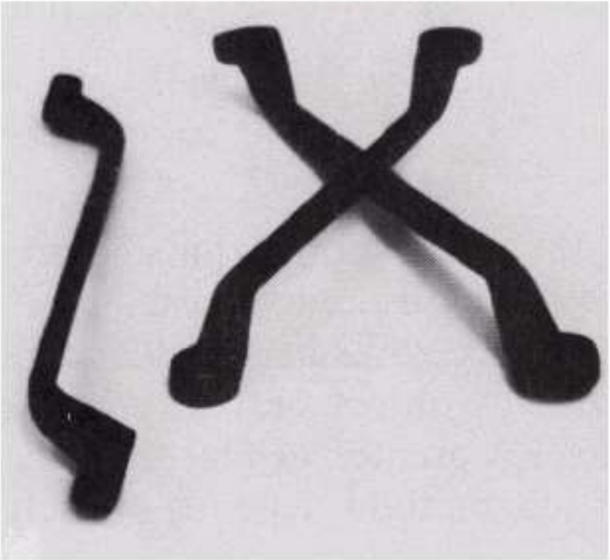
4. Compute a new threshold value:

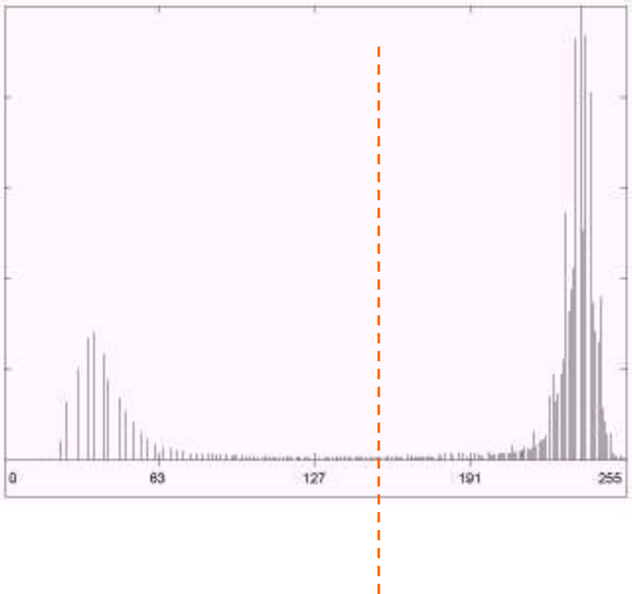
$$T = \frac{\mu_1 + \mu_2}{2}$$

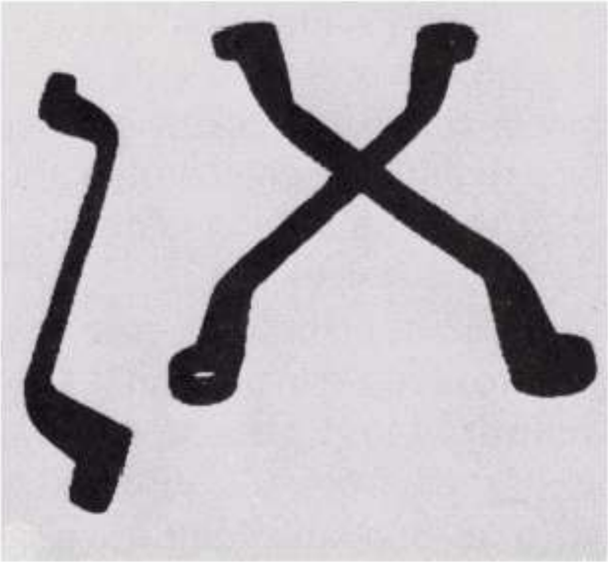
5. Repeat steps 2 – 4 until the difference in *T* in successive iterations is less than a predefined parameter dT

This algorithm works very well in a situation where there is a reasonable clear valley between the modes of histogram related to objects and background.

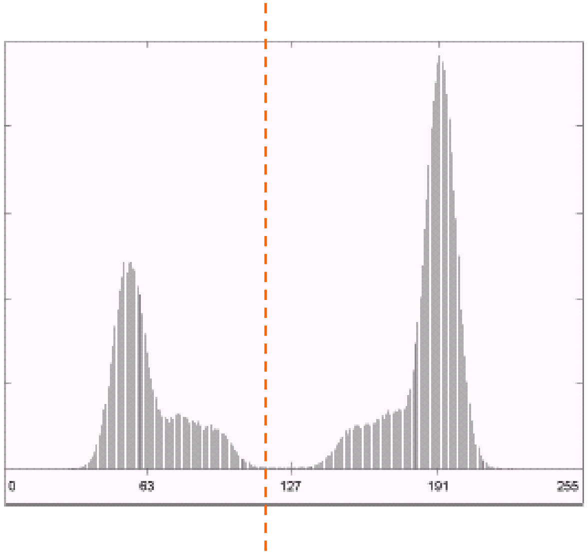
Thresholding Example 1







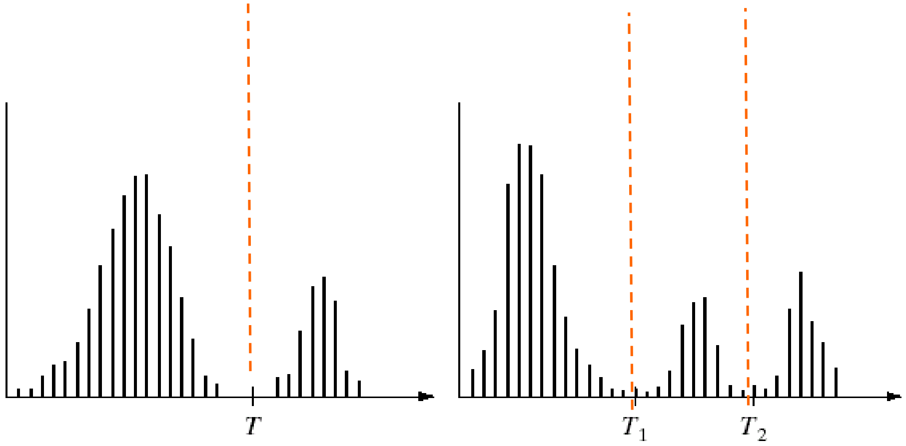
Thresholding Example 2



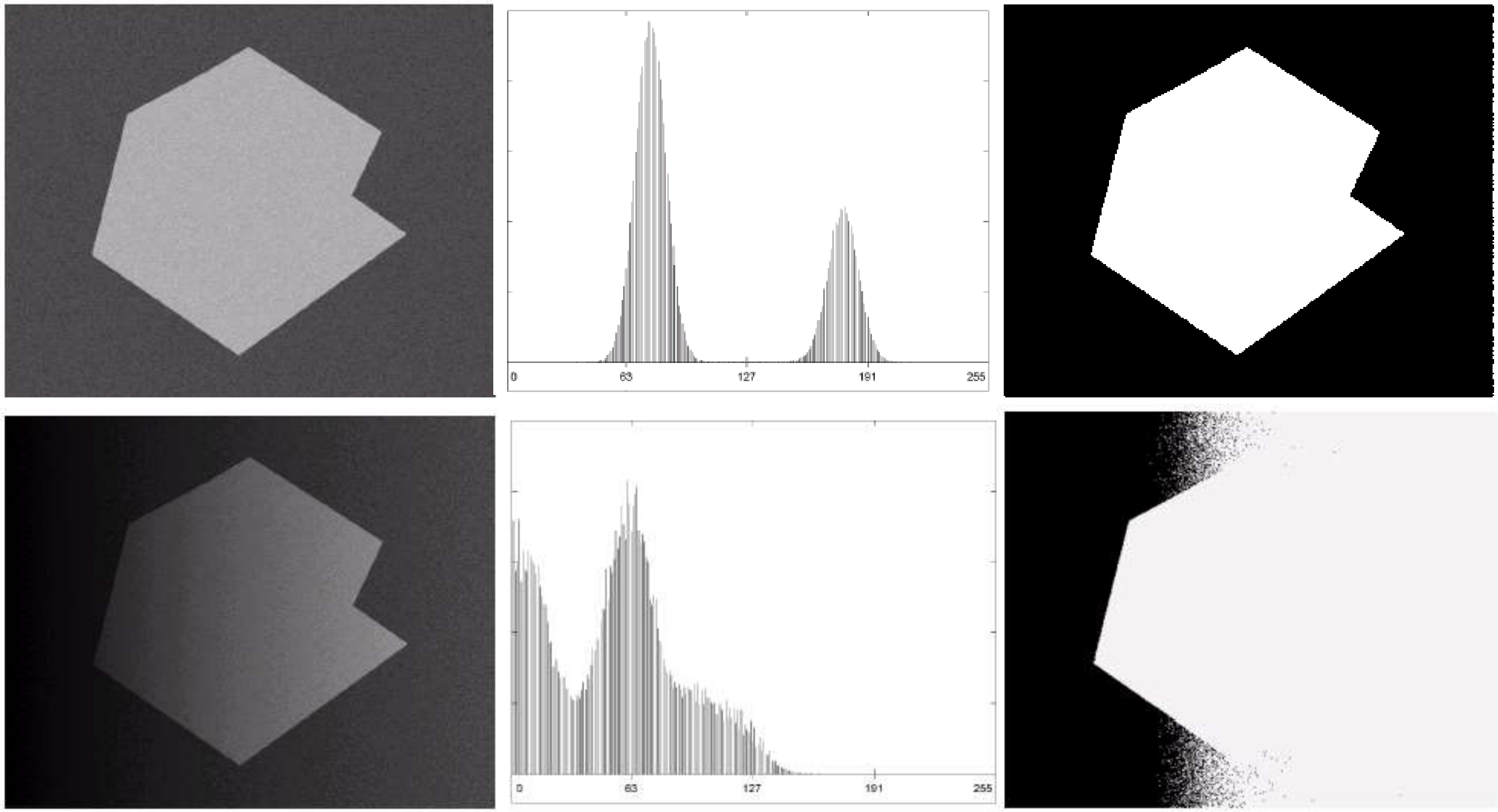
Problems With Single Value Thresholding

Single value thresholding only works for bimodal histograms.

Images with other kinds of histograms need more than a single threshold.



Single Value Thresholding and Illumination



Uneven illumination can really upset a single valued thresholding scheme

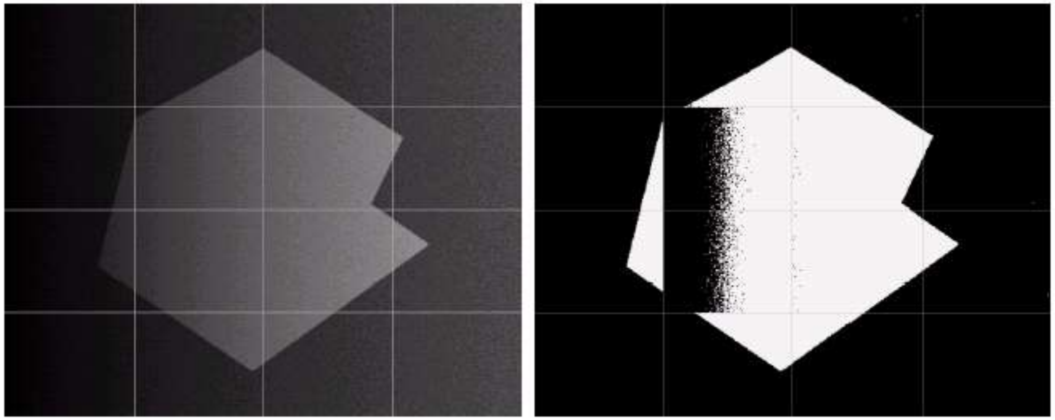
Basic Adaptive Thresholding

An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually.

Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive.*

Basic Adaptive Thresholding Example

The image below shows an example of using adaptive thresholding with the image shown previously.

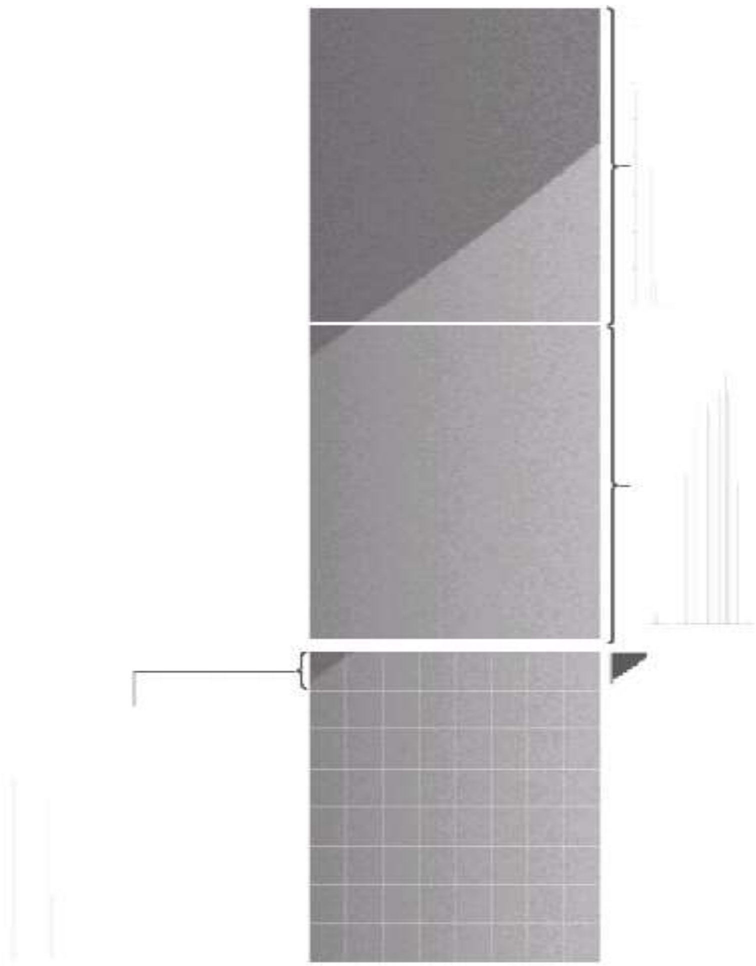


As can be seen success is mixed. But, we can further subdivide the troublesome sub images for more success.

Basic Adaptive Thresholding Example (cont…)

These images show the troublesome parts of the previous problem further subdivided.

After this sub division successful thresholding can be achieved.



Otsu's Thresholding Concept

Automatic global thresholding algorithms usually have the following steps.

1. Process the input image
2. Obtain image histogram (distribution of pixels)
3. Compute the threshold value *T*
4. Replace image pixels with white in those regions, where saturation is greater than *T* and into the black in the opposite cases.

– Usually, different algorithms differ in step 3.

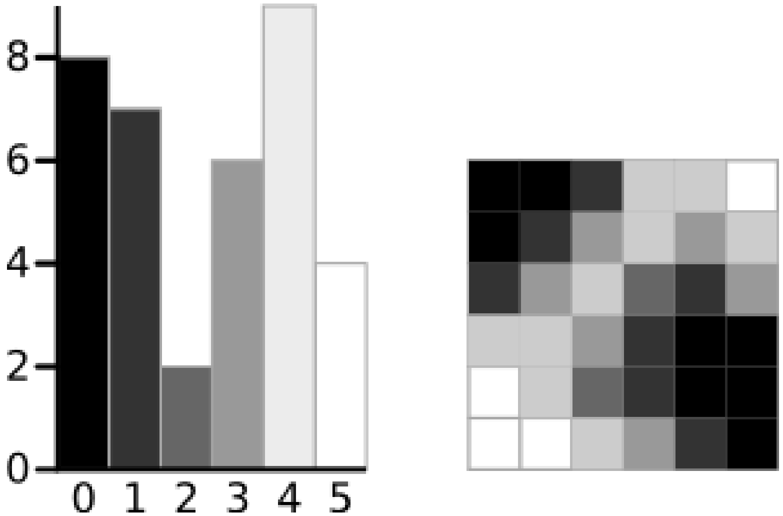
Otsu's Thresholding Concept

The core idea is to separate the image histogram into two clusters with a threshold defined as a result of minimization of the weighted variance of these classes denoted by σ_w^2

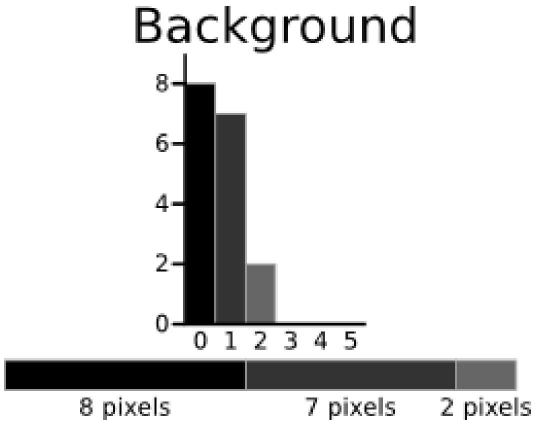
Where , W_b, W_f are the probabilities of the two classes divided by a threshold t, Which value is within the range from 0 to 255 inclusively.

$$\sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2$$

Otsu's Thresholding Concept



Otsu's Thresholding Concept

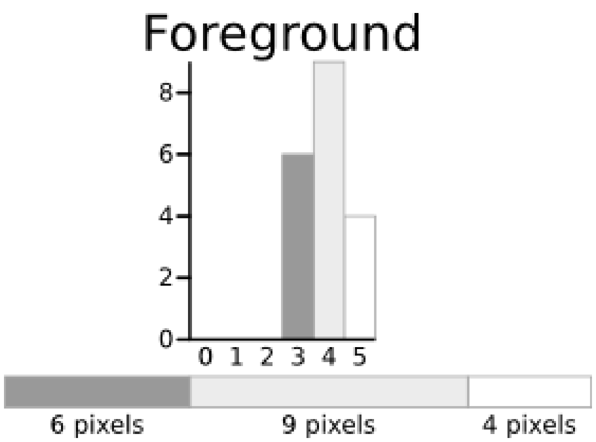


$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\begin{aligned}\text{Variance} \quad \sigma_b^2 &= \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17} \\ &= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17} \\ &= 0.4637\end{aligned}$$

Otsu’s Thresholding Concept



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

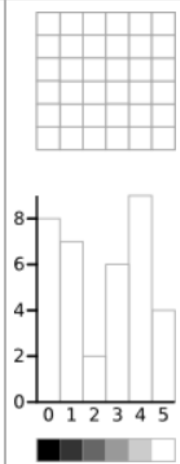
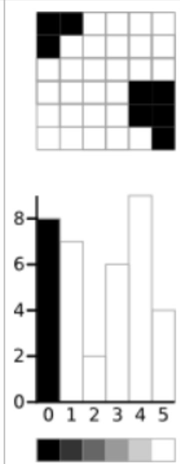
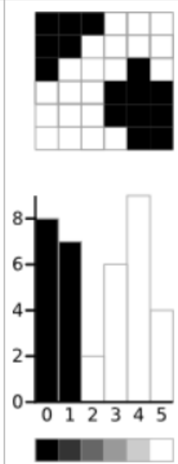
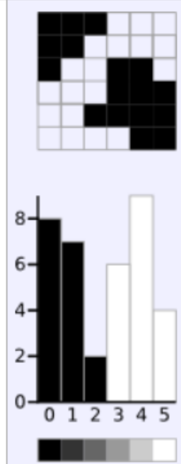
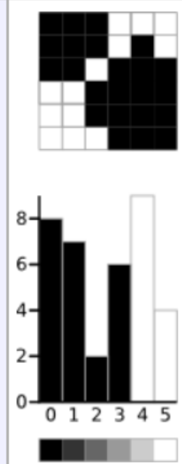
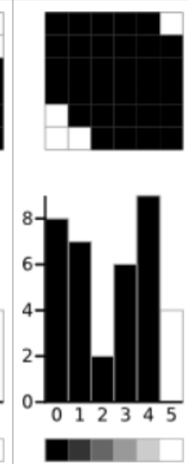
$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

$$= 0.5152$$

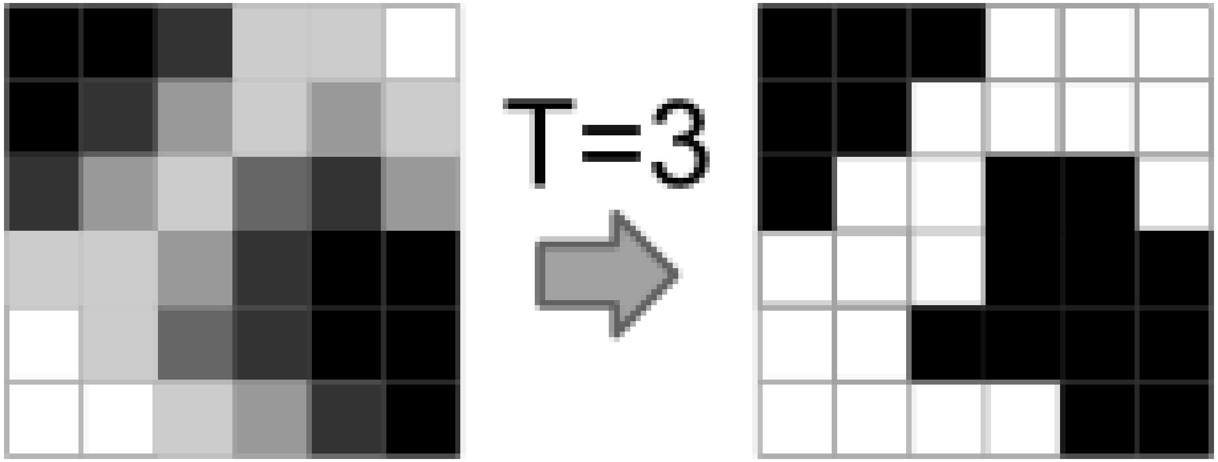
Otsu's Thresholding Concept

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152$$
$$= 0.4909$$

Otsu’s Thresholding Concept

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2{}_b = 0$	$\sigma^2{}_b = 0$	$\sigma^2{}_b = 0.2489$	$\sigma^2{}_b = 0.4637$	$\sigma^2{}_b = 1.4102$	$\sigma^2{}_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2{}_f = 3.1196$	$\sigma^2{}_f = 1.9639$	$\sigma^2{}_f = 0.7755$	$\sigma^2{}_f = 0.5152$	$\sigma^2{}_f = 0.2130$	$\sigma^2{}_f = 0$
Within Class Variance	$\sigma^2{}_W = 3.1196$	$\sigma^2{}_W = 1.5268$	$\sigma^2{}_W = 0.5561$	**$\sigma^2{}_W = 0.4909$**	$\sigma^2{}_W = 0.9779$	$\sigma^2{}_W = 2.2491$

Otsu's Thresholding Result



Result of Otsu's method