Image Segmentation

Michael A. Wirth, Ph.D.
University of Guelph
Computing and Information Science
Image Processing Group

Image Segmentation

- Image segmentation partitions an image into disjoint (non-overlapping) regions.
  - A region is a connected set of pixels
  - A connected set is one in which all the pixels are adjacent or touching
  - These regions correlate strongly with objects or features of interest.
  - Can also be regarded as a process of grouping together pixels that have similar attributes.

Image Segmentation

- Segmentation is generally the first stage in any attempt to analyze or interpret an image automatically.
- Applications involving the detection, recognition and measurement of objects in images.

Image Segmentation

- Segmentation texhniques can be classified as either contextual or non-contextual.
  - Non-contextual techniques ignore the relationships that exist between features in an image; pixels are simply grouped together on the basis of some global attribute

    e.g. intensity value
  - Contextual techniques exploit the relationships between image features → similar intensities, *and* spatial proximity

Image Thresholding

Michael A. Wirth, Ph.D.
University of Guelph
Computing and Information Science
Image Processing Group

Thresholding

- Thresholding is a simple, non-contextual technique
- Image thresholding is a segmentation technique which classifies pixels into two categories:
  - Those to which some property measured from the image falls below a threshold, and those at which the property equals or exceeds a threshold.
  - Thesholding creates a binary image → binarization

  e.g. perform cell counts in histological images

Thresholding

- Depends on the property being thresholded
- For edge detection → a measure of the strength of an edge (e.g. intensity gradient)
  - A value of 0 if the gradient falls below the threshold (not considered to be a "proper edge")
  - A non-zero value if the gradient matches or exceeds the threshold (indicates this pixel is a proper edge)
- *Fixed* or *adaptive* threshold values

Threshold Values & Histograms

- Thresholding usually involves analyzing the histogram
  - Different features give rise to distinct features in a histogram
  - In general the histogram peaks corresponding to two features will overlap. The degree of overlap depends on peak separation and peak width.
- An example of a threshold value is the mean intensity value

Intensity Thresholding

- Thresholding can be implemented in two ways:
  - Iterate over every pixel
  - Apply these equations once for all intensity values and store the results in a look-up table, which can be used to map the intensity level of each pixel to 0 or 1

Fixed Thresholding

- In fixed or global thresholding, the threshold value is held constant throughout the image:
  - Determine a single threshold value by treating each pixel independently of its neighborhood.
- Fixed thresholding is of the form:

$$g(x,y) = \begin{cases} 0 & f(x,y) < T \\ 1 & f(x,y) \geq T \end{cases}$$

where T is the threshold

- Assumes high-intensity pixels are of interest, and low-intensity pixels are not.

Fixed Thresholding

- To detect low-intensity features:

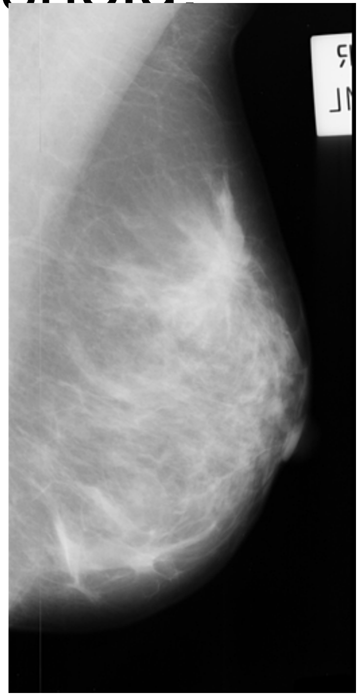
$$g(x,y)=\begin{cases}1 & f(x,y)\le T\\ 0 & f(x,y)>T\end{cases}$$

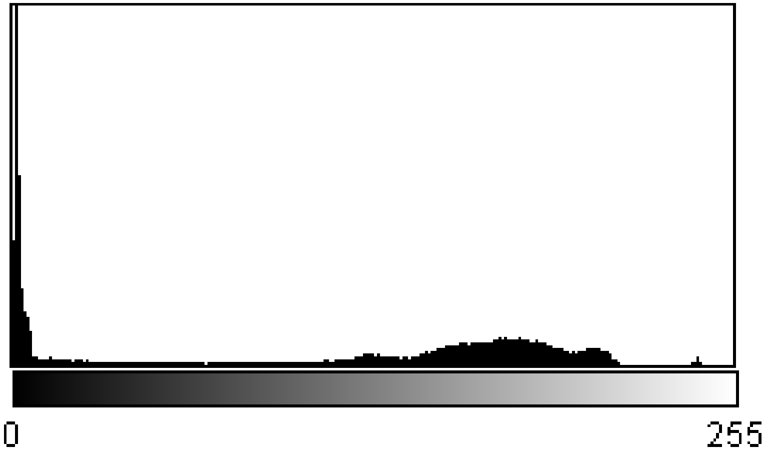
- A variation which uses two thresholds to define a range of intensity values

$$g(x,y)=\begin{cases}0 & f(x,y)<T_1\\ 1 & T_1\le f(x,y)\le T_2\\ 0 & f(x,y)>T_2\end{cases}$$

Fixed Thresholding

- The success of thresholding depends critically on the selection of an appropriate threshold.

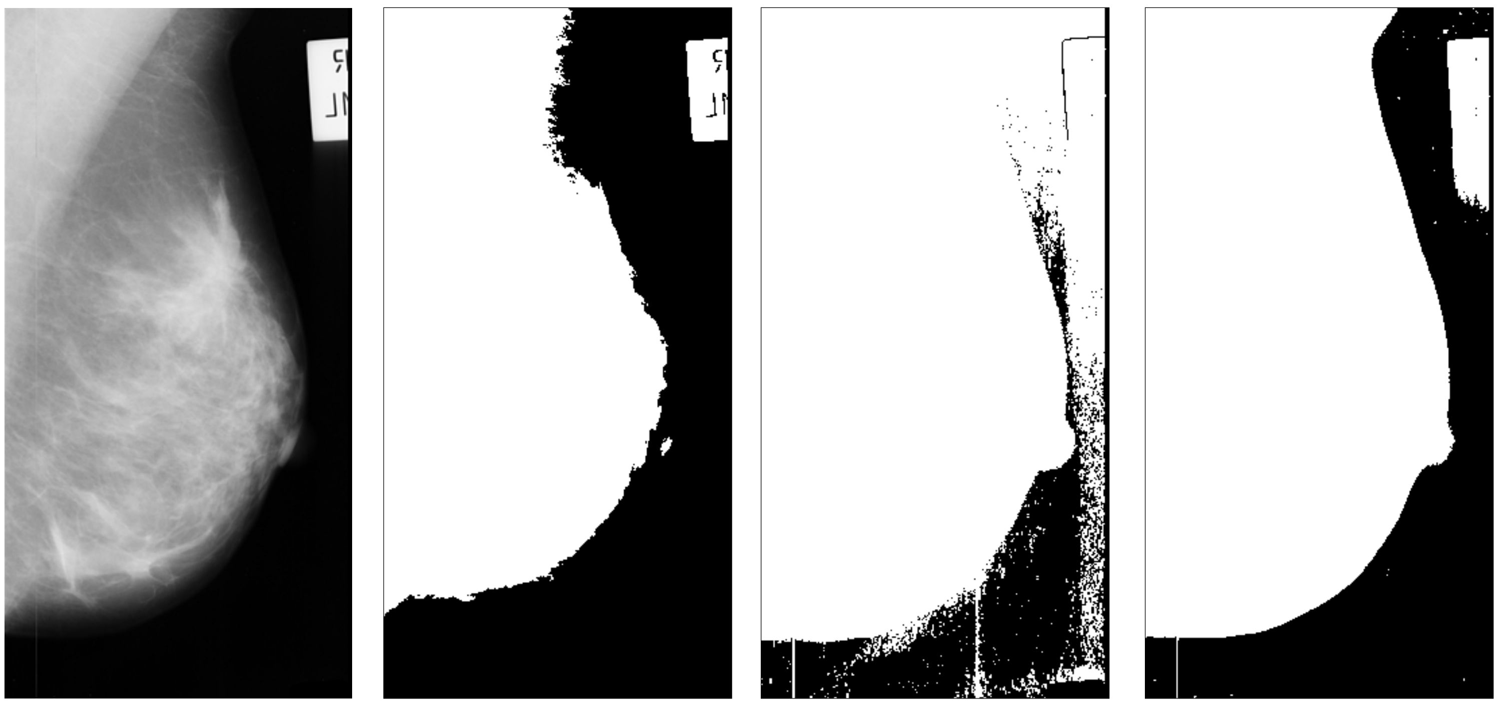




Fixed Thresholding:

Single Thresholds

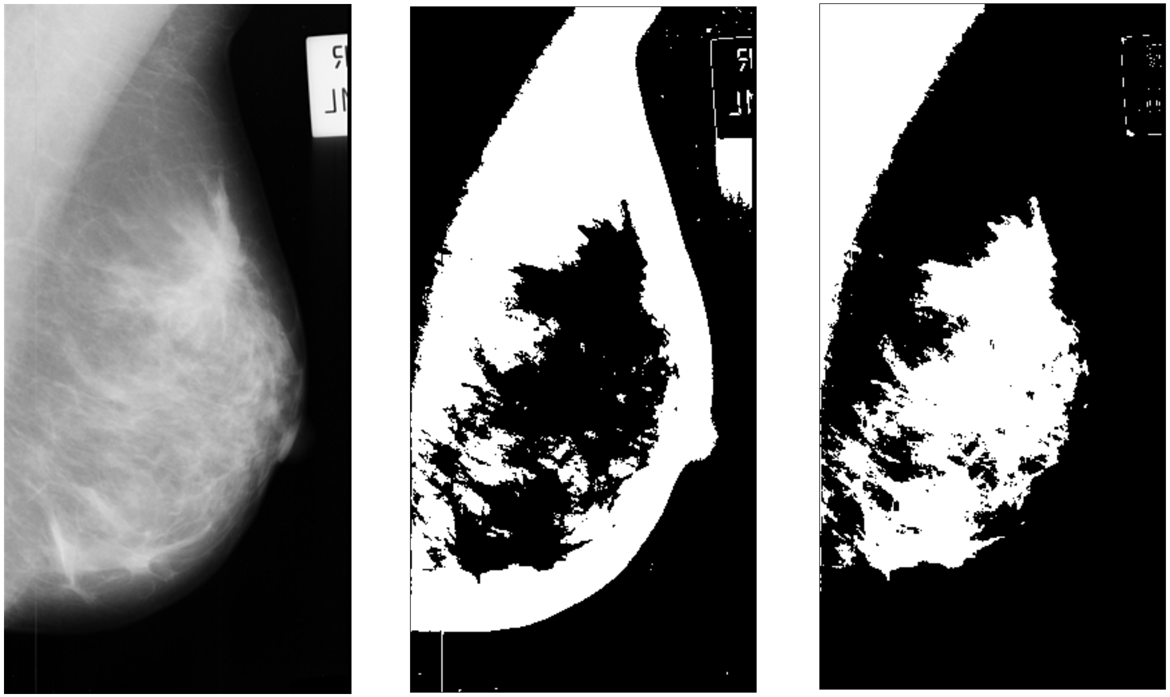
Too high Too low Correct



T=111 T=3 T=8

Fixed Thresholding:

Double Thresholds



T_1=8, T_2=169 T_1=169, T_2=223

Iterative Threshold Selection

1. Select an initial estimate of the threshold, *T*. (A good initial value is the mean intensity.
2. Partition the image into two groups, R_1 and R_2, using the threshold, *T*.
3. Calculate the mean intensity values μ_1 and μ_2 of the partitions R_1 and R_2.
4. Select a new threshold: $$T = \frac{(\mu_1 + \mu_2)}{2}$$
5. Repeat steps 2-4 until the mean values μ_1 and μ_2 do not change in successive iterations

Optimal Thresholding

- Histogram shape can be useful in locating the threshold.
  - However it is not reliable for threshold selection when peaks are not clearly resolved.
  - A “flat” object with no discernable surface texture, and no colour variation will give rise to a relatively narrow histogram peak.
  - An object with pronounced surface relief or significant variations in texture or colour across its surface will produce a broader peak that may overlap with the peak generated by the background.

Optimal Thresholding

- Choosing a threshold I the valley between two overlapping peaks, and inevitably some pixels will be incorrectly classified by the thresholding.

- In optimal thresholding, a *criterion function* is devised that yields some measure of separation between regions.
  - A criterion function is calculated for each intensity and that which maximizes this function is chosen as the threshold.

Otsu's Method

- Otsu's thresholding method is based on selecting the lowest point between two classes (peaks).
  - Formulated as discriminant analysis: a particular criterion function is used as a measure of statistical separation.
- Analysis of variance (variance=standard deviation^2)
  - Separately compute the variance of the two classes

σ_T^2 = total variance

σ_w^2 = within-class variance

  - The variation of the mean values for each class from the overall intensity mean of all pixels defines a between-classes variance: σ_b^2

Otsu's Method

- The criterion function involves minimizing the ratio of the between-classes variance to the total variance :

$$\eta(t) = \frac{\sigma_b^2}{\sigma_t^2}$$

- The value of *t* which gives the smallest value for *n* is the optima threshold.
  - σ_T^2 and the overall mean μ_T are derived from the image
  - The between-classes variance is calculated as:

$$\sigma_b^2 = w_0 w_1 (\mu_0 \mu_1)^2$$

Otsu's Method

where $$w_0 = \sum_{i=0}^{t} p_i \qquad w_1 = 1 - w_0$$

p_i is the probability of intensity value i, or the histogram value at i divided by the total number of pixels, and

$$\mu_0 = \frac{\mu_t}{w_0}, \qquad \mu_1 = \frac{\mu_T - \mu_t}{1 - w_0}, \qquad \mu_t = \sum_{i=0}^{t} i \cdot p_i$$

$\eta(t)$ is calculated for all possible values of t and the t that gives the smallest η is the optimal threshold

Entropy

- Entropy is a measure of information content
  - It serves as a measure of separation
  - Separates the information into two regions, above and below an intensity threshold, and measures the entropies of each class.
  - The separation is done for each intensity value and the value for which the sum of entropies of the two classes is maximum is the optimal threshold.

Entropy

- Using a threshold, t, the entropy associated with the pixels 0→t is:

$$H_b = -\sum_{i=0}^{t} p_i \log(p_i)$$

- whilst the entropy associated with the pixels t+1→ 255 is:

$$H_w = -\sum_{i=t+1}^{255} p_i \log(p_i)$$

- Find a threshold, t which maximizes $H = H_b + H_w$

Moment Preservation

- The objective is to choose a threshold such that the resulting thresholded image best preserves the mathematical moments of the original image.
  - Moments are calculated for the original image.
  - The moments are calculated for images resulting from every possible threshold.
  - The threshold value at which the original and thresholded images have the closest moments is said to be the optimal threshold.

Moment Preservation

- The first *k* moments of the grayscale image are evaluated directly from the intensity histogram:

$$m_k = \sum_{i=0}^{l-1} p_i i^k$$

p_i = probability of intensity value *i* between 0 and *i –1*, *k* is the order of the moment.

m_0 is defined to be 1.

Minimum Error

- Assumes the histogram is composed of two normally distributed classes of intensities.
  - Two normal distribution curves are determined by an iterative process to fit the two classes of pixels and minimize a specified classification error.
  - On each iteration, a prospective threshold value is tested by calculation of the means and variances from the histogram for the two classes separated by this threshold.
  - The criterion function is minimized to find the best fit between the statistical model and the histogram.

Minimum Error

- The value of t that minimizes $J(t)$ is the optimal threshold

$$J(t) = 1 + 2(P_1(t)\log\sigma_1(t) + P_2(t)\log\sigma_2(t)) \\ -2(P_1(t)\log P_1(t) + P_2(t)\log P_2(t))$$

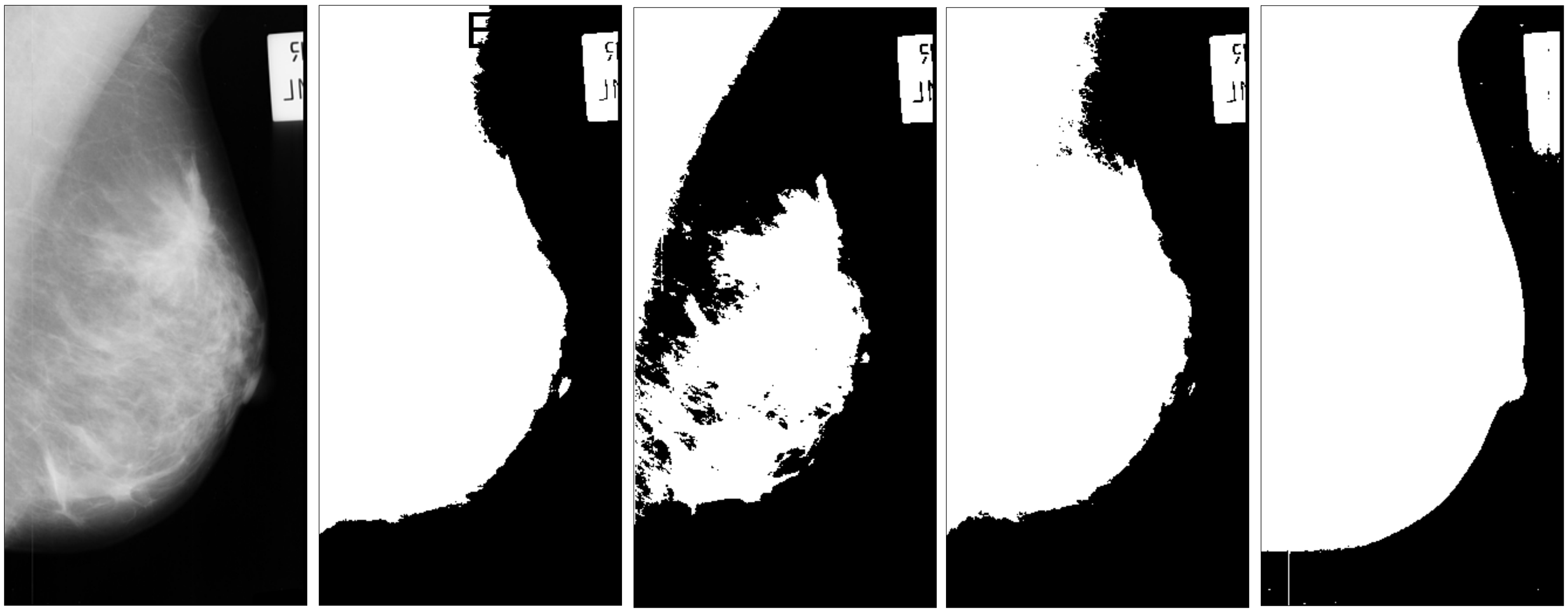
where

$$P_1(t) = \sum_{g=0}^{t} h(g) \qquad \mu_1(t) = \frac{\sum_{g=0}^{t} g \cdot h(g)}{P_1(t)} \qquad \sigma_1^2(t) = \frac{\sum_{g=0}^{t} h(g)(g-\mu_1(t))^2}{P_1(t)}$$

$$P_2(t) = \sum_{g=t+1}^{255} h(g) \qquad \mu_2(t) = \frac{\sum_{g=t+1}^{255} g \cdot h(g)}{P_2(t)} \qquad \sigma_2^2(t) = \frac{\sum_{g=t+1}^{255} h(g)(g-\mu_2(t))^2}{P_2(t)}$$

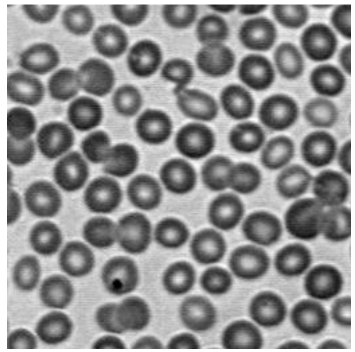
Comparing Threshold Values

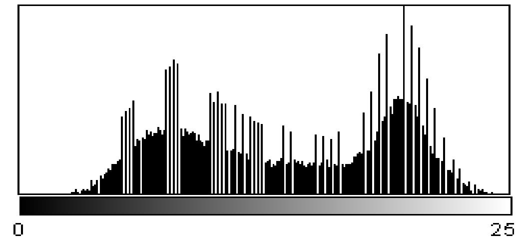
Otsu | Entropy | Moments | Minimum

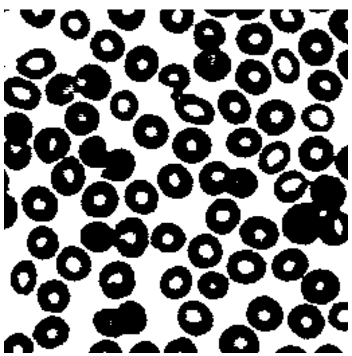
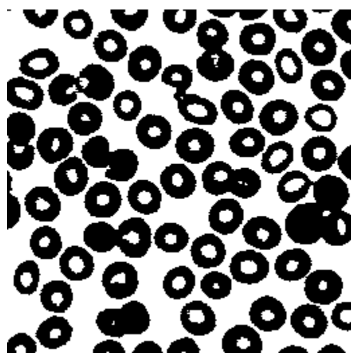
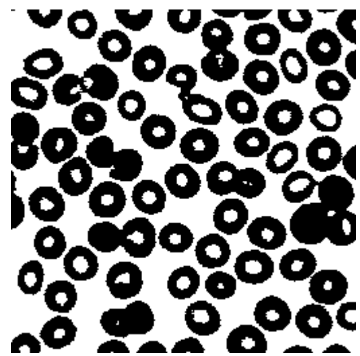
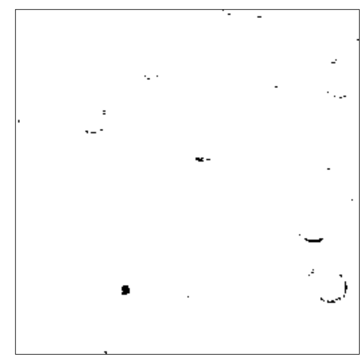


T=99 | T=161 | T=118 | T=10

Comparing Threshold Values





Otsu	Entropy	Moments	Minimum Error
			
T=138	T=135	T=135	T=32

Other Approaches

- Fuzzy Sets:
  – Huang, L-K., and Wang, M-J.J., “Image thresholding by minimizing the measures of fuzziness”, *Pattern Recognition*, 1995, **28**(1):pp.41-51

Adaptive Thresholding

- In adaptive thresholding, the threshold value varies throughout the image:
  - Sometimes known as regional thresholding
  - No single value can threshold the whole image.
  - Works when the background intensity level is not constant and the object varies within the image.
  - Examines the relationships between intensities of neighboring pixels to adapt the threshold according to the prevailing intensity statistics of different regions.

Thresholding Difficulties

- Poor image contrast (difficult to resolve the foreground from the background)
  - Corresponding pixels tend to overlap
- Spatial nonuniformities in the background itensity
  - Image appears light on one side and dark on the other
- Ambiguity between foreground (object) and background pixels

Edge Detection

Michael A. Wirth, Ph.D.
University of Guelph
Computing and Information Science
Image Processing Group

Contextual Segmentation

- Approaches to contextual segmentation are based on the concept of discontinuity or the concept of similarity.
  - Techniques based on discontinuity attempt to partition the image by detecting abrupt changes in intensity value.

    e.g. edge-detection techniques
  - Techniques based on similarity attempt to create uniform regions by grouping together pixels that satisfy predefined similarity criteria

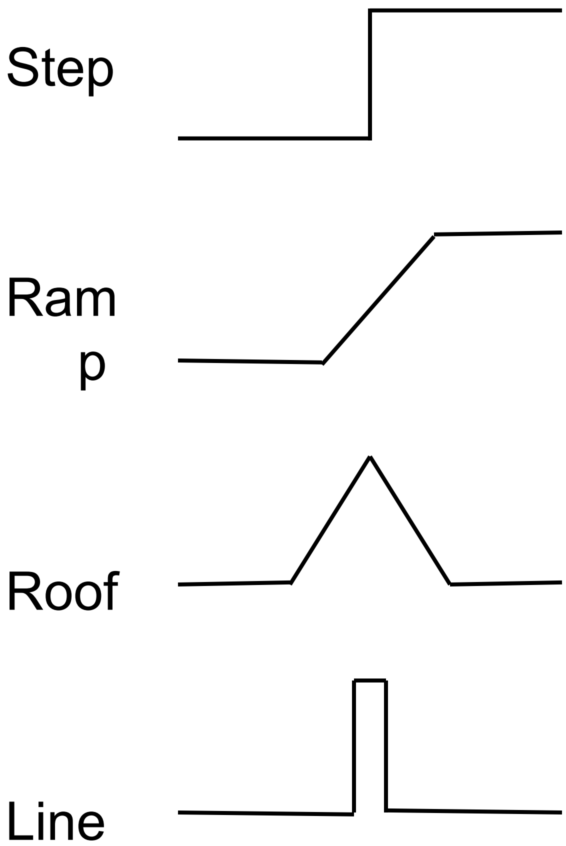
Edge Detection

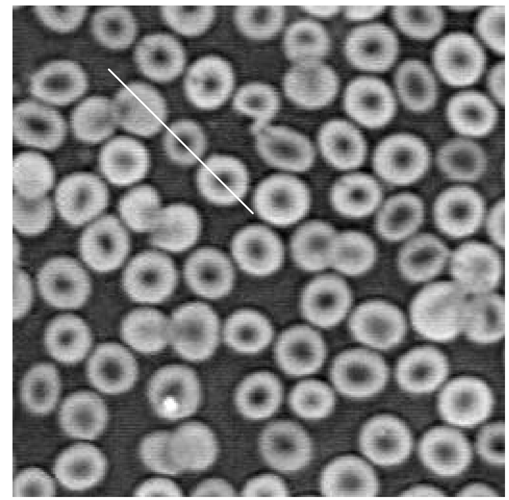
- Edges can be loosely defined as locations in an image where there is a sudden change (discontinuity) in the intensity of pixels.
  - significant local intensity changes in the image
  - important clues to separate regions within an object
  - no single best definition of what constitutes an edge

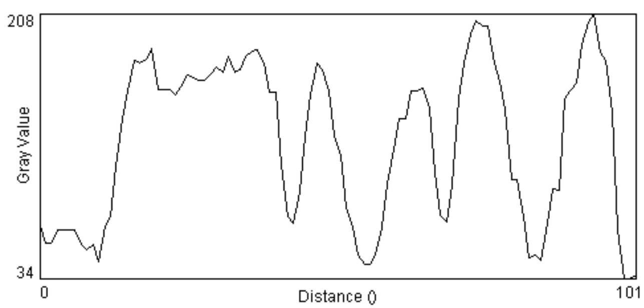
Types of Edges

- Discontinuities in the image intensity can be either:
  - *Step* edges where the intensity abruptly changes from one value on one side of the discontinuity to a different vaue on the opposite side
  - *Line* edges where the intensity abruptly changes value but then returns to the starting value within a short distance
  - Step edges become *ramp* edges and line edges become *roof* edges, where intensity changes are not instantaneous, but occur over a finite distance

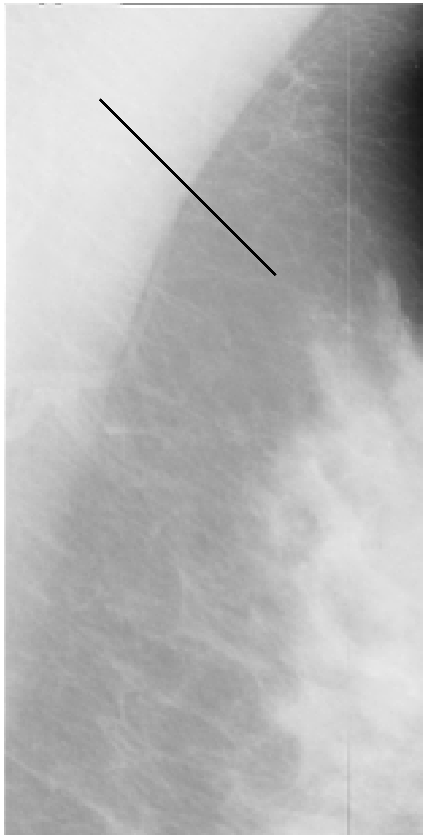
Types of Edges







Types of Edges



Steps in Edge Detection

- Edge detection techniques aim to detect significant local changes in an image.
- There are typically three steps to perform:
  - Noise reduction: suppress as much noise as possible, without smoothing away meaningful edges (more filtering to reduce noise results in a loss of edge strength)
  - Edge enhancement: apply a filter that responds strongly at edges and weakly elsewhere: edges may be identified as local maxima (sharpening filter)
  - Edge detection: decide which of the local maxima are meaningful edges, and which are caused by noise.

    e.g. thresholding

First-order Derivatives

- Imagine an image to be a surface with height corresponding to intensity level.
- 1^{st} order derivatives measure the local slope of this surface in the x and y directions

Gradient Edge Detection

- Express the gradient calculation as a pair of convolution operations:
  - One kernel responds maximally to a vertical edge, and the other to a horizontal edge.

$$g_x(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_x(i,j) f(x+i, y+j)$$

$$g_y(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_y(i,j) f(x+i, y+j)$$

w_x is the horizontal derivative filter

w_y is the vertical derivative filter

Gradient Edge Detection

- A gradient kernel is of the form:

$$\begin{bmatrix} i-1, j-1 & i-1, j & i-1, j+1 \\ i, j-1 & \boxed{i, j} & i, j+1 \\ i+1, j-1 & i+1, j & i+1, j+1 \end{bmatrix}$$

Gradient Edge Detection

- The two gradients computed using w_x and w_y can be regarded as the x and y components of a gradient vector:

$$G = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

- This vector is oriented along the direction of change, normal to the direction in which the edge runs.

Gradient Edge Detection

- The gradient magnitude and direction are given by:

$$G_M(x,y) = \sqrt{g_x(x,y) + g_y(x,y)}$$

$$G_\theta(x,y) = \tan^{-1}\left(\frac{g_y(x,y)}{g_x(x,y)}\right)$$

  where θ is measured relative to the *x*-axis
- The magnitude of the gradient is independent of the direction of the edge.

Gradient Edge Detection

- The gradient magnitude is sometimes approximated using:

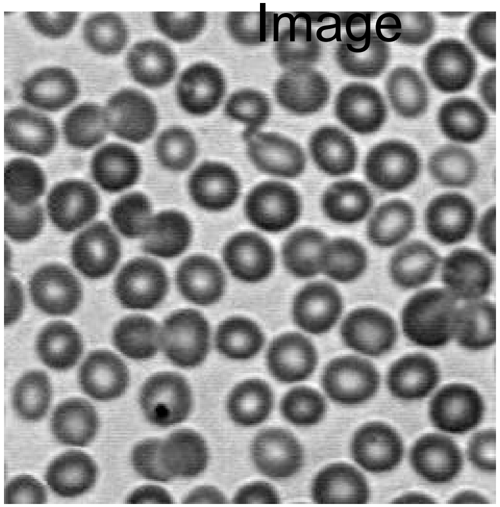
$$G_M(x,y) = \left|g_x(x,y)\right| + \left|g_y(x,y)\right|$$

or

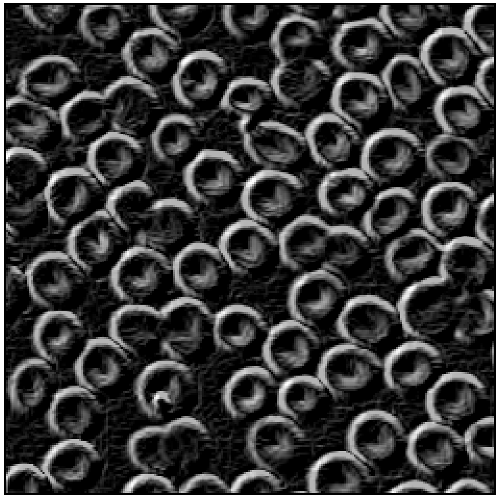
$$G_M(x,y) = \max\left(\left|g_x(x,y)\right|, \left|g_y(x,y)\right|\right)$$

Gradient Edge Detection

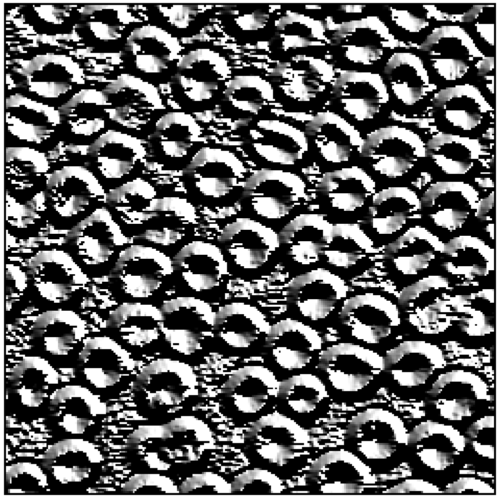
Original Image



Magnitude Image



Direction



Gradient Edge Detection

- The simplest gradient approximation is:

$$g_x(x,y) \cong f(x,y+1) - f(x,y)$$
$$g_y(x,y) \cong f(x,y) - f(x+1,y)$$

which corresponds to the simple convolution kernels:

$$w_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \qquad w_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Roberts Operator

- The Roberts cross-operator provides a simple approximation to the gradient magnitude
  - Responds well to sharp transitions in low-noise images

$$g_x(x,y) \cong f(x,y) - f(x+1, y+1)$$

$$g_y(x,y) \cong f(x+1, y) - f(x, y+1)$$

which corresponds to the simple convolution kernels:

$$w_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad w_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Prewitt Operator

- The Prewitt filter is an estimate of the maximum gradient.
- The two convolution kernels of the Prewitt filter have the form:

$$w_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad w_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Sobel Operator

- The Sobel filter is similar to the Prewitt filter, except that in estimating the maximum gradient it gives more weight to the pixels nearest (x,y).
- The two convolution kernels of the Sobel filter have the form:

$$w_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad w_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Kirsch Filter

- The Kirsch filter is a maximum response filter which is sensitive to edges at different orientations
  - Consists of eight convolution kernels oriented in directions 45° apart
  - Applies each of the eight orientations of the derivative kernel and retains the maximum value

$$G(x,y) = \max_{z=1,\ldots,8} \sum_{i=-1}^{1} \sum_{j=-1}^{1} w^{z}(i,j) f(x+i, y+j)$$

Kirsch Filter

- The eight convolution kernels of the Kirsch filter w^z are:

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

Frei & Chen Filter

- Apply a set of kernels to each point in an image
  - Each kernel extracts one kind of behavior in the image
  - The kernels are orthogonal or independent basis functions
- The results of applying each kernel to each pixel are summed to produce a ratio

$$G_R(x,y) = \sum_{z=0}^{8} \left[\sum_{i=-1}^{1} \sum_{j=-1}^{1} w^z(i,j) f(x+i, y+j) \right]$$

Frei & Chen Filter

- The cosine of the square-root of this value is effectively the vector projection of the information from the neighborhood in the direction of the "edgeness"

$$G(x,y) = \cos(\sqrt{G_R(x,y)})$$

- The advantage over Sobel is that the Frei & Chen operator is sensitive to a configuration of relative pixel values independent of the magnitude of the brightness
  - Frei, W., Chen, C.C., "Fast boundary detection: A generalization and a new algorithm", IEEE Transactions on Computing, 1977, **26**: pp.988-998

Frei & Chen Filter

- The convolution kernels of the Frei & Chen filter w^z are:

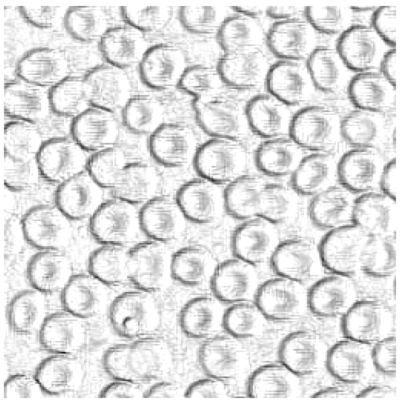
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ \sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & \sqrt{2} \end{bmatrix}$$

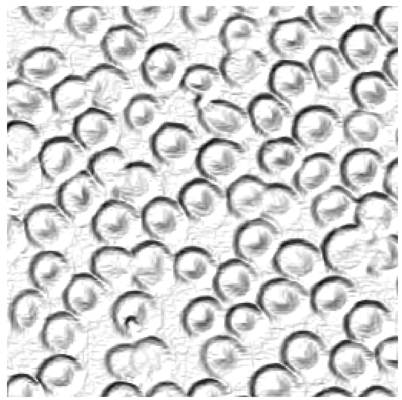
$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

Comparing Edge Detectors

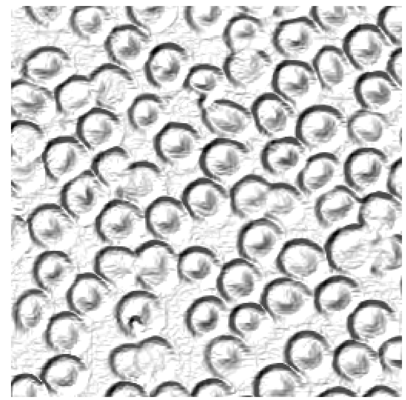
Roberts



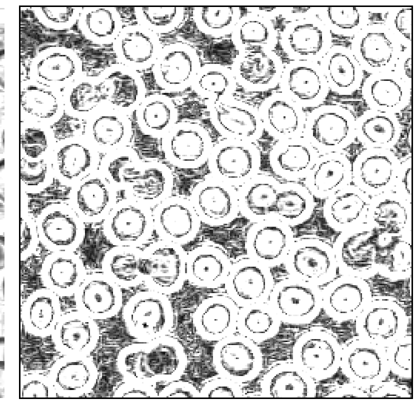
Prewitt

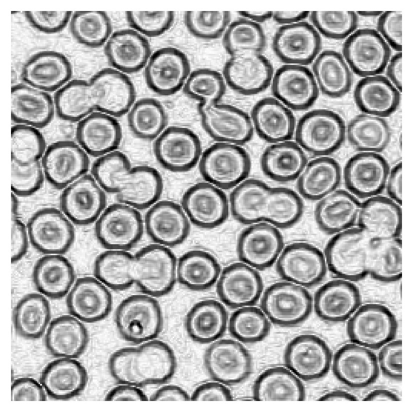


Sobel



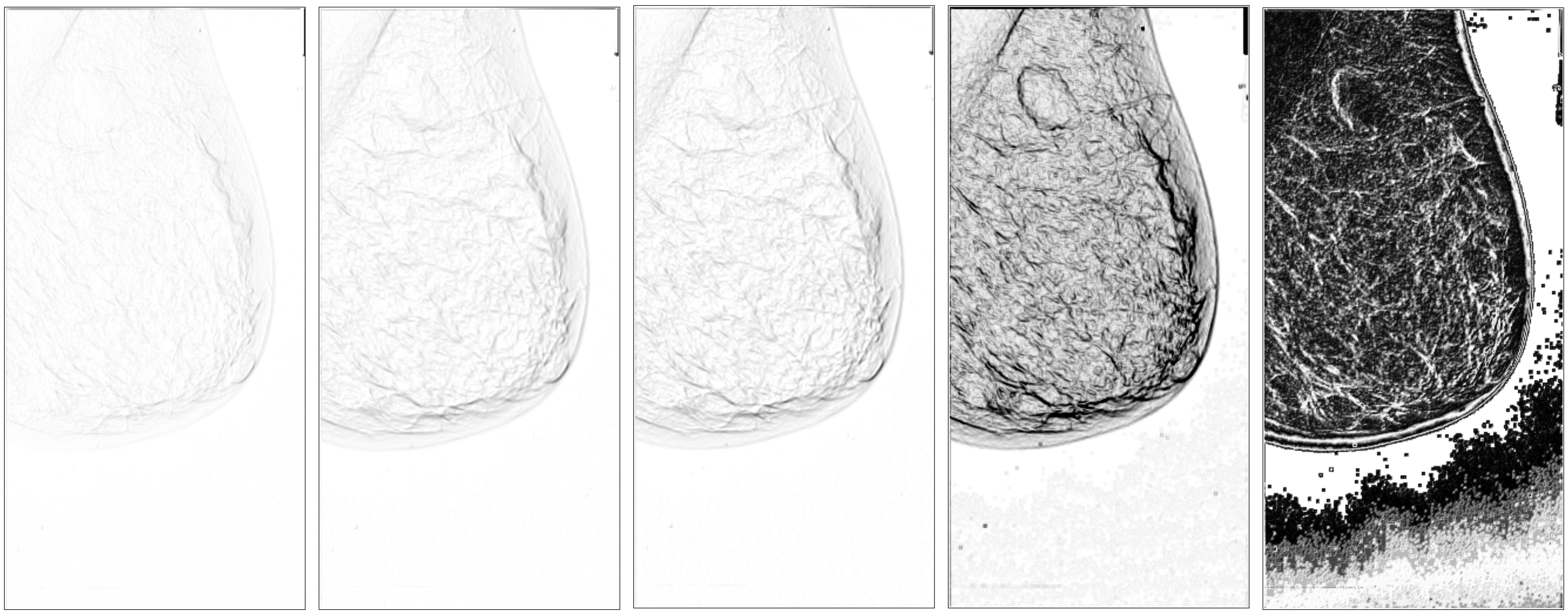
Kirsch





Comparing Edge Detectors

Roberts Prewitt Sobel Kirsch Frei & Chen



Separable Kernels

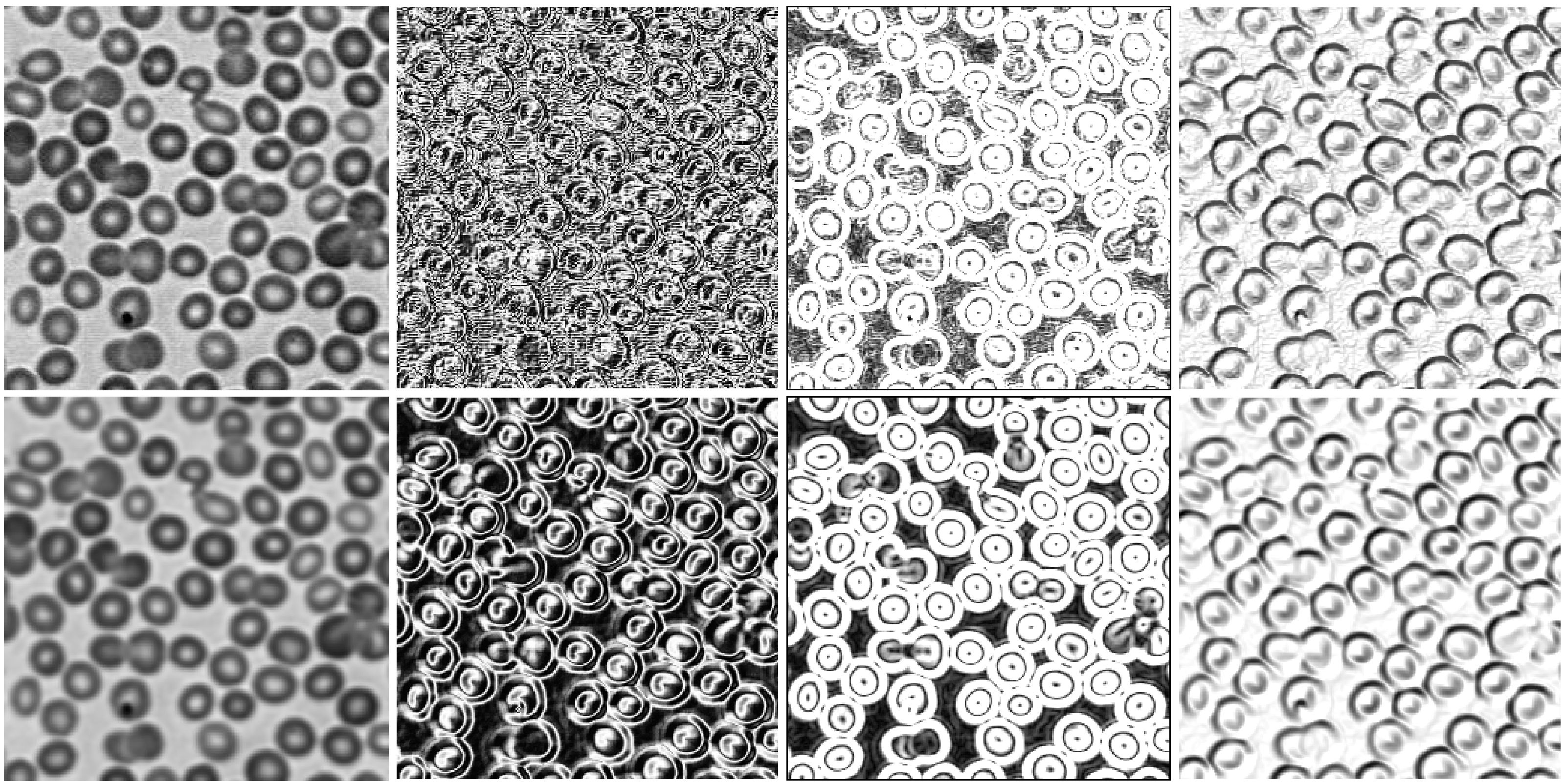
- In many cases both h_x and h_y are separable
  - Each filter takes the derivative in one direction and smoothes in the orthogonal direction

$$w_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$w_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Smoothing → Edge Detection

Original | Frei & Chen | Kirsch | Sobel



Smoothed | Frei & Chen | Kirsch | Sobel

Second-Derivative Operators

- 2^{nd} order derivatives measure the rate at which the slope of the intensity surface changes with distance travelled in the x and y directions.
  - At edge points there will be a peak in the first derivative and, equivalently, there will be a zero crossing in the second derivative
  - It changes sign at the centre of the edge
  - Edge points can be localised by finding the zero crossings of the second-derivative

Second-Derivative Operators

  - Second-derivative filters can be isotropic, and therefore responsive to edges in any direction
- There are two operators that correspond to the second derivative:
  - the Laplacian, and
  - the Second Directional Derivative

Laplacian Operator

- The Laplacian is the 2D equivalent of the 2nd derivative

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

- 1D approximation

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

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

Laplacian Operator

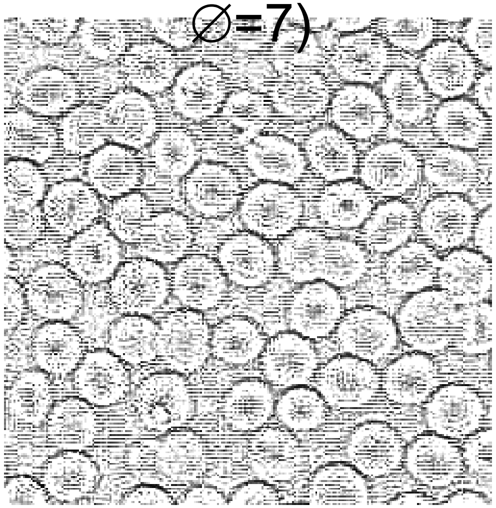
- 2D approximation

$$w = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad w = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad w = \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

Laplacian Operator

- Examples of the Laplacian (after contrast enhancement using histogram equalisation)

No smoothing



Smoothing (Gaussian $\varnothing$=7)