Chapter 10
Image Segmentation

Generally, image segmentation algorithms are based on one of two basic properties of intensity values:

1. **Discontinuity**
   - Point detection
   - Line detection
   - Edge detection

2. **Similarity**
   - Thresholding
   - Region growing
   - Region splitting and merging
A General 3x3 Mask

The response of the mask with respect to its center location is

\[ R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 \]

\[ = \sum_{i=1}^{9} w_i z_i \]

\( z_i \) is the gray level of the pixel associated with mask coefficient \( w_i \)
Point Detection

If \( |R| \geq T \), where \( T \) is a nonnegative threshold, then a point is detected.

\[
T = 90\% \times \text{max}
\]

Point detector mask

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>8</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>
## Line Detection

### Line Masks

<table>
<thead>
<tr>
<th>Horizontal</th>
<th>$+45^\circ$</th>
<th>Vertical</th>
<th>$-45^\circ$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 -1 -1</td>
<td>-1 -1 2</td>
<td>-1 2 -1</td>
<td>2 -1 -1</td>
</tr>
<tr>
<td>2 2 2</td>
<td>-1 2 -1</td>
<td>-1 2 -1</td>
<td>-1 2 -1</td>
</tr>
<tr>
<td>-1 -1 -1</td>
<td>2 -1 -1</td>
<td>-1 2 -1</td>
<td>-1 -1 2</td>
</tr>
</tbody>
</table>
Illustration of Line Mask

- 45° line detection

Absolute value of result after using -45° line detector

Thresholded image

Threshold = max. value in the left image

Figure 10.4
Illustration of line detection.
(a) Binary wire-bond mask.
(b) Absolute value of result after processing with -45° line detector.
(c) Result of thresholding image (b).
Edge Detection

An Ideal Digital Edge v.s. A Ramp Digital Edge

The slope of the ramp is inversely proportional to the degree of blurring in the edge. Blurred edges tend to be thick and sharp edges tend to be thin.

FIGURE 10.5
(a) Model of an ideal digital edge.
(b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.
Edge Detection

The magnitude of the first derivative can be used to detect if a point is on the ramp.

The sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge.
Edge Detection

Two additional properties of the second derivative:

1. It produces two values for every edge in an image.
2. An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. (can be used to locate the centers of thick edges)
First and Second Derivatives around a Noisy Edge

Free of noise

Corrupted by additive Gaussian noise, $m=0, \sigma =0.1$

Corrupted by additive Gaussian noise, $m=0, \sigma =1.0$

Corrupted by additive Gaussian noise, $m=0, \sigma =10.0$

The second order derivative is sensitive to the noise.
Gradient Operators

First-order derivatives in an image are computed using the gradient.

The gradient of an image \( f(x,y) \) at location \( (x,y) \) is defined as:

\[
\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}
\]

The magnitude of this vector is:

\[
\nabla f = mag(\nabla f) = \left[ G_x^2 + G_y^2 \right]^{\frac{1}{2}} \quad \nabla f \approx |G_x| + |G_y|
\]

The direction of the gradient vector:

\[
\alpha(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right)
\]
Gradient Operators

A 3x3 area in an image

\[ G_x = z_9 - z_5 \quad G_y = z_8 - z_6 \]

\[ G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \]
\[ G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \]

\[ G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \]
\[ G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \]
Masks for Detecting Diagonal Edges

**FIGURE 10.9** Prewitt and Sobel masks for detecting diagonal edges.
Illustrations of the Gradient and Its Components

**Figure 10.10**
(a) Original image. (b) $|G_x|$, component of the gradient in the $x$-direction. (c) $|G_y|$, component in the $y$-direction. (d) Gradient image, $|G_x| + |G_y|$.
Illustrations of the Gradient and Its Components

To smooth the contribution made by the wall bricks

Image smoothed by 5x5 averaging filter

\[ |G_x| \]

\[ |G_y| \]

\[ |G_x| + |G_y| \]

**FIGURE 10.11**
Same sequence as in Fig. 10.10, but with the original image smoothed with a 5 \( \times \) 5 averaging filter.
An Example of Diagonal Edge Detection

Using Diagonal Sobel Masks

**FIGURE 10.12**
Diagonal edge detection.
(a) Result of using the mask in Fig. 10.9(c).
(b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).
The Laplacian

Second-order derivatives in an image are obtained using the Laplacian.

The Laplacian of \( f(x,y) \) is defined as:

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]
Laplacian Masks

**FIGURE 10.13**

Laplacian masks used to implement Eqs. (10.1-14) and (10.1-15), respectively.

\[
\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)
\]

\[
\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9)
\]
The Role of the Laplacian in Segmentation

1. Using its zero-crossing property for edge location.

2. Using it for the complementary purpose of establishing whether a pixel is on the dark or light side of an edge.


Edge Finding by Zero Crossing

Gaussian function:

\[ h(r) = -e^{-\frac{r^2}{2\sigma^2}} \]

where, \( r^2 = x^2 + y^2 \) and \( \sigma \) is the standard deviation

The Laplacian of Gaussian (LoG) is:

\[ \nabla^2 h(r) = -\left[ \frac{r^2 - \sigma^2}{\sigma^4} \right] e^{-\frac{r^2}{2\sigma^2}} \]

The purpose of Gaussian is to smooth the image.
The purpose of the Laplacian operator is to provide an image with zero crossing for establishing the location of edges.
Laplacian of Gaussian

3-D plot

Zero crossing

5x5 approximation mask

FIGURE 10.14
Laplacian of a Gaussian (LoG).
(a) 3-D plot.
(b) Image (black is negative, gray is the zero plane, and white is positive).
(c) Cross section showing zero crossings.
(d) 5 × 5 mask approximation to the shape of (a).
Edge Finding by Zero Crossing

Original image

Gaussian smoothing function

Sobel gradient (for comparison)

Laplacian mask

LoG

Zero crossing

Thresholded LoG
Edge Linking and Boundary Detection

The edge detection algorithms typically are followed by linking procedures to assemble edge pixels into meaningful edges.

There are several approaches for this purpose:
- Local processing
- Global processing via the Hough Transform
- Global processing via Graph-theoretic techniques
Local Processing

An edge pixel with coordinates \((x_o,y_o)\) in a predefined neighborhood of \((x,y)\), is similar in magnitude to the pixel at \((x,y)\) if:

\[|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E\]

where \(E\) is a nonnegative threshold

An edge pixel at \((x_o,y_o)\) in the predefined neighborhood of \((x,y)\) has an angle similar to the pixel at \((x,y)\) if

\[|\alpha(x, y) - \alpha(x_0, y_0)| < A\]

where \(A\) is a nonnegative threshold

The direction of the edge at \((x,y)\) is perpendicular to the direction of the gradient vector at that point.

All points that are similar according to these predefined criteria are linked, forming an edge of pixels that share the criteria.
Edge Point Linking Based on Local Processing

Original image

\[ G_x \]

\[ G_y \]

Result of edge linking

**FIGURE 10.16**
(a) Input image.
(b) \( G_x \) component of the gradient.
(c) \( G_y \) component of the gradient.
(d) Result of edge linking. (Courtesy of Perceptics Corporation.)
Global Processing via the Hough Transform

xy-plane

ab-plane (parameter plane)

(a) xy-plane.
(b) Parameter space.

All points on this line have lines in parameter space that intersect at \((a', b')\)

\[ b = -x_i a + y_i \]

\[ b = -x_j a + y_j \]

\(a'\) is the slope and \(b'\) the intercept of the line containing both \((x_i, y_i)\) and \((x_j, y_j)\) in xy-plane.
Accumulator cells

(a_{\text{max}},a_{\text{min}}) and (b_{\text{max}},b_{\text{min}}) are the expected ranges of slope and intercept values.

The cell at coordinates (i,j), with accumulator value A(i,j), corresponds to the square associated with parameter space coordinates (a_i,b_j). Initially, A(i,j)=0.

1. For every point (x_k,y_k) in the image, let a equal each of the allowed subdivision values on the a-axis and solve b using b=ax_k + y_k, then round off b to its nearest value in b-axis.
2. If a_p results in b_q, then A(p,q)=A(p,q)+1.
3. A value of Q in A(i,j) corresponds to Q points in the xy-plane lying on the line y= a_i x + b_j.
The normal representation of a line is:

\[ x \cos \theta + y \sin \theta = \rho \]
Illustration of the Hough Transform

**“A”:** points 1, 3 and 5 lie on a straight line passing through \( \rho = 0 \) and \( \theta = -45^\circ \)

**“B”:** points 2, 3 and 4 lie on a straight line passing through \( \rho = 1/2 \) diagonal distance, \( \theta = 45^\circ \)

Hough transform has a reflective adjacency relationship at the right and left edge of the parameter space.
Using the Hough Transform for Edge Linking

1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the $\rho\theta$-plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Continuity: computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell. A gap at any point is significant if the distance between that point and its closest neighbor exceeds a certain threshold.
Using the Hough Transform for Edge Linking

The criteria for linking pixels:
1. The pixels belonged to one of the three accumulator cells with the highest count.
2. No gaps were longer than five pixels.
Global Processing via Graph-Theoretic Techniques

This method is based on representing edge segments in the form of a graph and searching the graph for low-cost paths that correspond to significant edges. It performs well in the presence of noise.

Definitions:
1. Graph $G=(N,U)$: a finite, nonempty set of nodes $N$, together with a set $U$ of unordered pairs of distinct elements of $N$.
2. Arc: each pair $(n_i, n_j)$ of $U$. A cost $c(n_i, n_j)$ is associated with it.
3. Directed graph: a graph in which the arcs are directed.
4. Successor, parent: if an arc is directed from node $n_i$ to node $n_j$, then $n_j$ is said to be a successor of the parent node $n_i$.
5. Path: a sequence of nodes $n_1, n_2, \ldots, n_k$, with each node $n_i$ being a successor of node $n_{i-1}$, is called a path from $n_1$ to $n_k$.
6. The cost of the path:
   \[
c = \sum_{i=2}^{k} c(n_{i-1}, n_i)
   \]
Edge element: boundary between two pixels $p$ and $q$, such that $p$ and $q$ are 4-neighbors.

The edge element here is defined by the pairs $(x_p, y_p)(x_q, y_q)$.
The cost for each edge element defined by pixels p and q:

\[ c(p, q) = H - [f(p) - f(q)] \]

H: the highest gray-level value in the image

f(p), f(q): gray level values of p and q.
Graph for Fig. 10.23

The lowest-cost path is shown dashed.
The Heuristic Graph Search Algorithm

Let \( r(n) \) be an estimate of the cost of a minimum-cost path from the start node \( s \) to a goal node, where the path is constrained to go through \( n \).

\[ r(n) = g(n) + h(n), \quad g(n) \text{ is the lowest-cost path from } s \text{ to } n, \quad h(n) \text{ is obtained by using any available heuristic information.} \]

**Step 1.** Mark the start node OPEN and get \( g(s) = 0 \).

**Step 2.** If no node is OPEN exit with failure; otherwise, continue.

**Step 3.** Mark CLOSED the OPEN node \( n \) whose estimate \( r(n) \) computed is smallest.

**Step 4.** If \( n \) is a goal node, exit with the solution path obtained by tracing back through the pointers; otherwise, continue.
The Heuristic Graph Search Algorithm

Step 5. Expand node \( n \), generating all of its successors. (If there are no successors go to Step 2.)

Step 6. If a successor \( n_i \) is not marked, set \( r(n_i) = g(n) + c(n, n_i) \) mark it OPEN, and direct pointers from it back to \( n \).

Step 7. If a successor \( n_i \) is marked CLOSED or OPEN, update its value by letting
\[
g'(n_i) = \min[g(n_i), g(n) + c(n, n_i)]
\]
mark OPEN those CLOSED successors whose \( g' \) values were thus lowered and redirect to \( n \) the pointers from all nodes whose \( g' \) were lowered. Go to step 2.

This algorithm does not guarantee a minimum-cost path, but it has higher speed.
The heuristic used at any point on the graph was to determine and use the optimum path for five levels down from that point.
Thresholding

(a) Single thresholding
if \( f(x, y) > T \), then \((x, y)\) is called an object point.

(b) Multilevel thresholding
if \( T_1 < f(x, y) \leq T_2 \), then \((x, y)\) belongs to one object.
if \( f(x, y) > T_2 \), then \((x, y)\) belongs to another object.
if \( f(x, y) \leq T_1 \), then \((x, y)\) belongs to the background.

FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.
Thresholding

Thresholding may be viewed as an operation that involves tests against a function $T$ of the form:

$$T = T[x, y, p(x, y), f(x, y)]$$

$f(x,y)$: gray level of point $(x,y)$
$p(x,y)$: a local property of $(x,y)$

The thresholded image:

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

Global thresholding: $T$ depends only on $f(x,y)$
Local thresholding: $T$ depends on both $f(x,y)$ and $p(x,y)$
Dynamic/Adaptive thresholding: $T$ depends on $x$ and $y$
The image resulting from poor (e.g. nonuniform) illumination could be quite difficult to segment.
Basic Global Thresholding

Original image

Thresholded image

$T$ is the midway between the max. and min. gray levels.

FIGURE 10.28
(a) Original image, (b) Image histogram, (c) Result of global thresholding with $T$ midway between the maximum and minimum gray levels.
Algorithm to Obtain T Automatically

1. Select an initial estimate for T.
2. Segment the image using T.
   - $G_1$ consists of all pixels with gray level values $> T$
   - $G_2$ consists of all pixels with gray level values $\leq T$
3. Compute a new threshold value:
   $$ T = \frac{1}{2}(\mu_1 + \mu_2) $$
   where $\mu_1$ and $\mu_2$ are the average gray level values for the pixels in regions $G_1$ and $G_2$ respectively.
4. Repeat 2 to 3 until the difference in T in successive iterations is smaller than a predefined parameter $T_0$. 
Image Segmentation Using an Estimated Global Thresholding

Original image

Thresholded image

3 iterations
T=125
Basic Adaptive Thresholding

**FIGURE 10.30**
(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.
Basic Adaptive Thresholding

Properly segmented subimage

Improperly segmented subimage

Subdivided the above sub-image into smaller subimages

Result of adaptively segmenting the left image
Optimal Global and Adaptive Thresholding

Estimating thresholds that produce the minimum average segmentation error

$p_1(z), p_2(z)$: probability density function (PDF) of the objects gray levels and background gray levels.

The PDF of the overall gray level variation in the image is:

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

$P_1$: the probability that a random pixel with value $z$ is an object pixel.

$P_2$: the probability that a random pixel with value $z$ is a background pixel.

$P_1 + P_2 = 1$
Optimal Global and Adaptive Thresholding

Let $T$ be the threshold. The probability of erroneously classifying a background point as an object point is:

$$E_1(T) = \int_{-\infty}^{T} p_2(z) dz$$

The probability of erroneously classifying an object point as background is:

$$E_2(T) = \int_{T}^{\infty} p_1(z) dz$$

The overall probability of error is:

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

The threshold value for which this error is minimal:

$$P_1 p_1(T) = P_2 p_2(T)$$
If we use Gaussian density, then

\[ p(z) = \frac{P_1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}} \]

The solution for the threshold \( T \):

\[ AT^2 + BT + C = 0 \]

where

\[ A = \sigma_1^2 - \sigma_2^2 \]
\[ B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \]
\[ C = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 \sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right) \]
If \( \sigma^2 = \sigma_1^2 = \sigma_2^2 \)

then

\[
T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \left( \frac{P_2}{P_1} \right)
\]
Use of Optimum Thresholding for Image Segmentation

3 preprocessing steps: 1. Log function: counter exponential effects caused by radioactive absorption. 2. (image captured after the medium was injected) – (image captured before the medium was injected): remove the spinal column present in both images. 3. Several images were summed: reduce random noise.
Use of Optimum Thresholding for Image Segmentation

black dots: histogram of region A in Fig.10.33(b).
“o’s” and “x’s”: two fits for the histogram by bimodal Gaussian density curves. Then the optimum thresholds were obtained by the equations for the Gaussian curves.
Boundaries were obtained by:
1. Obtaining the binary picture.
   \[ f(x, y) = \begin{cases} 
   1 & f(x, y) \geq T_{xy} \\
   0 & \text{otherwise} 
   \end{cases} \]
2. Taking the gradient of the binary picture.
Use of Boundary Characteristics for Histogram Improvement and Local Thresholding

Image is coded by the following equation:

$$s(x, y) = \begin{cases} 
0 & \text{if } \nabla f < T \\
+ & \text{if } \nabla f \geq T \text{ and } \nabla^2 f \geq 0 \\
- & \text{if } \nabla f \geq T \text{ and } \nabla^2 f < 0 
\end{cases}$$

For a dark image in a light background, it results in:
1. All pixels that are not on an edge are labeled “0”.
2. All pixels on the dark side of an edge are labeled “+”.
3. All pixels on the light side of an edge are labeled “-”.

The “+” and “-” will be reversed if a light object in on a dark background.
Image Segmentation by Local Thresholding

**Figure 10.37**
(a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)

**Figure 10.38**
Histogram of pixels with gradients greater than 5. (Courtesy of IBM Corporation.)

Original image

Image segmented by local thresholding (T at or near the midpoint of the valley shown in Fig.10.38)

Histogram of pixels with gradients greater than 5.
Thresholds Based on Several Variables

**Figure 10.39** (a) Original color image shown as a monochrome picture. (b) Segmentation of pixels with colors close to facial tones. (c) Segmentation of red components.
Region-Based Segmentation

Let R represent the entire image region. We may view segmentation as a process that partitions R into n subregions, R_1,R_2,……,R_n, such that:

(a) \[ \bigcup_{i=1}^{n} R_i = R. \]
(b) \( R_i \) is a connected region, i=1,2,…..,n.
(c) \( R_i \cap R_j = \emptyset \) for all i and j, i \( \neq \) j.
(d) \( P(R_i) = TRUE \) for i=1,2,…,n.
(e) \( P(R_i \cup R_j) = FALSE \) for i \( \neq \) j.

Here, \( P(R_i) \) is a logical predicate defined over the points in set \( R_i \) and \( \emptyset \) is the null set.
Region Growing

Criteria for a pixel to be annexed to a region: (1) The absolute gray-level difference between any pixel and the seed < 65. (2) The pixel had to be 8-connected to at least one pixel in that region.
Region Growing

Histogram of Fig. 10.40(a)
Region Splitting and Merging

1. Split into four disjoint quadrants any region $R_i$ for which $P(R_i)=\text{FALSE}$.
2. Merge any adjacent regions $R_j$ and $R_k$ for which $P(R_j \cup R_k)=\text{TRUE}$.
3. Stop when no further merging or splitting is possible.
Here, define \( P(R_i) = \text{TRUE} \) if at least 80% of the pixels in \( R_i \) have the property

\[
|z_j - m_i| \leq 2\sigma_i ,
\]

where \( z_j \) is the gray level of the jth pixel in \( R_i \), \( m_i \) is the mean gray level of that region, and \( \sigma_i \) is the standard deviation of the gray levels in \( R_i \). If \( P(R_i) = \text{TRUE} \) under this condition, the values of all the pixels in \( R_i \) were set equal to \( m_i \).
Segmentation by Morphological Watersheds

Watershed is based on visualizing an image in three dimensions - two spatial coordinates versus gray levels. (topographic view)

We consider three types of points:
1. Points belonging to a regional minimum.
2. Points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum. - catchment basin or watershed of that minimum.
3. Points at which water would be equally likely to fall to more than one such minimum. - divide lines or watershed lines.

The principal objectives of segmentation algorithms based on these concepts is to find the watershed lines.
Segmentation by Morphological Watersheds

FIGURE 10.44
(a) Original image. (b) Topographic view. (c)-(d) Two stages of flooding.
Segmentation by Morphological Watersheds

FIGURE 10.44 (Continued)
(c) Result of further flooding.  
(f) Beginning of merging of water from two catchment basins (a short dam was built between them).  
(g) Longer dams.  
(h) Final watershed (segmentation) lines.  
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed lines - continuous boundaries
FIGURE 10.45 (a) Two partially flooded catchment basins at stage $n - 1$ of flooding. (b) Flooding at stage $n$, showing that water has spilled between basins (for clarity, water is shown in white rather than black). (c) Structuring element used for dilation. (d) Result of dilation and dam construction.
Illustration of the Watershed Segmentation Algorithm

**Figure 10.46**
(a) Image of blobs. (b) Image gradient.
(c) Watershed lines.
(d) Watershed lines superimposed on original image.
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)
Illustration of Oversegmentation

FIGURE 10.47
(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris)
The Use of Markers

FIGURE 10.48
(a) Image showing internal markers (light gray regions) and external markers (watershed lines).
(b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)
The Use of Motion in Segmentation
- Spatial Techniques

ADI: Accumulative Difference Image – formed by comparing the reference image with every subsequent image in a sequence of image frames.

A counter for each pixel location in the accumulative image is incremented every time a difference occurs at that pixel location between the reference and an image in the sequence.

Three types of accumulative difference images:
- Absolute
- Positive
- Negative
Let \( f(x,y,k) \) denote the image at time \( t_k \), \( R(x,y) = f(x,y,1) \) denote the reference image. The values of the ADIs are counts. Assume that the gray-level values of the moving objects are larger than the background. Define:

**Absolute ADI**

\[
A_k(x, y) = \begin{cases} 
A_{k-1}(x, y) + 1 & \text{if} \quad |R(x, y) - f(x, y, k)| > T \\
A_{k-1}(x, y) & \text{otherwise}
\end{cases}
\]

**Positive ADI**

\[
P_k(x, y) = \begin{cases} 
P_{k-1}(x, y) + 1 & \text{if} \quad [R(x, y) - f(x, y, k)] > T \\
P_{k-1}(x, y) & \text{otherwise}
\end{cases}
\]

**Negative ADI**

\[
N_k(x, y) = \begin{cases} 
N_{k-1}(x, y) + 1 & \text{if} \quad [R(x, y) - f(x, y, k)] < -T \\
N_{k-1}(x, y) & \text{otherwise}
\end{cases}
\]
An Example of ADIs

ADIs of a rectangular object in a southeasterly direction

<table>
<thead>
<tr>
<th>Absolute ADI</th>
<th>Positive ADI</th>
<th>Negative ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Absolute ADI" /></td>
<td><img src="image2" alt="Positive ADI" /></td>
<td><img src="image3" alt="Negative ADI" /></td>
</tr>
</tbody>
</table>

**FIGURE 10.49** ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.
Building a Reference Image

**FIGURE 10.50** Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)
For a sequence of K digital images of size $M \times N$, the sum of the weighted projections onto the x axis at any integer instant of time is:

$$g_x(t, a_1) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y, t) e^{j2\pi a_1 x \Delta t} \quad t = 0, 1, \ldots, K-1$$

The 1-D Fourier transform is:

$$G_x(u_1, a_1) = \frac{1}{K} \sum_{t=0}^{K-1} g_x(t, a_1) e^{-j2\pi u_1 t / K} \quad u_1 = 0, 1, \ldots, K-1$$

The sum of the weighted projections onto the y axis is:

$$g_y(t, a_2) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y, t) e^{j2\pi a_2 y \Delta t} \quad t = 0, 1, \ldots, K-1$$

And the 1-D Fourier transform is:

$$G_y(u_2, a_2) = \frac{1}{K} \sum_{t=0}^{K-1} g_y(t, a_2) e^{-j2\pi u_2 t / K} \quad u_2 = 0, 1, \ldots, K-1$$
The frequency-velocity relationship is:

\[ u_1 = a_1 v_1 \quad u_2 = a_2 v_2 \]

The sign of the x-component of the velocity is obtained by computing:

\[ S_{1x} = \frac{d^2 \text{Re}[g_x(t, a_1)]}{dt^2} \bigg|_{t=n} \quad S_{2x} = \frac{d^2 \text{Im}[g_x(t, a_1)]}{dt^2} \bigg|_{t=n} \]

- If the velocity component \( v_1 \) is positive, then \( S_{1x} \) and \( S_{2x} \) will have the same sign at an arbitrary point in time \( n \).
- If \( v_1 \) is negative, then \( S_{1x} \) and \( S_{2x} \) will have the opposite sign.
- If either \( S_{1x} \) or \( S_{2x} \) is zero, we consider the next closest point in time \( t = n \pm \Delta t \).

Similar comments apply to computing the sign of \( v_2 \).
An Example of Detection of a Small Moving Object via the Frequency Domain

One of 32-frame sequence of LANDSAT images generated by adding white noise to a reference image.
An Example of Detection of a Small Moving Object via the Frequency Domain

Intensity plot of the previous image

FIGURE 10.52
Intensity plot of the image in Fig. 10.51, with the target circled. (Rajala, Riddle, and Snyder.)
An Example of Detection of a Small Moving Object via the Frequency Domain

\[ a_1 = 6 \]

\[ u_1 = 3 \text{ yields } v_1 = 0.5 \]

\textbf{FIGURE 10.53} Spectrum of Eq. (10.6-8) showing a peak at \( u_1 = 3 \). (Rajala, Riddle, and Snyder.)

© 2002 R. C. Gonzalez & R. E. Woods
An Example of Detection of a Small Moving Object via the Frequency Domain

\[ a_2 = 4 \]

**Figure 10.54**
Spectrum of Eq. (10.6-9) showing a peak at \( u_2 = 4 \). (Rajala, Riddle, and Snyder.)

\[ u_2 = 4 \text{ yields } v_2 = 1.0 \]