Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

Lecture 5

Image processing

Morphological operations

Morphological Operations

Mathematical Morphology

A methodology for image processing and image analysis based on set theory and topology

Points and vectors will be denoted by Latin lowercase letters
x, y, z

Sets will be denoted by Latin uppercase letters
X, Y, Z

The symbol ϕ denotes the empty set

Morphological Operations

Set inclusion: $Y \subset X$

If y is an element of Y, then y is also an element of X

Complement: X^c

This is the set of all elements which are not elements of X

Union:

The union of two sets X and Y, $X \cup Y$ is defined:

$$X \cup Y = \left\{x \mid x \in X \,\mathrm{or}\, x \in Y\right\}$$

Intersection:

The intersection of two sets X and Y, $X \cap Y$ is defined:

$$X \cap Y = \left(X^c \cup Y^c\right)^c$$

Morphological Operations

The translation of a set X by h is denoted by X_h

This is a set where each element (point) is translated by a vector h

Morphological Operations

A transformation of a set X is denoted by $\Psi(X)$

$\Psi(\ )$ is the set transformation and in an expression such as $Y = \Psi(X)$, Y is the transformed set

The dual of a transformation is defined $\Psi^{*}(X) \rightarrow \left(\Psi\left(X^{c}\right)\right)^{c}$

For example, intersection is the dual of union since

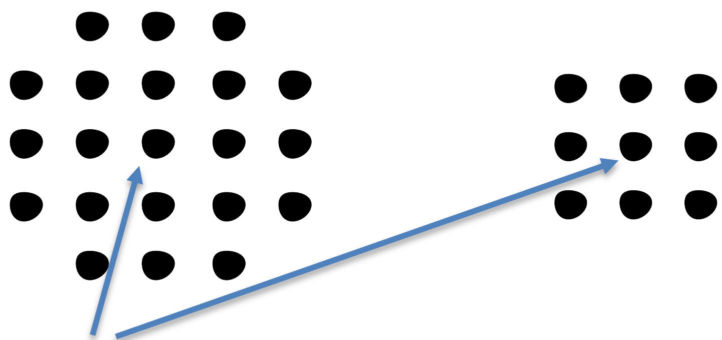
$$X \cap Y = \left(X^{c} \cup Y^{c}\right)^{c}$$

Morphological Operations

Structuring Elements

A structuring element B_x, centred at x, is a set of points which is used to “extract” structure in a set, X, say.

For example, the structuring element might be a square or a disk, or any other appropriate shape



Typically, x will be at the centre of the structuring element

Morphological Operations

Hit or Miss Transformations

The "point by point" transformation of a set X:

Choose, and fix, a structuring element B

Define

B_x^1 to be that subset of B_x whose elements belong to the foreground

B_x^2 to be the subset of B_x whose elements belong to the background

i.*e.* $B^1 \cap B^2 = \Phi$

Morphological Operations

Hit or Miss Transformations

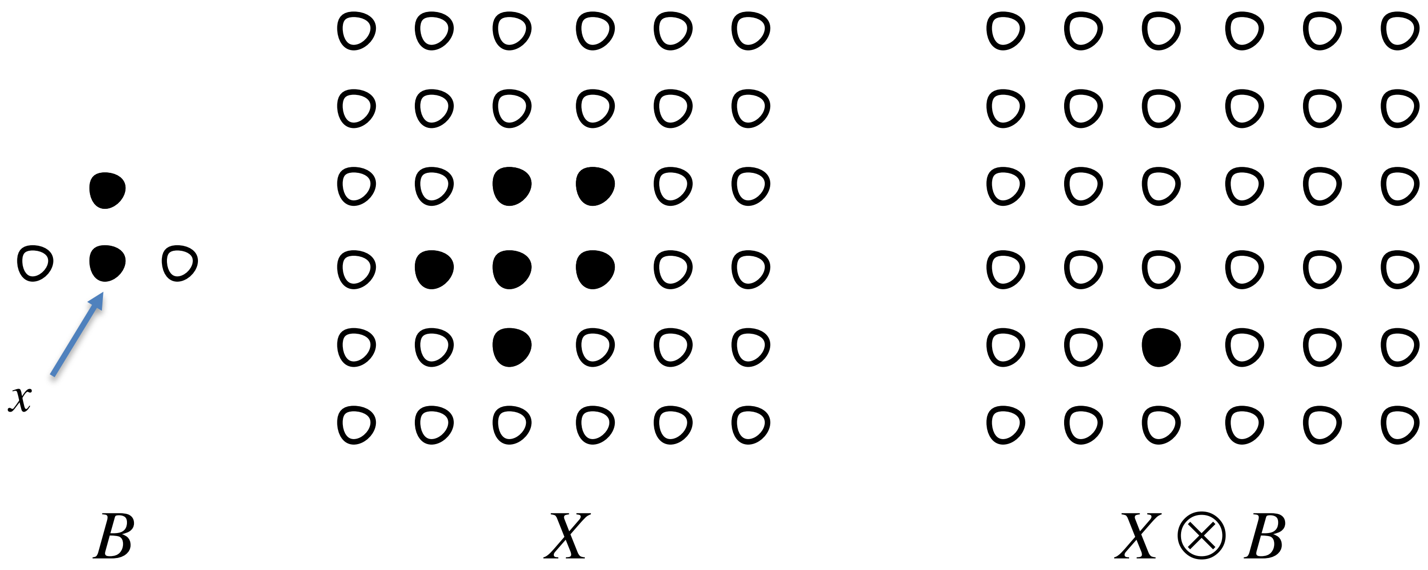
A point x belongs to the Hit or Miss transform, $X \otimes B$, if and only if B_x^1 is included in X and B_x^2 is included in X^c

$$X \otimes B = \left\{ x \mid B_x^1 \subset X;\ B_x^2 \subset X^c \right\}$$

Thus $X \otimes B$ defines the points where the structuring element B exactly matches (hits) the set X, *i.e.* the image

Morphological Operations

Hit or Miss Transformations



Morphological Operations

Erosion

Let $\breve{B}$ be the transposed set of B, *i.e.* the symmetrical set of B with respect to its origin

The erosion operation is denoted Θ and the erosion of a set X with $\breve{B}$ is defined:

$$X \Theta \breve{B} = \left\{ x \mid B_x \subset X \right\}$$

Morphological Operations

Erosion

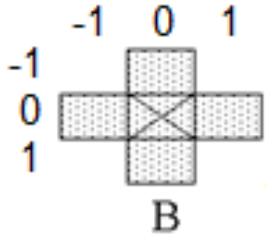
Note that this is equivalent to a Hit or Miss transformation

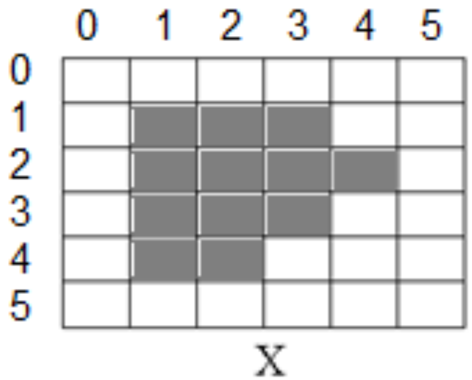
of X with $\breve{B}$ where $B^2 = \Phi$, *i.e.* where there are no background points

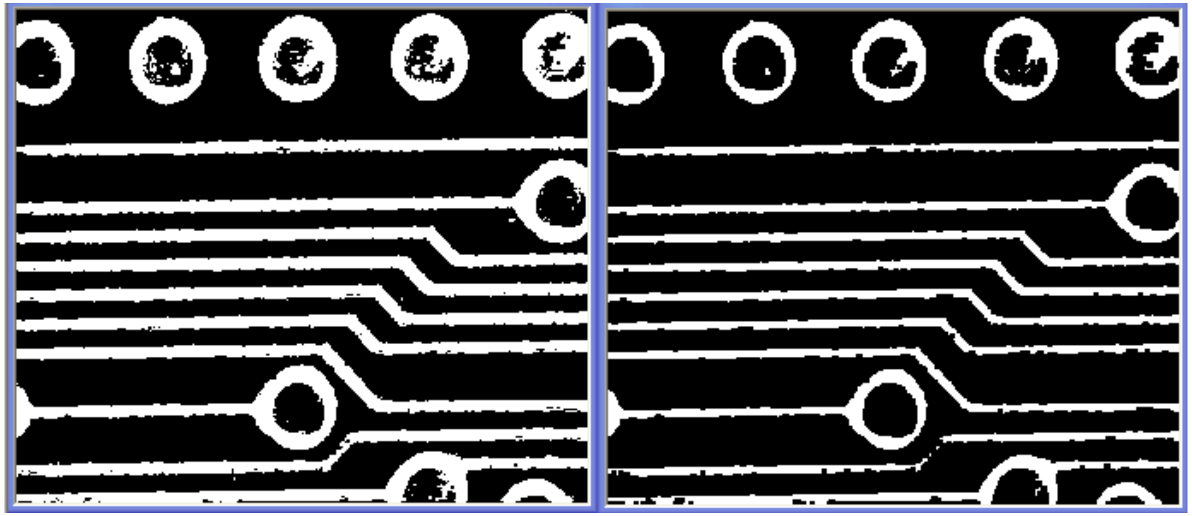
Note also that $X \Theta B$ is not an erosion:
it is the Minkowski Subtraction of B from X

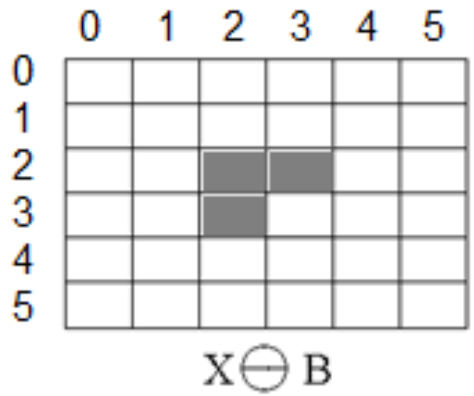
Morphological Operations

Erosion









Kenneth Dawson-Howe, A Practical Introduction to Computer Vision with OpenCV, © Wiley & Sons Inc. 2014

Morphological Operations

Erosion

Intuitively:

the erosion of a set by a structuring element amounts to

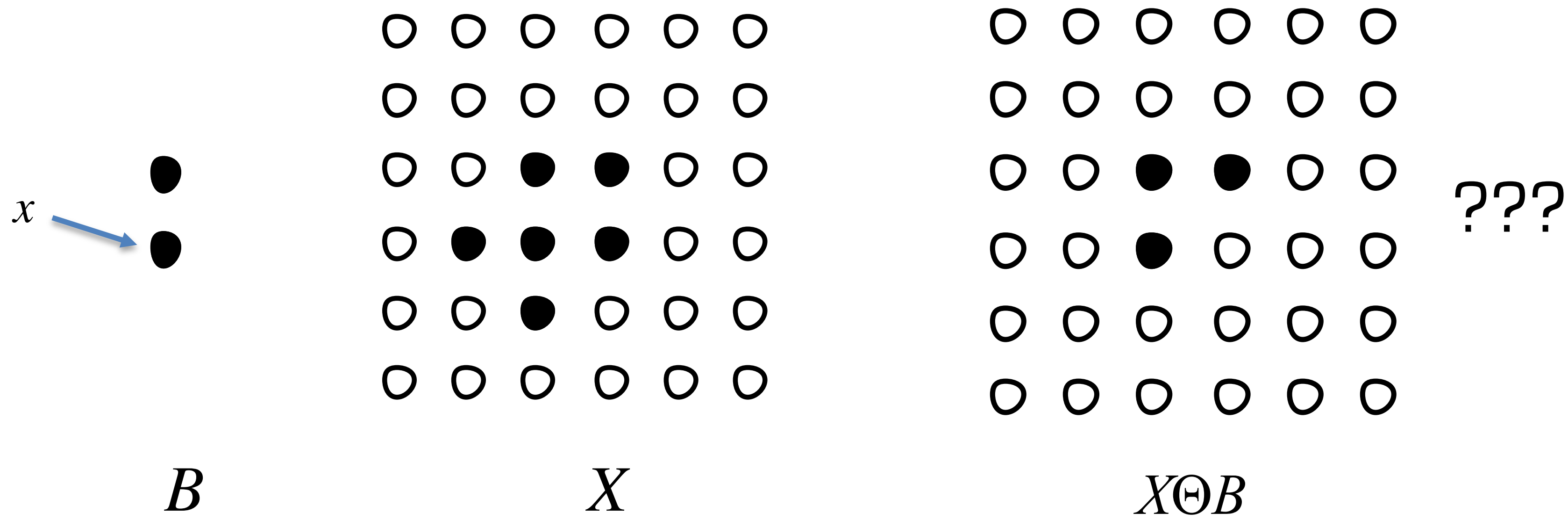
the generation of a new (transformed/eroded) set

where each element in the transformed set

is a point where the structuring element is included in the original set X

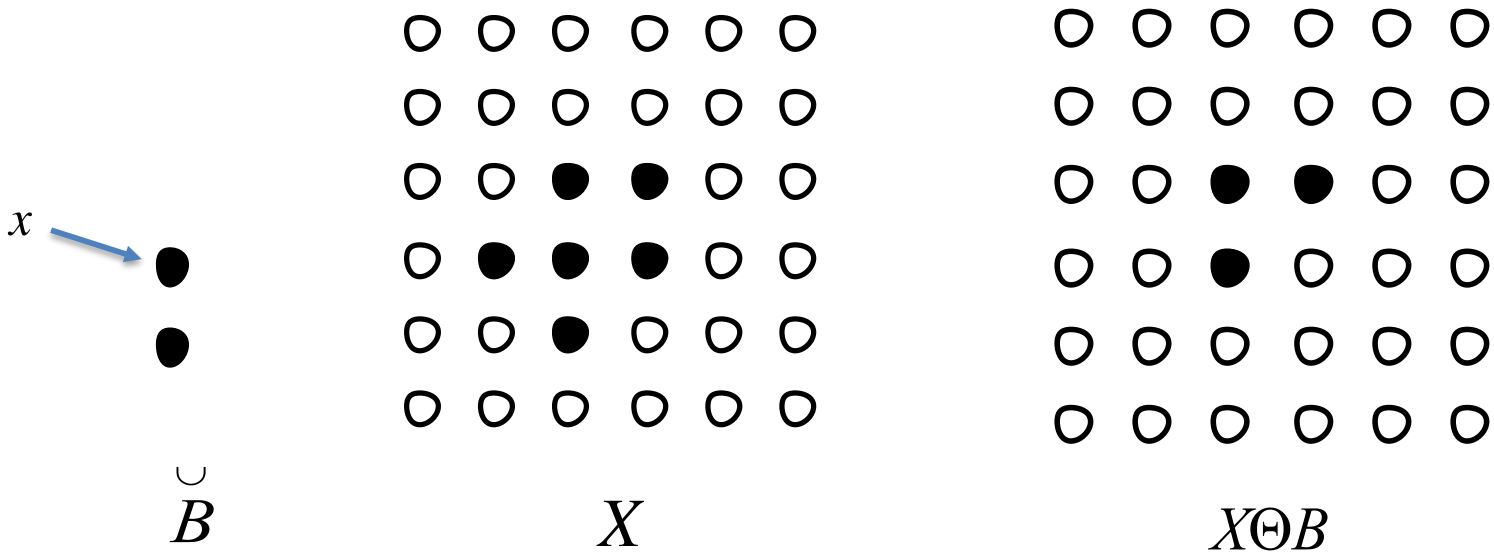
Morphological Operations

Erosion



Morphological Operations

Erosion



Morphological Operations

Dilation

Dilation is dual of erosion

The dilation operation is denoted by the symbol $\oplus$

$$X \oplus \breve{B} = \left(X^c \Theta \breve{B} \right)^c$$

That is, the dilation of X is the erosion of X^c

Morphological Operations

Dilation

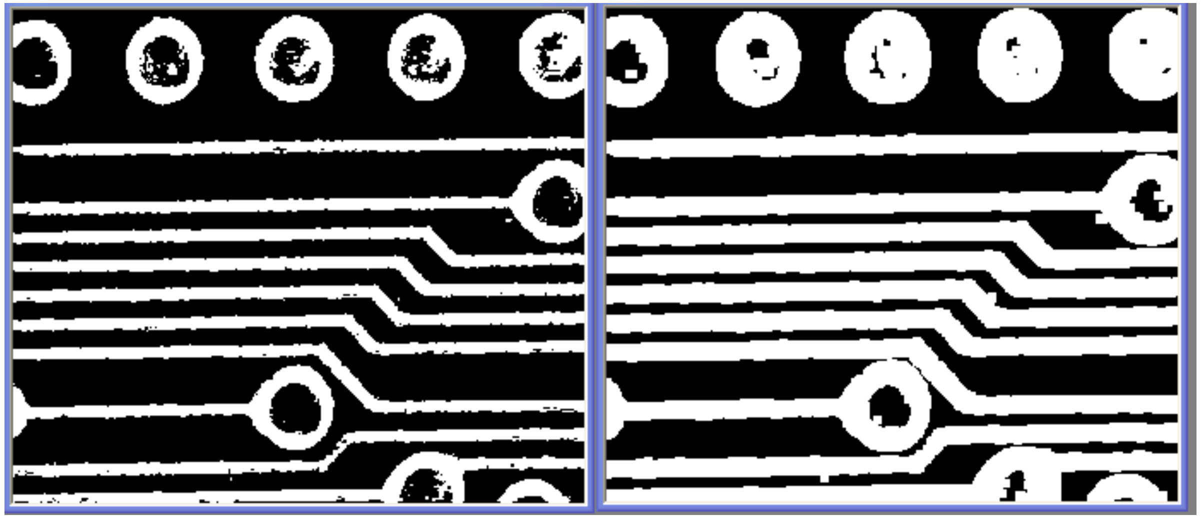
The erosion of a set (*e.g.* an image of an object) with a given structuring element

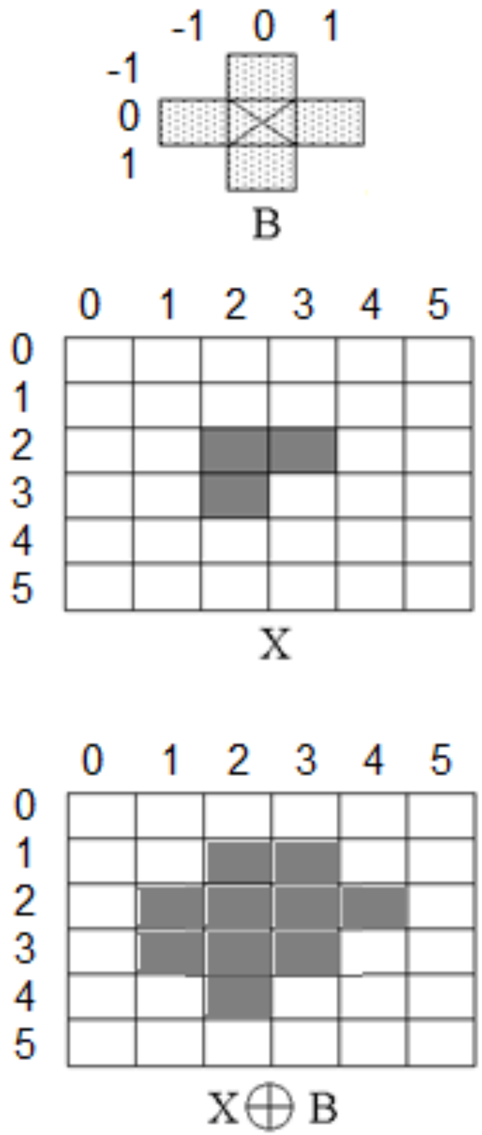
is equivalent to the dilation of its complement (*i.e.* its background) with the same structuring element,

and *vice versa*

Morphological Operations

Dilation





Kenneth Dawson-Howe, A Practical Introduction to Computer Vision with OpenCV, © Wiley & Sons Inc. 2014

Morphological Operations

Dilation

After having eroded X by B, it is not possible in general to recover the initial set by dilating the eroded set $X \Theta B$ by the same B

This dilate reconstitutes only a part of X,

- which is simpler and has less details
- but may be considered as that part which is most essential to the structure of X

Morphological Operations

Opening

This new set (*i.e.* the results of erosion followed by dilation) filters out (*i.e.* generates) a new subset of X which is extremely rich in morphological and size distribution properties.

This transformation is called an Opening X_B

$$X_B = \left(X \Theta \breve{B} \right) \oplus B$$

Morphological Operations

Opening

The opening is the domain swept out by all the translates of B which are included in X

This effects

- a smoothing of the contours of X
- cuts narrow isthmuses
- suppresses small islands and sharp edges in X

Morphological Operations

Closing

This new set produced by dilation followed by erosion is called an Closing X^B

$$X^B = (X \oplus B) \Theta \breve{B}$$

Morphological Operations

Opening vs Closing

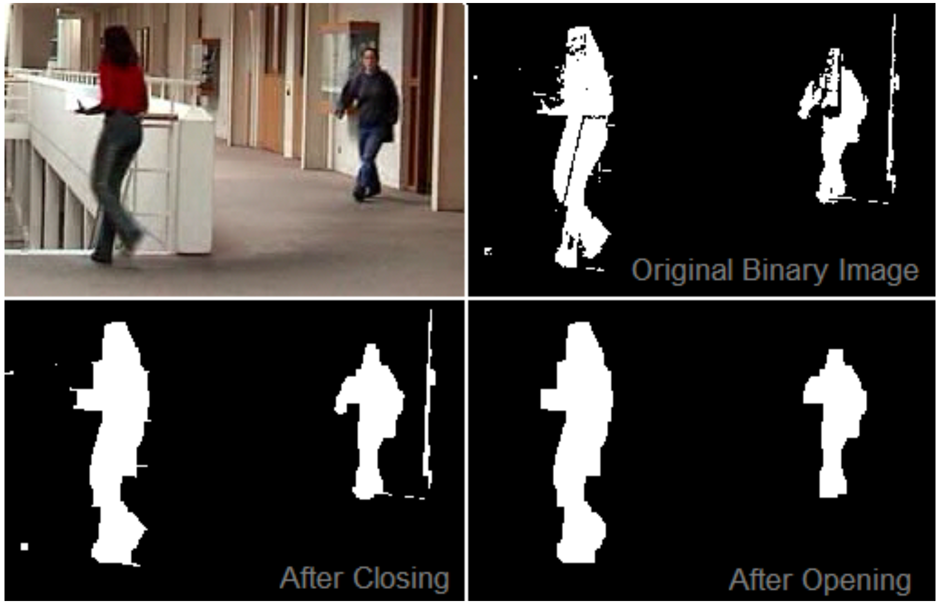
Opening is the dual of closing, *i.e.* :

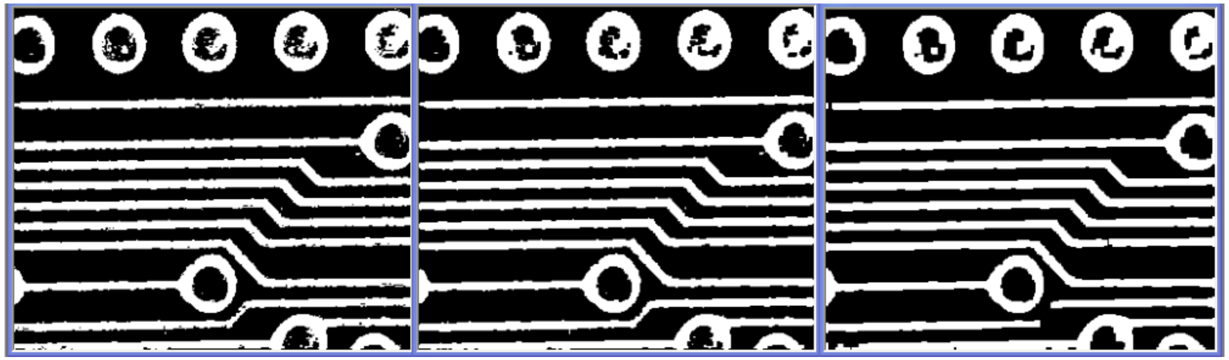
$$\left(X^c\right)^B = \left(X_B\right)^c$$

and

$$\left(X_B\right)^c = \left(X^c\right)^B$$

Morphological Operations





Kenneth Dawson-Howe, A Practical Introduction to Computer Vision with OpenCV, © Wiley & Sons Inc. 2014

Morphological Operations

Thinning

- Thinning is an iterative neighbourhood operation which generates a skeletal representation of an object
- The skeleton of an object may be thought of as a generalised axis of symmetry of the object.
- It is a suitable representation for objects which display obvious axial symmetry
- The Medial Axis Transform (MAT) proposed by Blum is one of the studied techniques for generating the skeleton

Morphological Operations

Thinning

The skeleton exhibits three topological properties :

- Connectedness (one object generates one skeleton)
- Invariance to scaling and rotation
- Information preservation in the sense that the object can be reconstructed from the medial axis

Morphological Operations

Thinning

The concept of thinning a binary image of an object is related to such medial axis transformations in that

- it generates a representation of an approximate axis of symmetry of a shape

- by successive deletion of pixels from the boundary of the object

Morphological Operations

Thinning

In general, this thinned representation is not formally related to the original object shape and it is not possible to reconstruct the original boundary from the object

Morphological Operations

Thinning

- a logical neighbourhood operation
- where object pixels are removed from an image

Morphological Operations

Thinning

- Set of conditions for pixel removal:
- The first restriction is that the pixel must lie on the border of the object
- This implies that it has at least one 4-connected neighbouring pixel which is a background pixel

Aside: Adjacency Conventions

- A convention defining the neighbours of a given pixel
- Consider the 3*3 neighbourhood in an image where the pixels are labelled 0 through 8

3	2	1
4	8	0
5	6	7

Which pixels does pixel 8 touch?

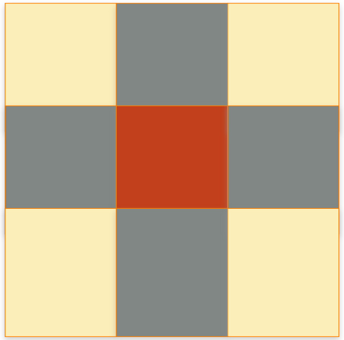
Aside: Adjacency Conventions

Adjacency conventions

- A pixel p at coordinates (i, j) has four horizontal and vertical neighbors at coordinates
  $(i\text{-}1, j)$ $(i+1, \mathrm{j})$ $(i, j\text{-}1)$ $(i, j+1)$

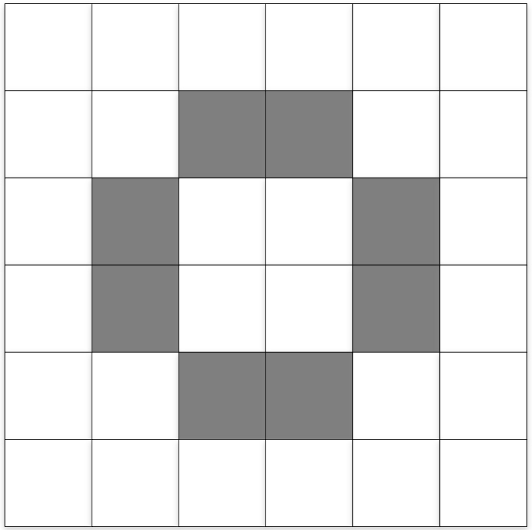
- This set is called 4-neighborhood $N_4(p)$

- The pixel also has four diagonal neighbors:
  $(i\text{-}1, j\text{-}1)$ $(i+1, j\text{-}1)$ $(i+1, j\ \text{-}1)$ $(i+1, j+1)$



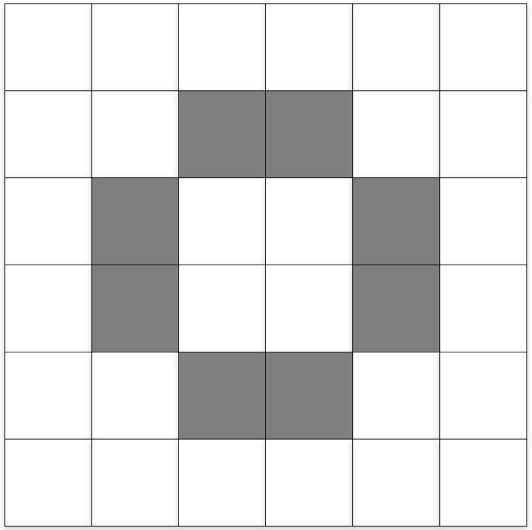
- The 8 points together form a 8-neighborhood $N_8(p)$

Aside: Adjacency Conventions



If figure and ground are both 8-connected it means the hole in the 'ring' is connected to the region surrounding the 'ring'

Aside: Adjacency Conventions



It is normal practice to use both conventions

- one for an object
- one for the background on which it rests

This can be extended quite generally

- adjacency conventions are applied alternatively to image regions which are recursively nested (or embedded) within other regions as one goes from level to level in the nesting

Morphological Operations

Thinning

- The removal of pixels from all borders simultaneously would cause difficulties:
- For example, an object two pixels thick will vanish if all border pixels are removed simultaneously
- A solution to this is to remove pixels of one border orientation only on each pass of the image by the thinning operator
- Opposite border orientations are used alternatively to ensure that the resultant skeleton is as close to the medial axis as possible

Morphological Operations

Thinning

Second restriction

– The deletion of a pixel should not destroy the object's connectedness

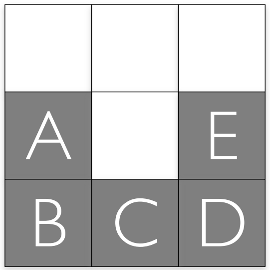
– i.e. the number of skeletons after thinning should be same as the number of objects in the image before thinning

Morphological Operations

Thinning

Second restriction

- The pixel C connects the two object segments AB and ED, that is, if C were removed then this would break the object in two; this pixel is critically connected



Morphological Operations

Thinning

- Given a pixel, labelled 8 and its eight adjacent neighbours, labelled 0-7

3	2	1
4	8	0
5	6	7

- Assume that writing the pixel number (*e.g.* 7) indicates presence, *i.e.* it is an object pixel

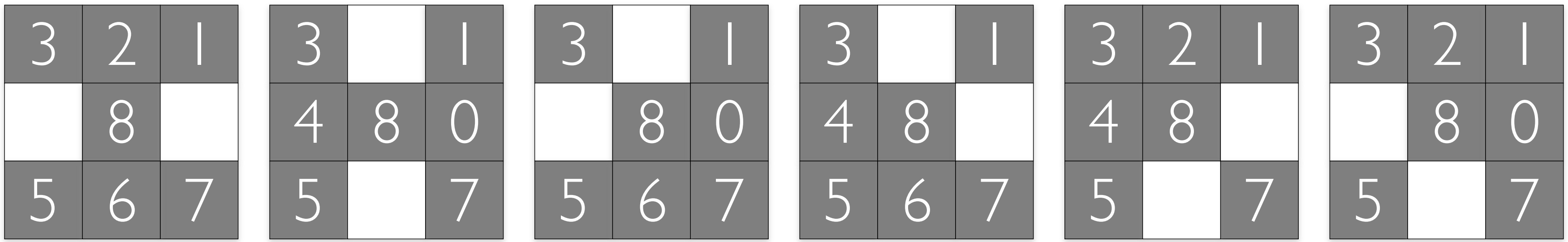
- Whereas writing it with an over-bar (e.g. $\overline{7}$) indicates absence, it is a background pixel

Morphological Operations

Thinning

Pixel 8 is critically connected if the following expression is true

$$
\begin{aligned}
8.\{&\left[(1+2+3).(5+6+7).\bar{4}.\bar{0}\right]\\
+&\left[(1+0+7).(3+4+5).\bar{2}.\bar{6}\right]\\
+&\left[3.(5+6+7+0+1).\bar{2}.\bar{4}\right]\\
+&\left[1.(3+4+5+6+7).\bar{2}.\bar{0}\right]\\
+&\left[7.(1+2+3+4+5).\bar{0}.\bar{6}\right]\\
+&\left[5.(7+0+1+2+3).\bar{4}.\bar{6}\right]\}
\end{aligned}
$$



Morphological Operations

Thinning

- A thinning algorithm should also preserve an object's length
- To facilitate this, a third restriction must be imposed
  - Arc-ends, i.e., object pixels which are adjacent to just one other pixel, must not be deleted

Morphological Operations

Thinning

- A thinned image should be invariant under the thinning operator
- The application of a thinning algorithm to a fully thinned image should produce no changes
- This provides us with a condition for stopping the thinning algorithm
- Since the pixels of a fully thinned image are either critically-connected or are arc-ends, imposing restrictions 2 and 3 allows this property to be fulfilled

Morphological Operations

Thinning

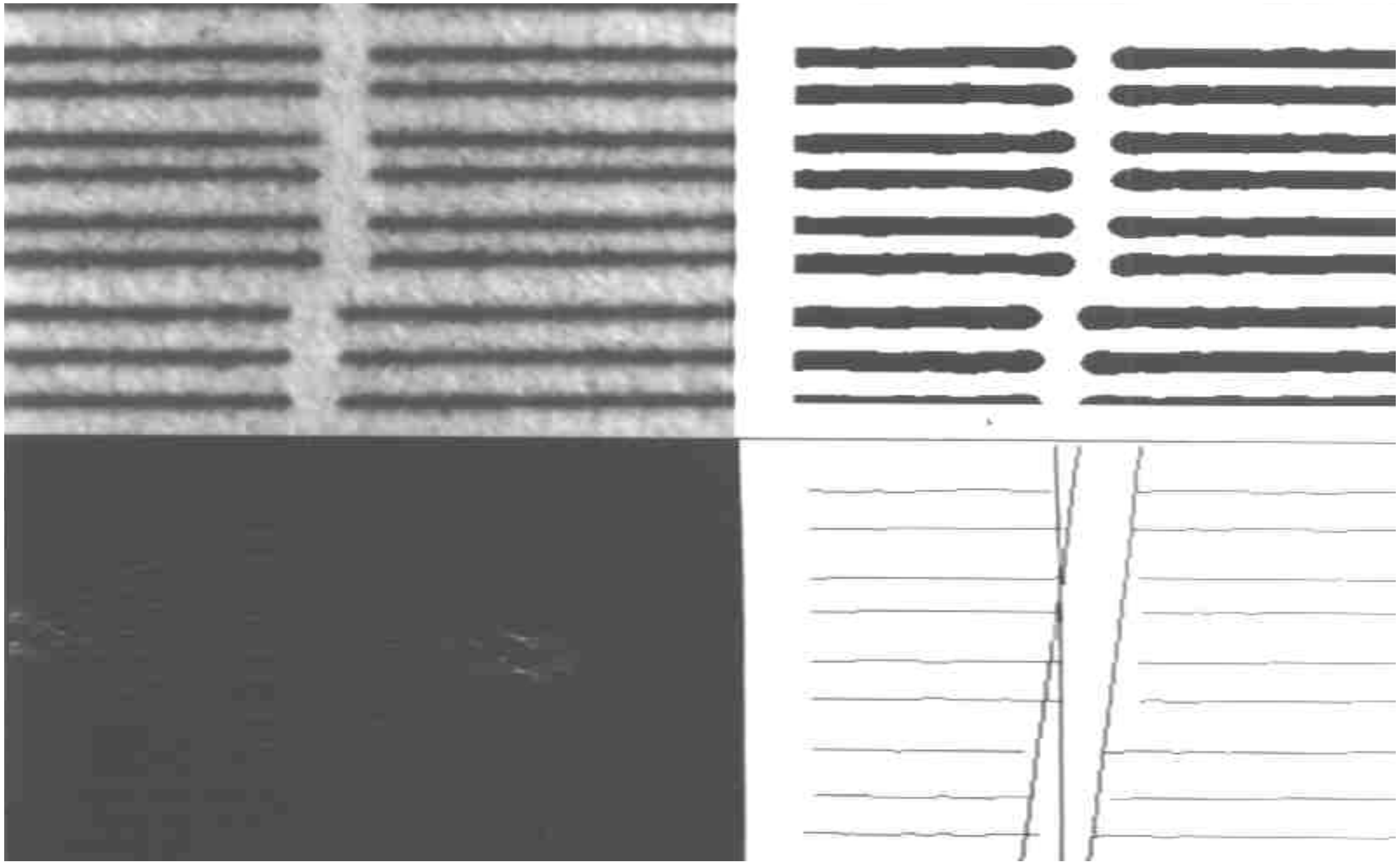
The final thinning algorithm

- scan the image in a raster fashion
- removing of all object pixels according to these three restrictions
- varying border from pass to pass

The image is thinned until four successive passes (corresponding to the four border orientations) producing no changes to the image are made, at which stage, thinning ceases

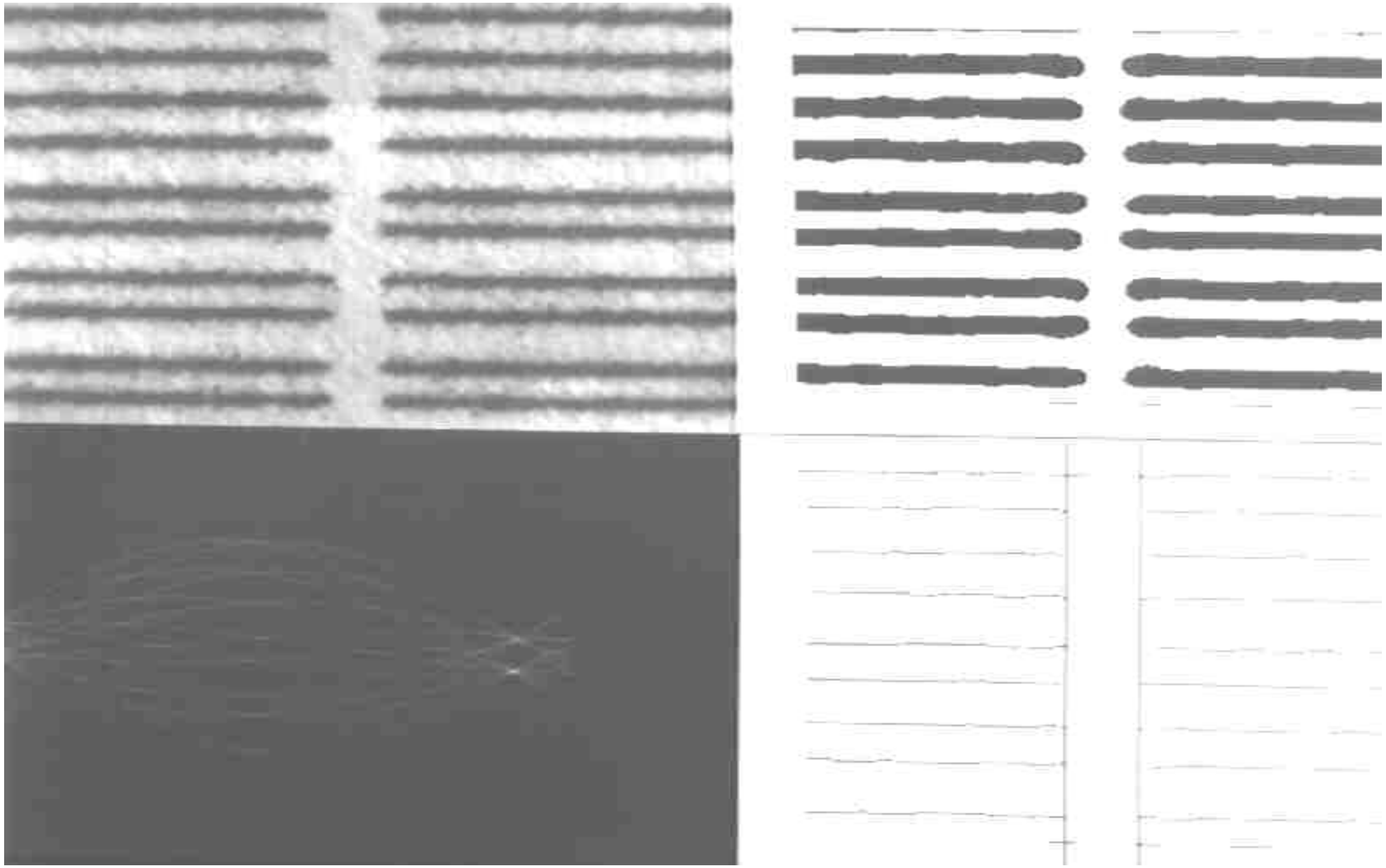
Morphological Operations

Thinning



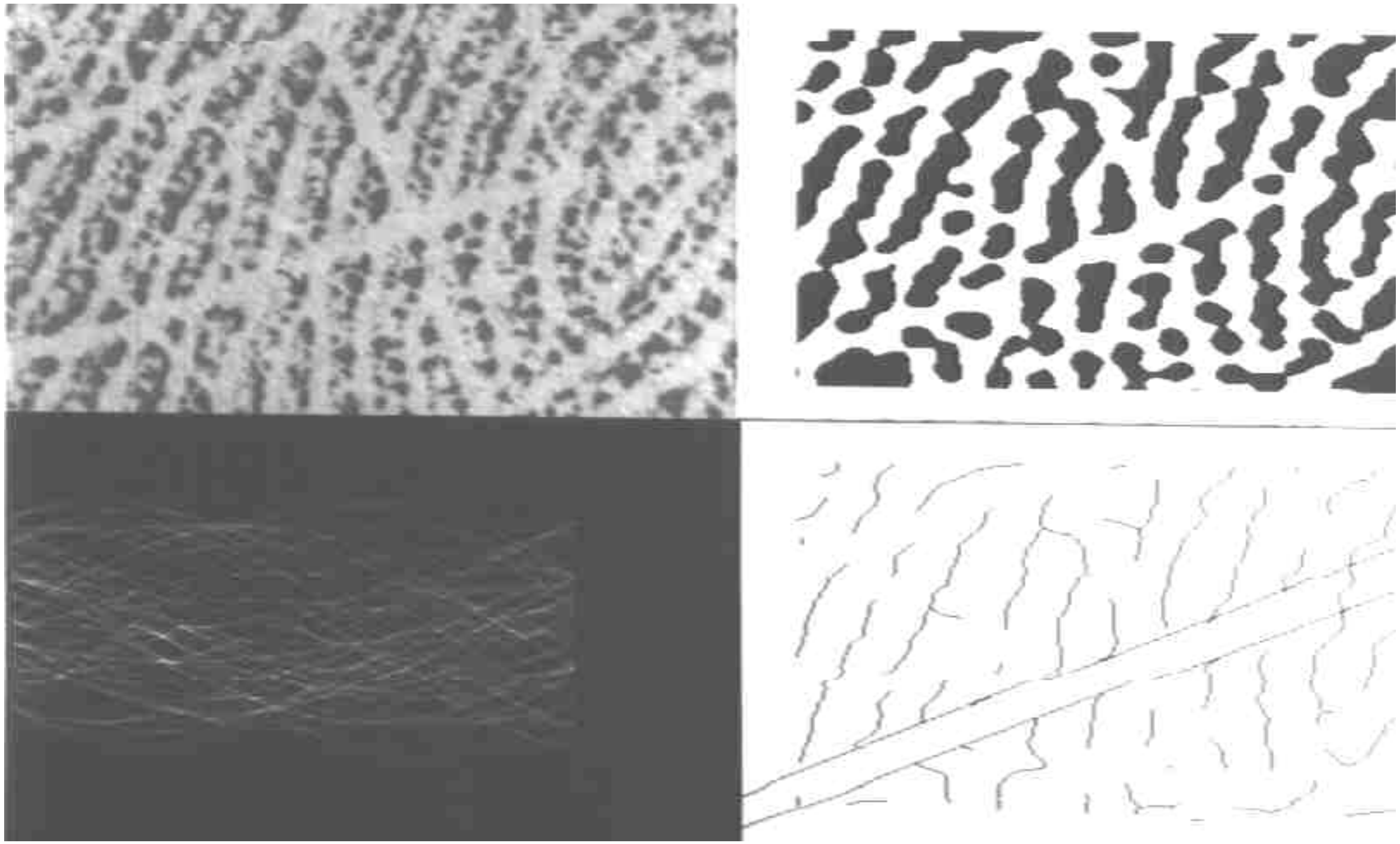
Morphological Operations

Thinning



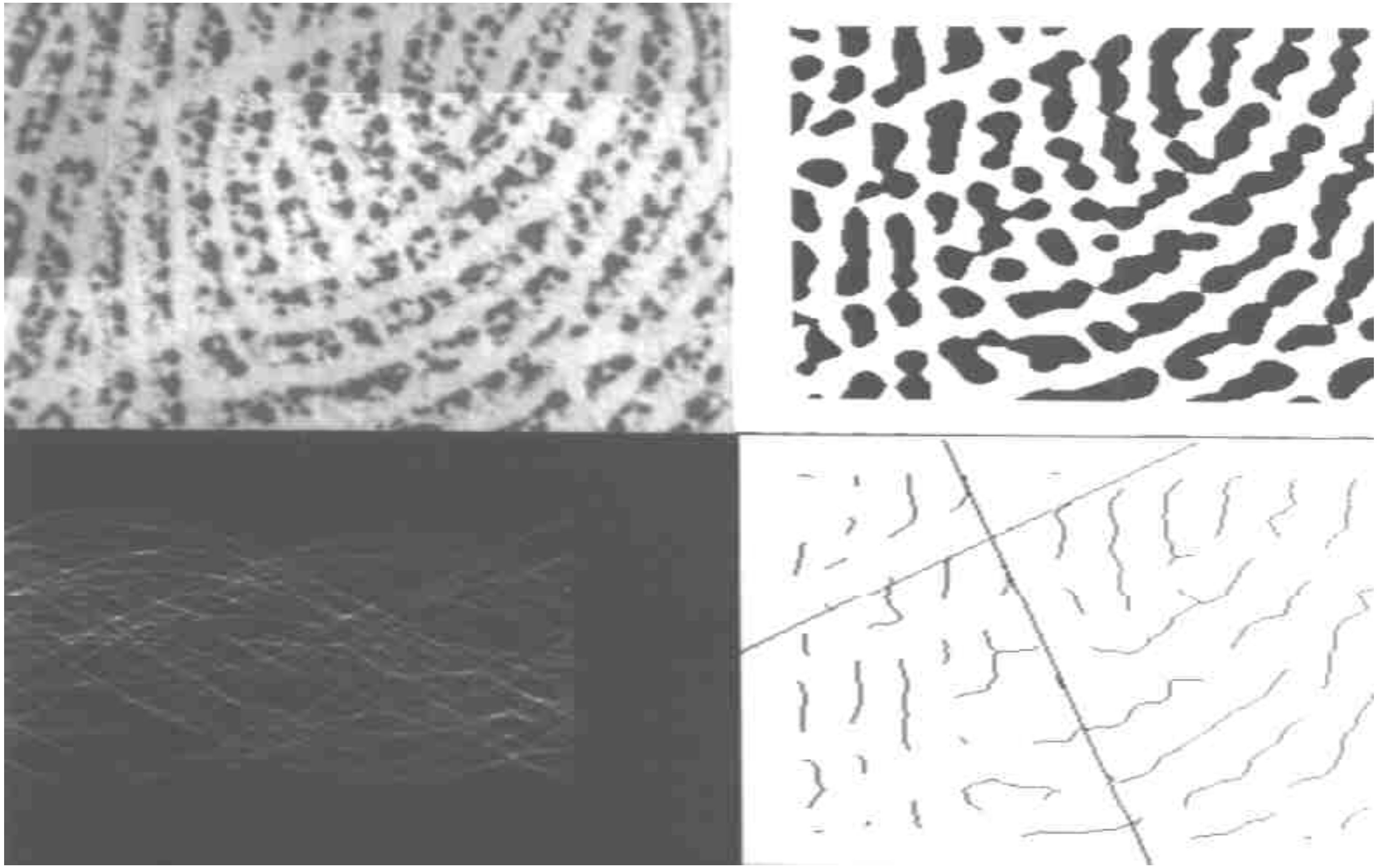
Morphological Operations

Thinning



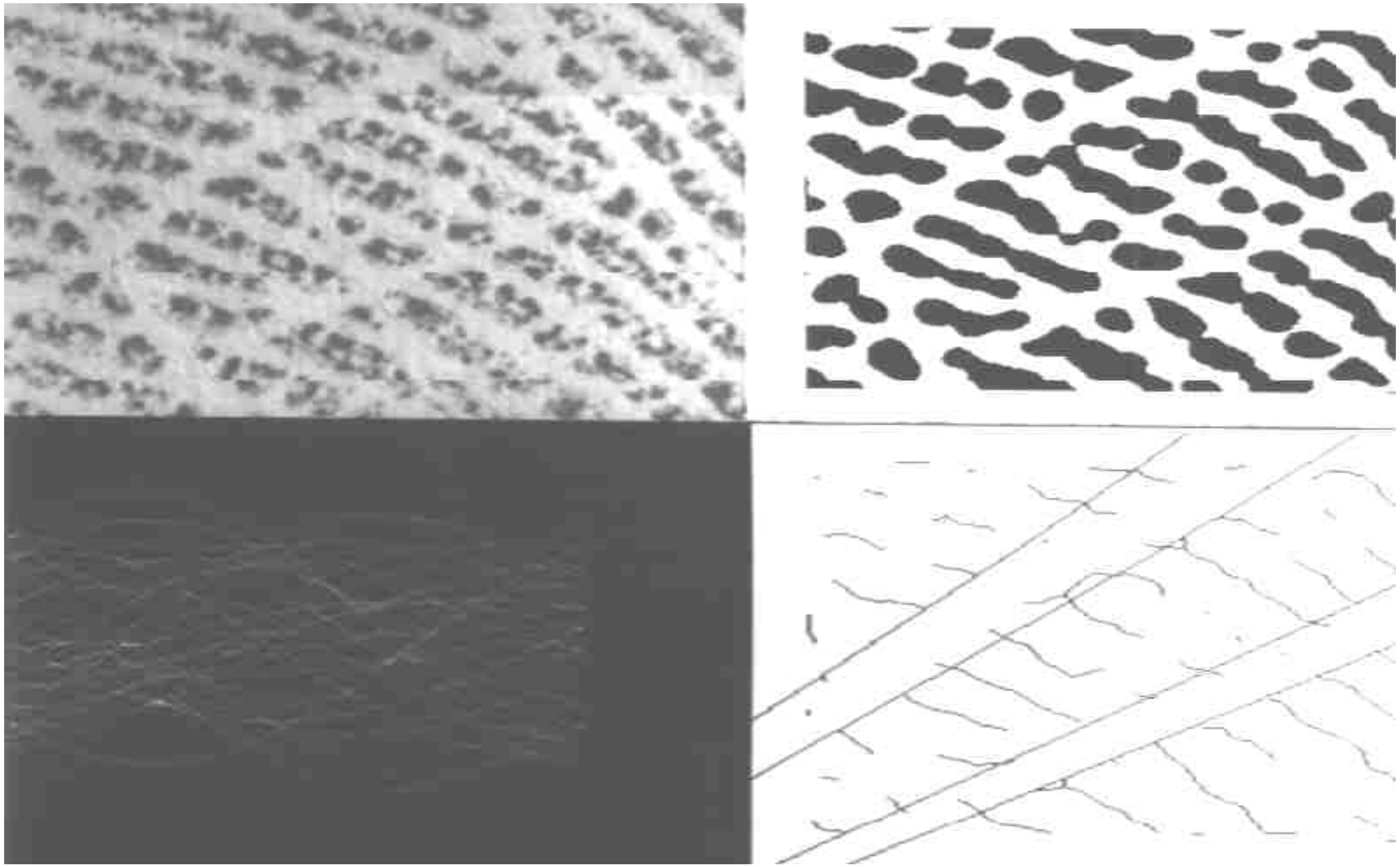
Morphological Operations

Thinning



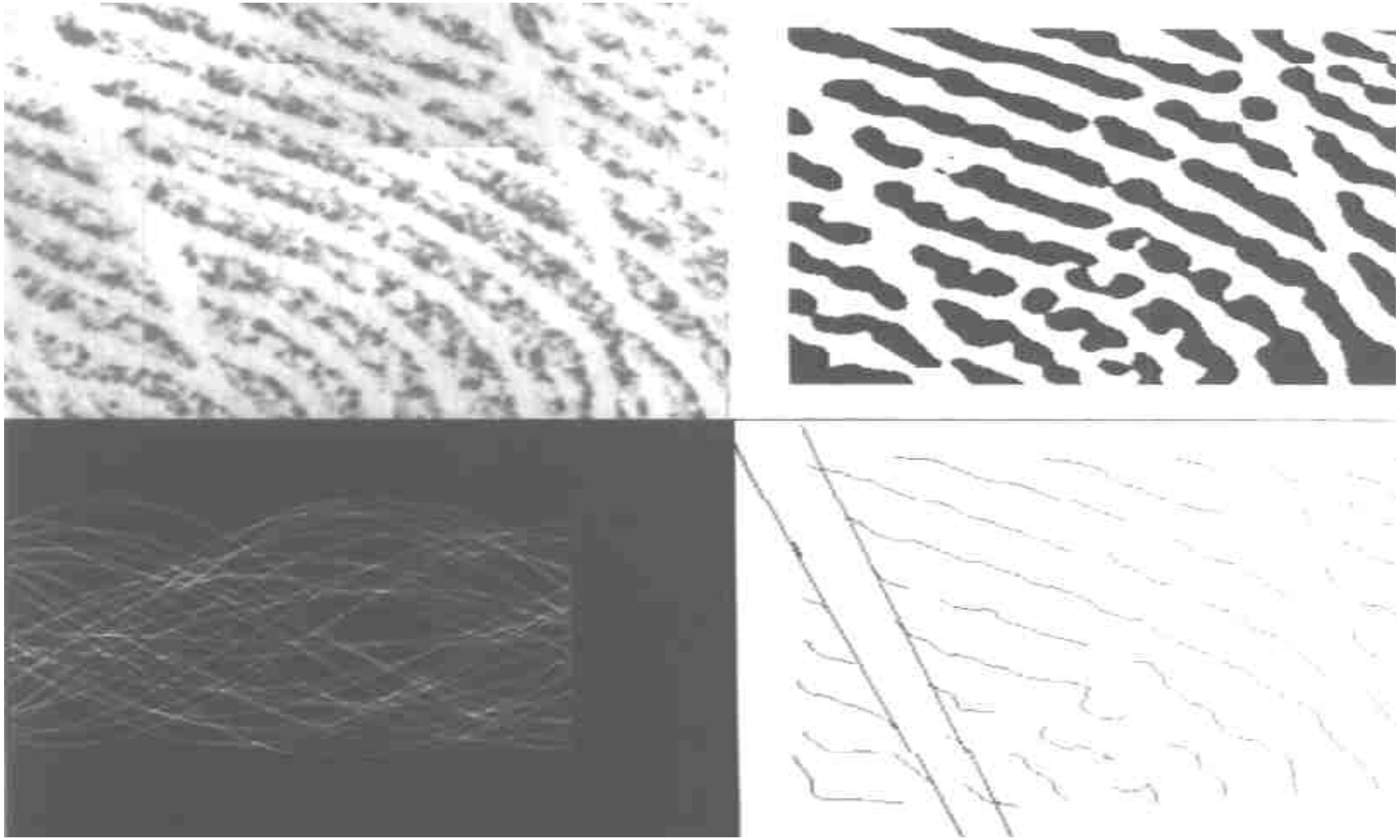
Morphological Operations

Thinning



Morphological Operations

Thinning



Morphological Operations

Thinning

From the perspective of mathematical morphology,
the thinning of a set X by a sequence of structuring elements L, is

$$X\mathrm{O}\left\{L^i\right\} \quad \left(\left(\left(\ldots\left(X\mathrm{O}L^1\right)\mathrm{O}L^2\right)\mathrm{O}L^3\right)\ldots\mathrm{O}L^i\right)$$

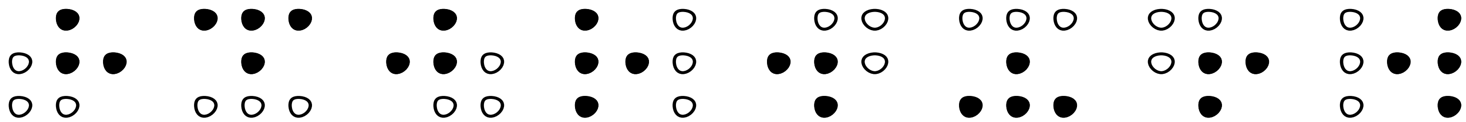
$$X\mathrm{O}L = X \ / \ X \otimes L$$

That is, the set X less the set of points in X which hit L

Morphological Operations

Thinning

If $X \otimes L$ identifies border points,
and L is appropriately structured to maintain connectivity of a set,
then repeated application of the thinning process successively removes border points from a set until the skeleton is achieved

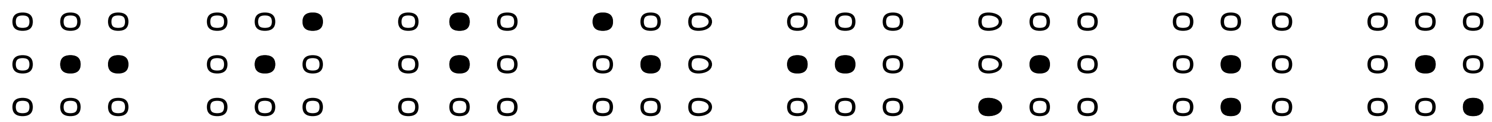


L^1 L^2 L^3 L^4 L^5 L^6 L^7 L^8

Morphological Operations

End Points

Given a skeleton set X, we can identify the endpoints, *i.e.* points which are connected to just one other point, using the Hit or Miss Transform and an appropriate set of structuring elements $\{E\}$



E^1 E^2 E^3 E^4 E^5 E^6 E^7 E^8

The endpoints of the skeleton are given by: $Y = \bigcup_{i=0}^{8} X \otimes E^i$

i.e. the union of all those points which hit with one of these end-point structuring elements

Demos

The following code is taken from the morphologicalErosion project in the lectures directory of the ACV repository

See:

```
morphologicalErosion.h
morphologicalErosionImplementation.cpp
morphologicalErosionApplication.cpp
```

```
/*
 * function processThresholdAndMorphologyOperation
 * Trackbar callback - binary threshold, input from user
 * Trackbar callback - size of structuring element, input from user
*/

void processThresholdAndMorphologyOperation(int, void*) {

   extern Mat input_image;
   extern int threshold_value;
   extern int half_structuring_element_size;
   extern char* binary_window_name;
   extern char* processed_window_name;

   Mat greyscale_image;
   Mat binary_image;
   Mat filtered_image;

   int structuring_element_size;

   structuring_element_size = half_structuring_element_size * 2 + 1;

   if (threshold_value < 1)  // the trackbar has a lower value of 0 which is invalid
      threshold_value = 1;
```

```
  /* convert from colour to greyscale if necessary */

  if (input_image.type() == CV_8UC3) { // colour image
     cvtColor(input_image, greyscale_image, CV_BGR2GRAY);
  }
  else {
     greyscale_image = input_image.clone();
  }

  /* perform binary thresholding */

  binary_image.create(greyscale_image.size(), CV_8UC1);

  threshold(greyscale_image, binary_image, threshold_value, 255,THRESH_BINARY);

  /* perform erosion */

  Mat structuring_element(structuring_element_size,structuring_element_size,CV_8U,Scalar(1)); // create structuring element

  erode(binary_image,filtered_image,structuring_element);

  imshow(binary_window_name,    binary_image);
  imshow(processed_window_name, filtered_image);
}
```

Demos

The following code is taken from the morphologicalDilation project in the lectures directory of the ACV repository

See:

```
morphologicalDilation.h
morphologicalDilationImplementation.cpp
morphologicalDilationApplication.cpp
```

```
if (input_image.type() == CV_8UC3) { // colour image
   cvtColor(input_image, greyscale_image, CV_BGR2GRAY);
}
else {
   greyscale_image = input_image.clone();
}

binary_image.create(greyscale_image.size(), CV_8UC1);

/* perform manual binary thresholding */

threshold(greyscale_image, binary_image, threshold_value, 255,THRESH_BINARY);

/* perform dilation */

Mat structuring_element(structuring_element_size,structuring_element_size,CV_8U,Scalar(1));  // create structuring element
dilate(binary_image,filtered_image,structuring_element);

imshow(binary_window_name,    binary_image);
imshow(processed_window_name, filtered_image);
}
```

Demos

The following code is taken from the morphologicalClosing project in the lectures directory of the ACV repository

See:

```
morphologicalClosing.h
morphologicalClosingImplementation.cpp
morphologicalClosingApplication.cpp
```

```
   /* convert from colour to greyscale if necessary */

   if (input_image.type() == CV_8UC3) { // colour image
      cvtColor(input_image, greyscale_image, CV_BGR2GRAY);
   }
   else {
      greyscale_image = input_image.clone();
   }

   /* perform binary thresholding */

   binary_image.create(greyscale_image.size(), CV_8UC1);
   threshold(greyscale_image, binary_image, threshold_value, 255,THRESH_BINARY);

   /* perform closing */

   Mat structuring_element(structuring_element_size,structuring_element_size,CV_8U,Scalar(1)); // create structuring element
   morphologyEx(binary_image,filtered_image,MORPH_CLOSE,structuring_element);

   imshow(binary_window_name,    binary_image);
   imshow(processed_window_name, filtered_image);
}
```

Demos

The following code is taken from the morphologicalOpening project in the lectures directory of the ACV repository

See:

```
morphologicalOpening.h
morphologicalOpeningImplementation.cpp
morphologicalOpeningApplication.cpp
```

```
/*
 * function processThresholdAndMorphologyOperation
 * Trackbar callback - size of structuring element, input from user
*/

void processMorphologyOperation(int, void*) {

   extern Mat input_image;
   extern int half_structuring_element_size;
   extern char* processed_window_name;

   Mat greyscale_image;
   Mat filtered_image;

   int structuring_element_size;

   structuring_element_size = half_structuring_element_size * 2 + 1;

   /* perform opening */

   Mat structuring_element(structuring_element_size,structuring_element_size,CV_8U,Scalar(1)); // create structuring element
   morphologyEx(input_image,filtered_image,MORPH_OPEN,structuring_element);  //MORPH_CLOSE, MORPH_DILATE, MORPH_ERODE

   imshow(processed_window_name, filtered_image);
 }
```

Exercises

1. Read OpenCV documentation for all OpenCV functions in sample code

2. Study utility functions in sample code

Reading

R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

Section 3.3 More neighborhood operations

Section 3.3.2 Morphology

Section 3.3.3 Distance transform