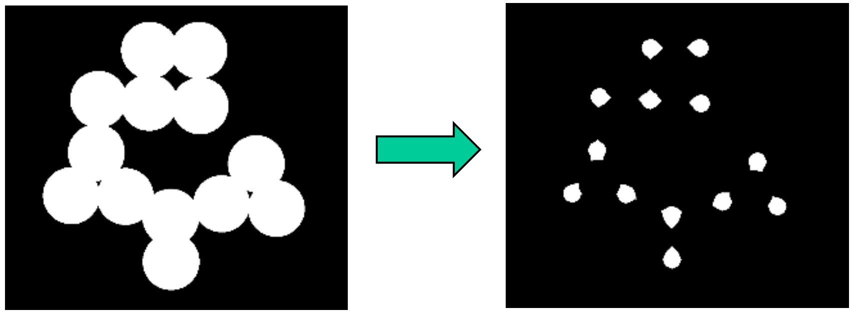
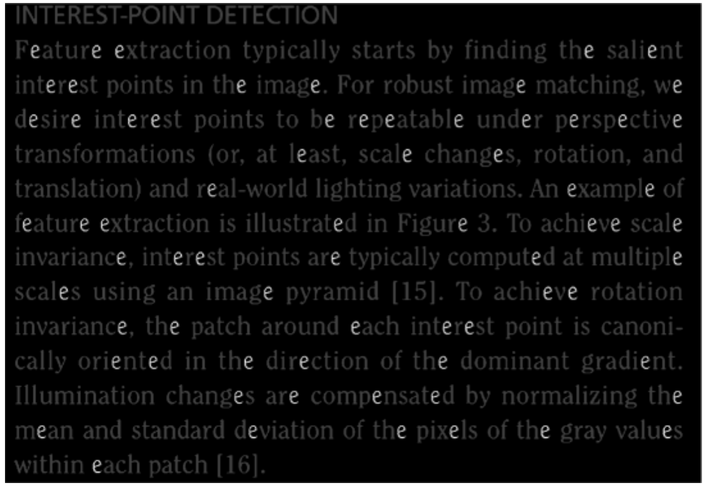
Morphological Image Processing

- Binary dilation and erosion
- Set-theoretic interpretation
- Opening, closing, morphological edge detectors
- Hit-miss filter
- Morphological filters for gray-level images
- Cascading dilations and erosions
- Rank filters, median filters, majority filters





INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Binary image processing

- Binary images are common
  - Intermediate abstraction in a gray-scale/color image analysis system
    - Thresholding/segmentation
    - Presence/absence of some image property
  - Text and line graphics, document image processing
- Representation of individual pixels as 0 or 1, convention:
  - foreground, object = 1 (white)
  - background = 0 (black)
- Processing by logical functions is fast and simple
- Shift-invariant logical operations on binary images: "*morphological*" *image processing*
- Morphological image processing has been generalized to gray-level images via level sets

Shift-invariance

- Assume that digital images $f[x,y]$ and $g[x,y]$ have infinite support

$$[x,y] \in \{\cdots,-2,-1,0,1,2,\cdots\} \times \{\cdots,-2,-1,0,1,2,\cdots\}$$

. . . then, for all integers a and b

$$f[x,y] \longrightarrow \boxed{\text{Shift-invariant system}} \longrightarrow g[x,y]$$

$$f[x-a,y-b] \longrightarrow \boxed{\text{Shift-invariant system}} \longrightarrow g[x-a,y-b]$$

- Shift-invariance does **not** imply linearity (or vice versa).

Structuring element

- Neighborhood "window" operator

180 degree rotation

"structuring element"

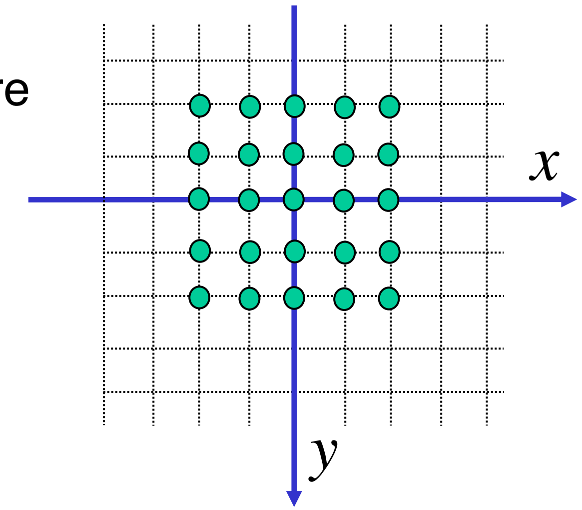
$$W\{f[x,y]\} = \{f[x-x', y-y'] : [x', y'] \in \Pi_{xy}\}$$

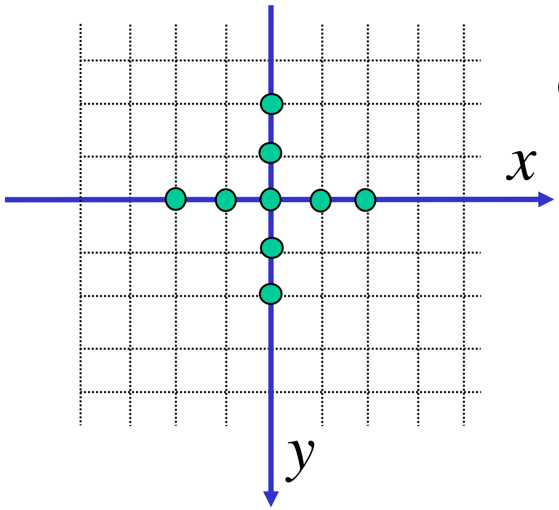
$$\hat{W}\{f[x,y]\} = \{f[x+x', y+y'] : [x', y'] \in \Pi_{xy}\}$$

The "hat" notation indicates upright structuring element (i.e., not rotated!)

- Example structuring elements Π_{xy}:

5x5 square

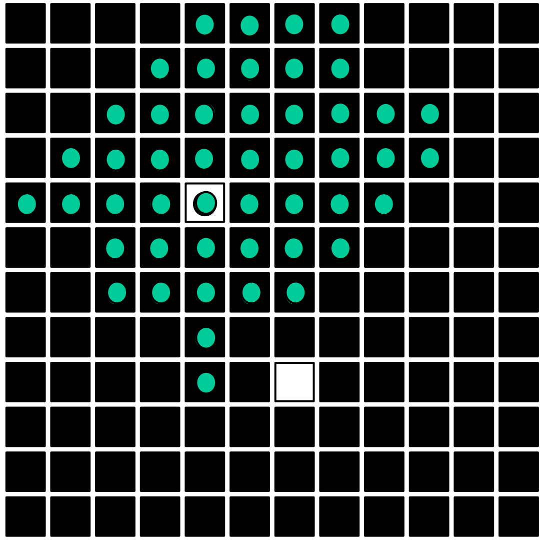




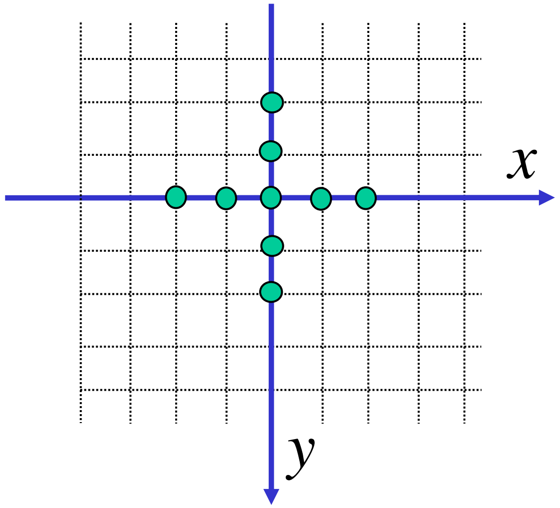
cross

Binary dilation (expanding foreground)

$$g[x,y] = OR\left[W\{f[x,y]\}\right] := dilate(f,W)$$



$f[x,y]$

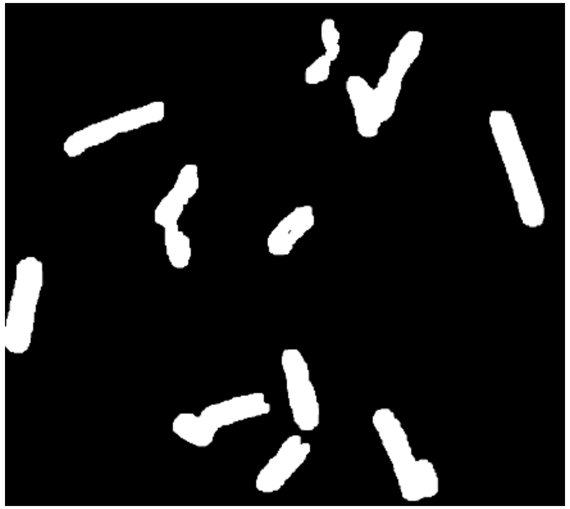


Π_{xy}

Binary dilation with square structuring element

$$g[x,y] = OR\left[W\left\{f[x,y]\right\}\right] := dilate(f,W)$$

Original (701x781)



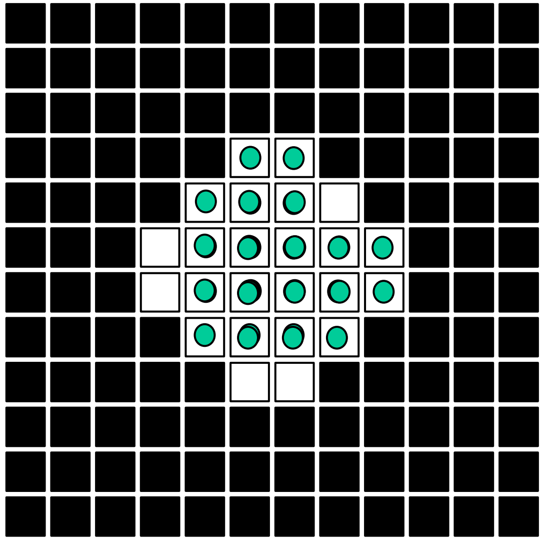
dilation with
3x3 structuring element

dilation with
7x7 structuring element

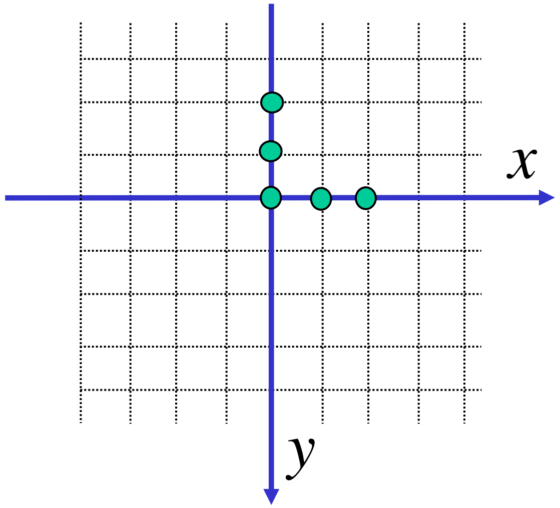
- Expands the size of 1-valued objects
- Smoothes object boundaries
- Closes holes and gaps

Binary erosion (shrinking foreground)

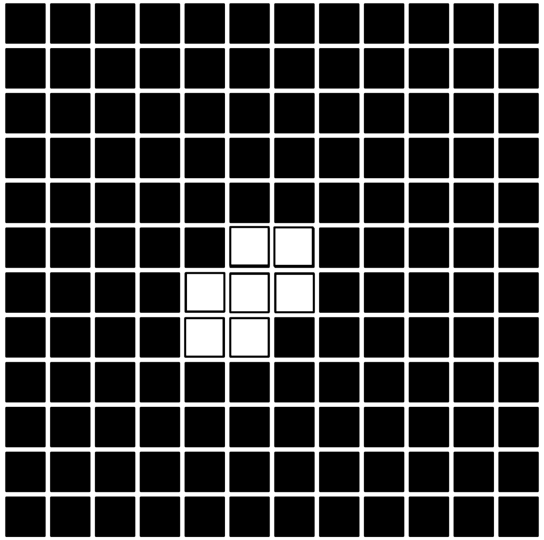
$$g[x,y] = AND\left[\hat{W}\{f[x,y]\}\right] := erode(f,W)$$



$f[x,y]$



Π_{xy}



$g[x,y]$

<u>Caveat:</u> There is another definition of erosion in the literature, which flips the structuring element, as for dilation. The Lagunita online videos use that alternative definition. Matlab function *imerode* uses the definition on this slide. To the best of our knowledge, there is no such discrepancy defining dilation.

Binary erosion with square structuring element

$$g[x,y] = AND\left[\hat{W}\{f[x,y]\}\right] := erode(f,W)$$

Original (701x781)

erosion with
3x3 structuring element

erosion with
7x7 structuring element

- Shrinks the size of 1-valued objects
- Smoothes object boundaries
- Removes peninsulas, fingers, and small objects

Relationship between dilation and erosion

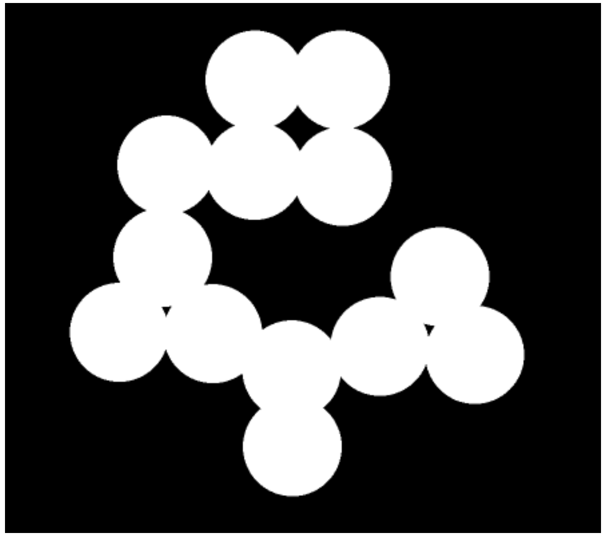
- Duality: erosion is dilation of the background

$$dilate(f, W) = NOT\left[erode\left(NOT[f], \hat{W}\right)\right]$$
$$erode(f, W) = NOT\left[dilate\left(NOT[f], \hat{W}\right)\right]$$

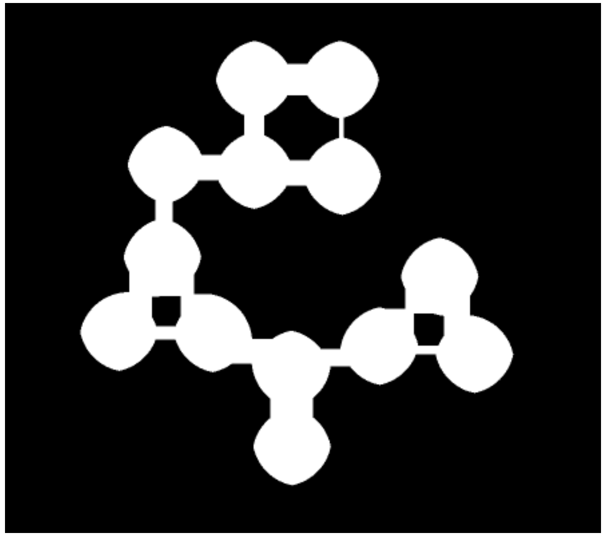
- But: erosion is <u>not</u> the inverse of dilation

$$\begin{aligned} f[x,y] &\neq erode\left(dilate\left(f, W\right), W\right) \\ &\neq dilate\left(erode\left(f, W\right), W\right) \end{aligned}$$

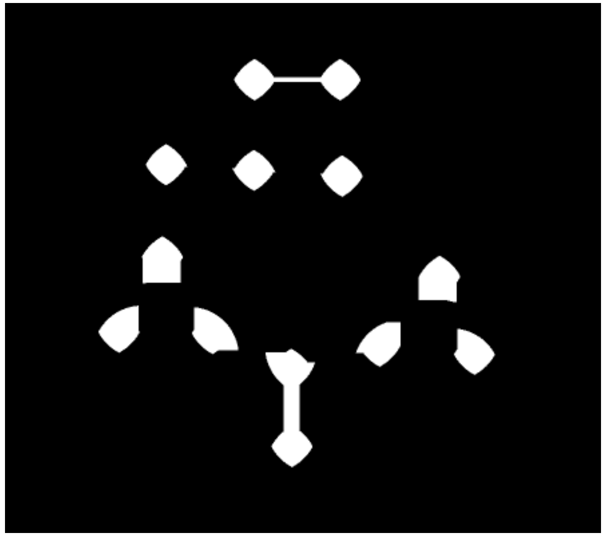
Example: blob separation/detection by erosion



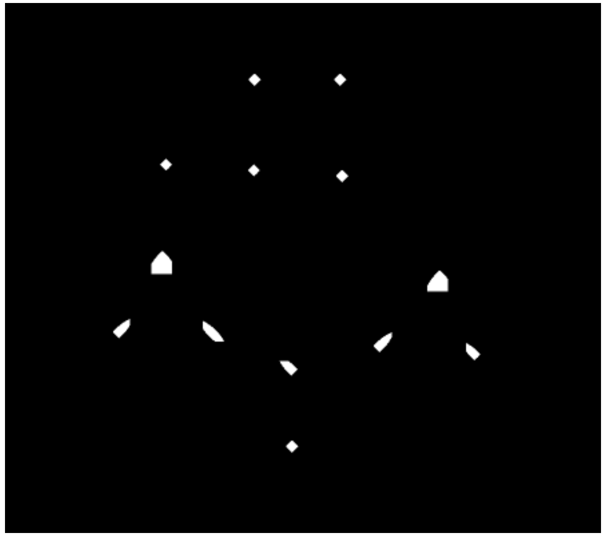
Original binary image
Circles (792x892)



Erosion by 30x30
structuring element

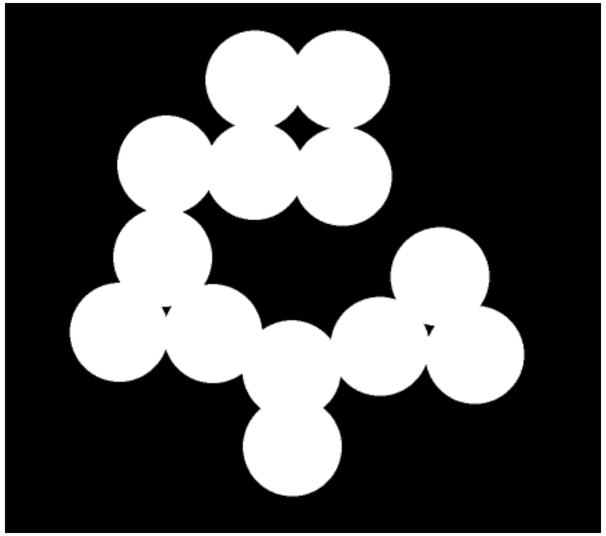


Erosion by 70x70
structuring element

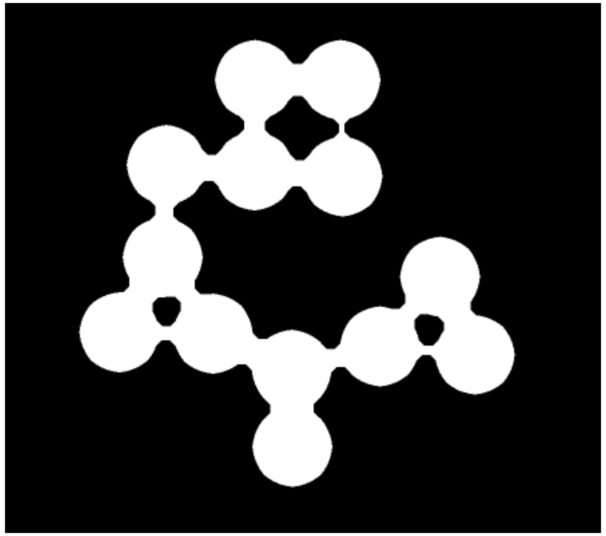


Erosion by 96x96
structuring element

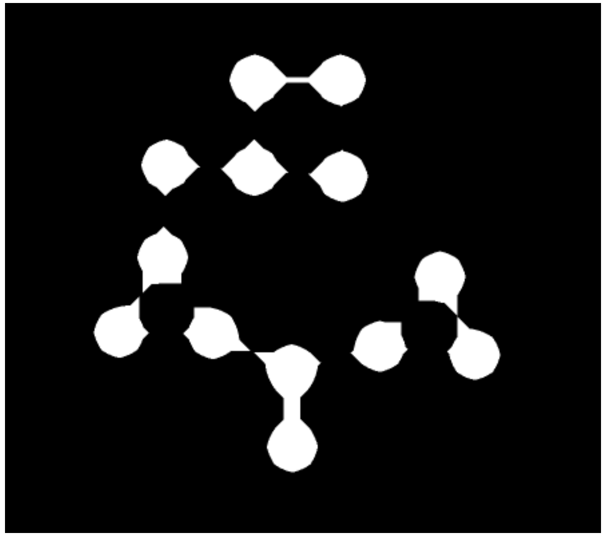
Example: blob separation/detection by erosion



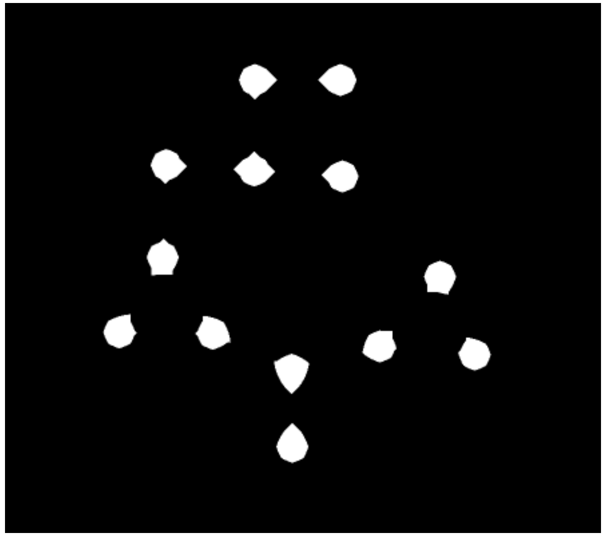
Original binary image
Circles (792x892)



Erosion by disk-shaped structuring element
Diameter=15

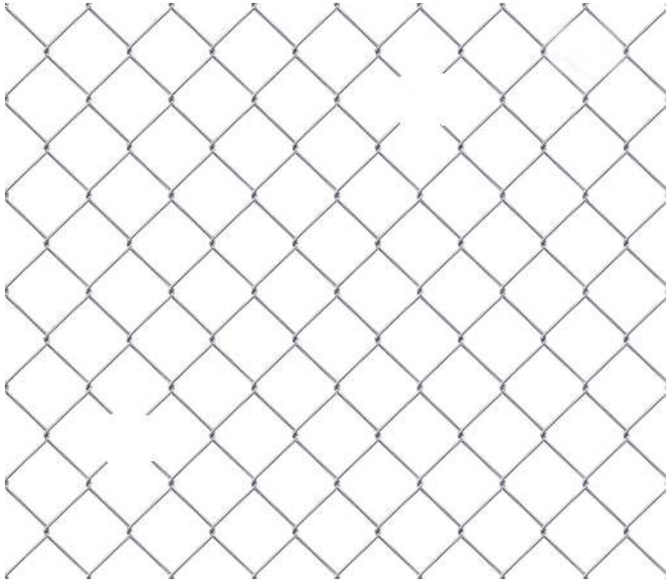


Erosion by disk-shaped structuring element
Diameter=35

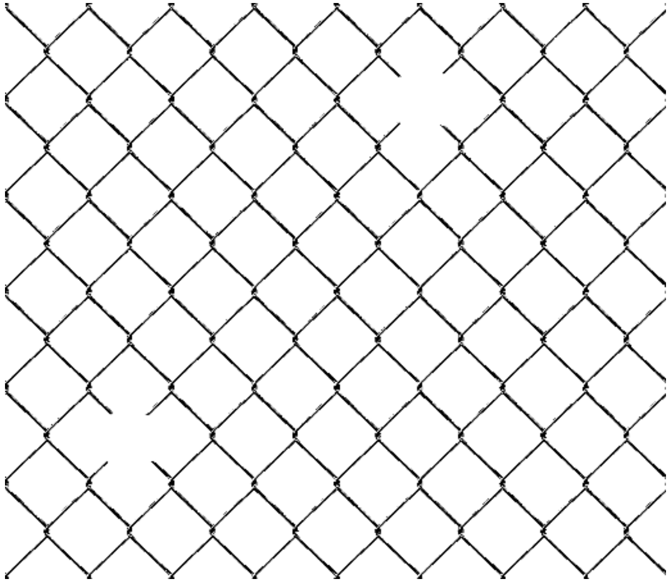


Erosion by disk-shaped structuring element
Diameter=48

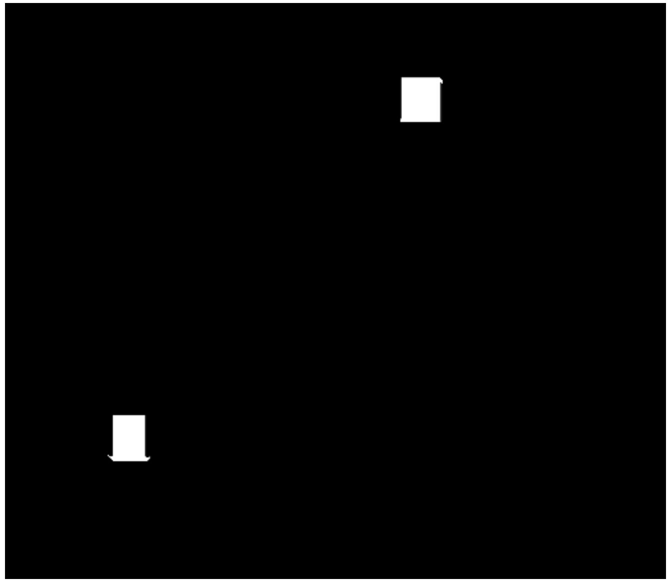
Example: chain link fence hole detection



Original grayscale image
Fence (1023 x 1173*)*



Fence thresholded
using Otsu's method



Erosion with 151x151
"cross" structuring element

Set-theoretic interpretation

- Set of object pixels

$$F \equiv \{(x,y) : f(x,y) = 1\}$$

Continuous (x,y). *Works for discrete* [x,y] *in the same way.*

- Background: complement of foreground set

$$F^c \equiv \{(x,y) : f(x,y) = 0\}$$

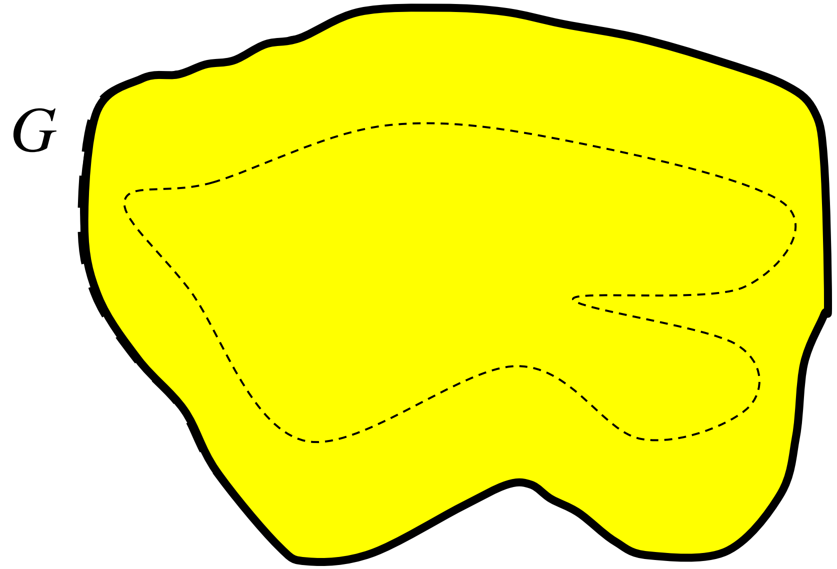
- Dilation is Minkowski set addition

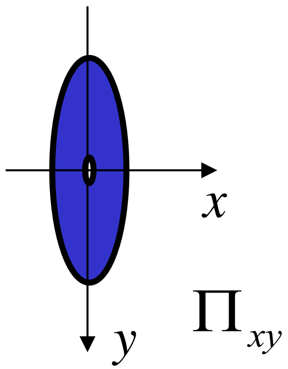
Commutative and associative!

$$\begin{aligned} G &= F \oplus \Pi_{xy} \\ &= \{(x+p_x, y+p_y) : (x,y) \in F, (p_x,p_y) \in \Pi_{xy}\} \\ &= \bigcup_{(p_x,p_y)\in\Pi_{xy}} F_{+(p_x,p_y)} \end{aligned}$$

translation of F by vector (p_x, p_y)

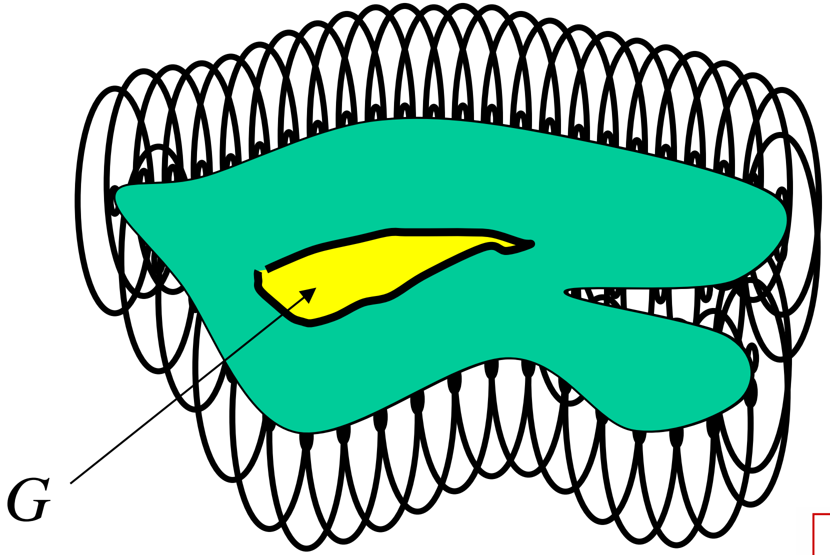
Set-theoretic interpretation: dilation

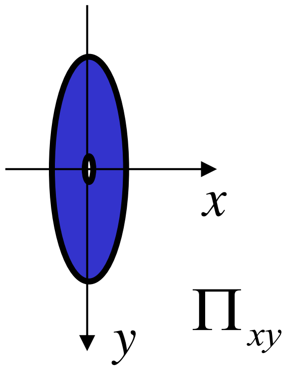




$$
\begin{aligned}
G &= F \oplus \Pi_{xy} \\
&= \left\{ \left(x + p_x, y + p_y \right) : \left(x, y \right) \in F, \left(p_x, p_y \right) \in \Pi_{xy} \right\} \\
&= \bigcup_{\left(p_x, p_y \right) \in \Pi_{xy}} F_{+\left(p_x, p_y \right)}
\end{aligned}
$$

Set-theoretic interpretation: erosion





Not commutative!
Not associative!

<u>Minkowski set subtraction</u>

$$G = \bigcap_{(-p_x,-p_y)\in\Pi_{xy}} F_{+(p_x,p_y)} = F \ominus \Pi_{xy}$$

Opening and closing

- Goal: smoothing without size change
- Open filter

$$open(f,W) = dilate(erode(f,W),W)$$

- Close filter

$$close(f,W) = erode(dilate(f,W),W)$$

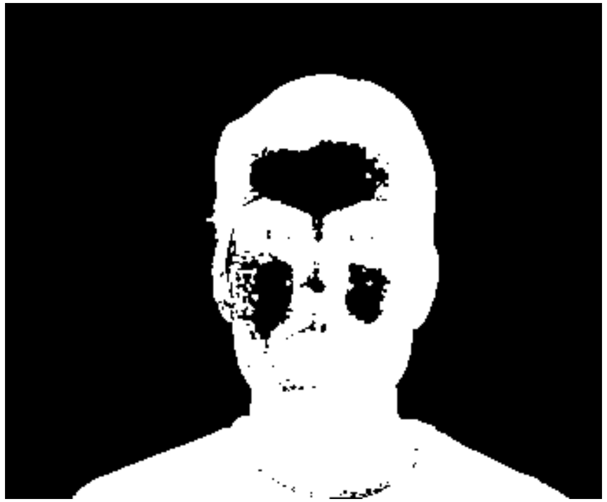
- Open filter and close filter are biased
  - Open filter removes small 1-regions
  - Close filter removes small 0-regions
  - Bias is often desired for enhancement or detection!
- Unbiased size-preserving smoothers

$$close-open(f,W) = close(open(f,W),W)$$

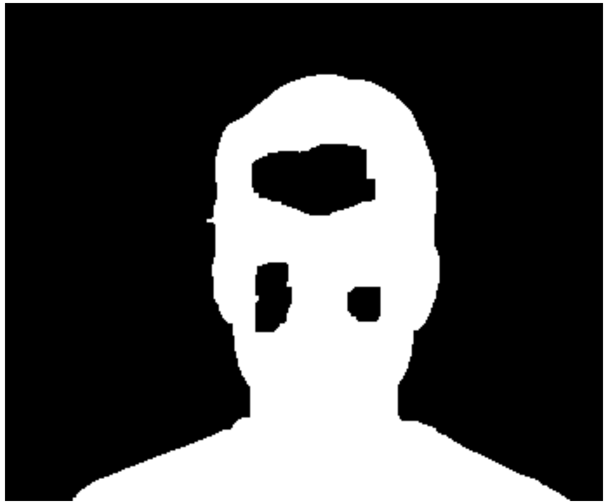
$$open-close(f,W) = open(close(f,W),W)$$

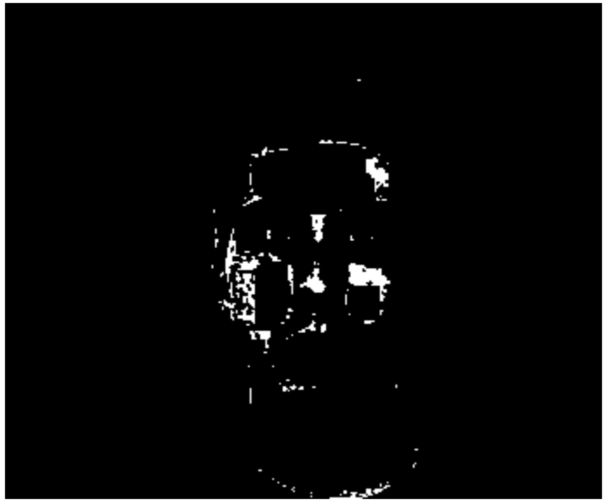
- *close-open* and *open-close* are duals, but not inverses of each other.

Small hole removal by closing

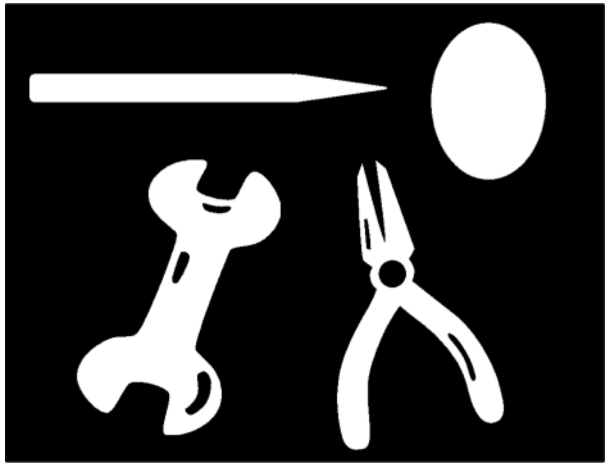

Original binary mask


Dilation
10x10

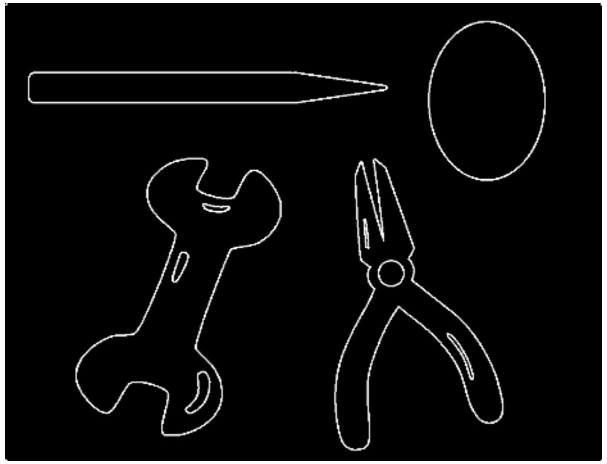

Closing 10x10


Difference to original mask

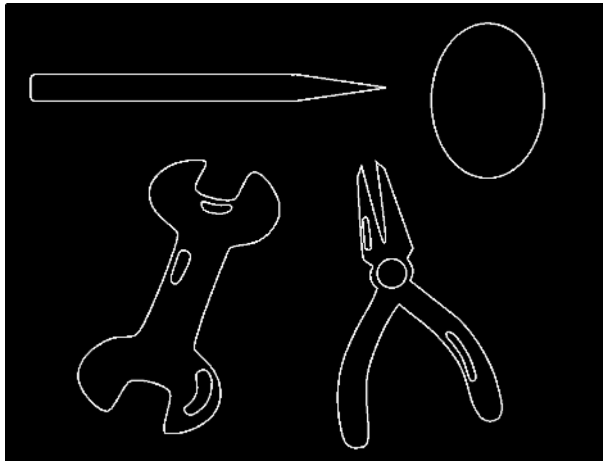
Morphological edge detectors



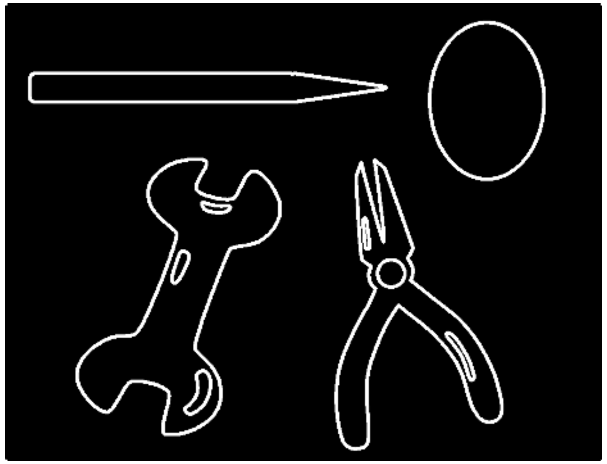
$f[x,y]$



$dilate(f,W) \neq f$



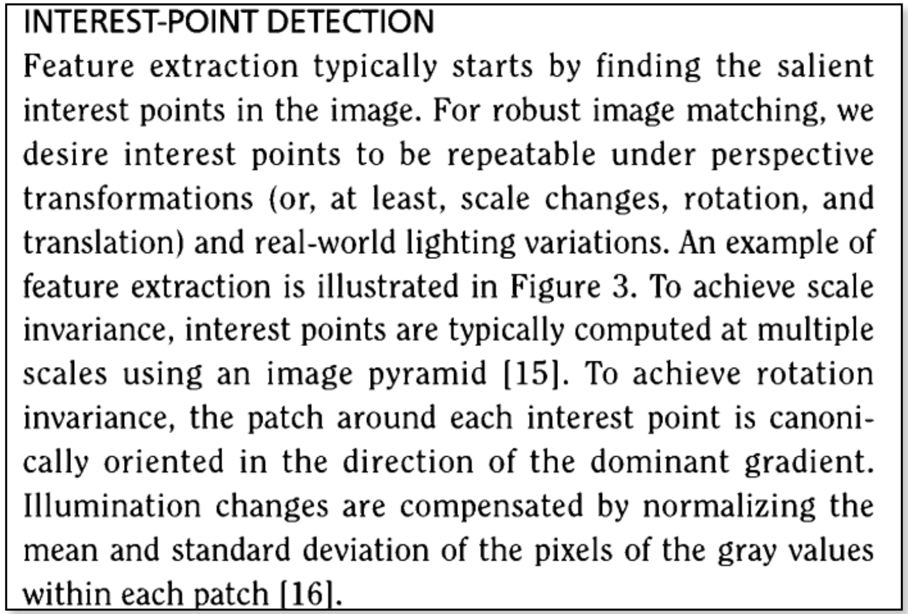
$erode(f,W) \neq f$



$dilate(f,W) \neq erode(f,W)$

Recognition by erosion

Binary image f

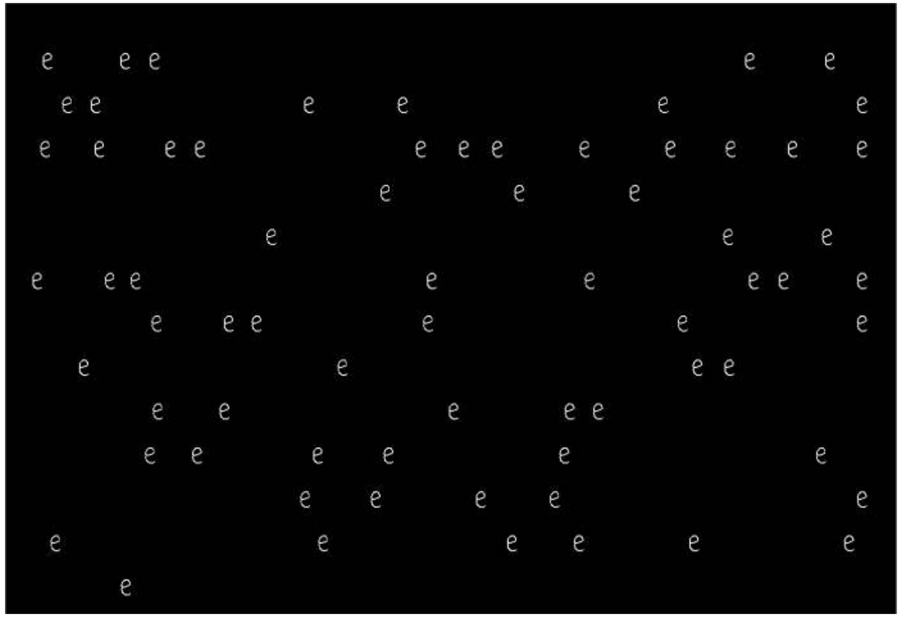

INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

1400

2000

$$\mathrm{dilate}\left(\mathrm{erode}\left(\mathrm{NOT}\left[f\right],W\right),W\right)$$

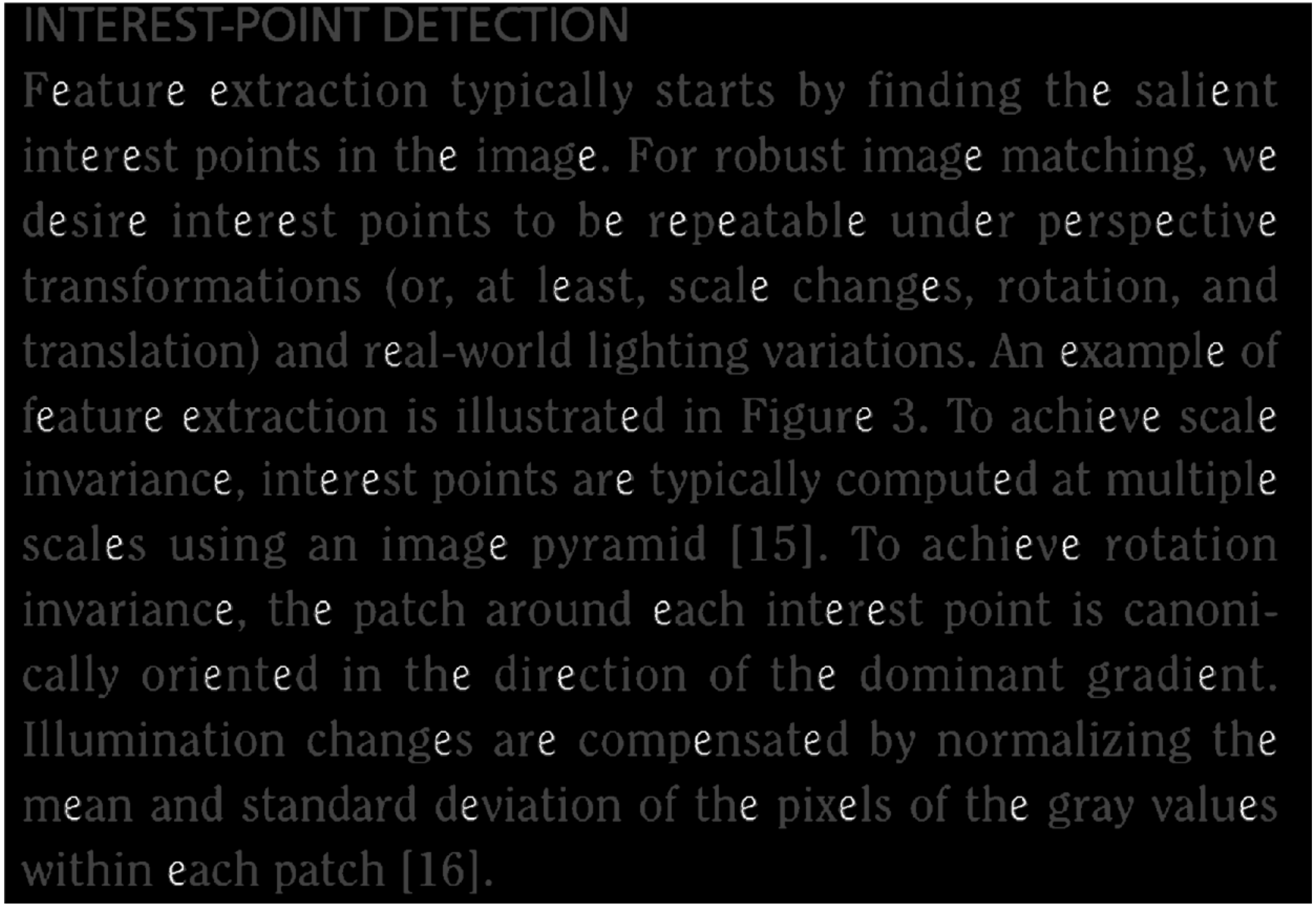


Structuring element W

44

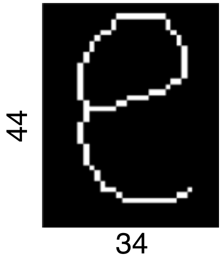
34

Recognition by erosion


INTEREST-POINT DETECTION

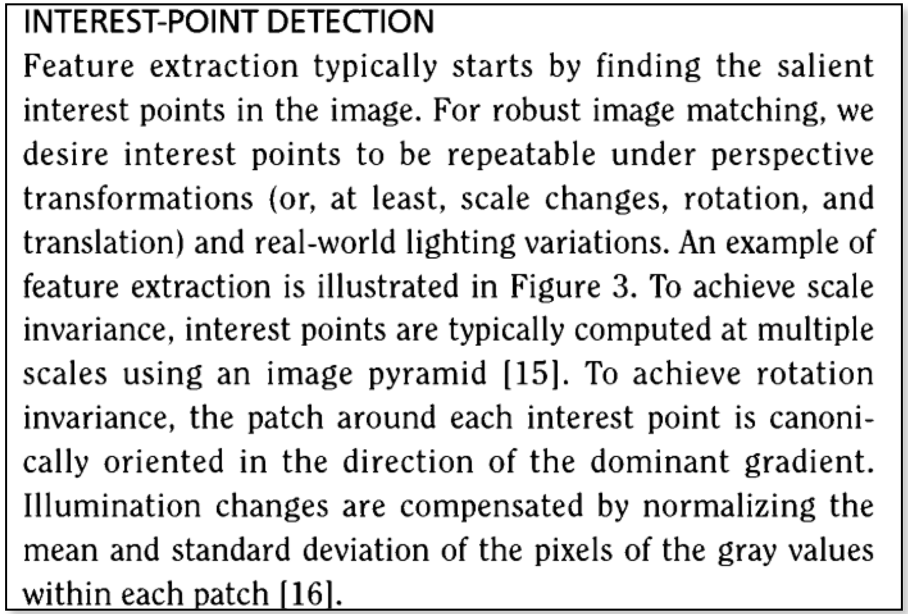
Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Structuring element W



Recognition by erosion

Binary image f

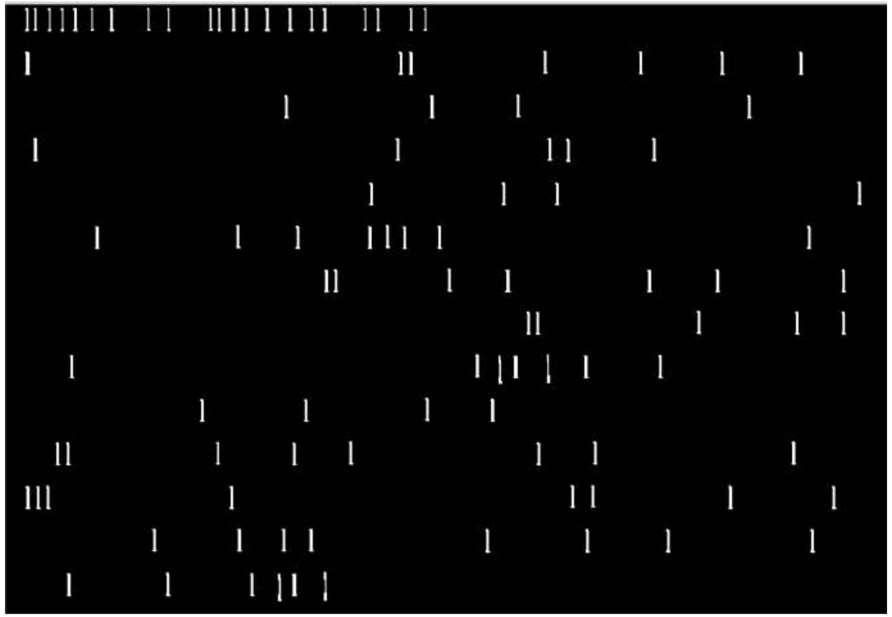

INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

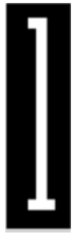
1400

2000

$$\mathrm{dilate}\left(\mathrm{erode}\left(\mathrm{NOT}\left[f\right],W\right),W\right)$$



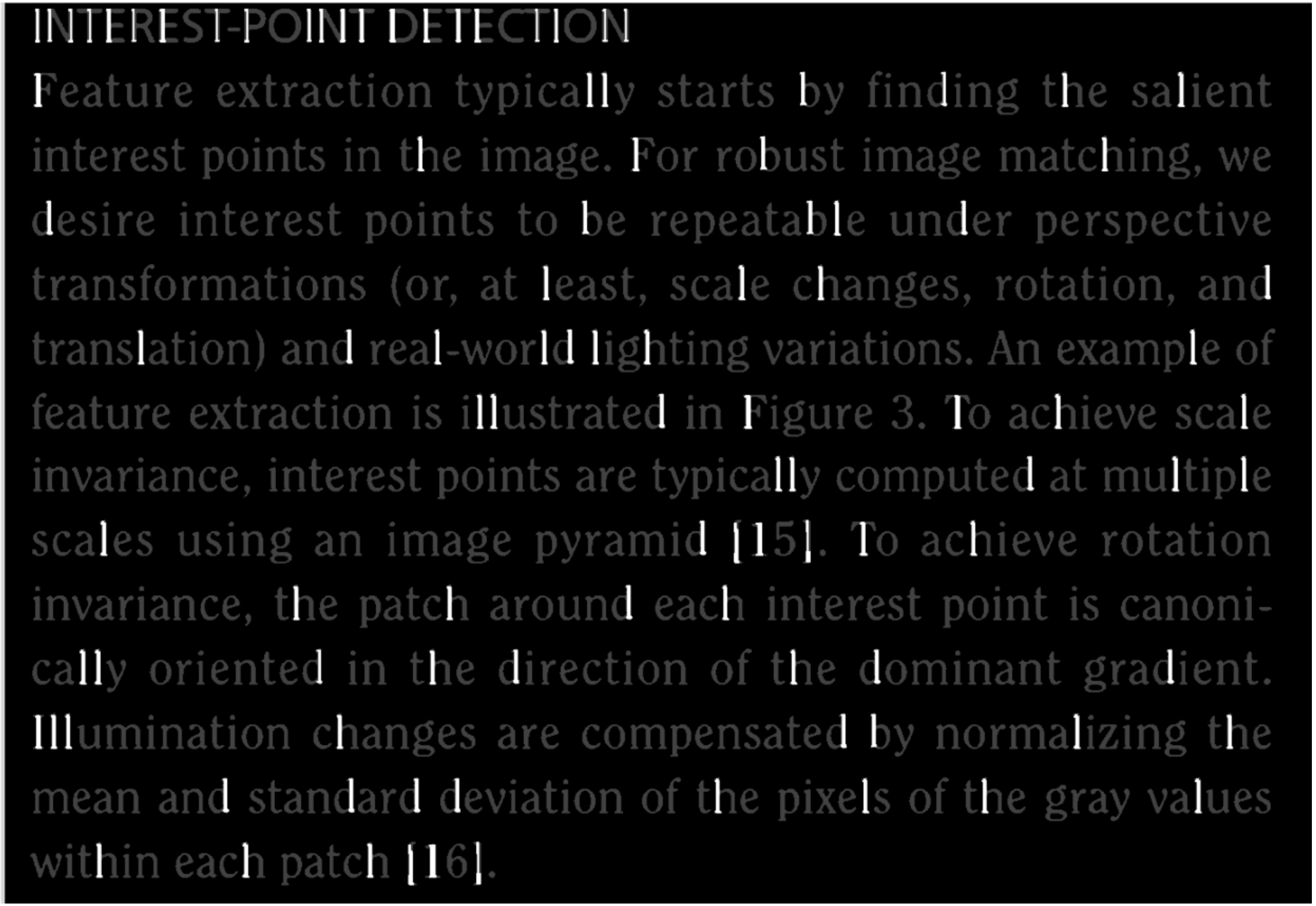
Structuring element W



62

18

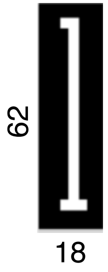
Recognition by erosion



INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Structuring element W

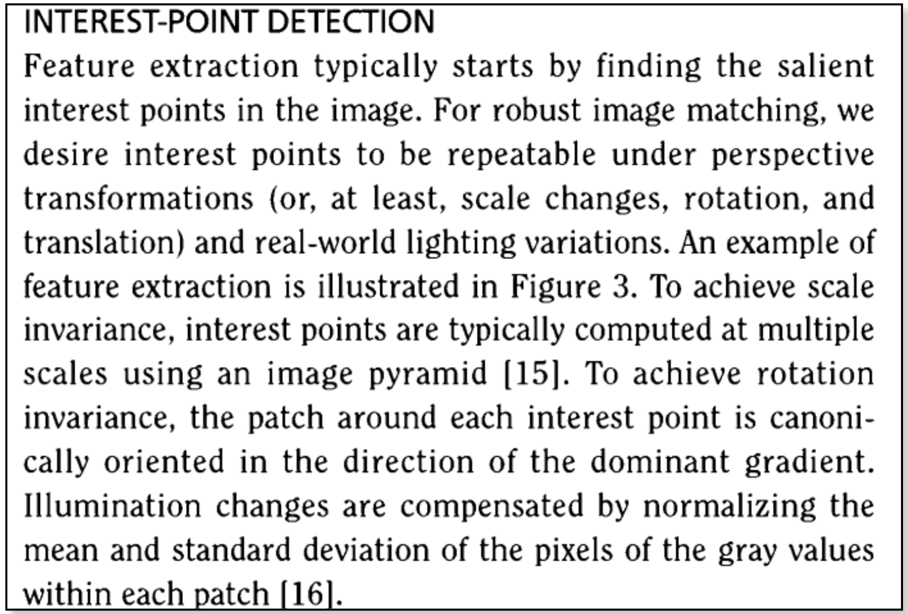


Hit-miss filter

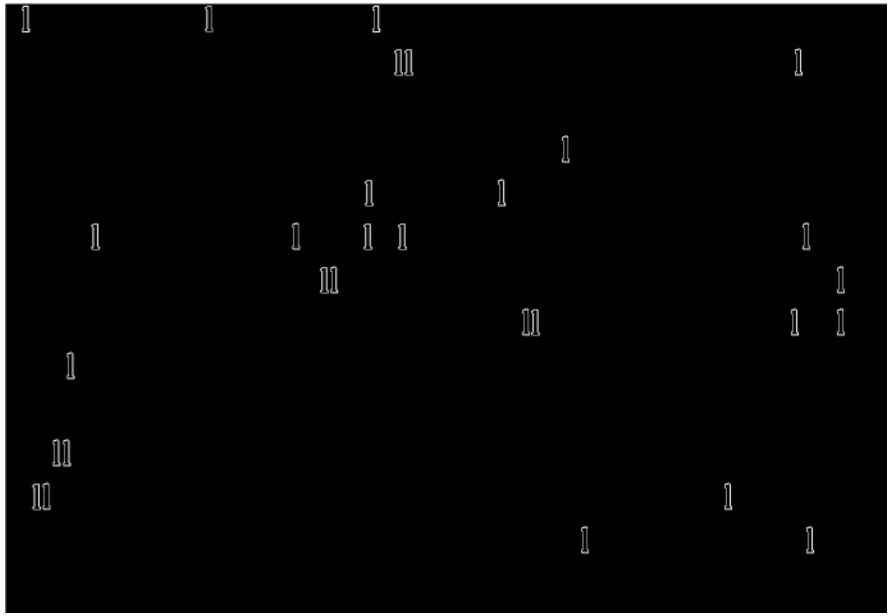
Binary image f

$$dilate\left(erode\left(NOT\left[f\right],V\right) \,\&\, erode\left(f,W\right),W\right)$$

1400


INTERESTPOINT DETECTION

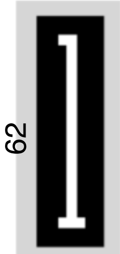
Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].



2000

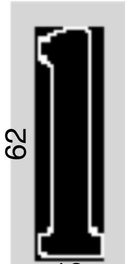
Structuring element V

62



18

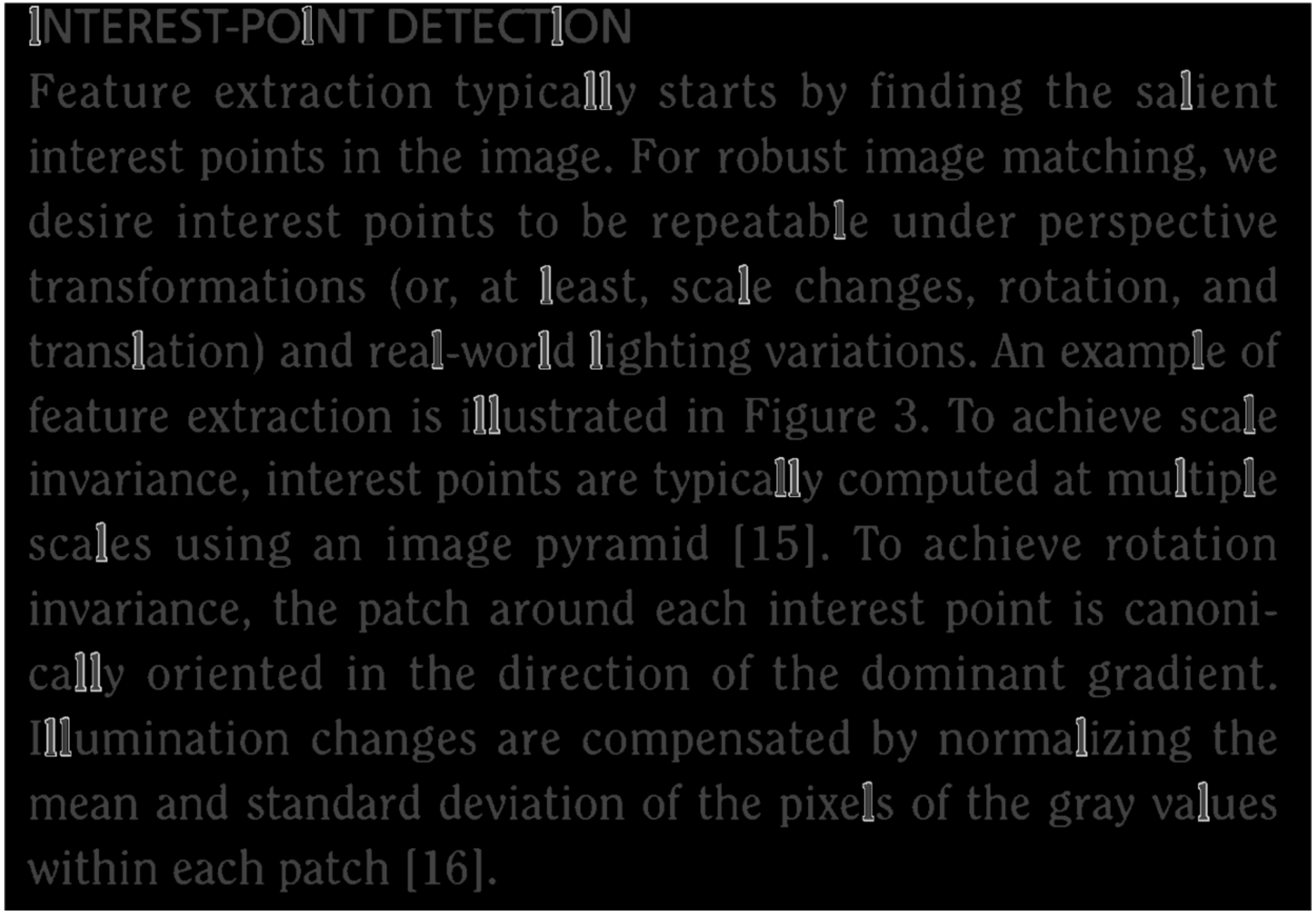
62



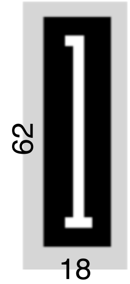
18

Structuring element W

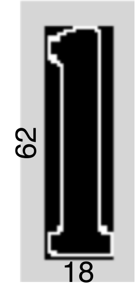
Hit-miss filter



Structuring element V



Structuring element W



Morphological filters for gray-level images

- Threshold sets of a gray-level image $f[x,y]$ (aka level sets)

$$\mathrm{T}_\theta\left(f\left[x,y\right]\right)=\left\{\left[x,y\right]:f\left[x,y\right]\geq\theta\right\}, \qquad -\infty<\theta<+\infty$$

- Reconstruction of original image from threshold sets

$$f\left[x,y\right]=\sup\left\{\theta:\left[x,y\right]\in\mathrm{T}_\theta\left(f\left[x,y\right]\right)\right\}$$

- Idea of morphological operators for multi-level (or continuous-amplitude) signals
  - Decompose into threshold sets
  - Apply binary morphological operator to each threshold set
  - Reconstruct via supremum operation
  - Gray-level operators thus obtained: ***flat operators***

➔ *Flat morphological operators and thresholding are commutative*

Dilation/erosion for gray-level images

- Explicit decomposition into threshold sets not required in practice
- Flat dilation operator: local maximum over window W

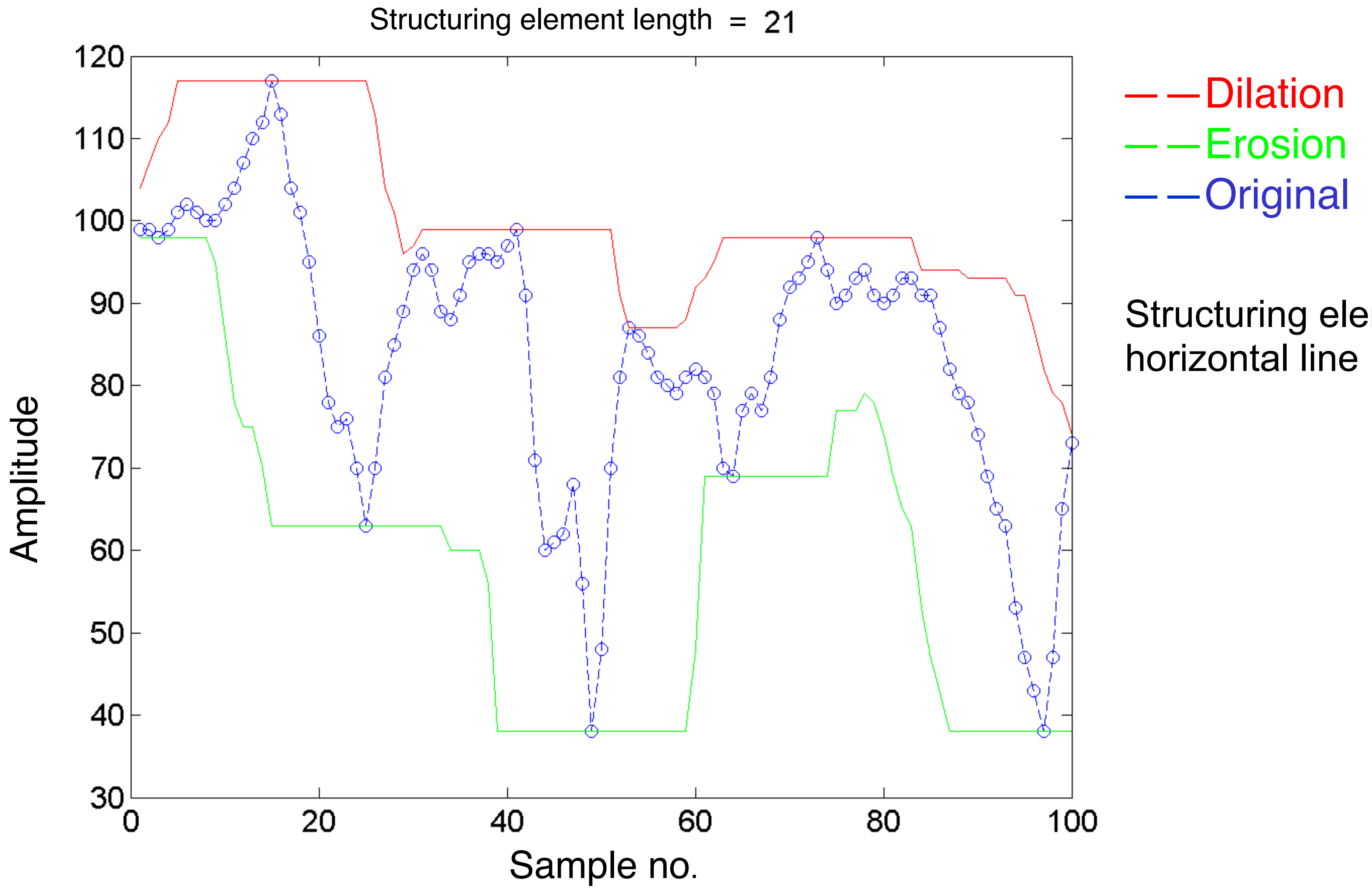
$$g[x,y] = \max\{W\{f[x,y]\}\} := dilate(f, W)$$

- Flat erosion operator: local minimum over window W

$$g[x,y] = \min\{\hat{W}\{f[x,y]\}\} := erode(f, W)$$

- Binary dilation/erosion operators contained as special case

1-d illustration of erosion and dilation



Structuring element: horizontal line

Image example

Original 394 x 305

Dilation 10x10 square

Erosion 10x10 square

Flat dilation with different structuring elements

Original

Diamond

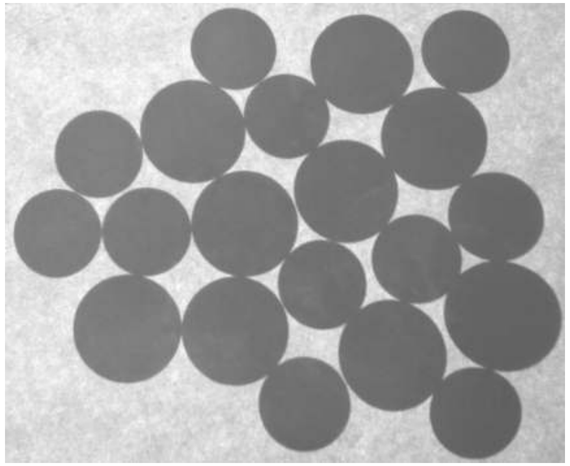
Disk

20 degree line

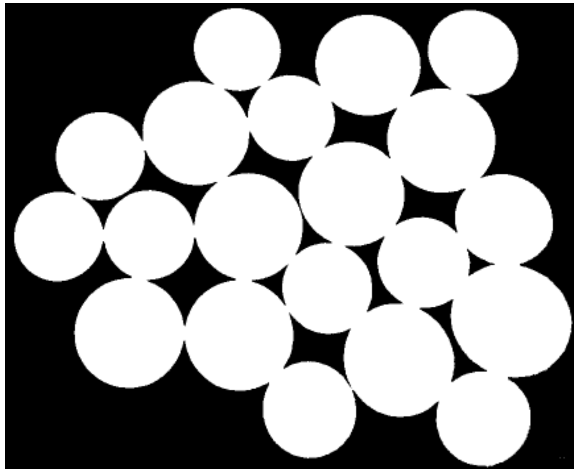
9 points

2 horizontal lines

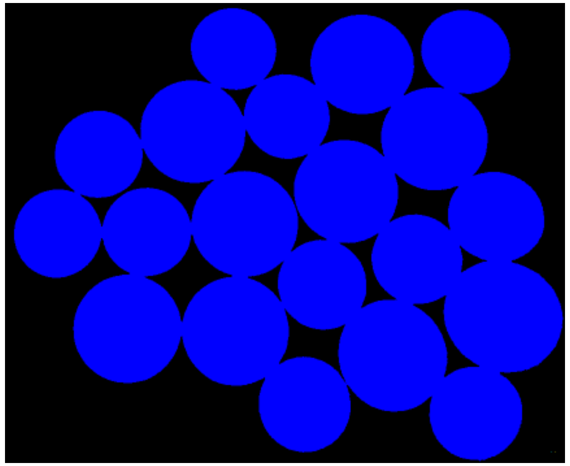
Example: counting coins



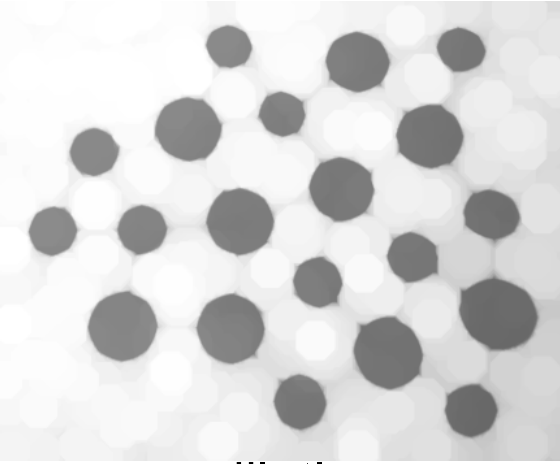
Original



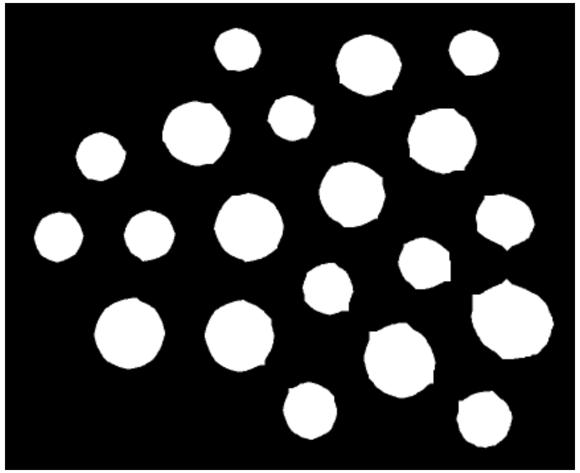
thresholded



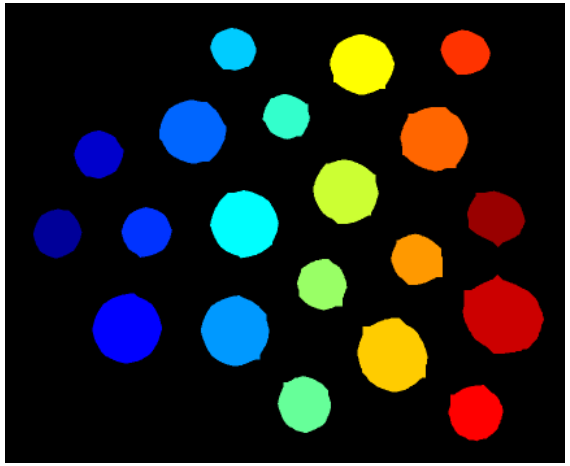
1 connected component



dilation

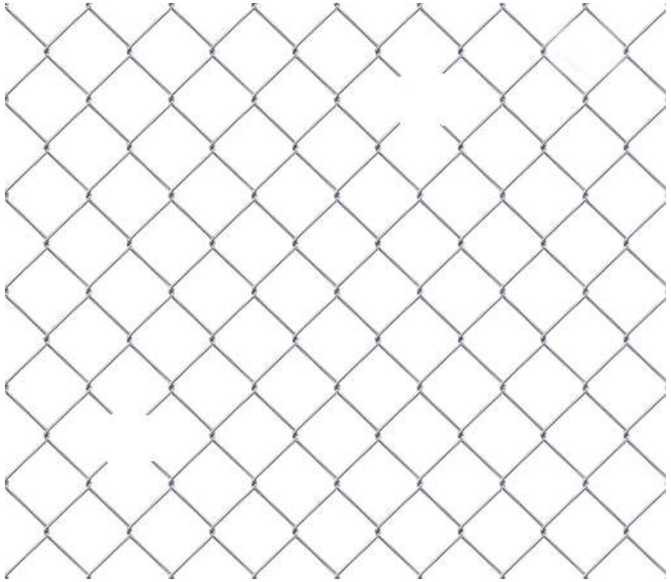


thresholded after dilation

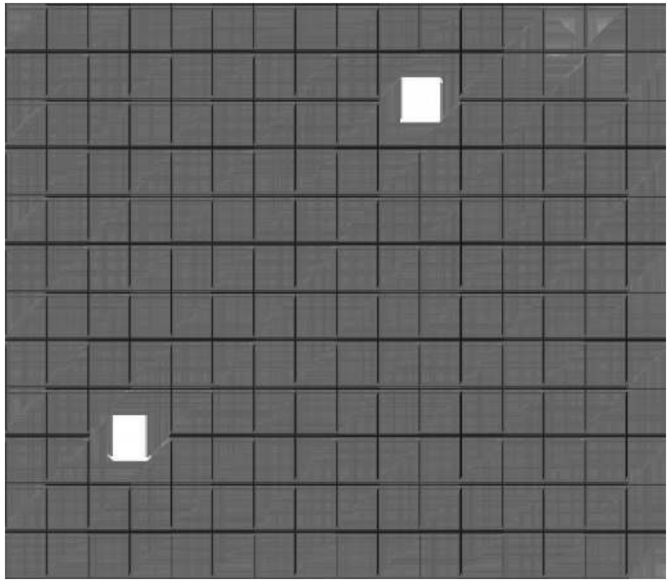


20 connected components

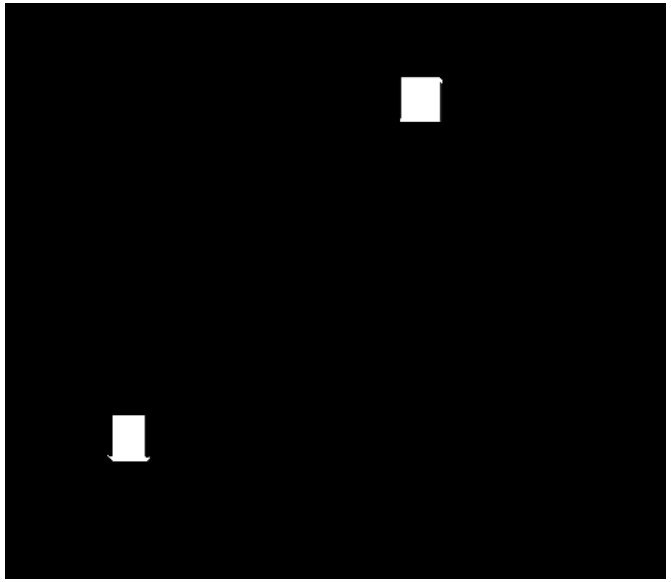
Example: chain link fence hole detection



Original grayscale image
Fence (1023 x 1173*)*



Flat erosion with 151x151
“cross” structuring element



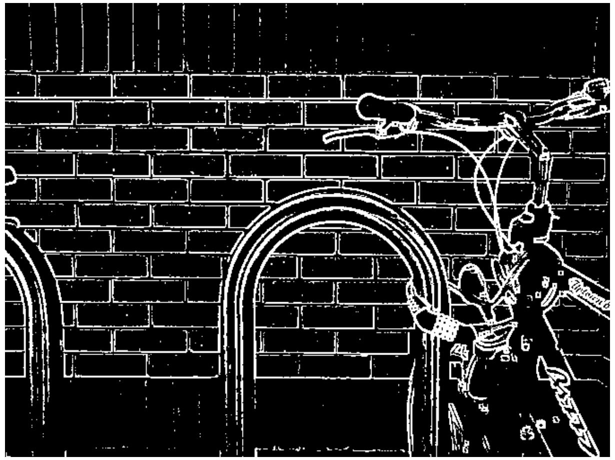
Binarized by Thresholding

Morphological edge detector

original f

dilation g

$g - f$



$g - f$ thresholded

Beyond flat morphological operators

- General dilation operator

$$g[x,y] = \sup_{\alpha,\beta}\left\{ f[x-\alpha, y-\beta] + w[\alpha,\beta] \right\} = \sup_{\alpha,\beta}\left\{ w[x-\alpha, y-\beta] + f[\alpha,\beta] \right\}$$

- Like linear convolution, with $\sup$ replacing summation, addition replacing multiplication
- Dilation with “unit impulse”

$$d[\alpha,\beta] = \begin{cases} 0 & \alpha = \beta = 0 \\ -\infty & \text{else} \end{cases}$$

does not change input signal:

$$f[x,y] = \sup_{\alpha,\beta}\left\{ f[x-\alpha, y-\beta] + d[\alpha,\beta] \right\}$$

Flat dilation as a special case

- Find $w[\alpha,\beta]$ such that

$$f[x,y] = \sup_{\alpha,\beta}\left\{ f[x-\alpha, y-\beta] + w[\alpha,\beta] \right\} = dilate(f, W)$$

- Answer:

$$w[\alpha,\beta] = \begin{cases} 0 & [\alpha,\beta] \in \Pi_{xy} \\ -\infty & \text{else} \end{cases}$$

- Hence, write in general

$$g[x,y] = \sup_{\alpha,\beta}\left\{ f[x-\alpha, y-\beta] + w[\alpha,\beta] \right\}$$
$$= dilate(f, w) = dilate(w, f)$$

General erosion for gray-level images

- General erosion operator

$$g[x,y] = \inf_{\alpha,\beta}\left\{ f[x+\alpha, y+\beta] - w[\alpha,\beta]\right\} = erode(f,w)$$

- Dual of dilation

$$g[x,y] = \inf_{\alpha,\beta}\left\{ f[x+\alpha, y+\beta] - w[\alpha,\beta]\right\}$$

$$= -\sup_{\alpha,\beta}\left\{ -f[x+\alpha, y+\beta] + w[\alpha,\beta]\right\} = -dilate(-f,\hat{w})$$

- Flat erosion contained as a special case

Cascaded dilations

f

$dilate(f, w_1)$

$dilate(f, w_1, w_2)$

$dilate(f, w_1, w_2, w_3)$

$$dilate\left[dilate(f, w_1), w_2\right] = dilate(f, w)$$

$$\text{where } w = dilate(w_1, w_2)$$

Cascaded erosions

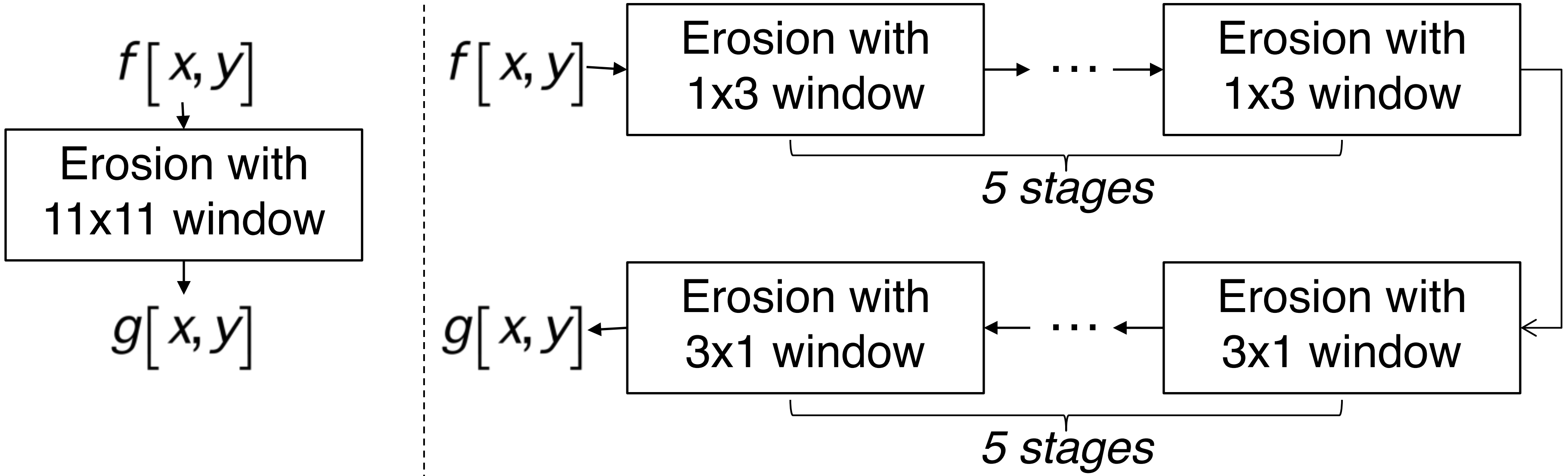
- Cascaded erosions can be lumped into single erosion

$$
\begin{aligned}
erode\left[erode(f, w_1), w_2\right] &= erode\left[-dilate(-f, \hat{w}_1), w_2\right] \\
&= -dilate\left[dilate(-f, \hat{w}_1), \hat{w}_2\right] \\
&= -dilate(-f, \hat{w}) \\
&= erode(f, w) \\
\text{where} \quad w &= dilate(w_1, w_2)
\end{aligned}
$$

- New structuring element (SE) is <u>not</u> the erosion of one SE by the other, but dilation.

Fast dilation and erosion

- Idea: build larger dilation and erosion operators by cascading simple, small operators
- Example: binary erosion by 11x11 window



120 AND per pixel

2x10 = 20 AND per pixel

Rank filters

- Generalisation of flat dilation/erosion: in lieu of min or max value in window, use the p-th ranked value
- Increases robustness against noise
- Best-known example: median filter for noise reduction
- Concept useful for both gray-level and binary images
- All rank filters are commutative with thresholding

Median filter

- Gray-level median filter

$$g[x,y] = median\left[W\{f[x,y]\}\right] := median(f,W)$$

- Binary images: majority filter

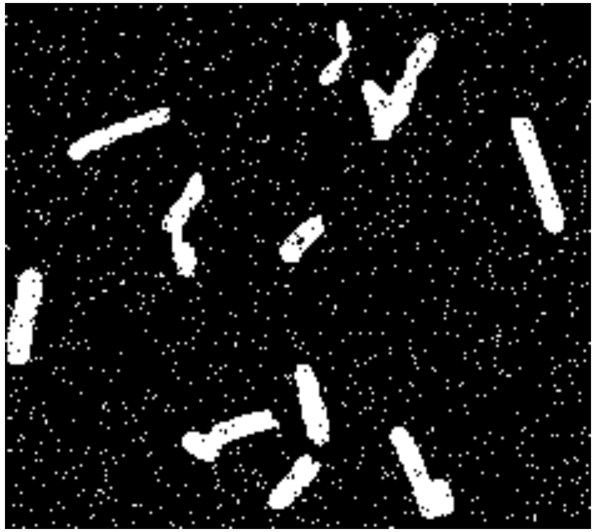
$$g[x,y] = MAJ\left[W\{f[x,y]\}\right] := majority(f,W)$$

- Self-duality

$$median(f,W) = -\left[median(-f,W)\right]$$

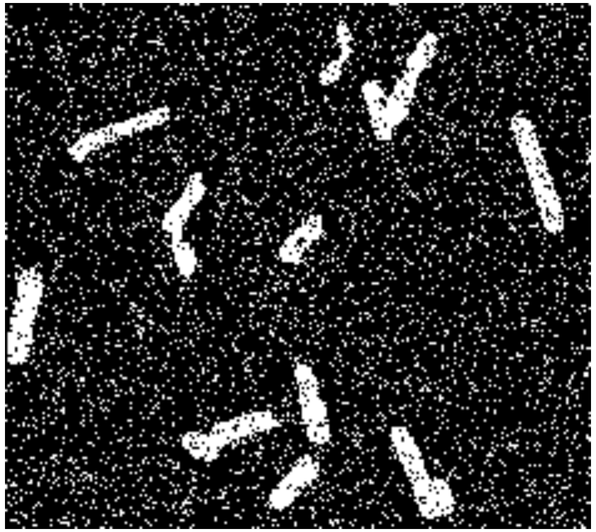
$$majority(f,W) = NOT\left[majority(NOT[f],W)\right]$$

Majority filter: example



Binary image with
5% 'Salt&Pepper' noise

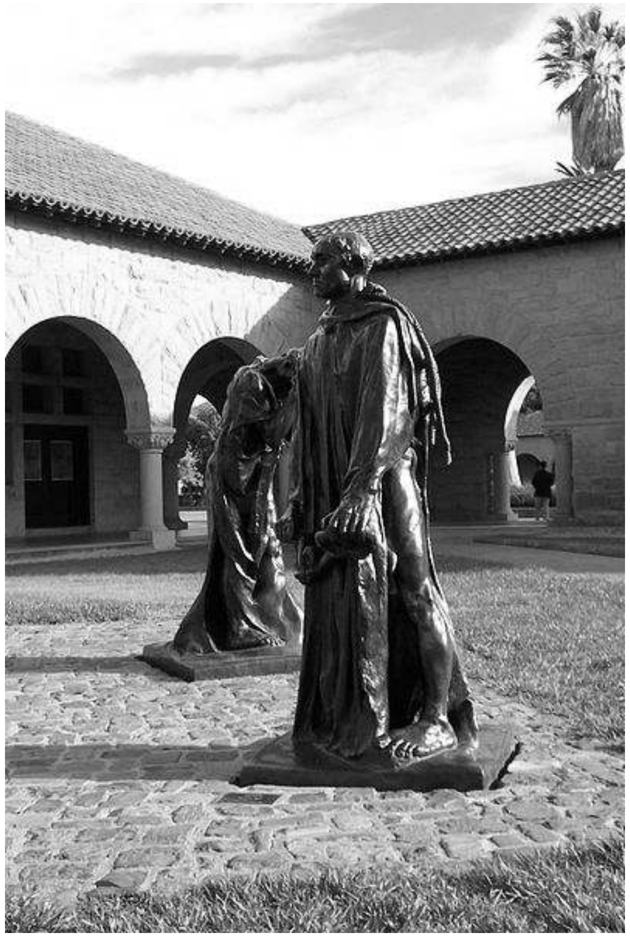
3x3 majority filter



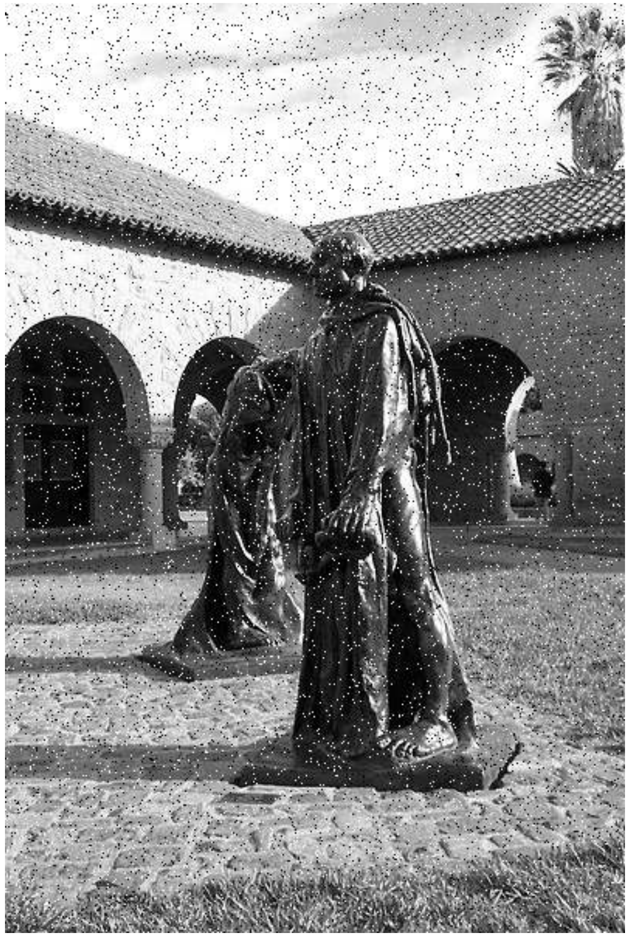
20% 'Salt&Pepper' noise

3x3 majority filter

Median filter: example



Original image

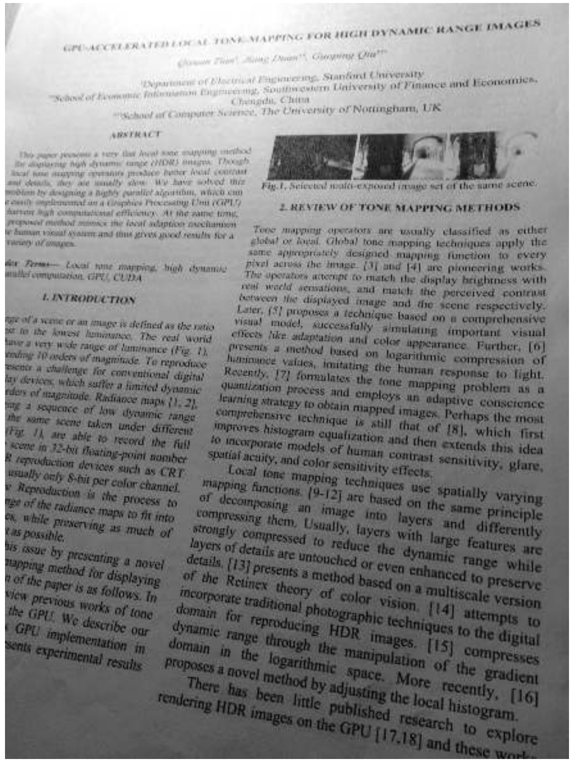


5% ‘Salt&Pepper’ noise

3x3 median filtering

7x7 median filtering

Example: non-uniform lighting compensation



GPU-ACCELERATED LOCAL TONE-MAPPING FOR HIGH DYNAMIC RANGE IMAGES

Department of Electrical Engineering, Stanford University

School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, China

School of Computer Science, The University of Nottingham, UK

ABSTRACT

Fig.1. Selected multi-exposed image set of the same scene.

1. INTRODUCTION

2. REVIEW OF TONE MAPPING METHODS

Tone mapping operators are usually classified as either global or local. Global tone mapping techniques apply the same appropriately designed mapping function to every pixel across the image. [3] and [4] are pioneering works. The operators attempt to match the display brightness with real world sensations, and match the perceived contrast between the displayed image and the scene respectively. Later, [5] proposes a technique based on a comprehensive visual model, successfully simulating important visual effects like adaptation and color appearance. Further, [6] presents a method based on logarithmic compression of luminance values, imitating the human response to light. Recently, [7] formulates the tone mapping problem as a quantization process and employs an adaptive conscience learning strategy to obtain mapped images. Perhaps the most comprehensive technique is still that of [8], which first improves histogram equalization and then extends this idea to incorporate models of human contrast sensitivity, glare, spatial acuity, and color sensitivity effects.

Local tone mapping techniques use spatially varying mapping functions. [9-12] are based on the same principle of decomposing an image into layers and differently compressing them. Usually, layers with large features are strongly compressed to reduce the dynamic range while layers of details are untouched or even enhanced to preserve details. [13] presents a method based on a multiscale version of the Retinex theory of color vision. [14] attempts to incorporate traditional photographic techniques to the digital domain for reproducing HDR images. [15] compresses dynamic range through the manipulation of the gradient domain in the logarithmic space. More recently, [16] proposes a novel method by adjusting the local histogram.

There has been little published research to explore rendering HDR images on the GPU [17,18] and these works

Original image

1632x1216 pixels

Dilation (local max)

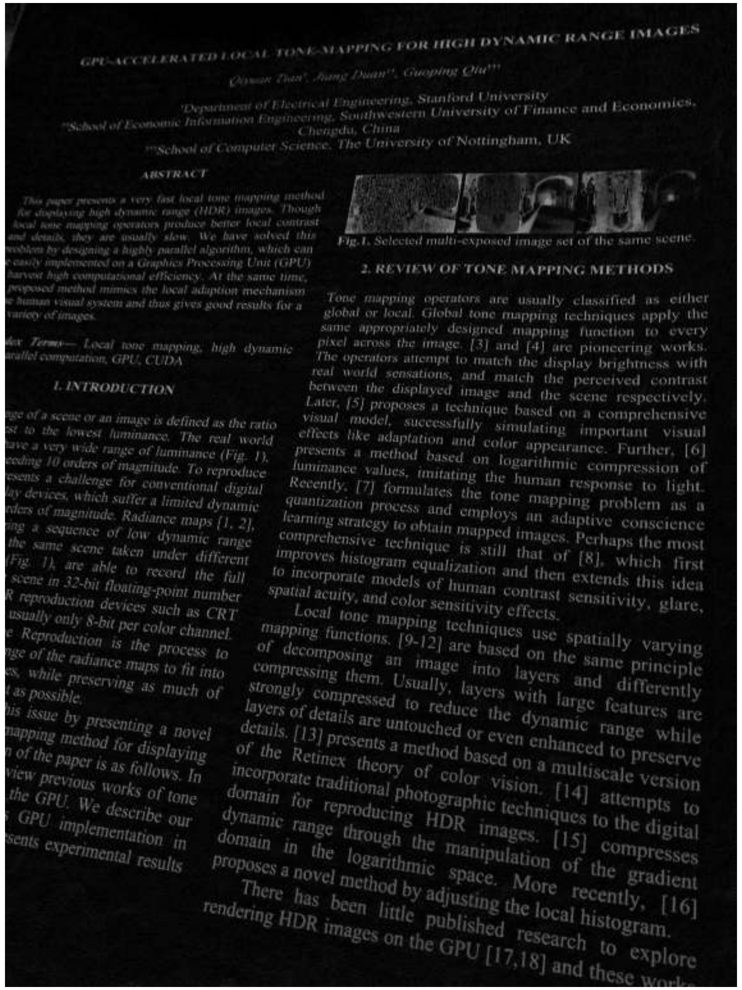
61x61 structuring element

Rank filter

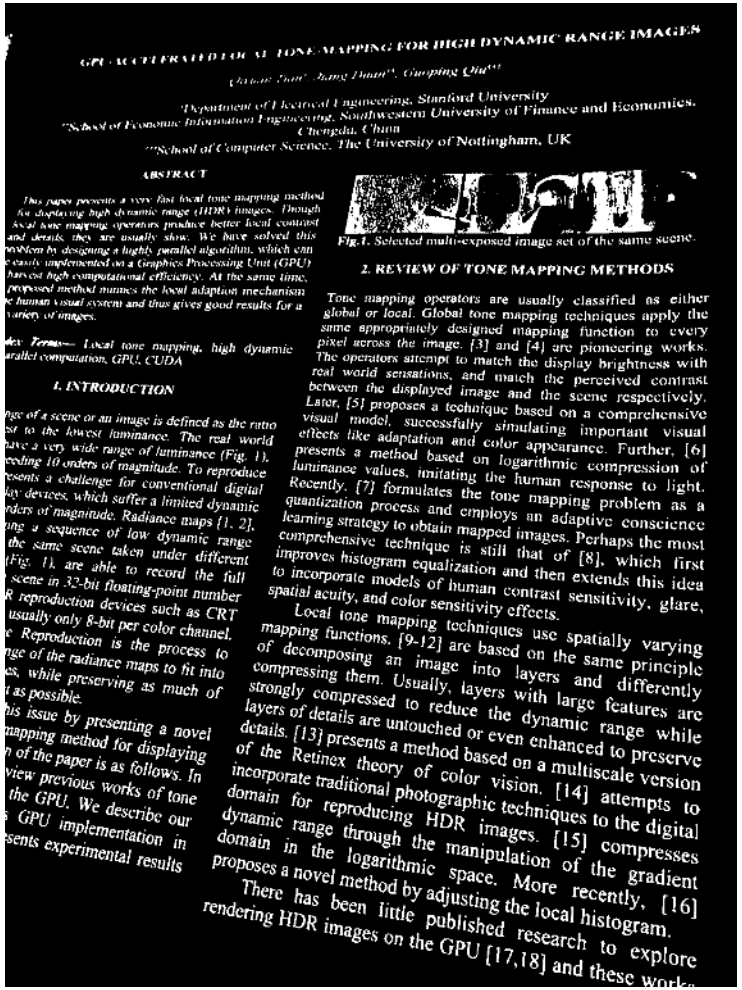
10st brightest pixel

61x61 structuring element

Example: non-uniform lighting compensation



Background – original image



After global thresholding