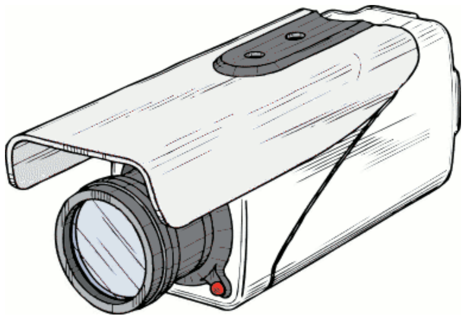
Univerza v Ljubljani

Machine perception
Image formation & Image processing 1

Matej Kristan

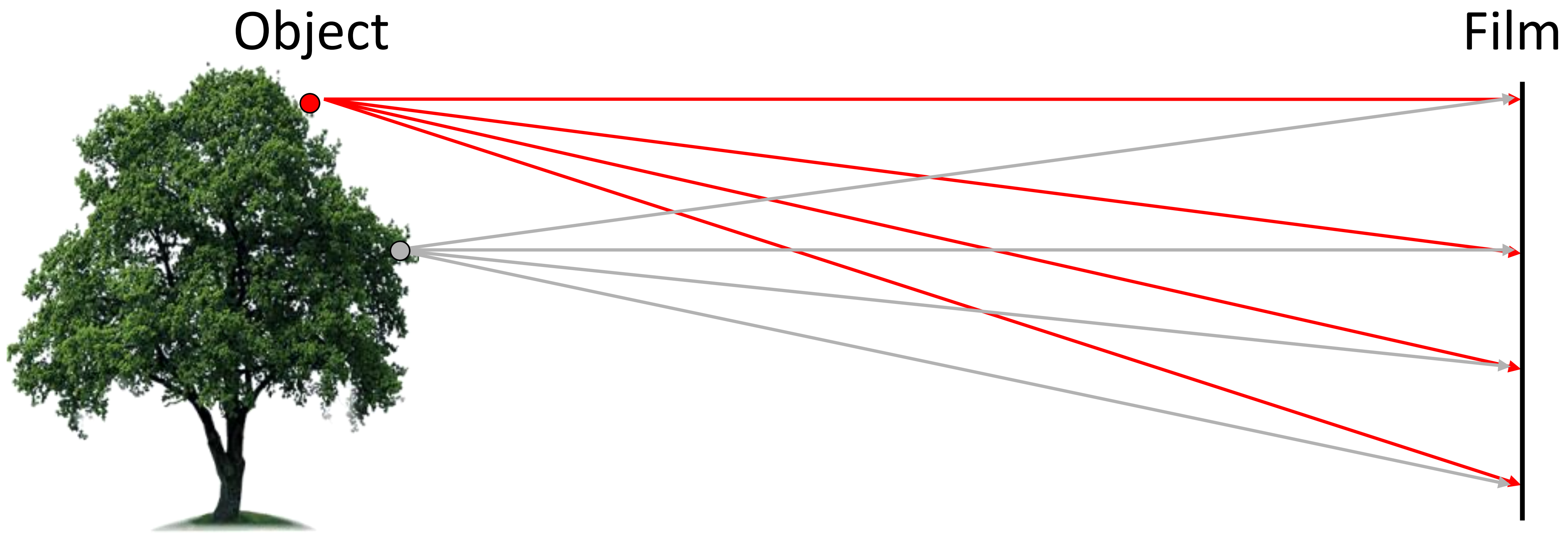


Laboratorij za Umetne Vizualne Spoznavne Sisteme,
Fakulteta za računalništvo in informatiko,
Univerza v Ljubljani

Machine perception

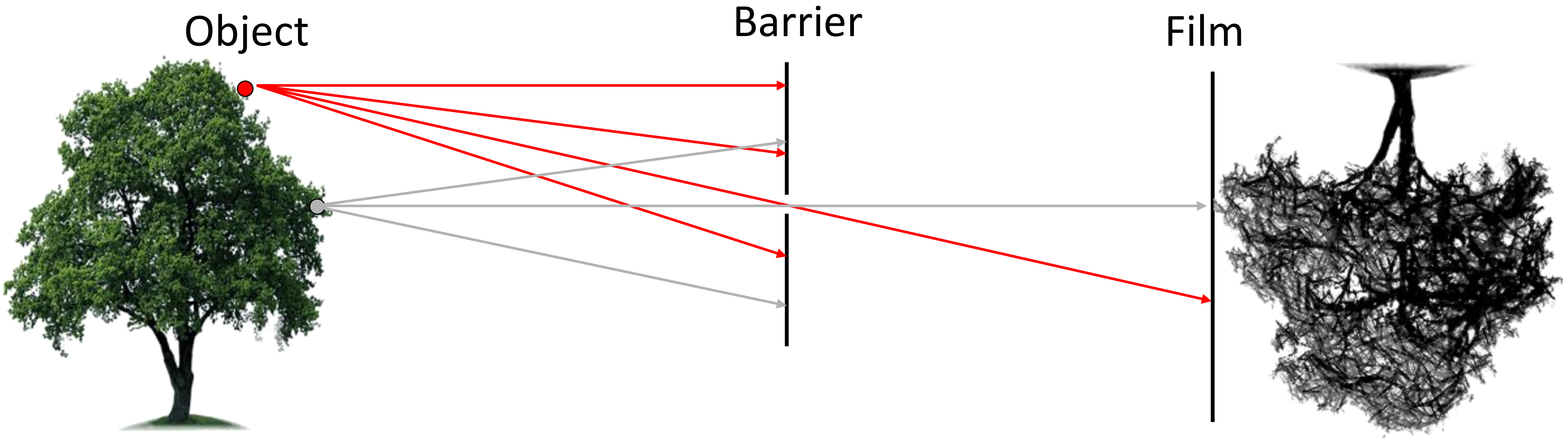
IMAGE FORMATION

Let’s design a camera!



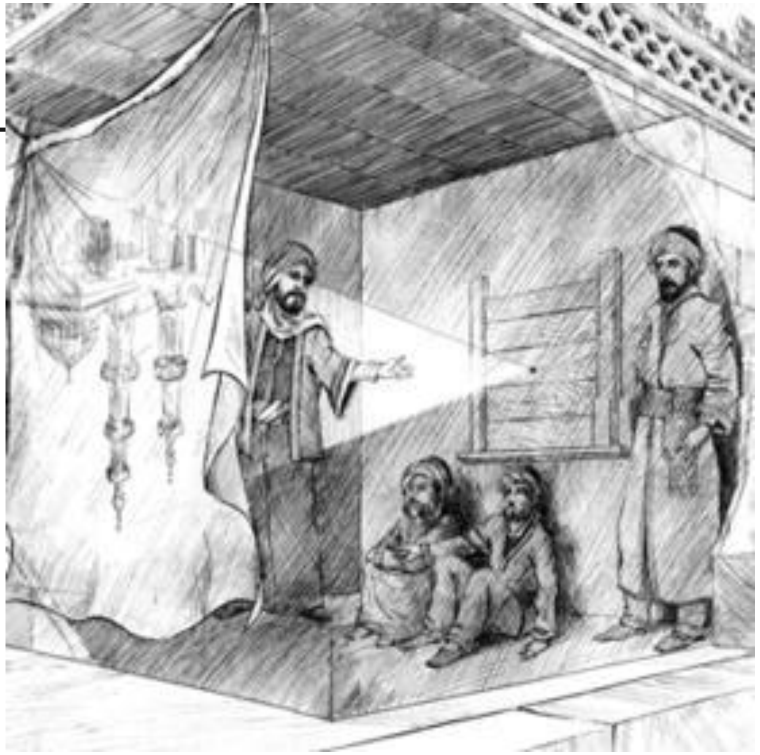
- Idea 1: put an object in front of a film...
- Do we get a good image of the object?

Let’s design a camera!

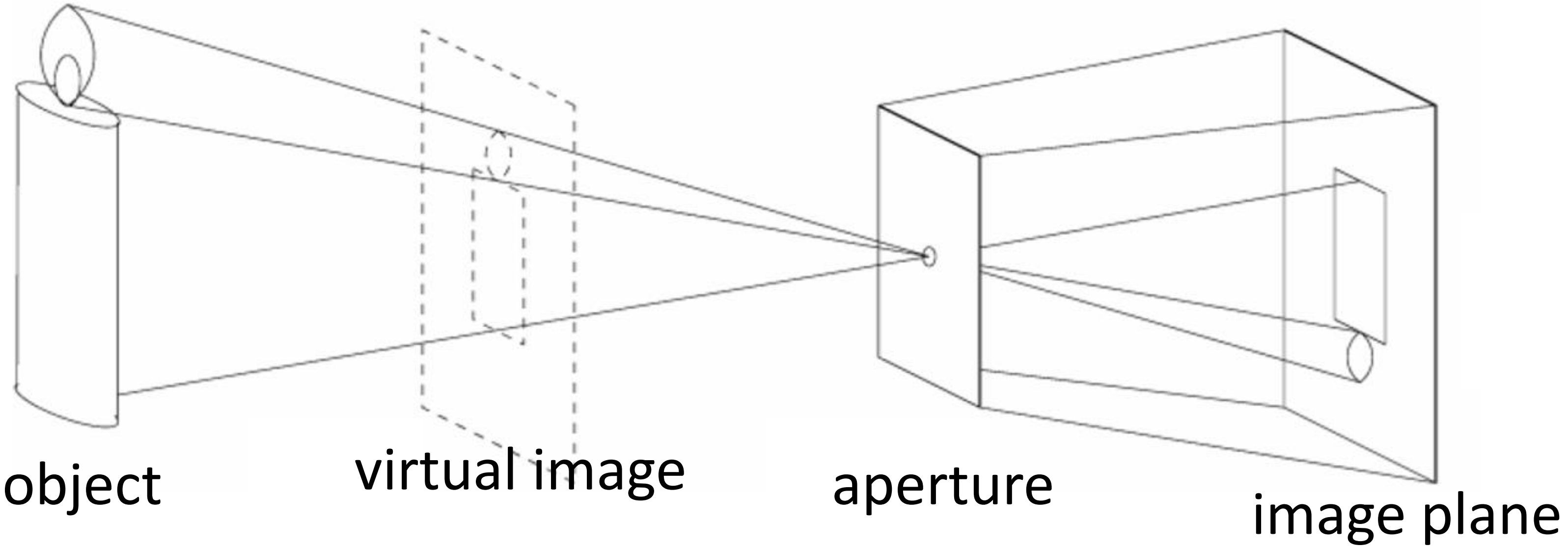


- Add a punctured barrier that blocks most of the rays
  - Significantly reduces blurring
  - The „hole“ is known as aperture

A pinhole camera

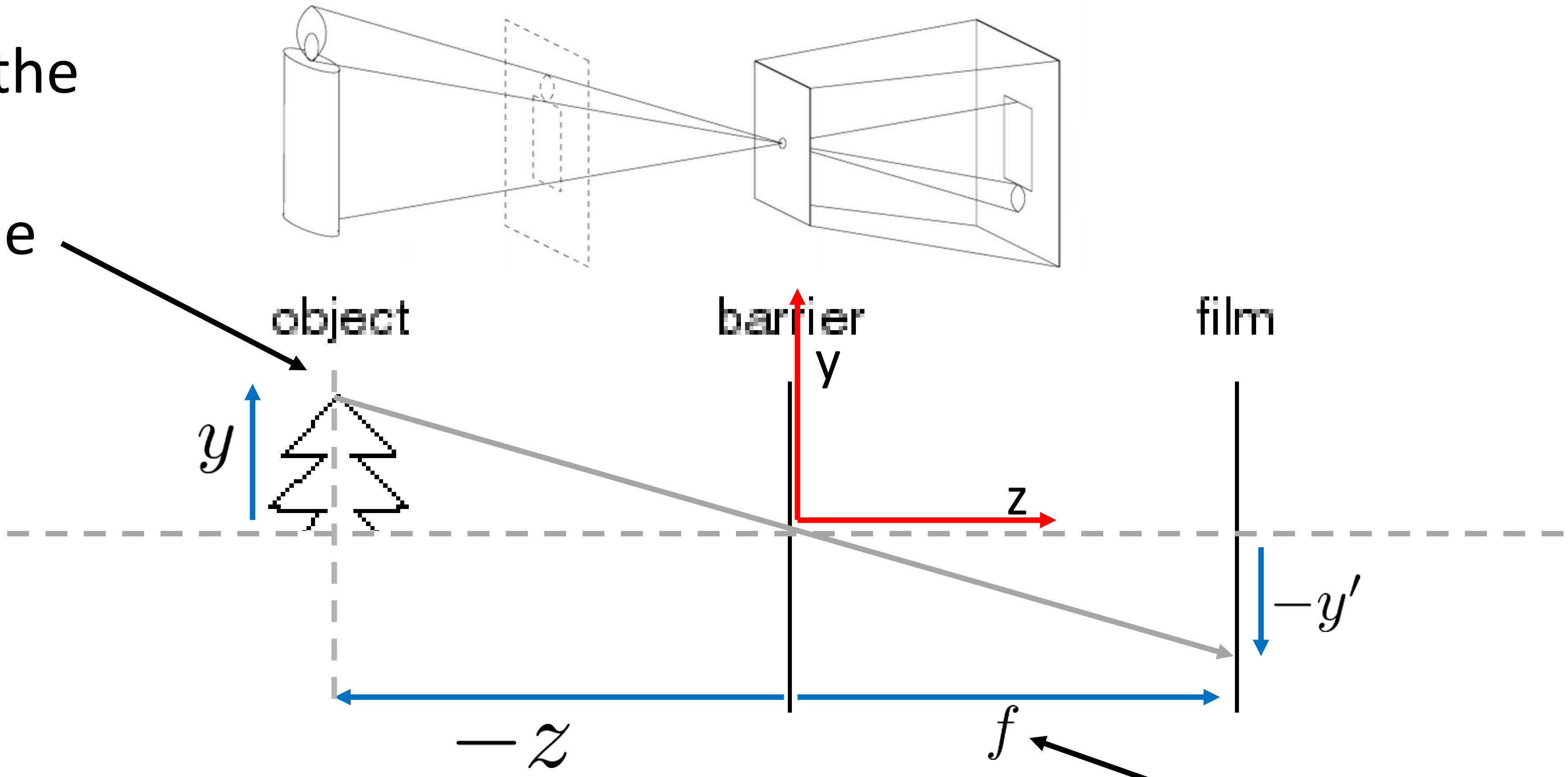


- Earliest and remarkably correct written description: ~500 BC Mohist canon (ancient Chinese texts)
- A simple standard camera model
  - A box with a small aperture
  - Works in practice



A pinhole camera: a taste of geometry

How large will the image of this object be on the film?



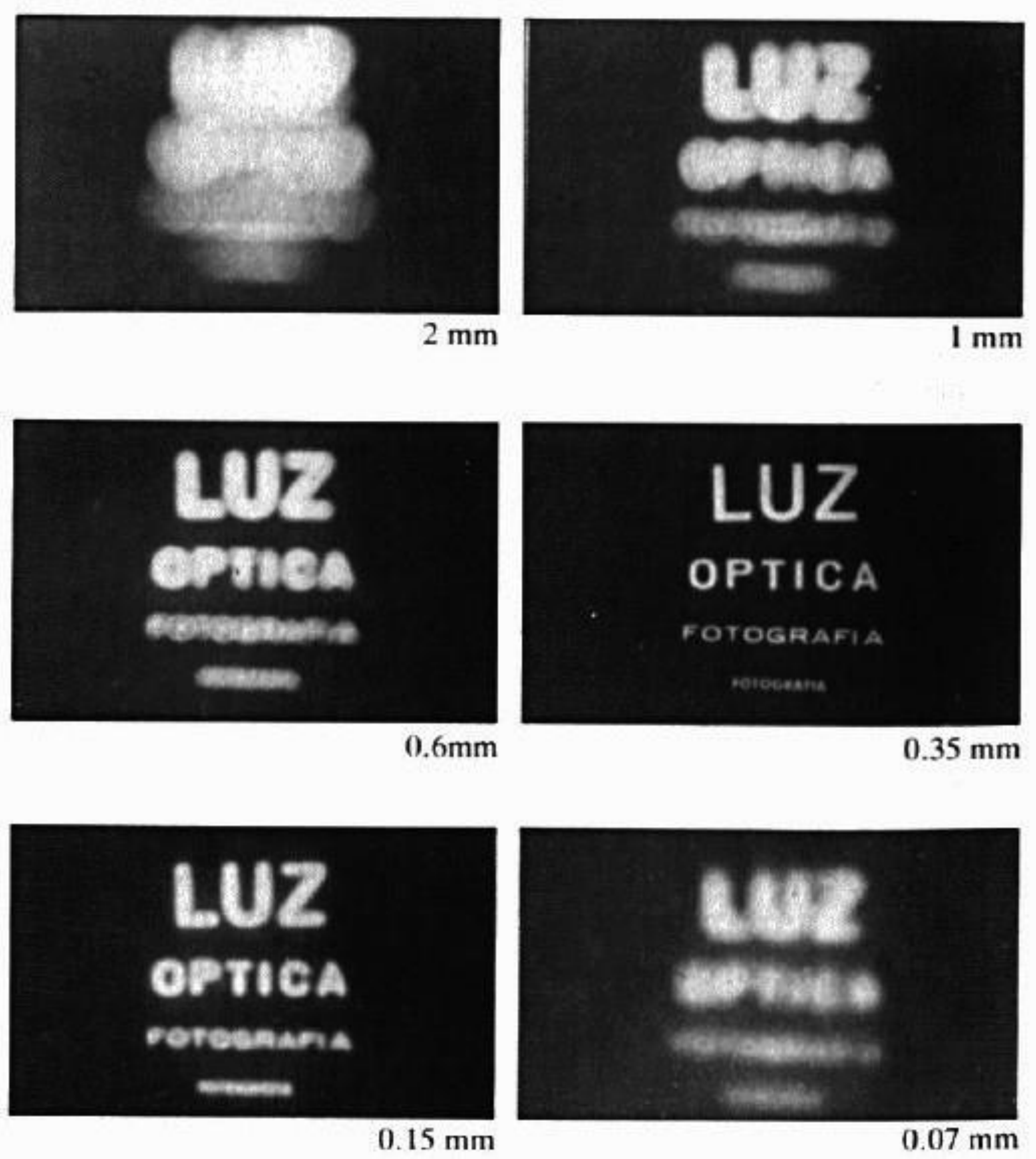
Goriščna razdalja
(angl., focal length)

Perspective projection equation

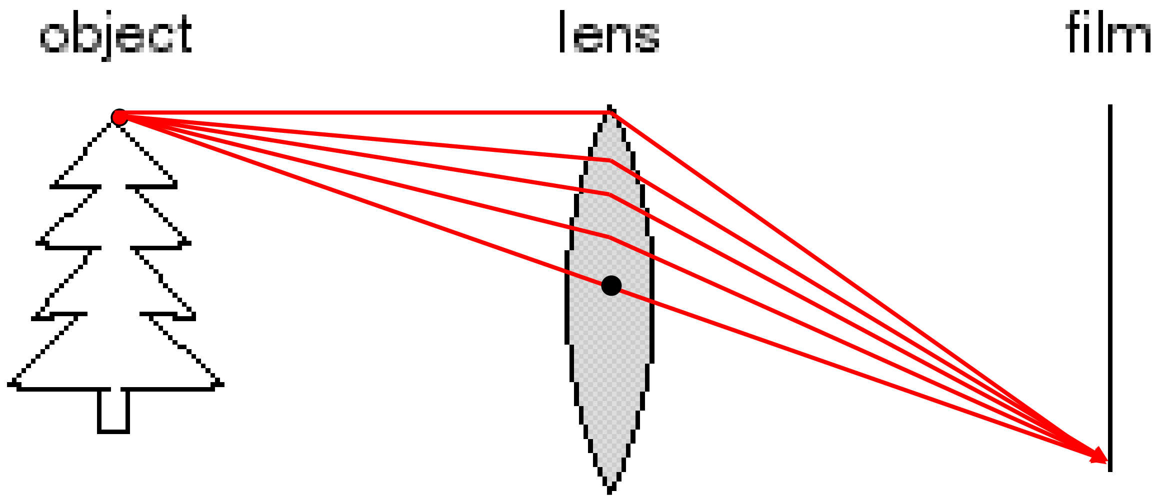
$$\frac{f}{-y'} = \frac{-z}{y}$$

Effects of the aperture size

- Too large – multiple directions averaging, resulting in a blurred image.
- Too small – light starts diffracting, causing blurred image.
- In general small number of rays hit the film, which results in a dark image.
- How do we deal with this?

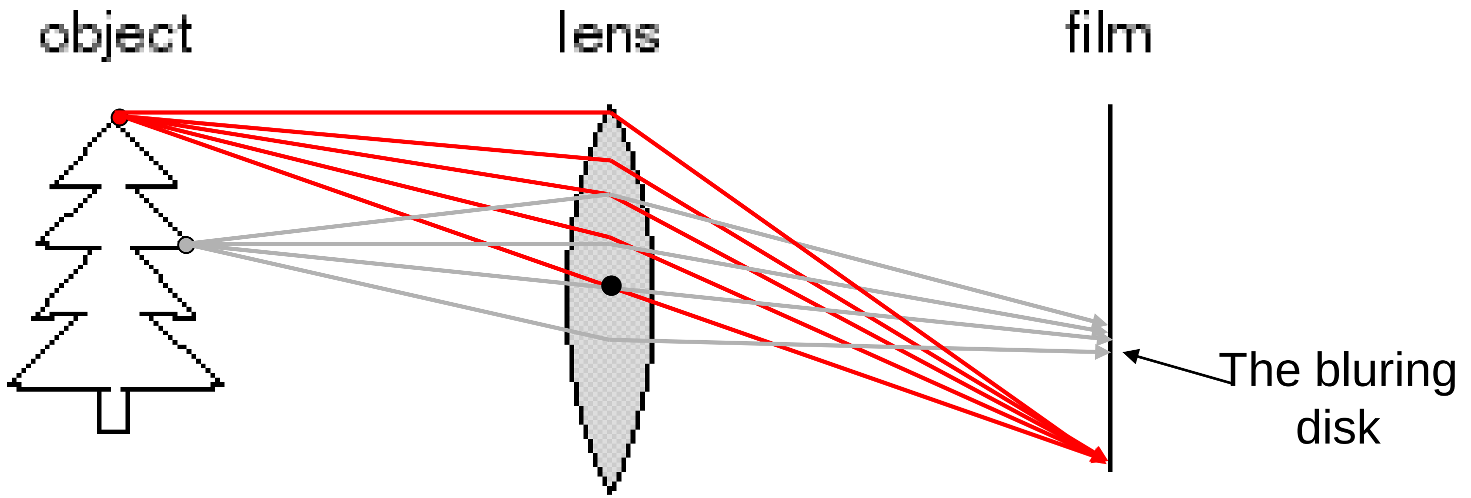


Let's add a lens…



- The lens focuses light to film
  - The rays that travel through the center do not refract.

Let's add a lens...

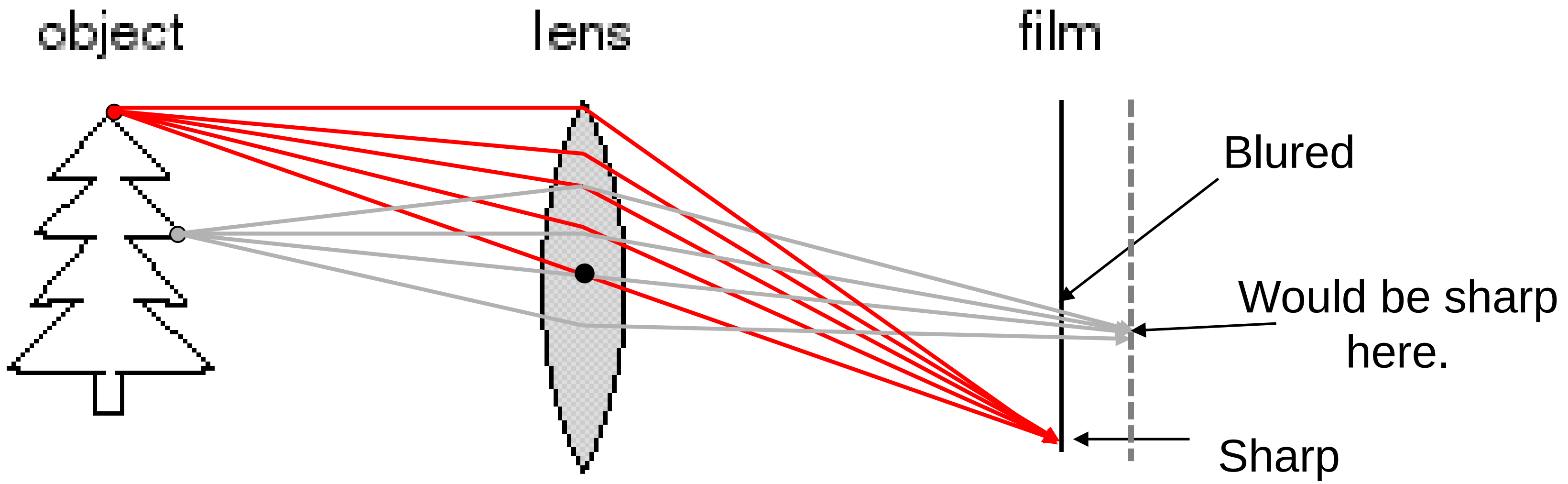


- The lens focuses light to film
  - The rays that travel through the center do not refract.
  - Points at particular distance remain in-focus.
  - Points at other distances are blurred.

Focus and the depth-of-field

- Thin lens: Points at different depths get focused at different depths of image plane.

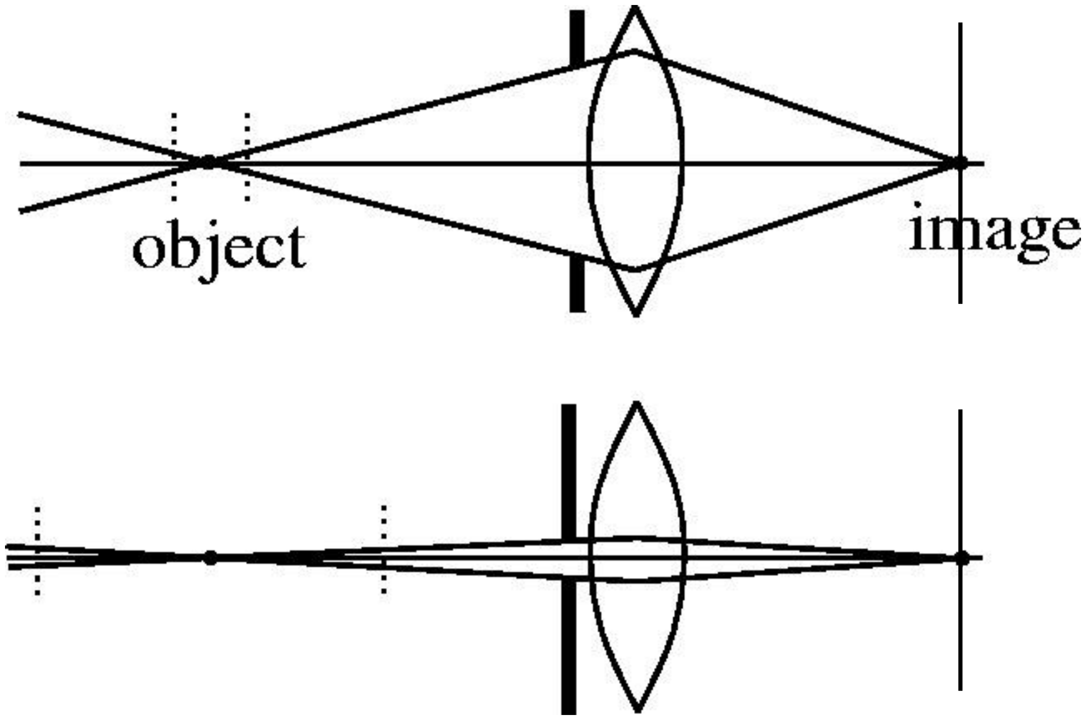
  (Real-world lens have a greater depth of field)



- Depth of field: distance between image planes at which the blurring effect is sufficiently small..

Focus and the depth-of-field

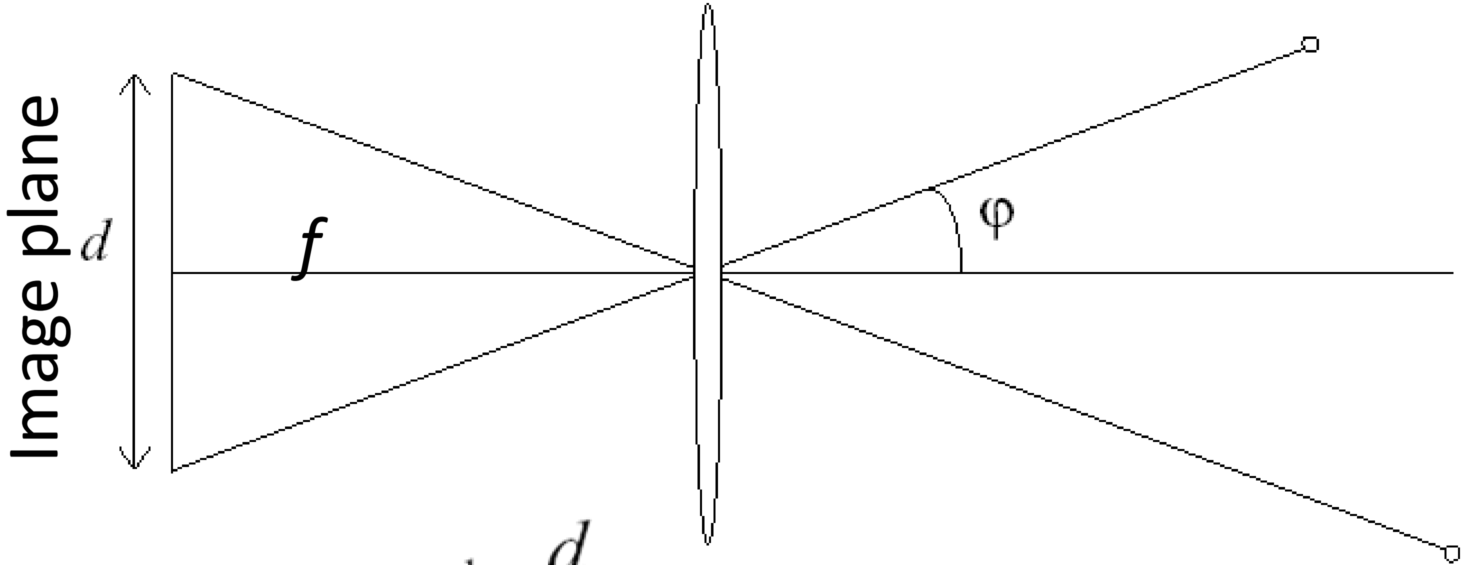
- Effects of aperture on the depth-of-field



- Small aperture increases the depth-of-field.
- But due to reduced illumination we have to increase the exposure time.

Field of view

- Field of view (FOV) $(2 \times \varphi)$ is an angular measure of space perceived by the camera.



$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

- Larger focal length → Smaller field of view

Field of view

- Small *f* results in wide-angle image (Large field of view) →
  - More 3D points project to the sensor.
- Large *f* results in a telescopic image (small FOV) →
  - Smaller portion of 3D scene is projected to the sensor.


28 mm lens, 65.5° × 46.4°

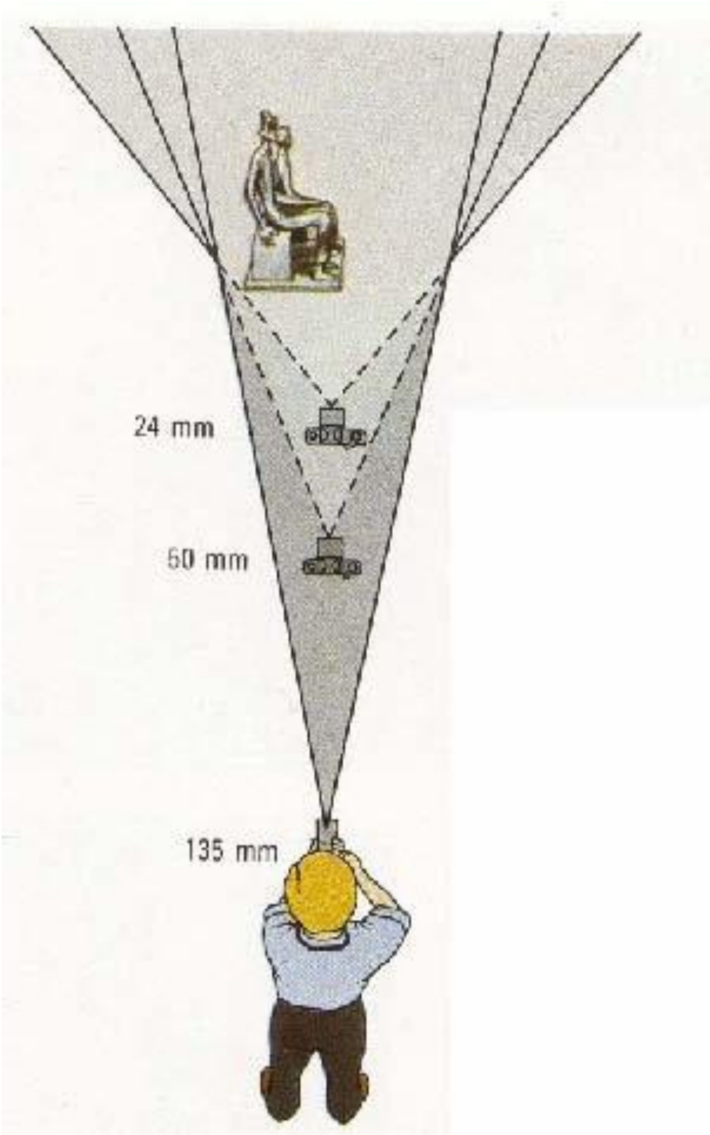

50 mm lens, 39.6° × 27.0°


70 mm lens, 28.9° × 19.5°


210 mm lens, 9.8° × 6.5°

$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

Field of view and focal length

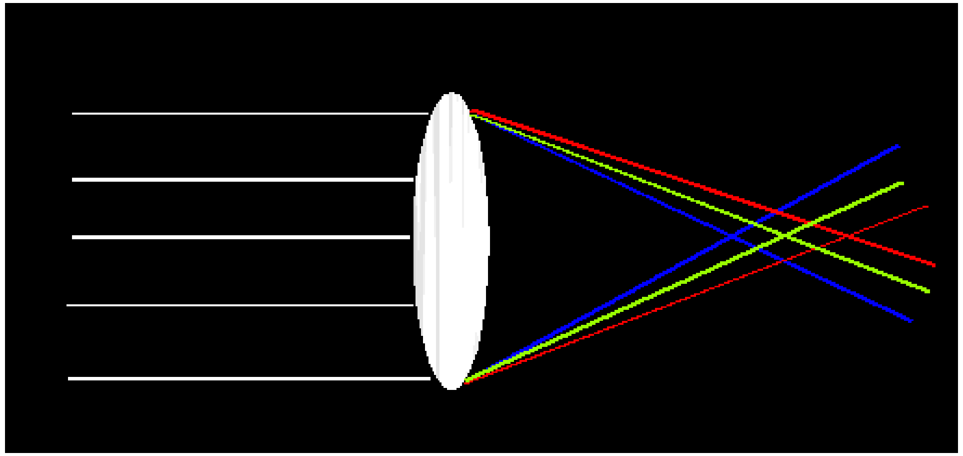


Large FOV, small *f*
Camera close to the car

Small FOV, large *f*
Camera far away from the car

Chromatic aberration

- Different wave-lengths refract at different angle and focus at slightly different distances:

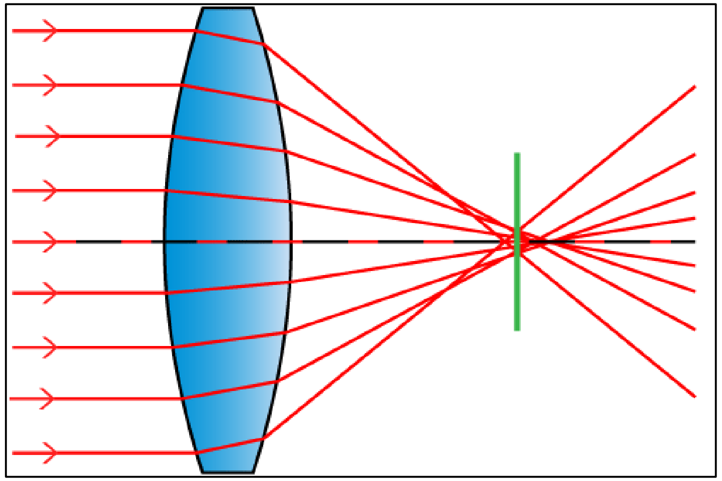


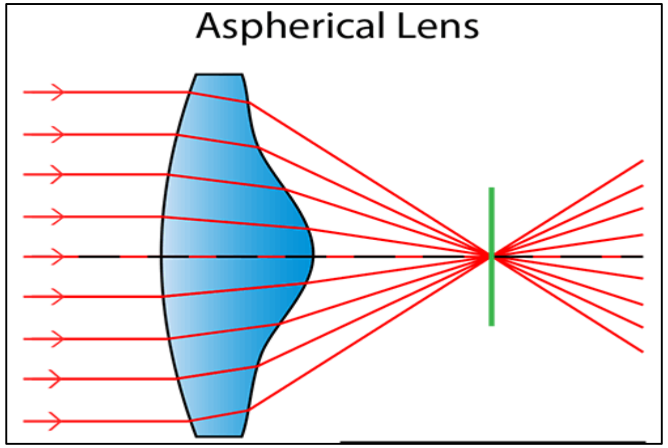
Close to image center

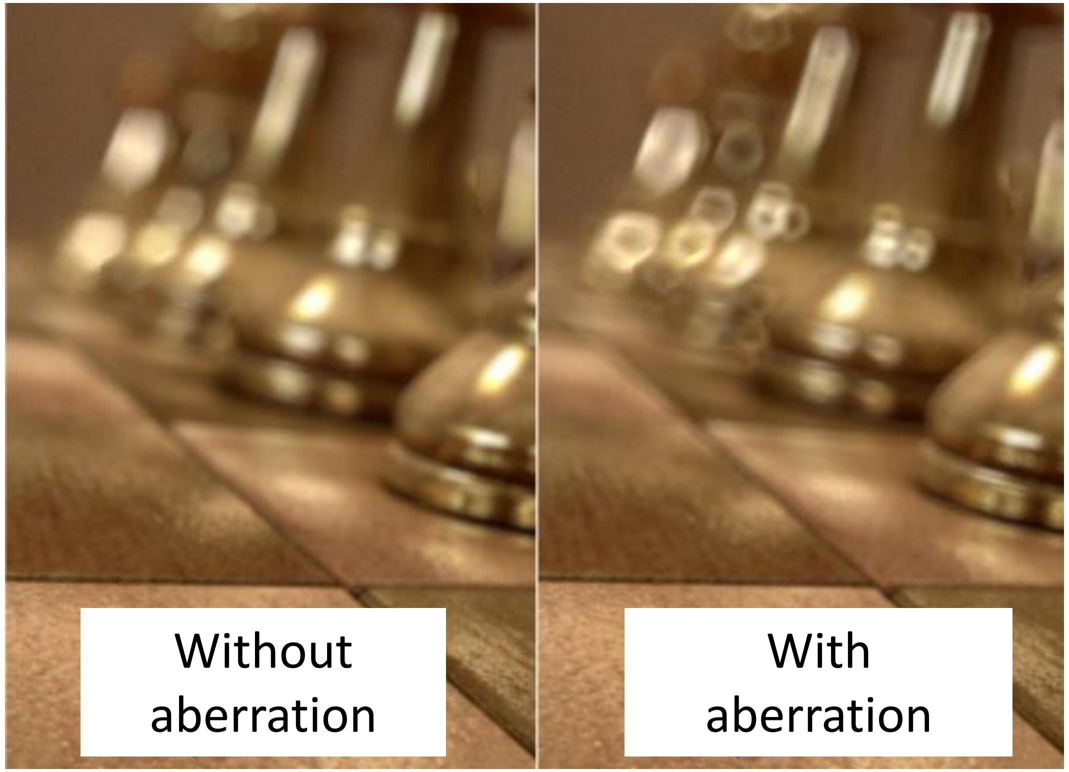
Close to image edge

Spherical aberration

- Spherical lenses do not focus the light perfectly.
- Rays close to lens edge focus closer than those at the center.



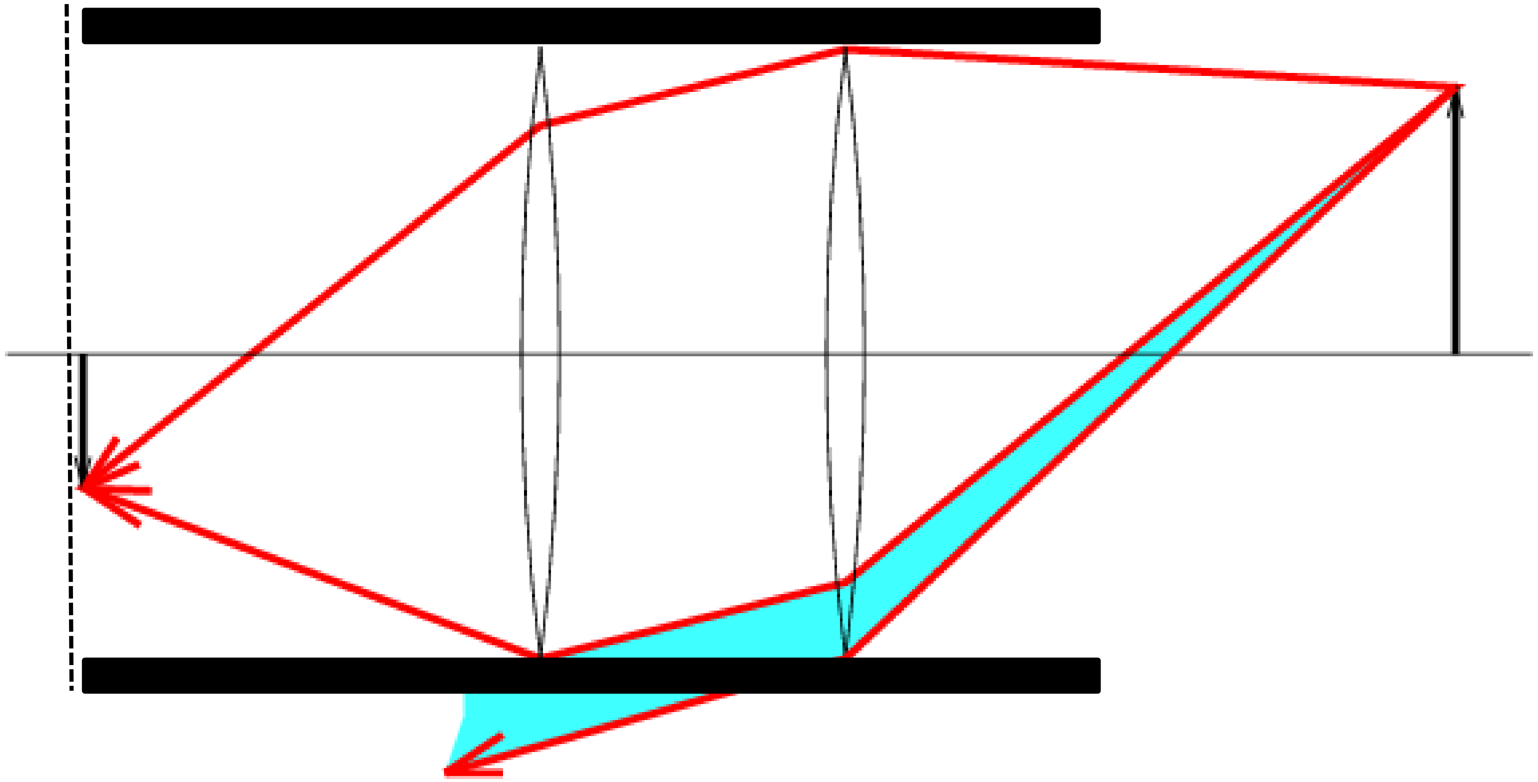




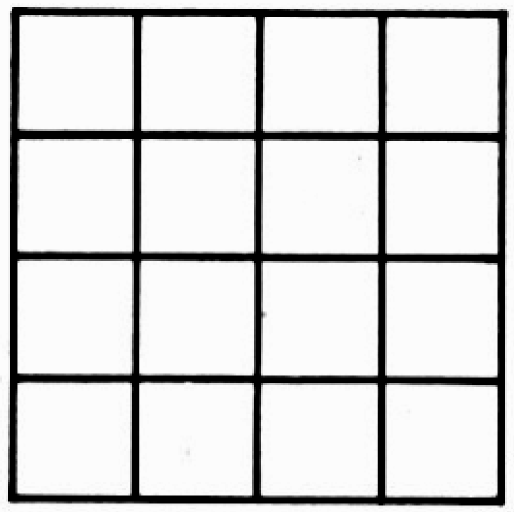
http://photographylife.com/what-is-spherical-aberration

http://www.dofpro.com/sagallery.htm

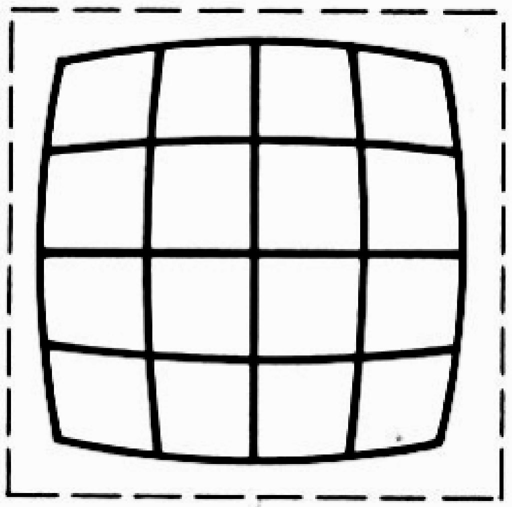
Vignetting



Radial distortion



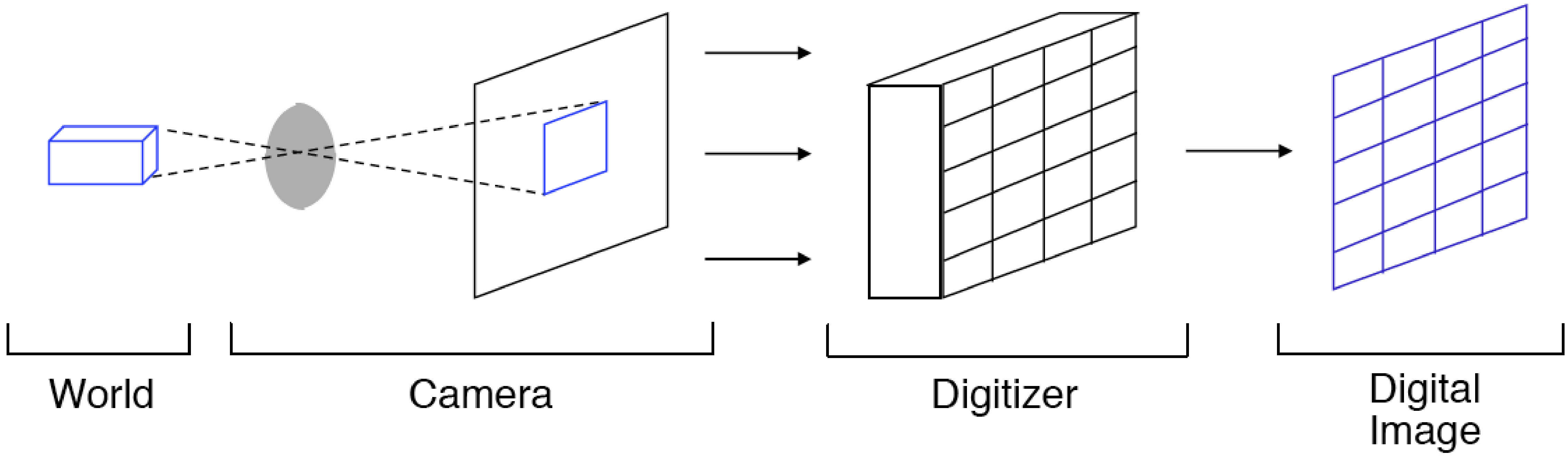
Without distortion



Barrel distortion

- Due to lens imperfections or fisheye.
- Most apparent at the edge of the image.

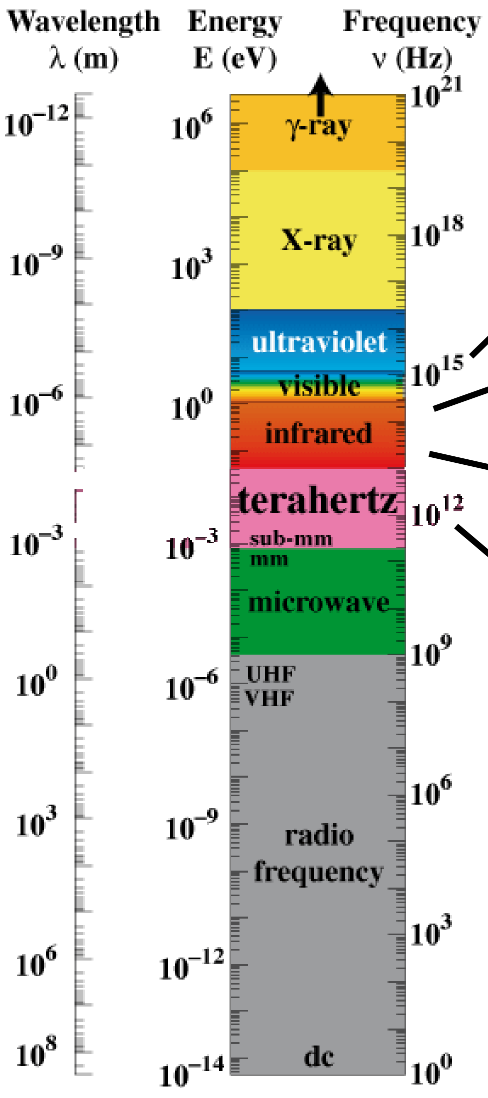
Digital image

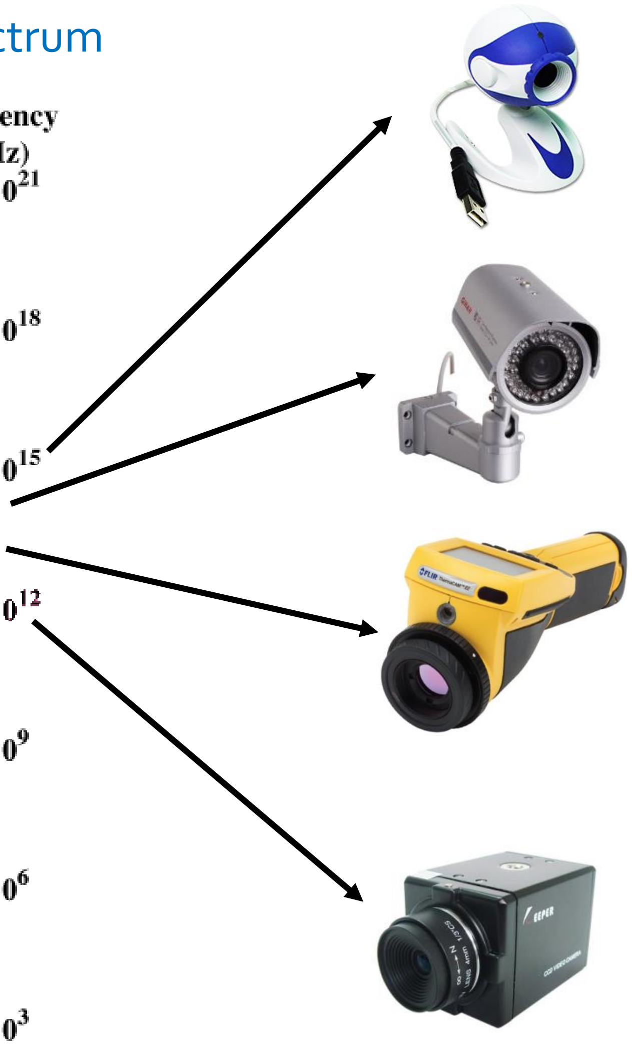


- Instead of film, use matrix (array) of sensors.
- *Discretize* image into pixels.
- *Quantize* light into intensity levels.

Sensor: Camera

Electromagnetic spectrum

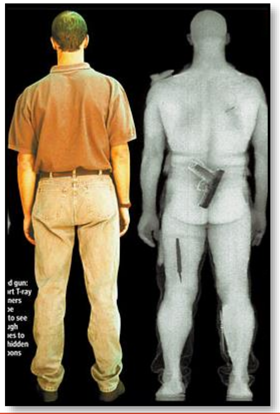




Visible light

Near-infrared light

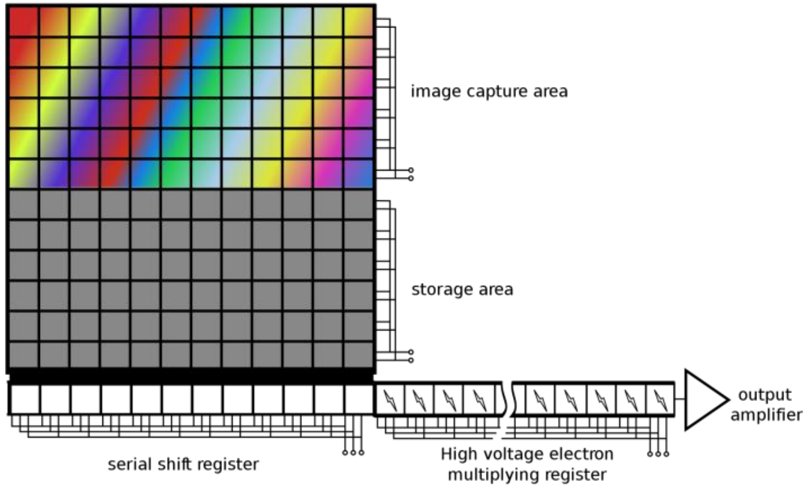
Far-infrared light



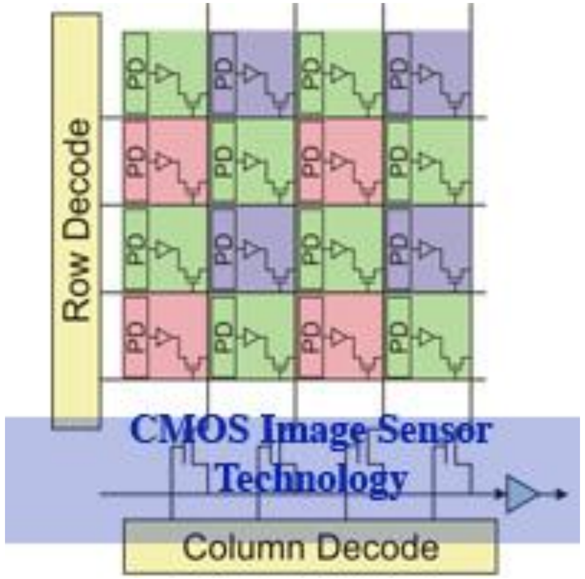
Terahertz light

Visible light cams: CCD vs CMOS

Charge coupled device (CCD)



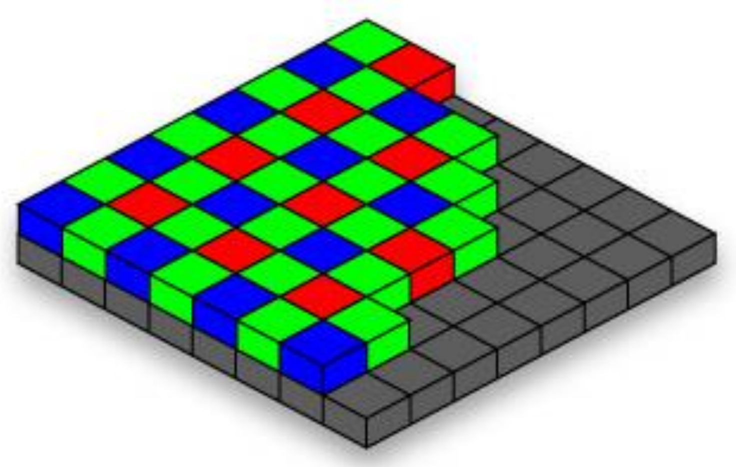
Complementary metal–oxide–semiconductor (CMOS)

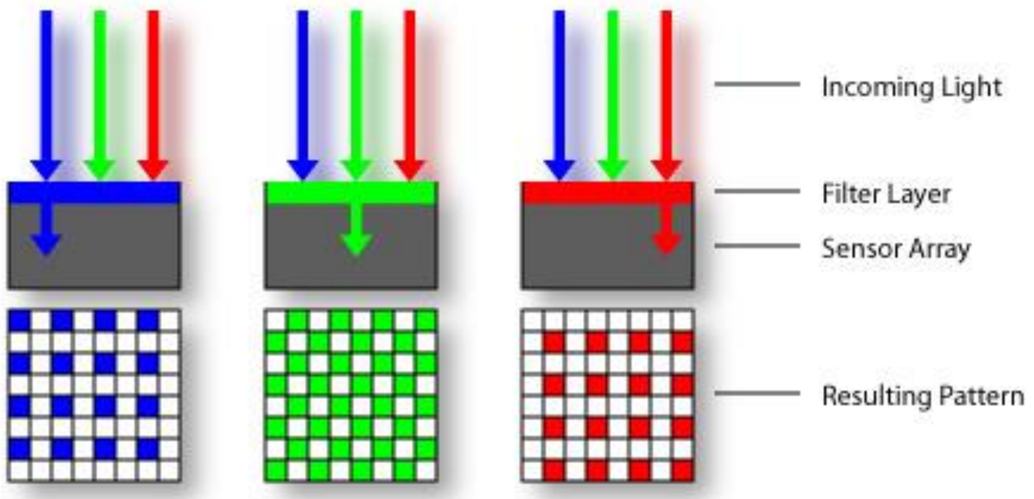


- In both: Photons cause charge on each sensor „cell“.
- CCD reads out the charge (FIFO) serially and digitizes.
- CMOS performs digitization on each cell separately.
- CCD used to deliver better images, but CMOS technology has progressed.
- CMOS is cheaper to produce and is thus wide-spread.

Color perception in digital cameras

Bayer sensor





In classical design, we cannot read out R, G and B channel at a single pixel.

Why twice as many greens compared to blue and red?

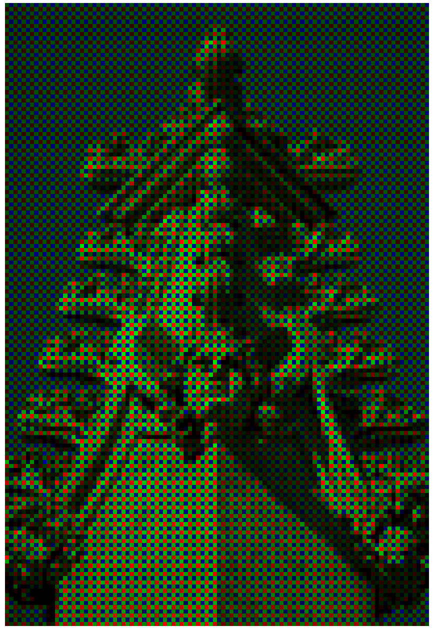
Luminance is mostly determined by the green values.

Human visual system much more sensitive to changes in intensity than in chroma (color).

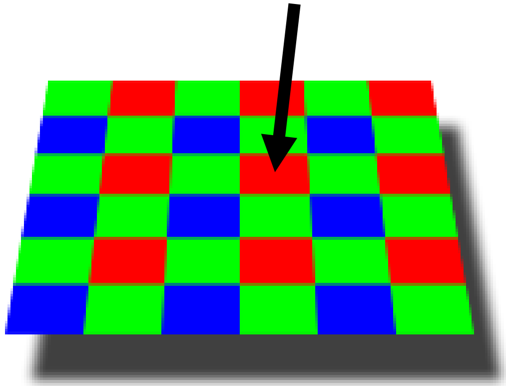
Color perception in digital cameras

What you see

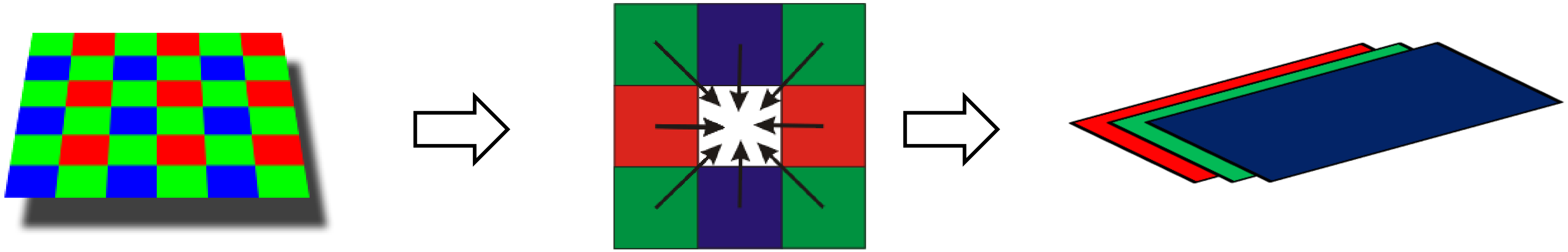
Your camera sees



Missing green!



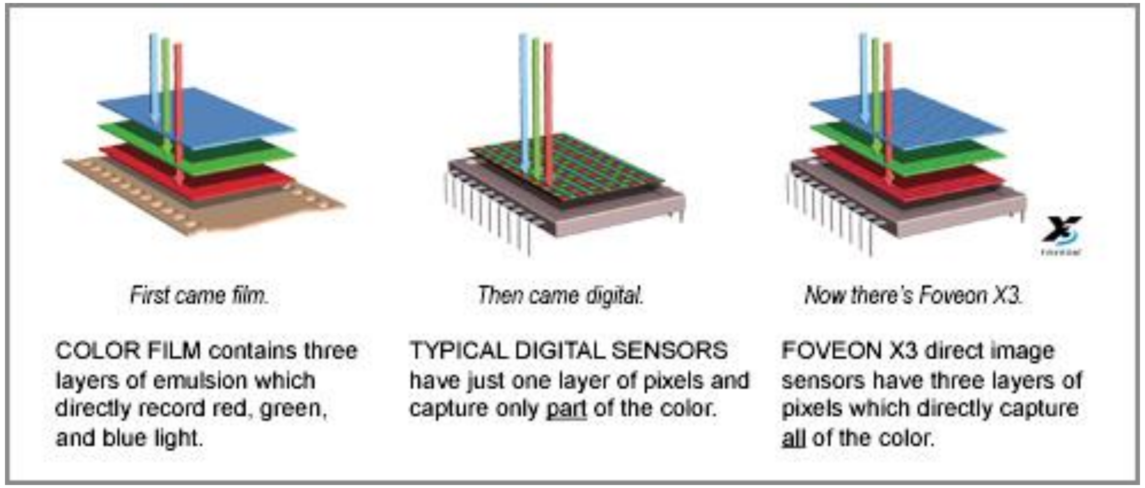
De-mosaicking: The missing color channels at a pixel need to be interpolated!

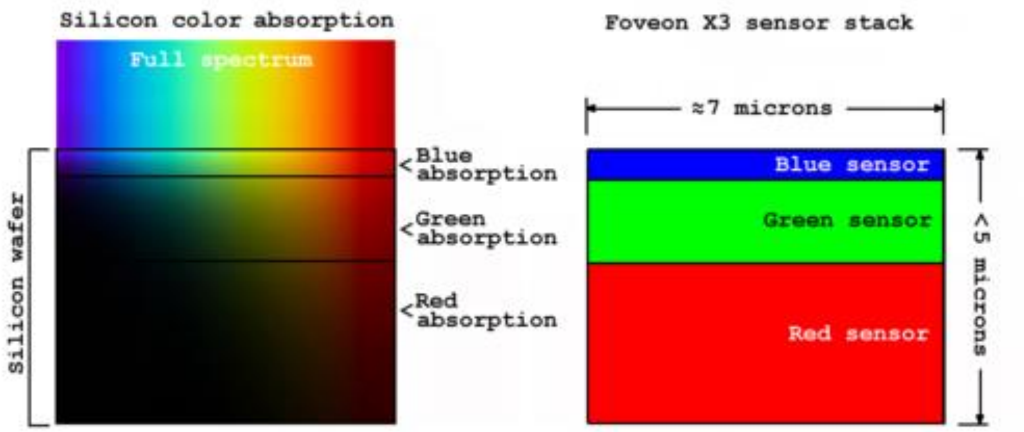


http://www.cambridgeincolour.com/tutorials/camera-sensors.htm

Color perception : Foveon X3

- CMOS-based sensor.
- Based on the fact, that red, green and blue color penetrate the silicon at different depths.

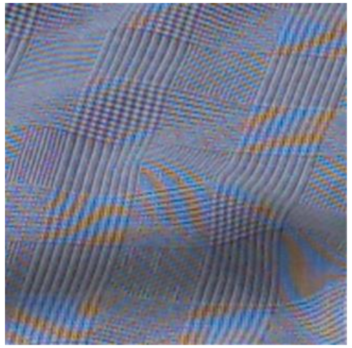
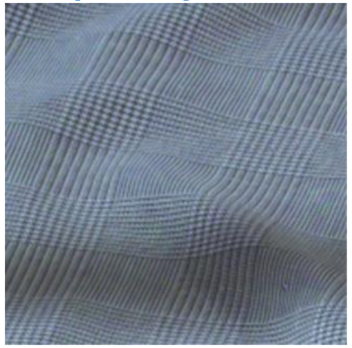




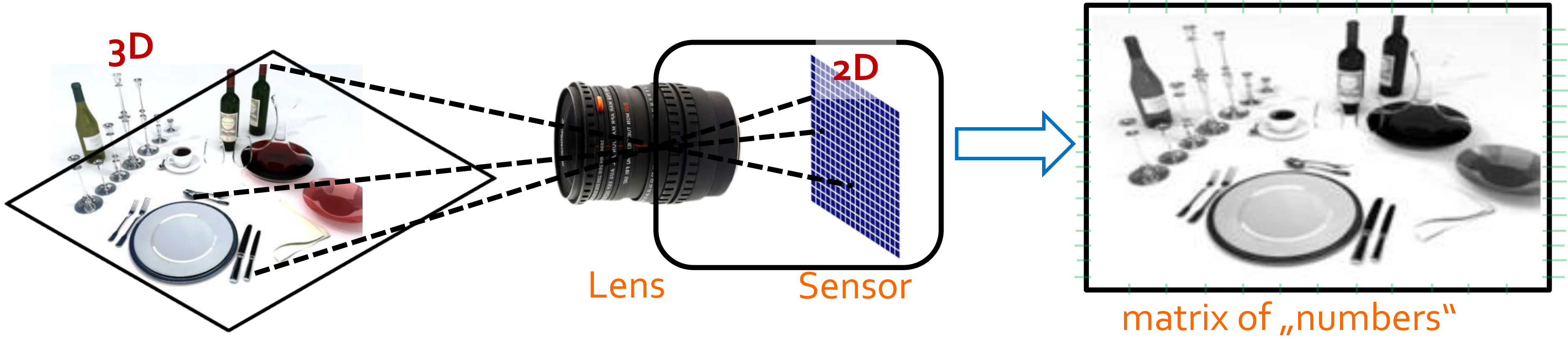
http://www.foveon.com/article.php?a=67

http://en.wikipedia.org/wiki/Foveon_X3_sensor

Better image quality

Bayer-like   Foveon X3

From camera to perception



- How does a human perceive the bottles, plates, forks,..., using only brightness?
- How do we perceive depth?
- Can a computer program do that?

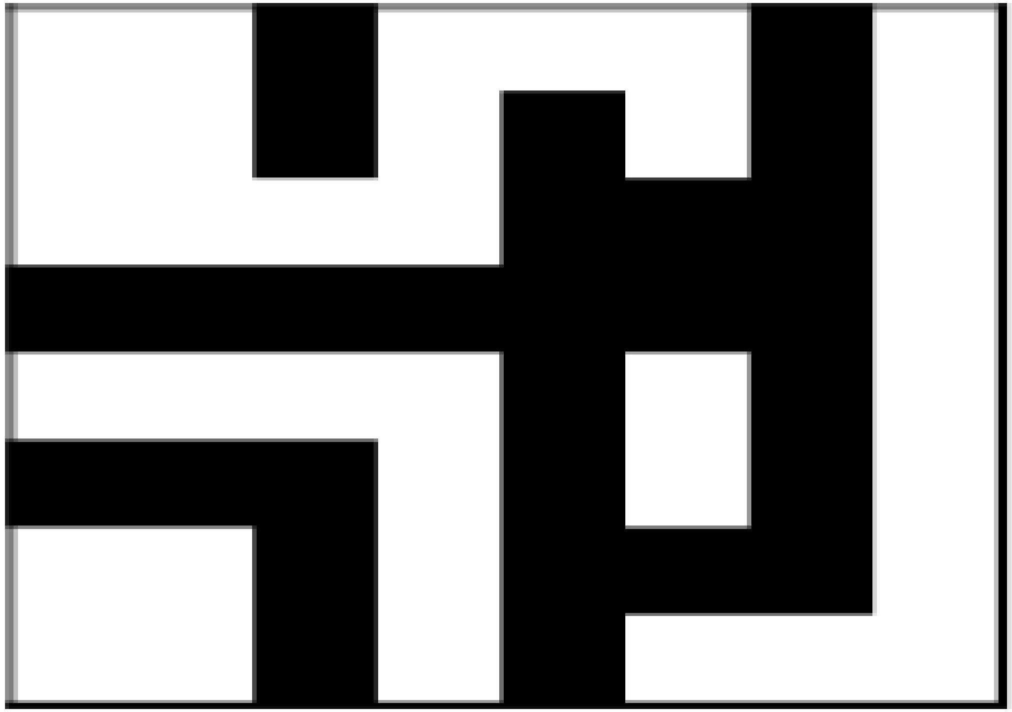
Machine perception

IMAGE PROCESSING 1

Binary images

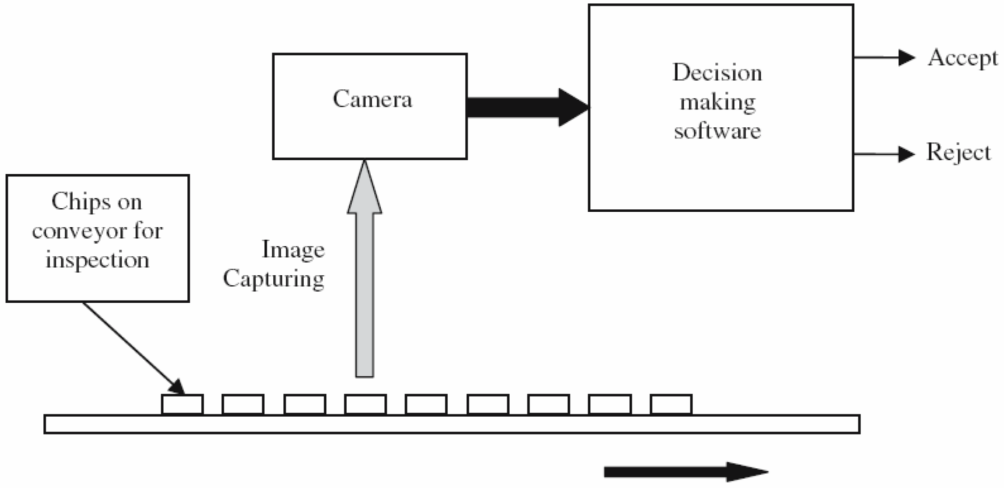
- Only two possible gray levels
- Foreground vs. background

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1



Usage: Machine vision, OCR, etc.

Fig. 3 Schematic diagram of marking inspection setup at Texas Instruments



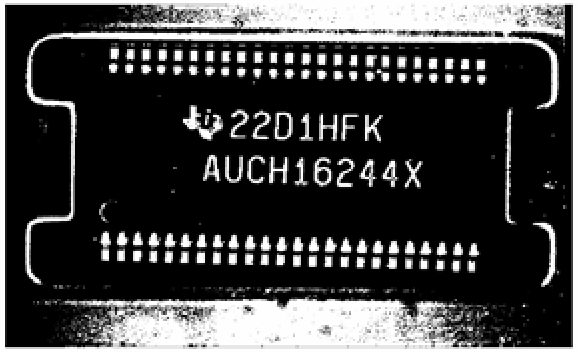
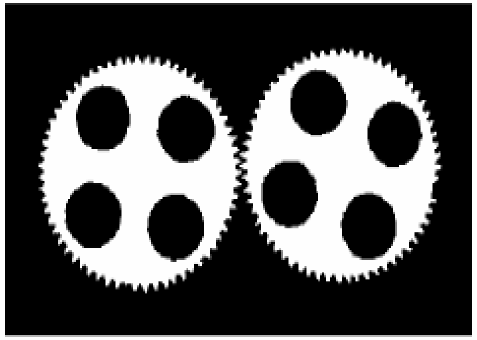
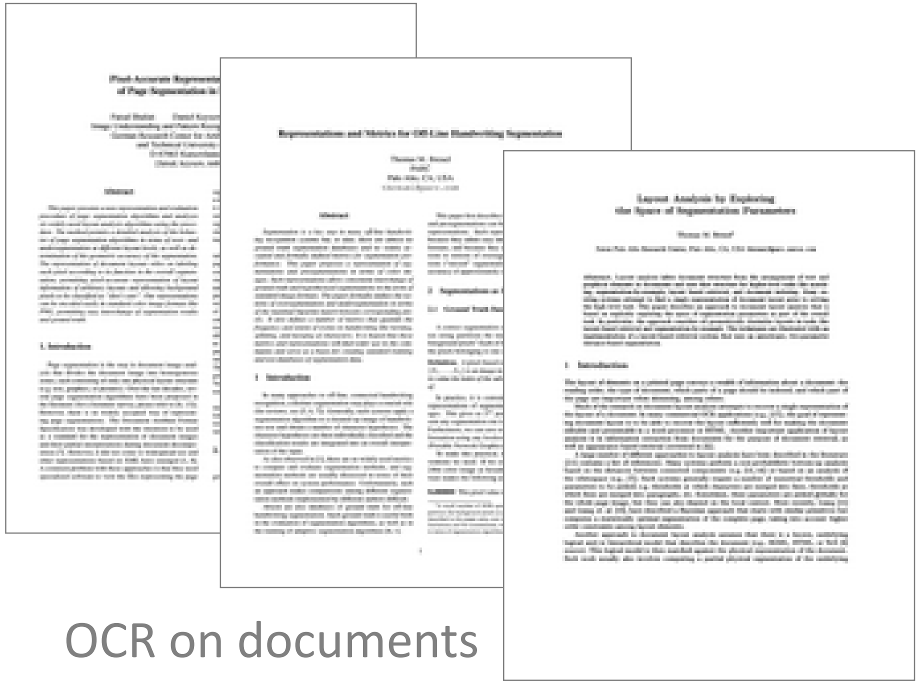


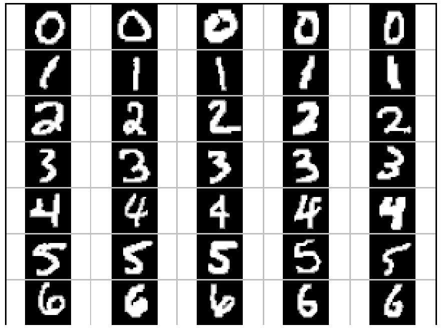
Fig. 7 Binarized image





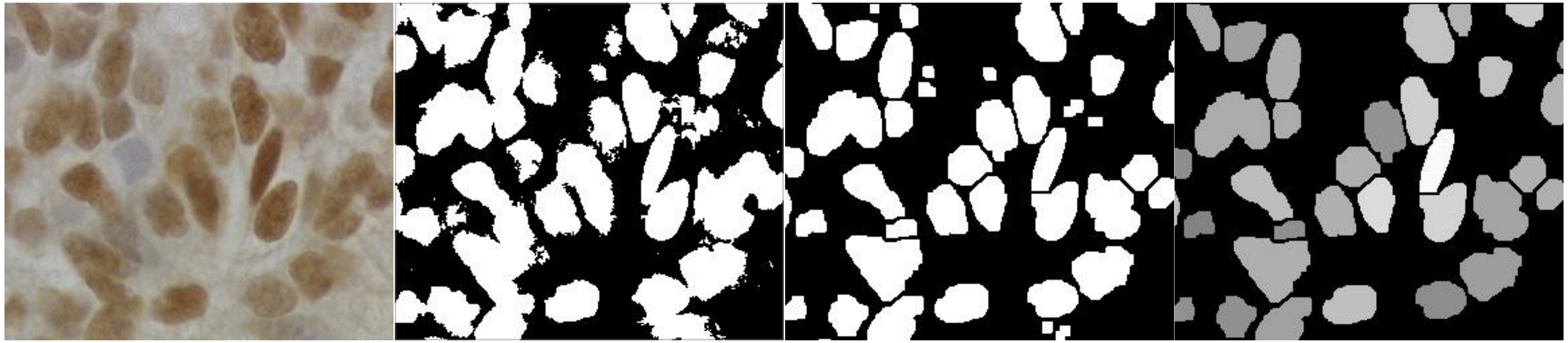
OCR on documents

Hand written numbers

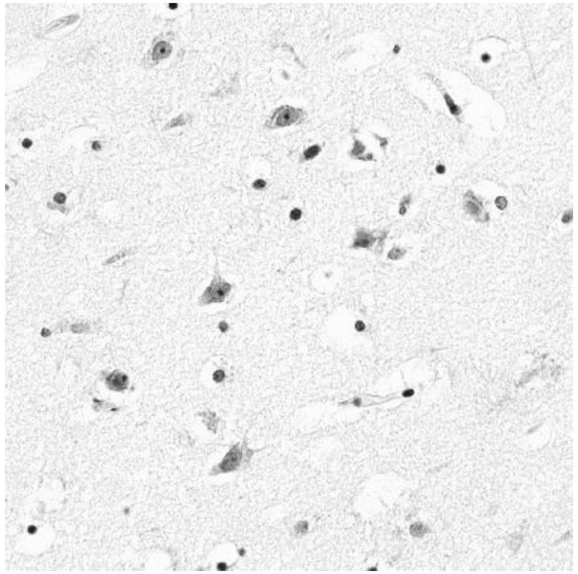


R. Nagarajan et al. “A real time marking inspection scheme for semiconductor industries“, 2006

Usage: Medical imaging

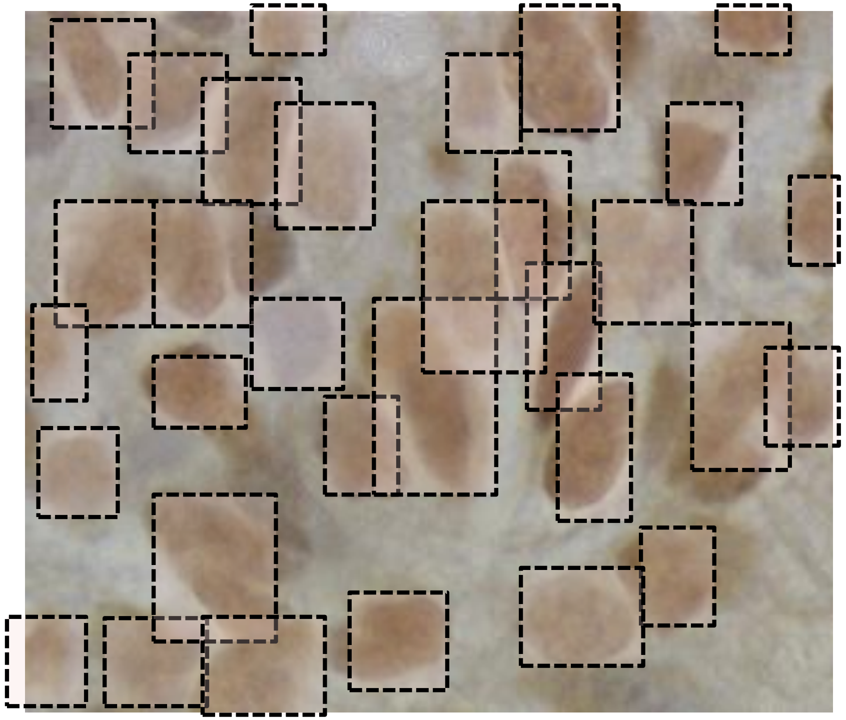


Source: D. Kim et al., Cytometry 35(1), 1999

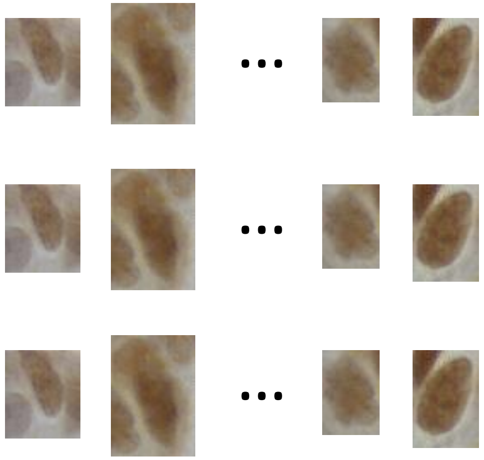


Use case: Count the "round" cells

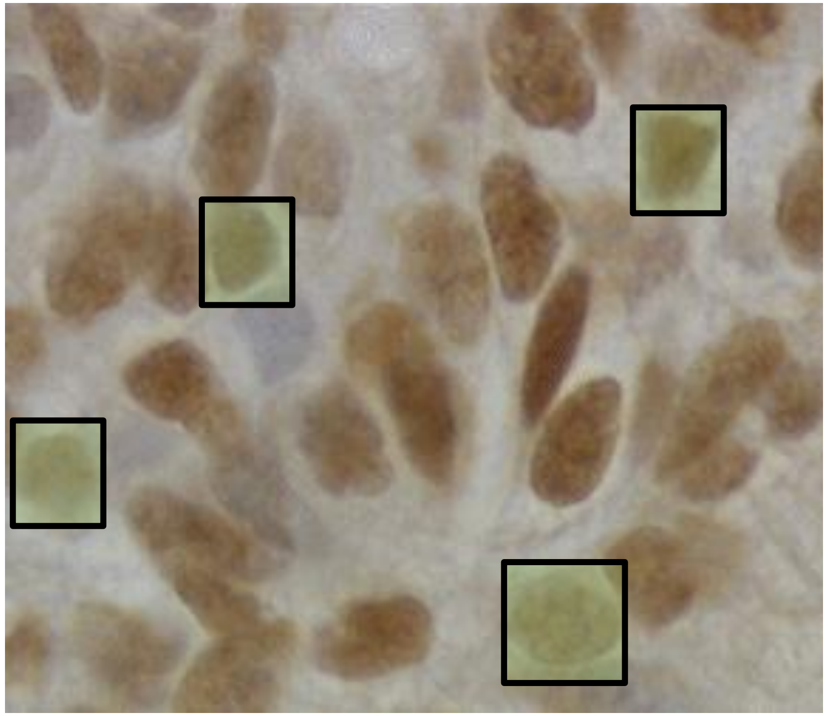
Generate hypotheses



Classify each region into a "round" and "not round"



Keep "round" regions

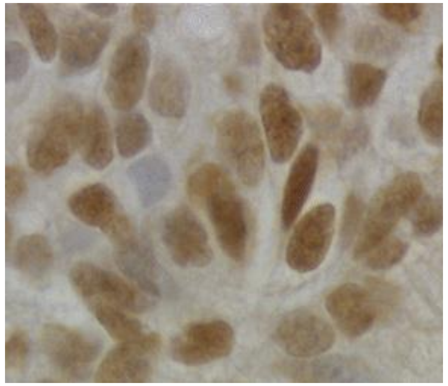


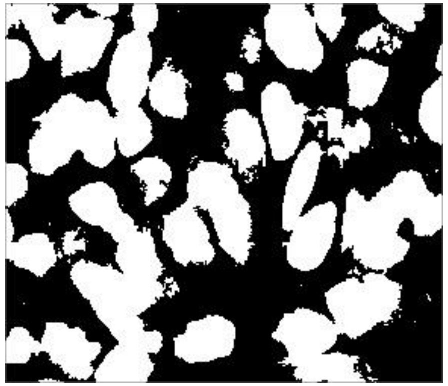
Localize, Describe, Classify

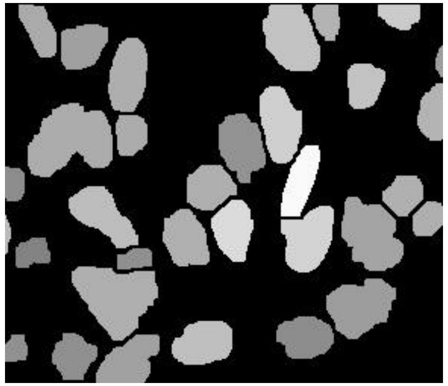
Localize: Sequence of processing steps

- Convert gray image to a binary image
  - Thresholding
- Clean binary image
  - Morphologic filtering
- Extract individual regions
  - Connected components

… then describe each localized region and classify

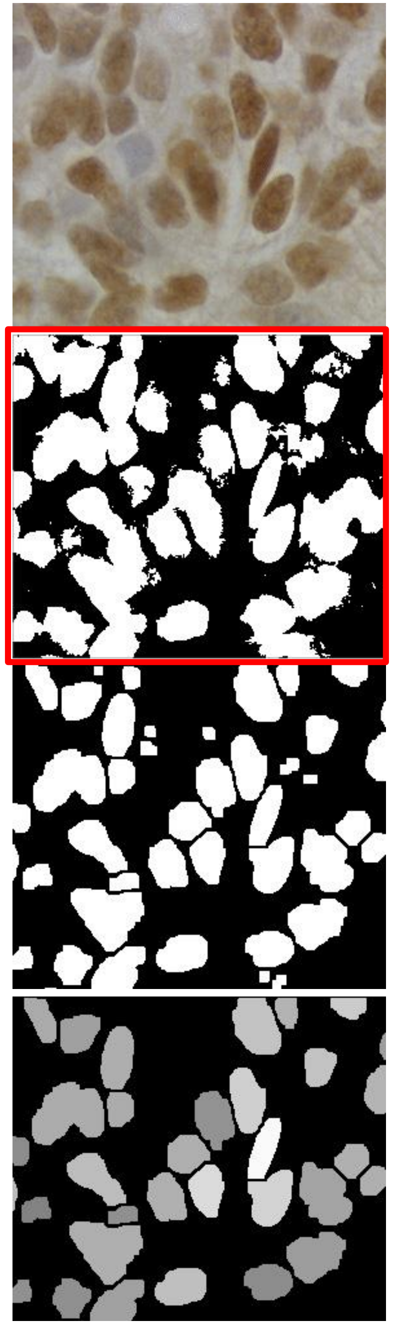






Machine perception

IMAGE THRESHOLDING



Thresholding

- Transform an image into a Binary Mask
- Various approaches
  - Apply a single threshold

$$F_T[i,j] = \begin{cases} 1, \text{if } F[i,j] \leq T \\ 0, \text{otherwise} \end{cases}$$

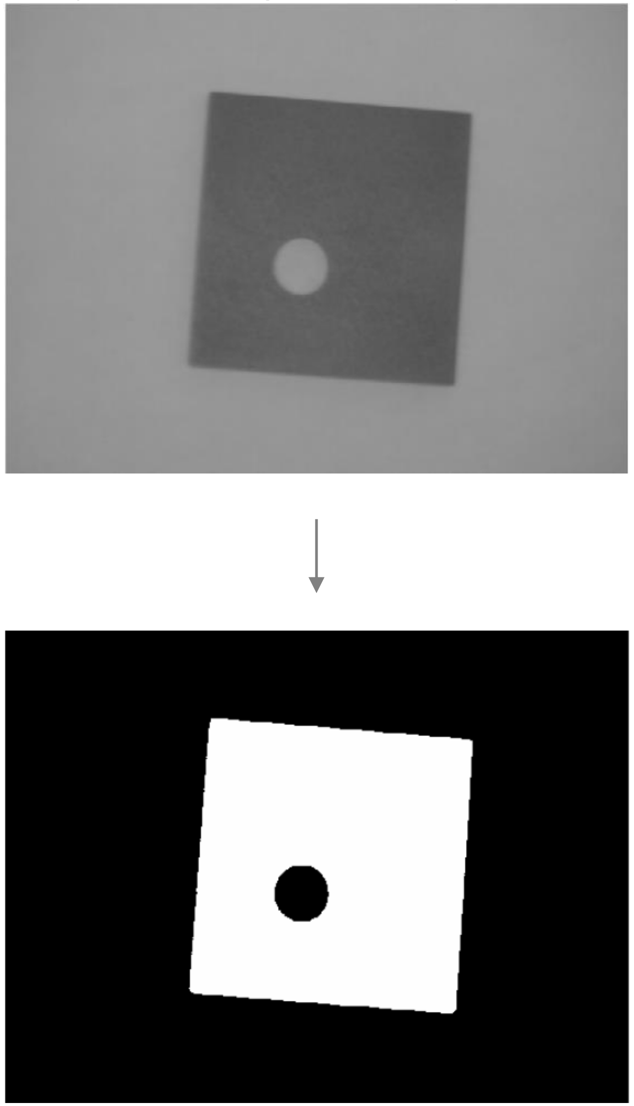
  - Apply two thresholds

$$F_T[i,j] = \begin{cases} 1, & \text{if } T_1 \leq F[i,j] \leq T_2 \\ 0, & \text{otherwise} \end{cases}$$

  - A general view: apply a classifier

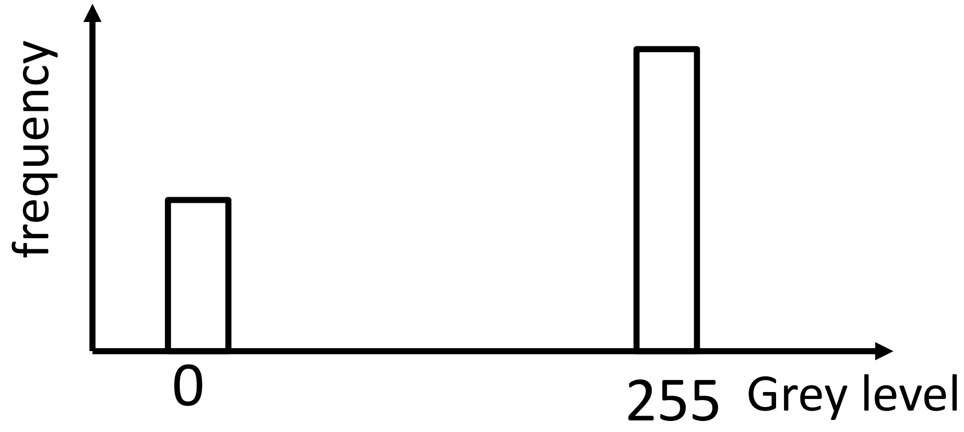
$$F_T[i,j] = \begin{cases} 1, & \text{if } F[i,j] \in Z \\ 0, & \text{otherwise} \end{cases}$$

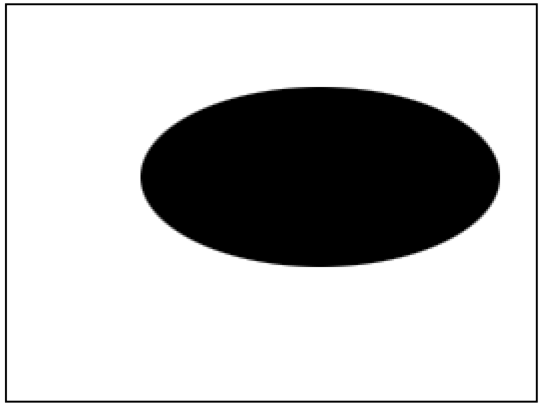
Object/background separation



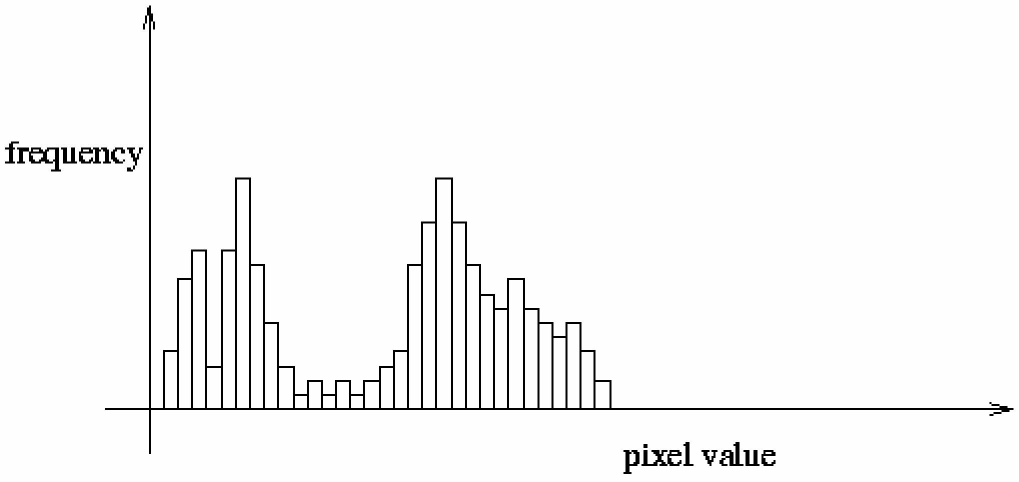
Source: Bastian Leibe

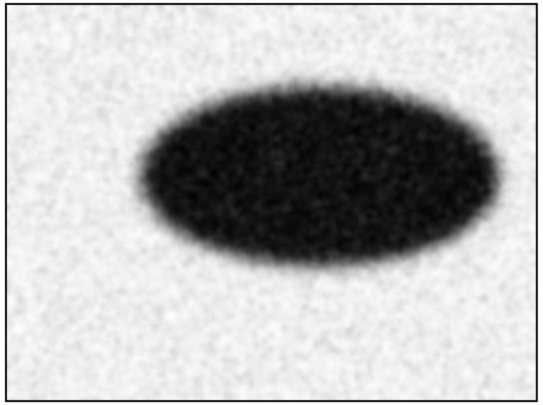
A simple example: Bimodal histogram





Ideal case:
bright object on
dark background.

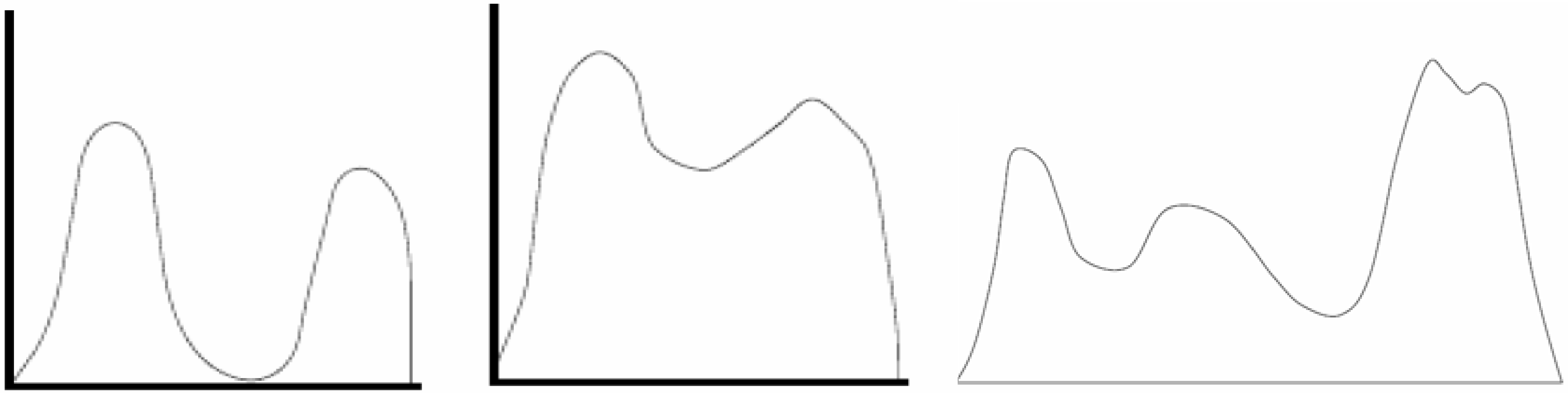




A more realistic noisy image.

A not so simple example...

- What to do here?



Separate modes | Overlapping modes | Multiple modes

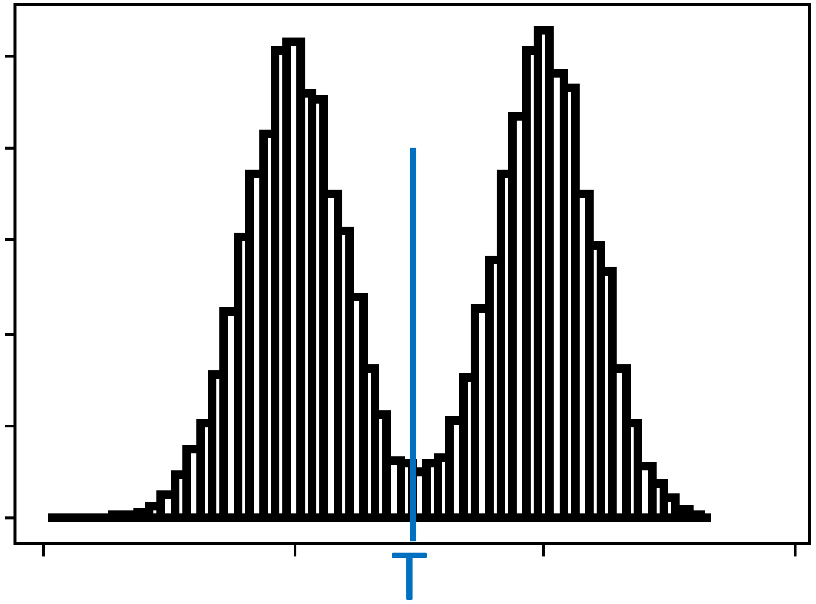
- Generally thresholding is a difficult problem
  - Domain knowledge helps a great deal.
  - E.g., the portion on letters on a page.
  - E.g., size of the structure we want to detect...

Global binarization [Otsu '79]

- Find a threshold *T*, that minimizes intensity variances within classes separated by *T*:

$$\sigma^2_{within}(T) = n_1(T)\sigma_1^2(T) + n_2(T)\sigma_2^2(T)$$

$$n_1(T) = \left|\{I_{(x,y)} < T\}\right|, n_2(T) = \left|\{I_{(x,y)} \geq T\}\right|$$



- This equals to maximization of between class variance $\sigma_{between}$:

$$\sigma^2_{between}(T) = \sigma^2 - \sigma^2_{within}(T)$$
$$= n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2$$

Otsu, N (1979), "A threshold selection method from gray-level histograms", IEEE SMC

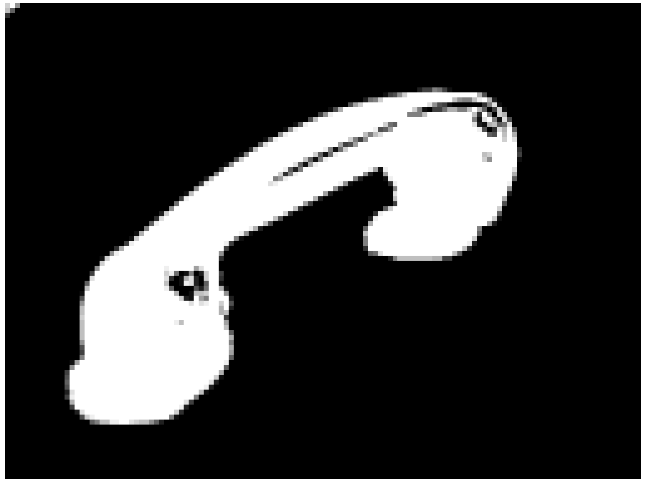
Otsu's Algorithm

For threshold value *T*

1. Separate the pixels into two groups by intensity threshold *T*
2. For each group get an average intensity and calculate $\sigma^2_{between}$.

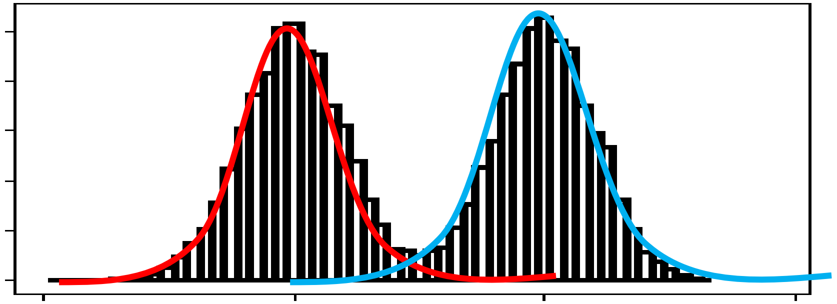
Select the T*, that maximizes the variance:

$$T^* = \arg\max_T [\sigma^2_{between}(T)]$$



Used in several thousand modern algorithms in particular in medical imaging

State-of-the-art: Generalization of Otsu (CVPR2020)



- Recently, Otsu's method revisited:
- Formulate the problem as fitting 2 Gaussians to the histogram with priors on means and variances (Bayesian view)
- Efficiently computed by a single pass through the histogram (like Otsu)
- Outperforms all single-pass algorithms and all deep learning algorithms on the text binarization benchmark

(a) Input Image (b) True Binarization (c) Oracle Binarization

(d) Otsu's Method (e) MET (f) GHT

Barron, J.T., [A Generalization of Otsu's Method and Minimum Error Thresholding](), CVPR2020 ; [link to video]()

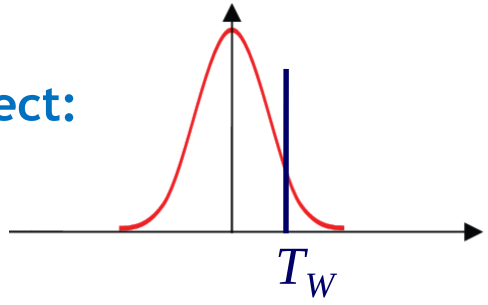
Local binarization [Niblack'86]

- Estimate a local threshold in neighborhood W:

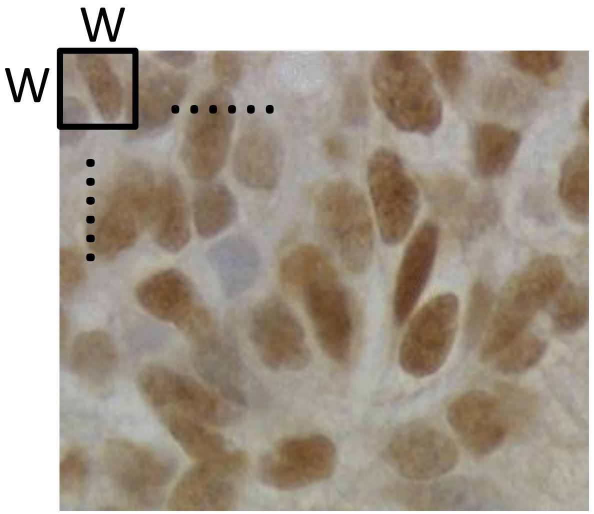
$$T_W = \mu_W + k \cdot \sigma_W$$

with $k \in [-1,1]$ set by user.

Effect:

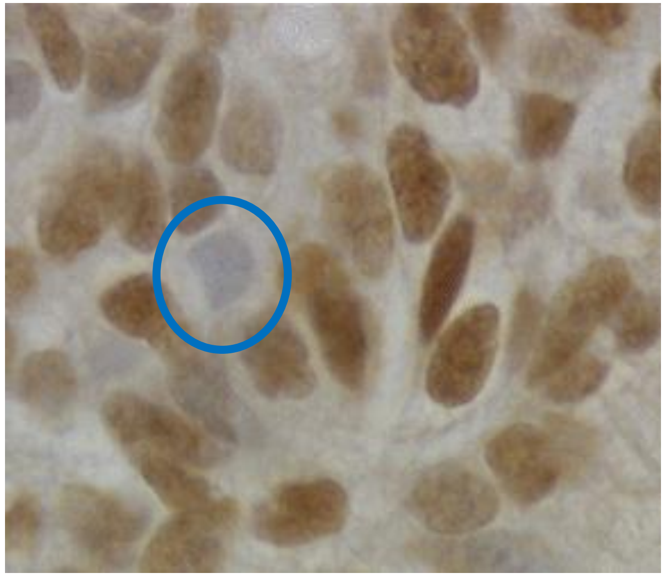


- Calculate the threshold separately for each pixel.

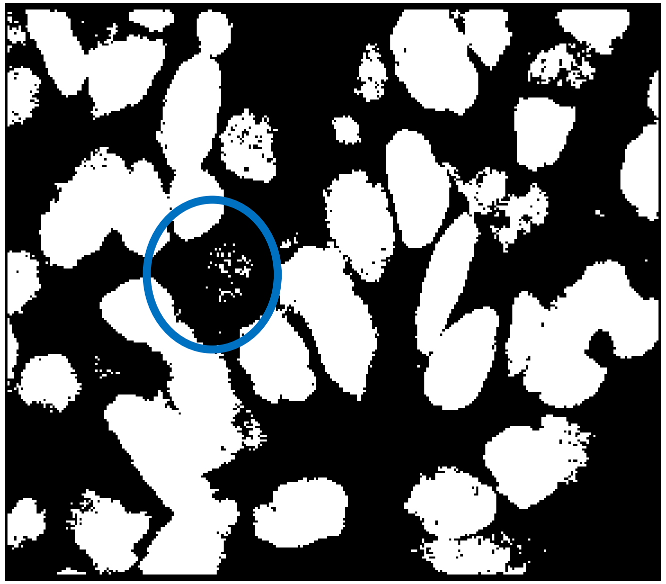


Niblack, W (1986), An introduction to Digital Image Processing, Prentice-Hall

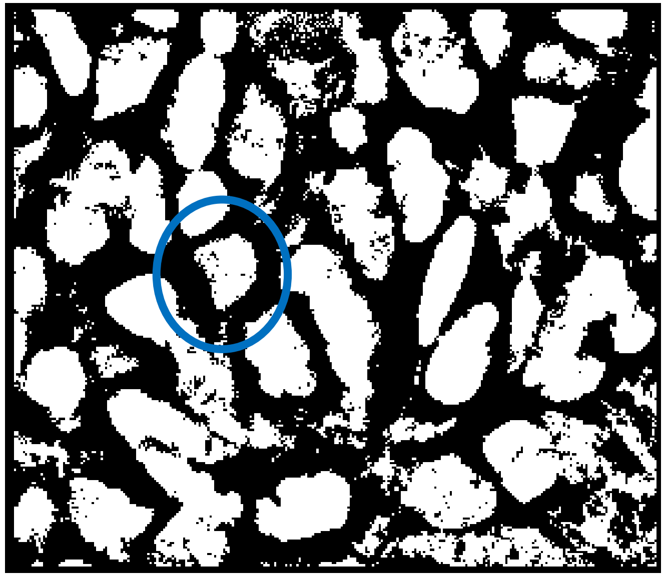
Examples of thresholding



Original



Global (Otsu)



Local (Niblack)

Additional improvements

- The shade in documents is often smooth...

⇒*Try to model it by a polynomial!*

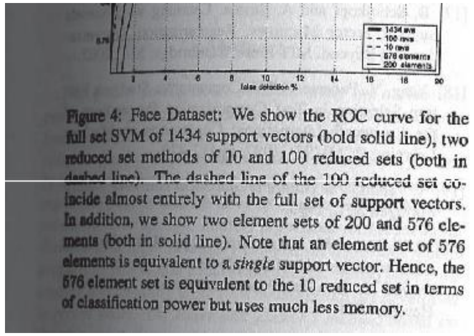
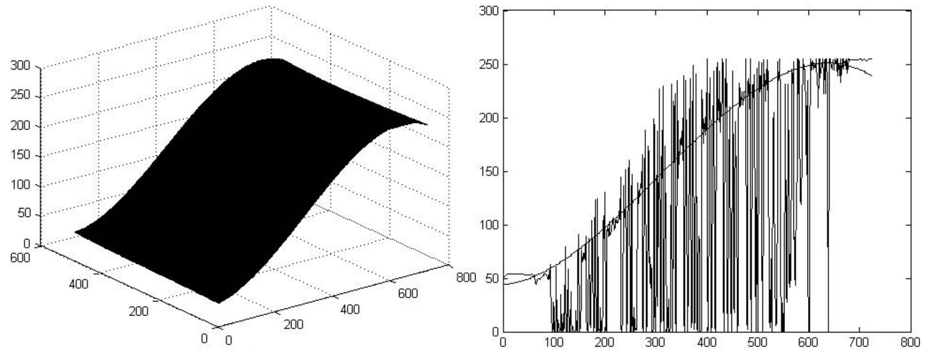


Figure 4: Face Dataset: We show the ROC curve for the full set SVM of 1434 support vectors (bold solid line), two reduced set methods of 10 and 100 reduced sets (both in dashed line). The dashed line of the 100 reduced set coincide almost entirely with the full set of support vectors. In addition, we show two element sets of 200 and 576 elements (both in solid line). Note that an element set of 576 elements is equivalent to a *single* support vector. Hence, the 576 element set is equivalent to the 10 reduced set in terms of classification power but uses much less memory.

Original



Fitted surface

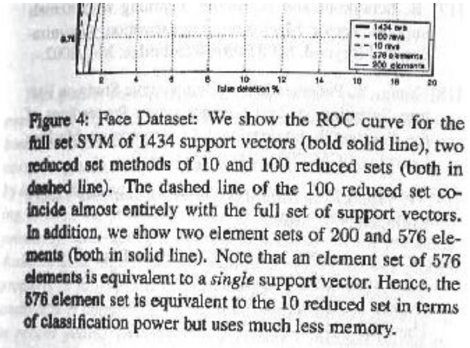


Figure 4: Face Dataset: We show the ROC curve for the full set SVM of 1434 support vectors (bold solid line), two reduced set methods of 10 and 100 reduced sets (both in dashed line). The dashed line of the 100 reduced set coincide almost entirely with the full set of support vectors. In addition, we show two element sets of 200 and 576 elements (both in solid line). Note that an element set of 576 elements is equivalent to a *single* support vector. Hence, the 576 element set is equivalent to the 10 reduced set in terms of classification power but uses much less memory.

Shadow compensation

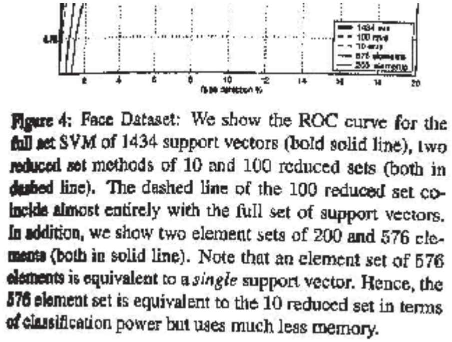


Figure 4: Face Dataset: We show the ROC curve for the full set SVM of 1434 support vectors (bold solid line), two reduced set methods of 10 and 100 reduced sets (both in dashed line). The dashed line of the 100 reduced set coincide almost entirely with the full set of support vectors. In addition, we show two element sets of 200 and 576 elements (both in solid line). Note that an element set of 576 elements is equivalent to a *single* support vector. Hence, the 576 element set is equivalent to the 10 reduced set in terms of classification power but uses much less memory.

Binarized result

Comparison of results

Original image

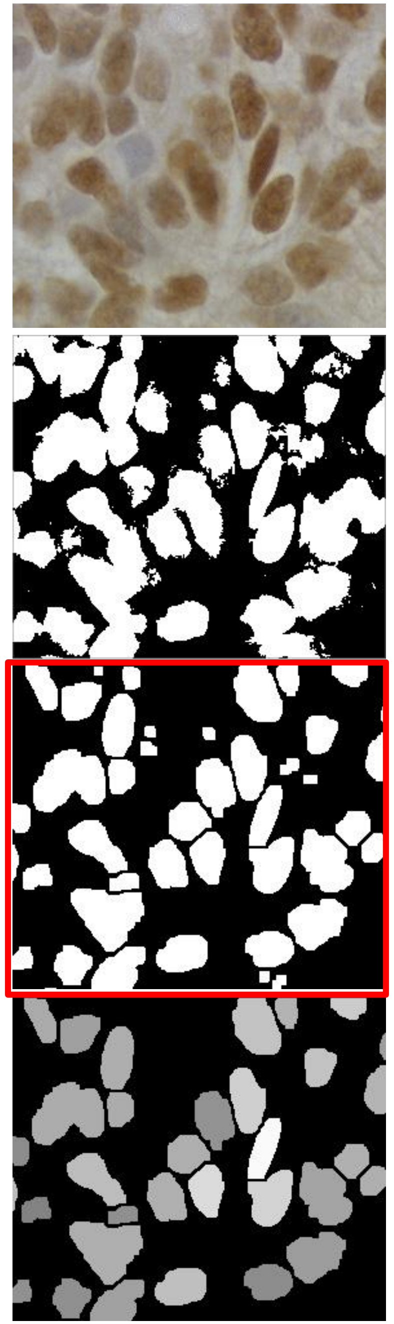
Global (Otsu)

Local (Sauvola)

Polynomial + global

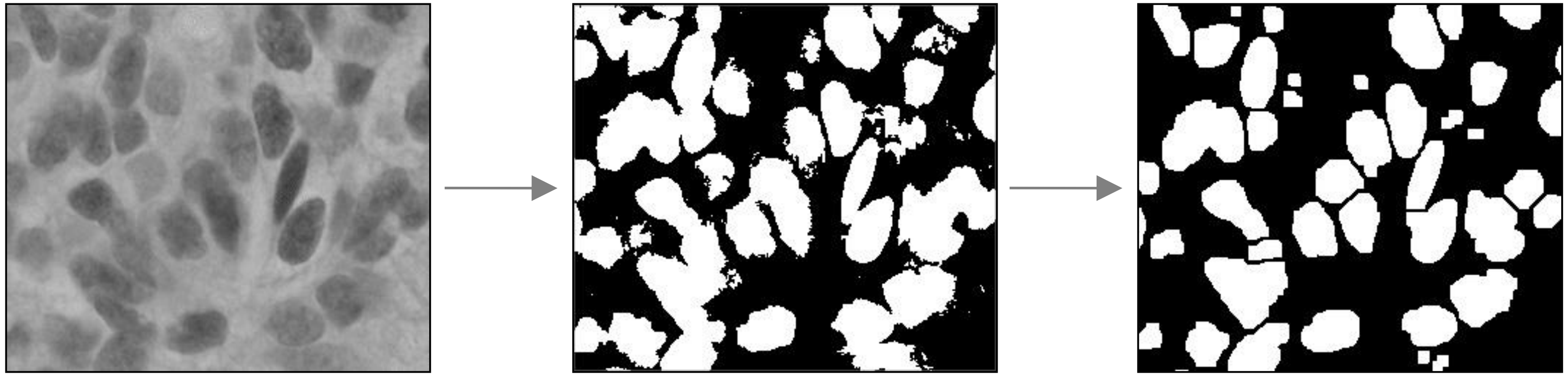
Machine perception

CLEANING THE IMAGE



Cleaning the binary image

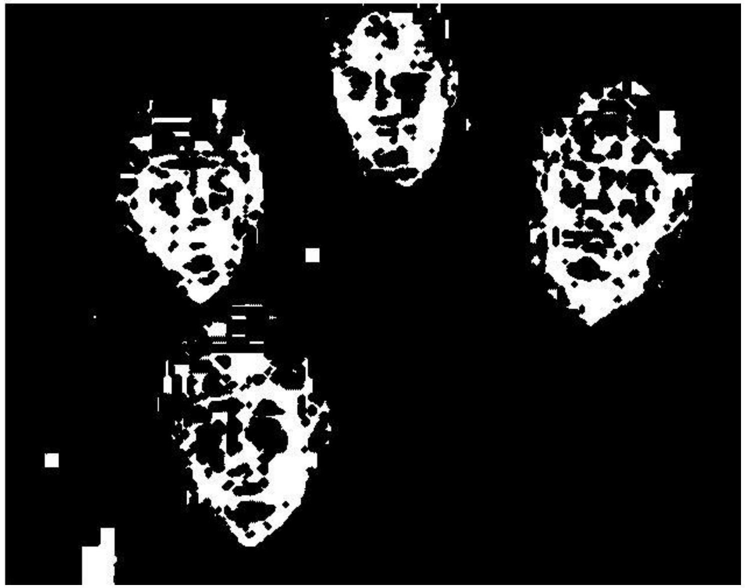
- Thresholded image still includes noise



- Require post-processing to remove artefacts
- Morphological operators
  - Remove isolated points and small structures
  - Fill holes

Dilation: A sneak peak preview

- Dilate the regions of „white“ pixels
- Increases the size of the structures
- Fills holes in regions



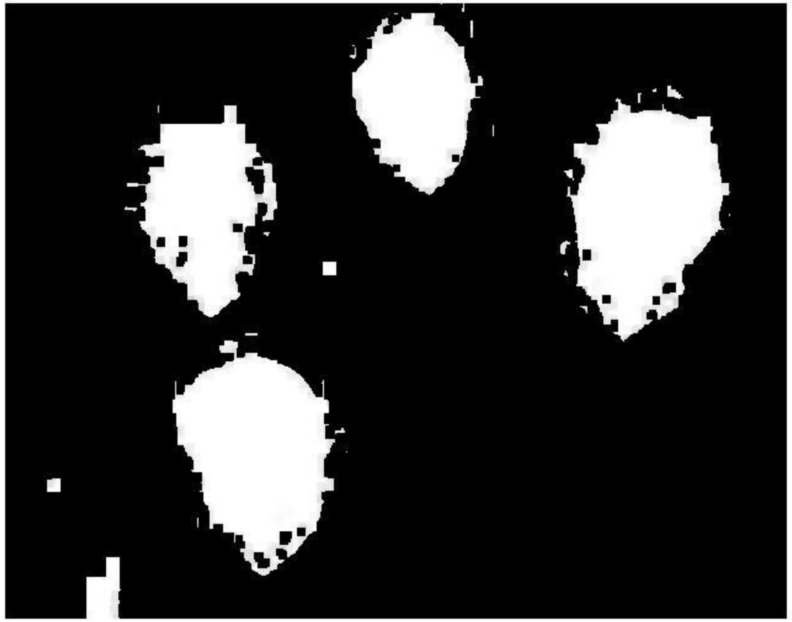
Before dilation

After dilation

Erosion: A sneak peak preview

- Erode the regions of „white“ pixels
- Reduce the size of structures
- Remove bridges, branches, noise

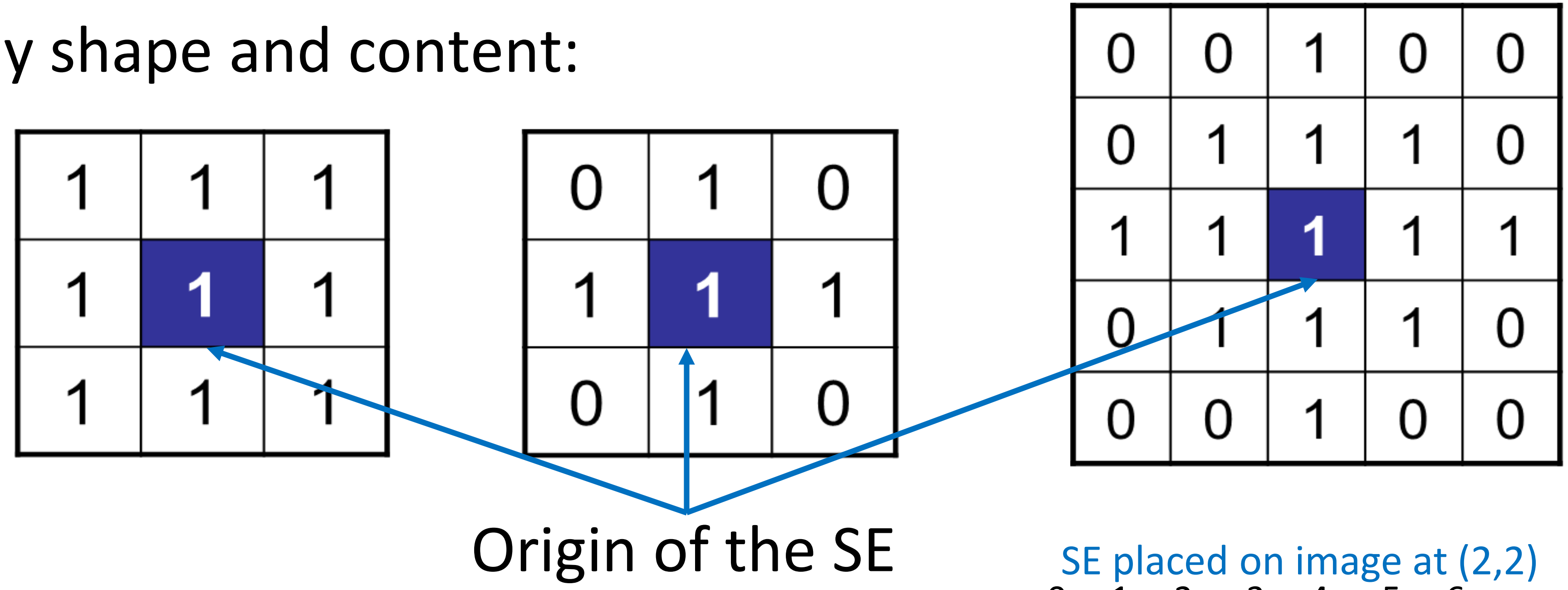
Before erosion



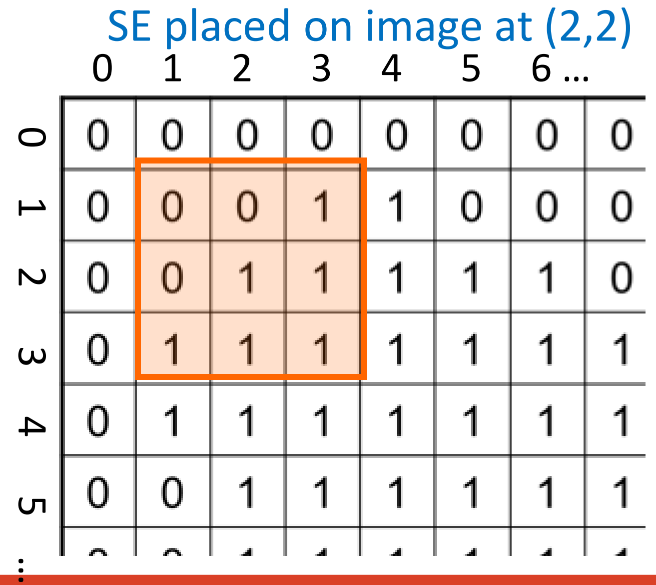
After erosion

Central to morphology: Structuring element (SE)

- Can be any shape and content:



- Fit: All “1” pixels in SE cover “1” pixels in the image.
- Hit: Any “1” pixels in SE cover “1” pixels in the image.



Fitting & Hitting

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1 **B**	1	1	1	0	0 **C**	1	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1 **A**	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

0	1	0
1	1	1
0	1	0

Structuring Element

Fit / Hit?

A:

B:

C:

Fit : All “1” elements in SE cover 1

Hit: Any “1” element in SE cover 1

Erosion

- Erosion of image f by structuring element s is given by $g = f \ominus s$.
- The structuring element s is positioned with its origin at (x, y) and the new pixel value is determined using the rule:

$$g(x,y) = \begin{cases} 1 \text{ if } s \text{ fits } f \\ 0 \text{ otherwise} \end{cases}$$

s

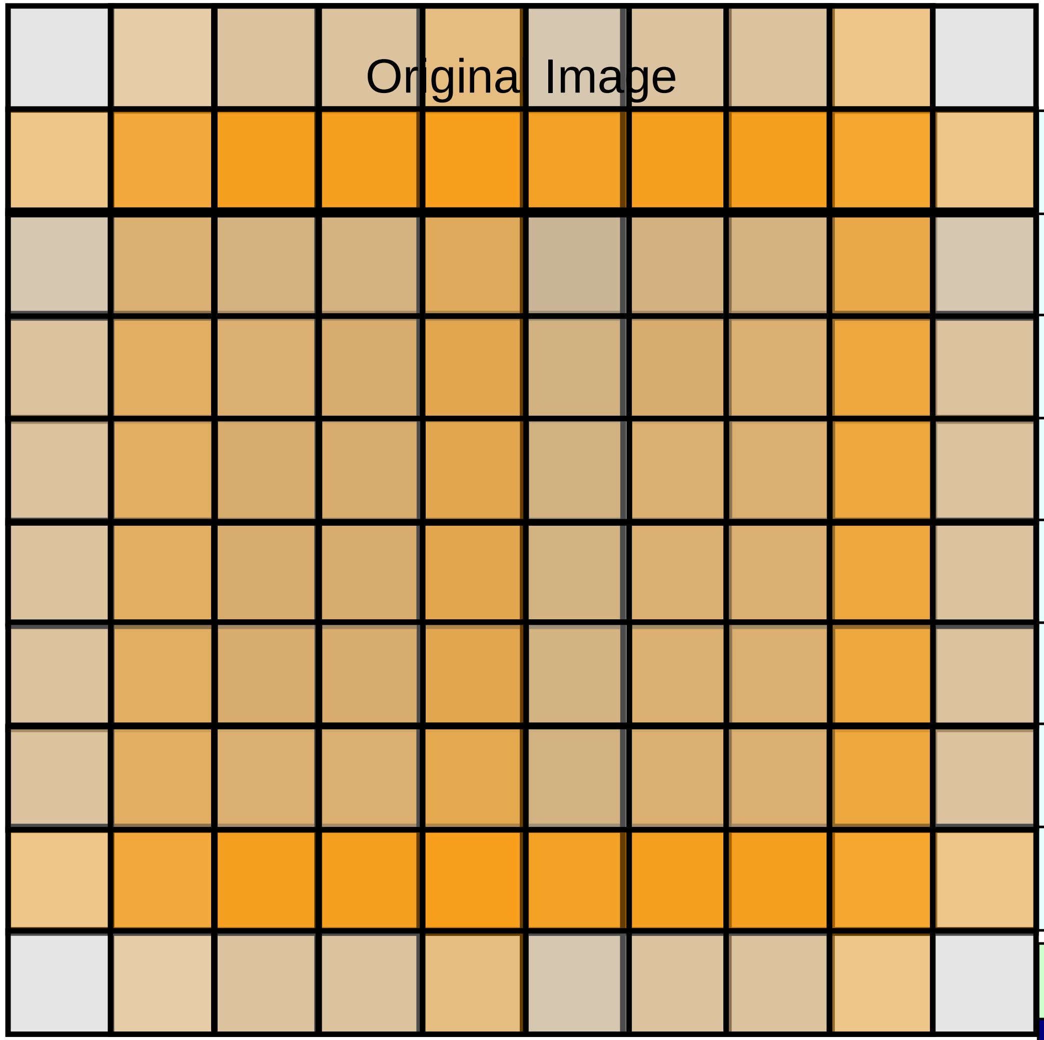
0	1	0
1	1	1
0	1	0

SE placed on image at (2,2)

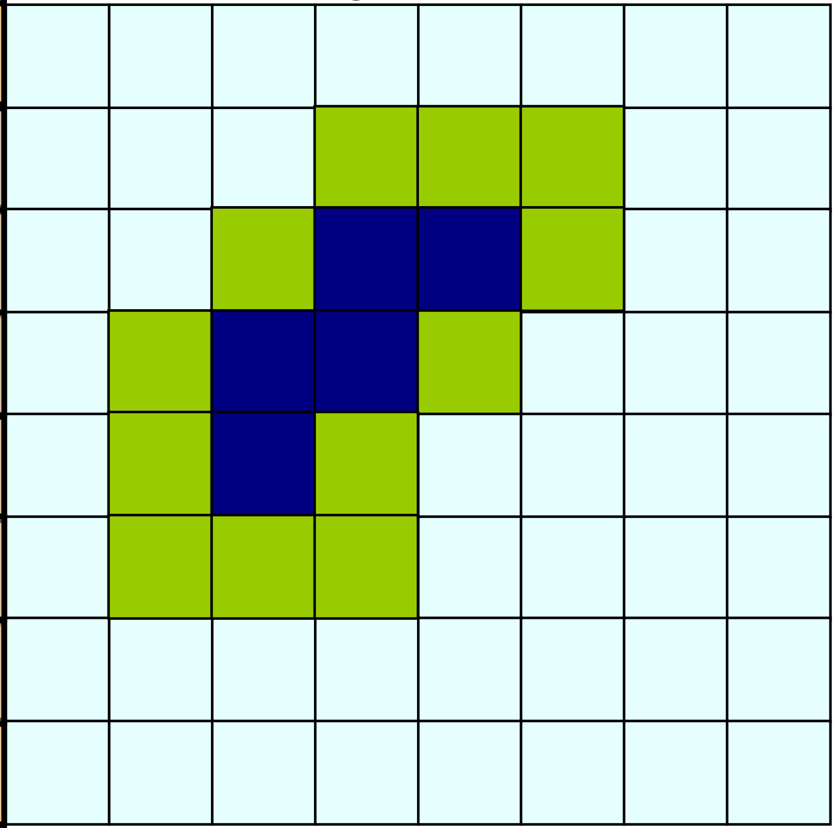
	0	1	2	3	4	5	6 ...	
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0
2	0	0	1	1	1	1	1	0
3	0	1	1	1	1	1	1	1
4	0	1	1	1	1	1	1	1
5	0	0	1	1	1	1	1	1
...								

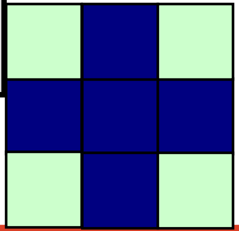
Fit: All “1” pixels in SE cover “1” pixels in the image.

Erosion Example



Processed Image With Eroded Pixels



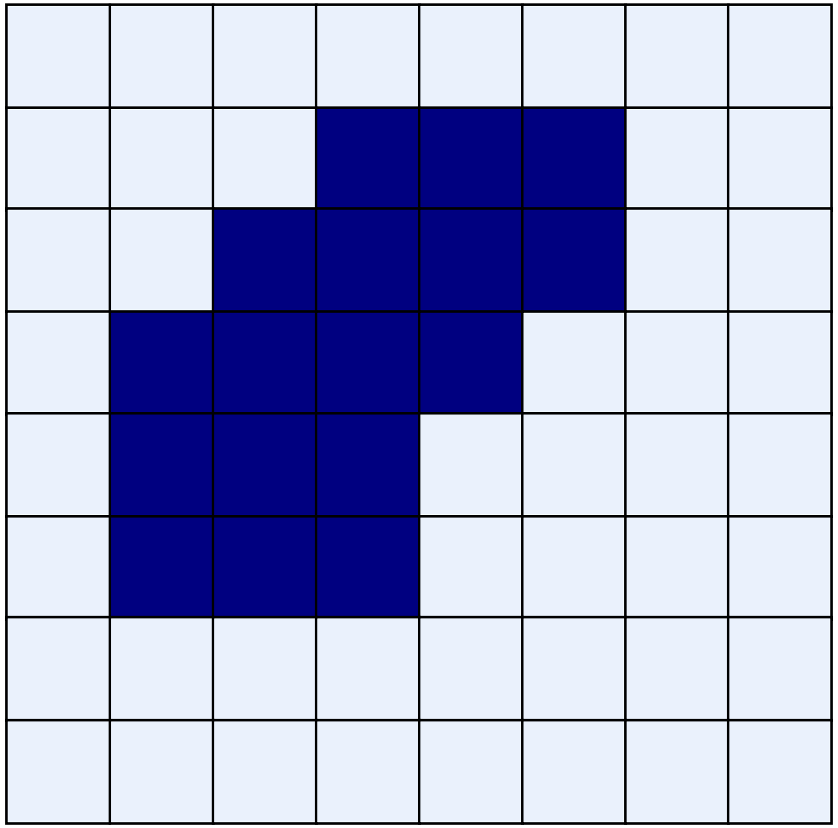


Structuring Element

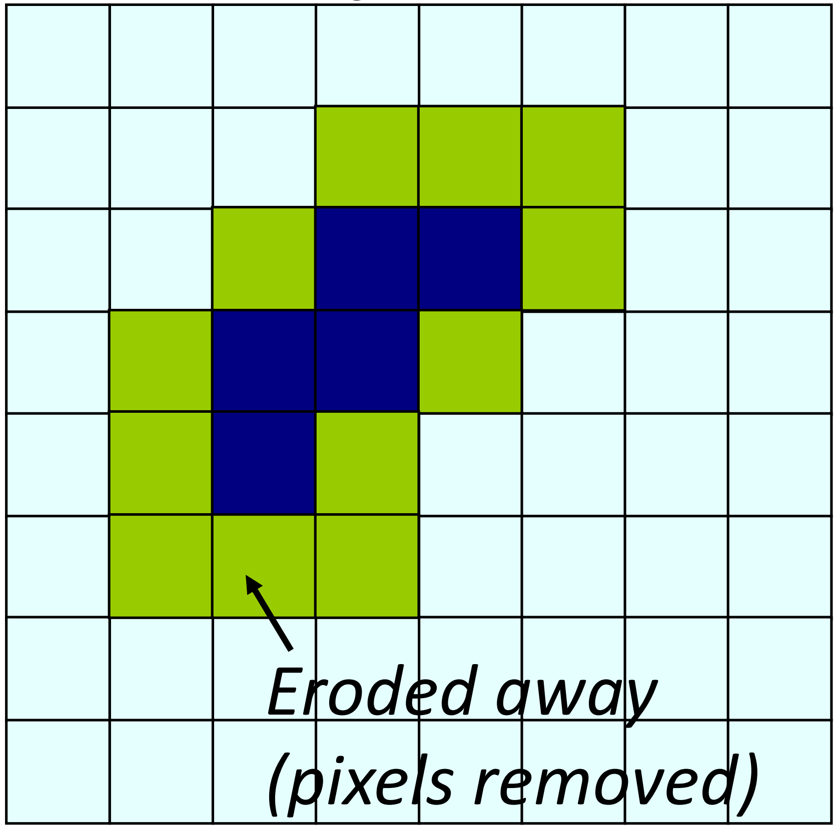
Fit :All 1 in SE covered in image

Erosion Example

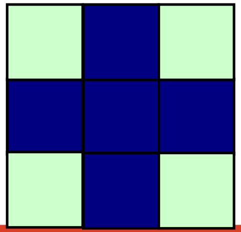
Original Image



Processed Image With Eroded Pixels



Eroded away (pixels removed)



Structuring Element

Fit :All 1 in SE covered in image

Dilation

- Dilation of image f by structuring element s is given by $g = f \oplus s$.
- The structuring element s is positioned with its origin at (x, y) and the new pixel value is determined using the rule:

$$g(x,y) = \begin{cases} 1 \text{ if } s \text{ hits } f \\ 0 \text{ otherwise} \end{cases}$$

s

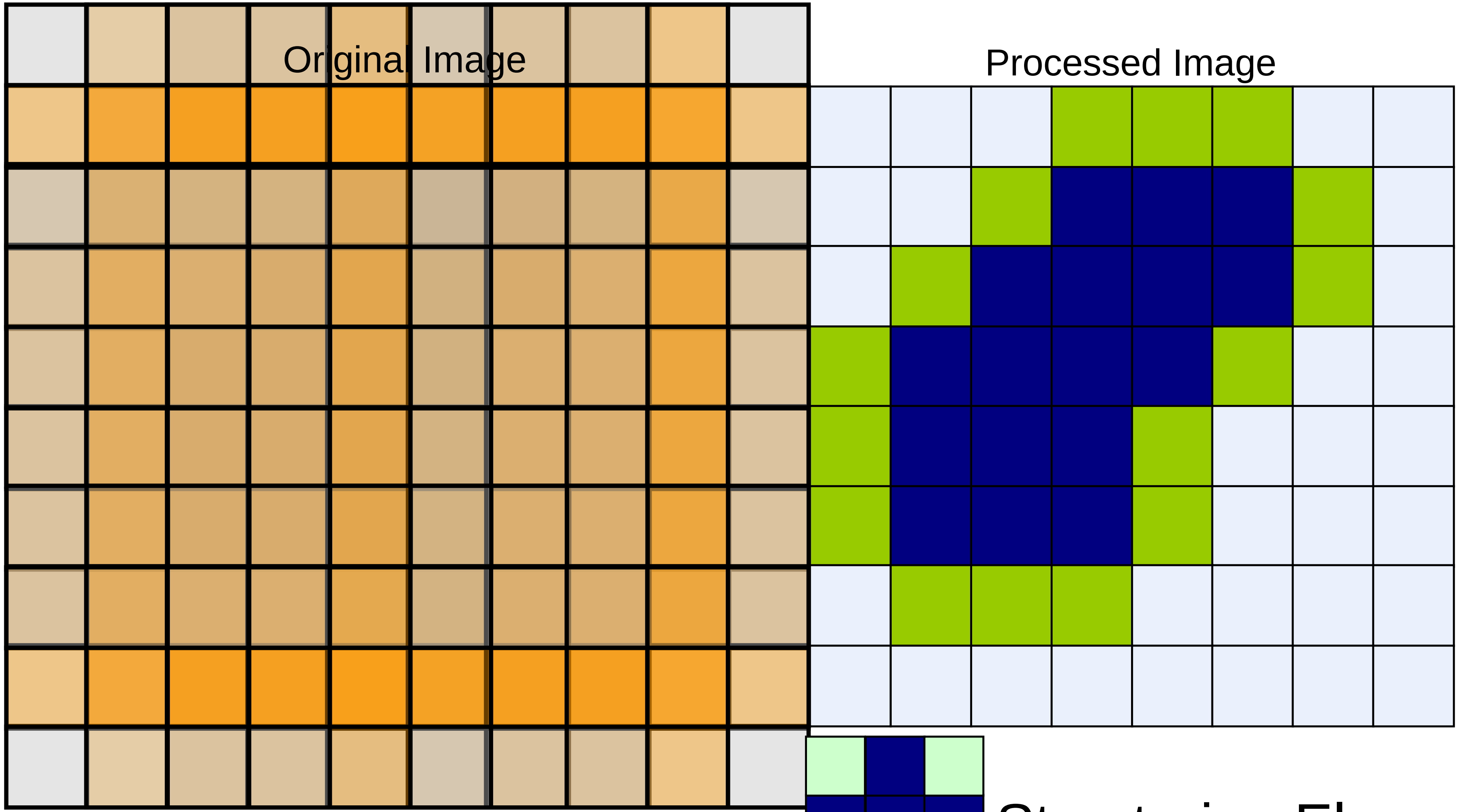
0	1	0
1	1	1
0	1	0

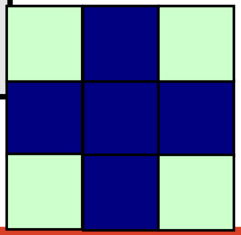
SE placed on image at (2,2)

	0	1	2	3	4	5	6 ...	
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0
2	0	0	1	1	1	1	1	0
3	0	1	1	1	1	1	1	1
4	0	1	1	1	1	1	1	1
5	0	0	1	1	1	1	1	1
...								

Hit: Any "1" pixels in SE cover "1" pixels in the image.

Dilation Example

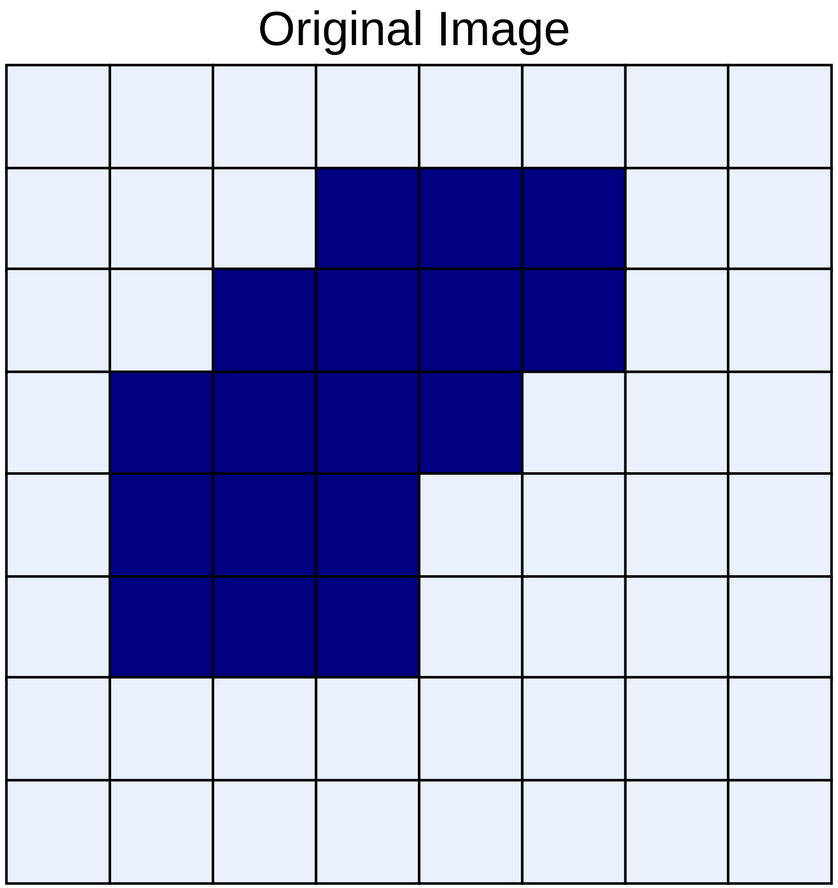


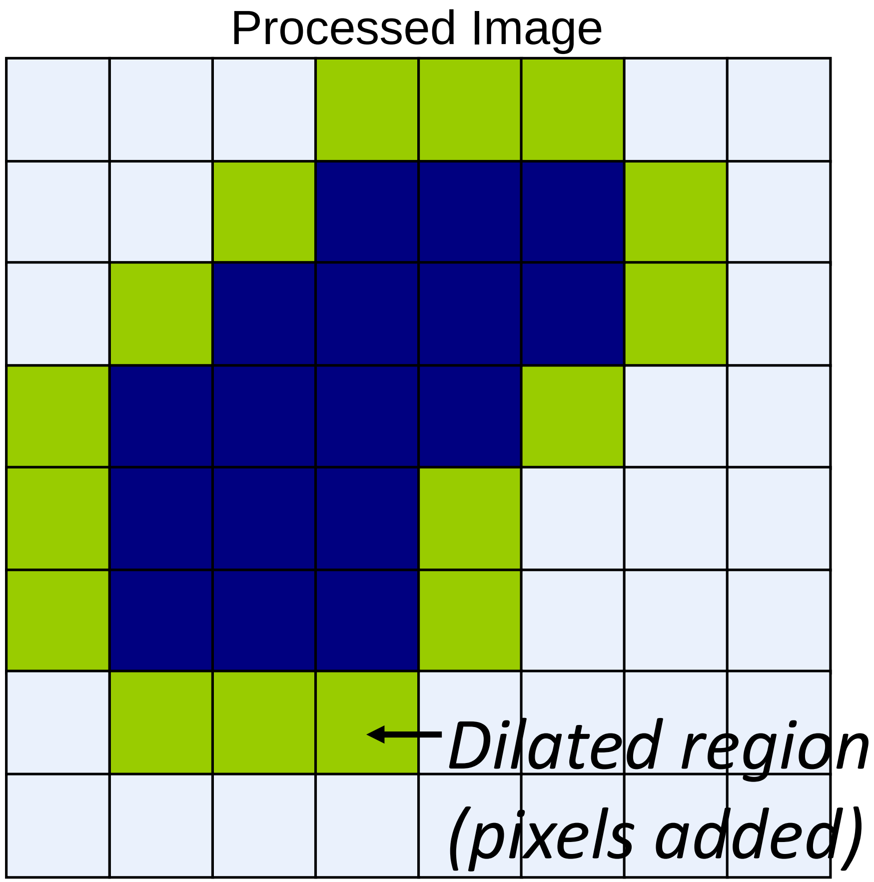


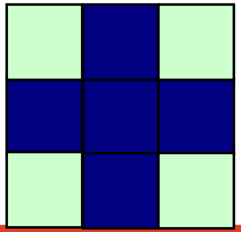
Structuring Element

Hit: Any 1 in SE covered in image

Dilation Example



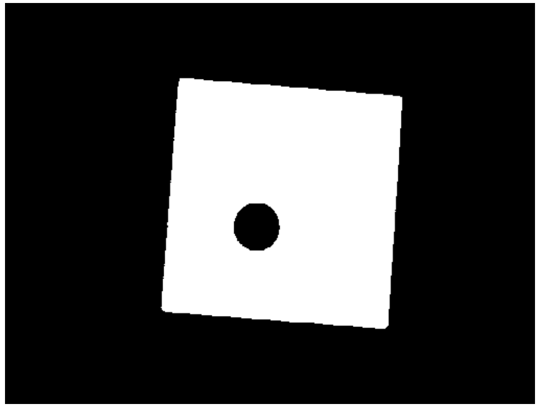




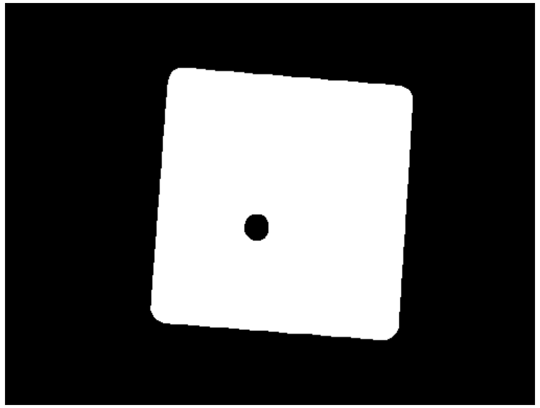
Structuring Element

Hit: Any 1 in SE covered in image

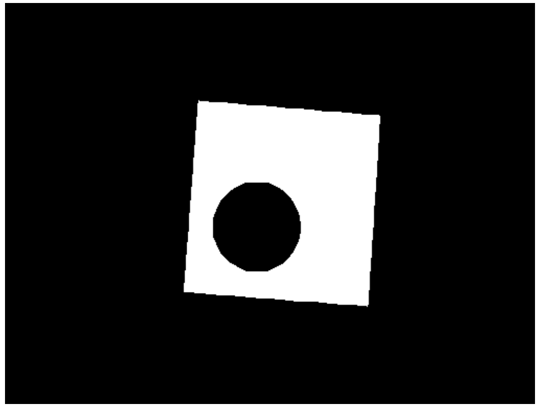
Effects of erosion and dilation



Original



Dilation by a round structuring element.



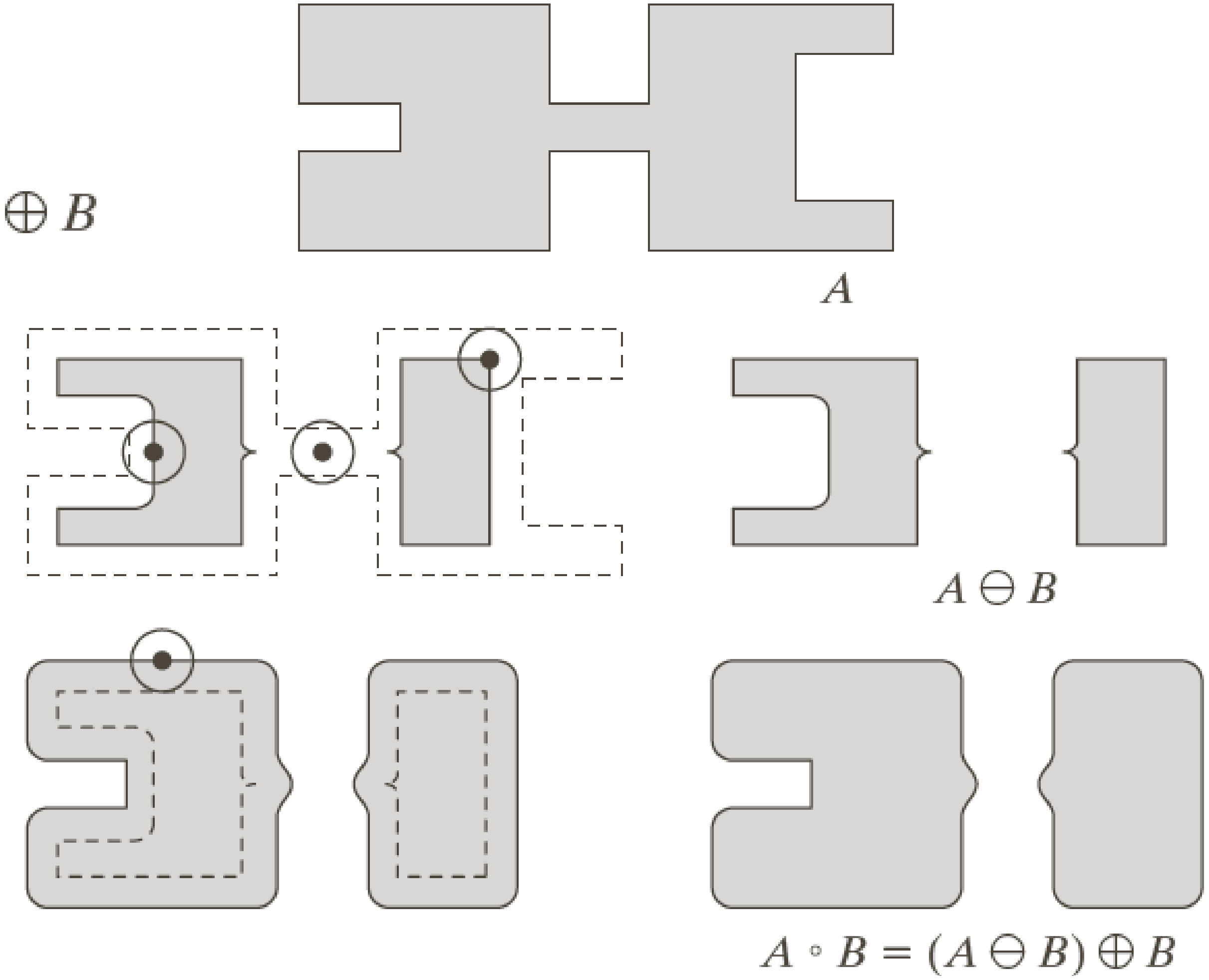
Erosion by a round structuring element.

Source of images: http://homepages.inf.ed.ac.uk/rbf/HIPR2/

Combined operations: Opening

- Definition
  - Apply erosion then dilation

$$A \circ B = (A \ominus B) \oplus B$$

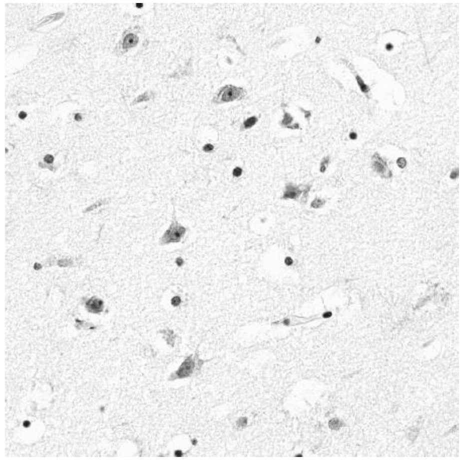


- Effect:

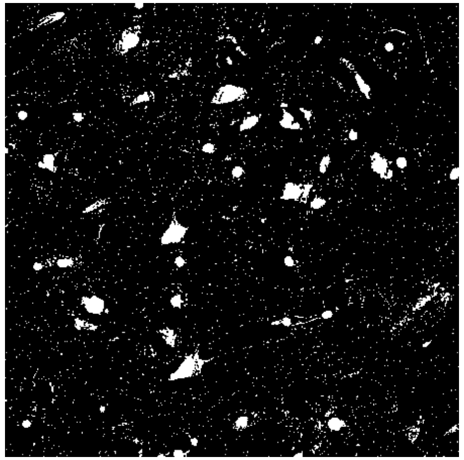
  $\Rightarrow$ Removes small objects,
  preserves rough shape.

Effects of opening

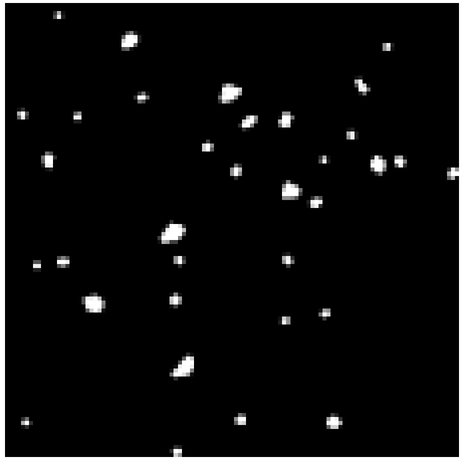
- Can filter out structures by selecting the size of structuring element.



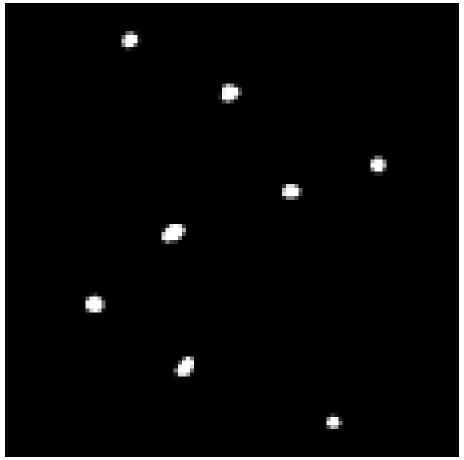
Original



Thresholded



Opening by a small structuring element

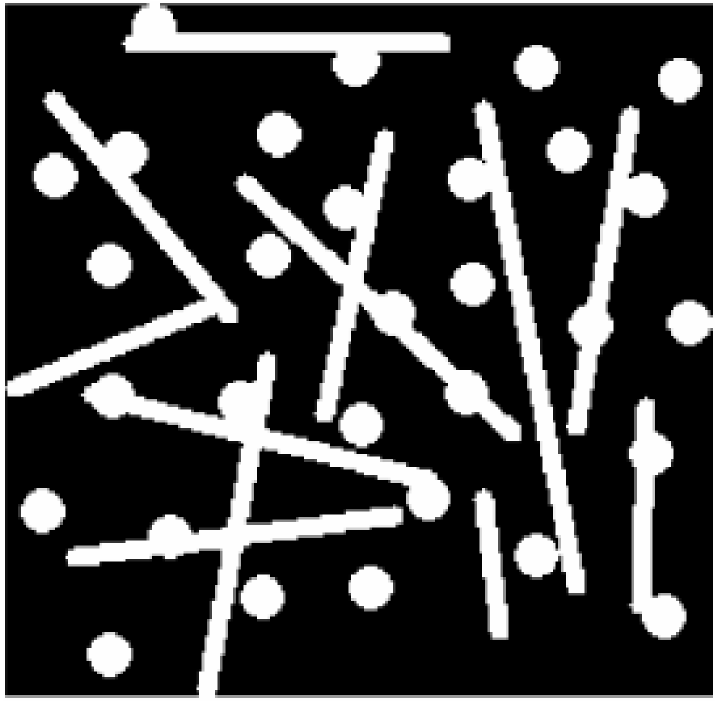


Opening by a large structuring element

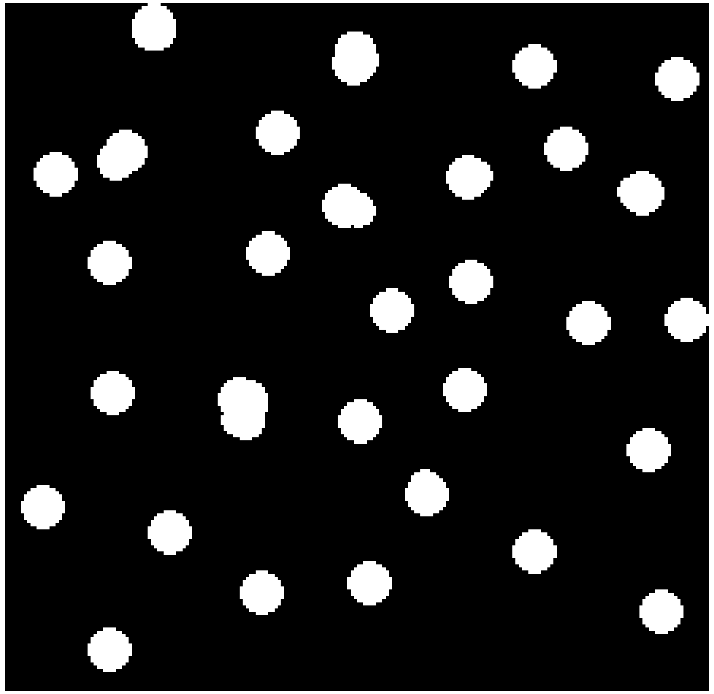
Source of images: http://homepages.inf.ed.ac.uk/rbf/HIPR2/

Effects of opening

- Choose the structure in image by choosing the shape of the structuring element.



Original image

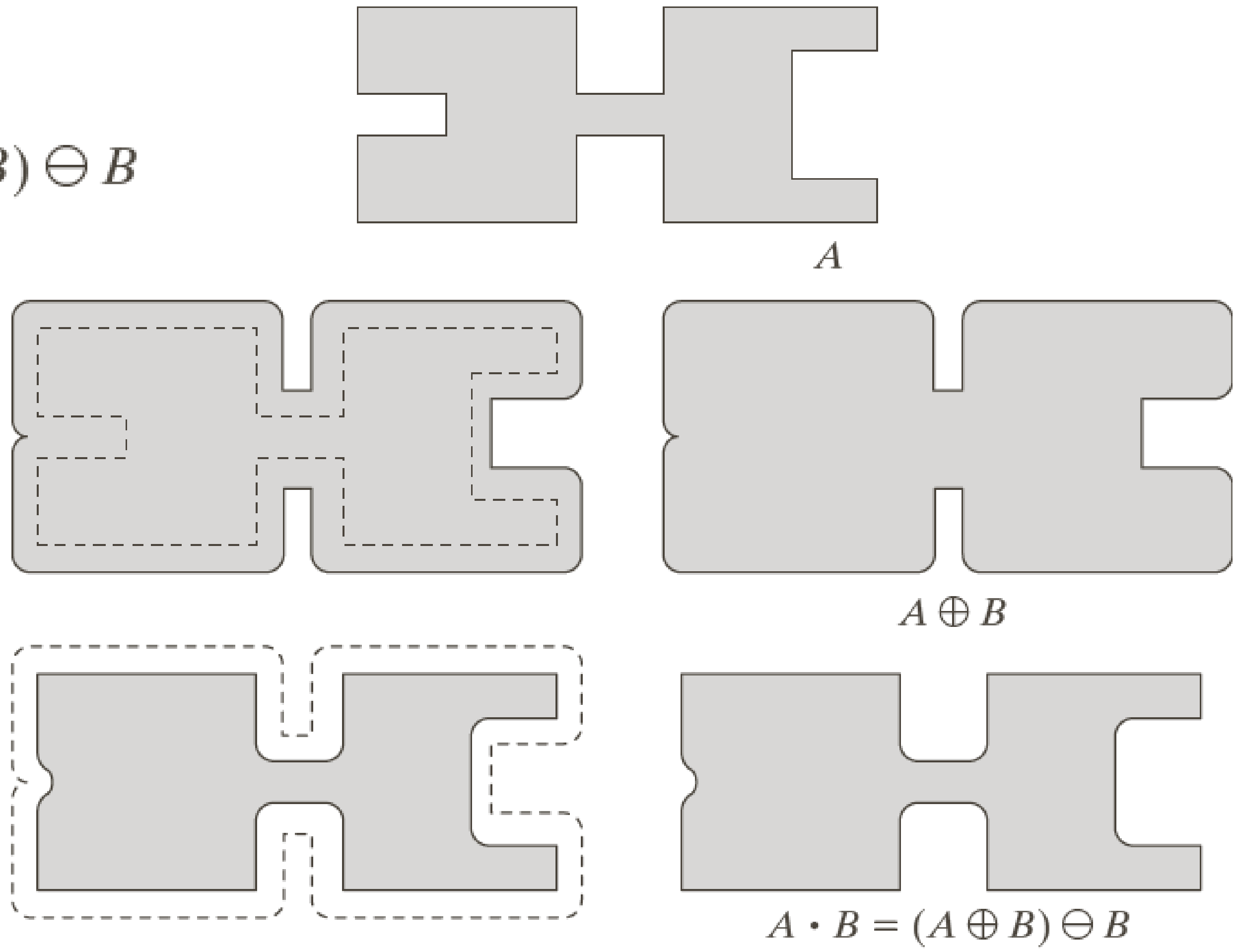


Opening by a round structuring element

Combined operations: Closing

- Definition
  - Apply dilation then erosion

$$A \cdot B = (A \oplus B) \ominus B$$



- Effect

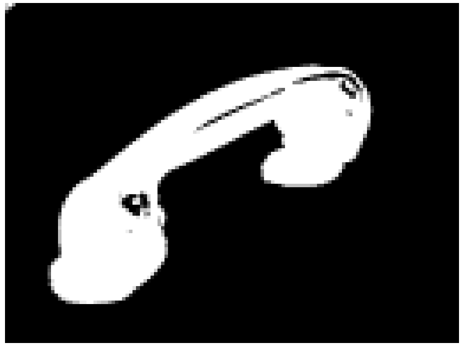
  $\Rightarrow$ Fill holes, preserves the original shape.

Effects of closing

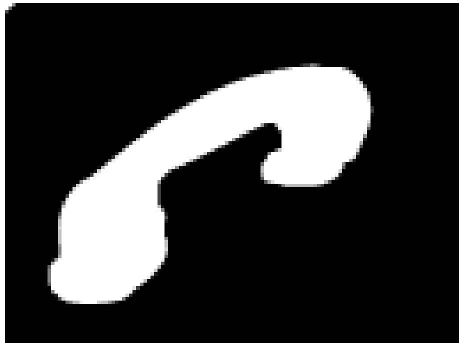
- Fill holes in thresholded image

  (*eg., reflections*)

Original

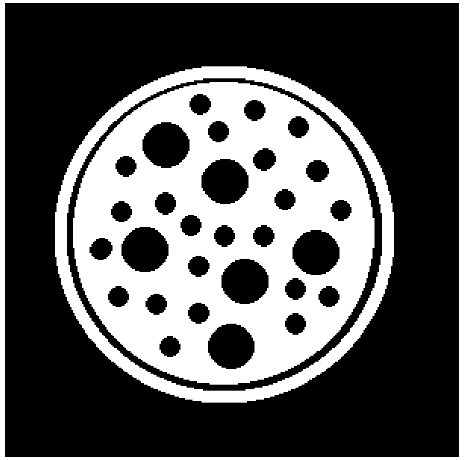


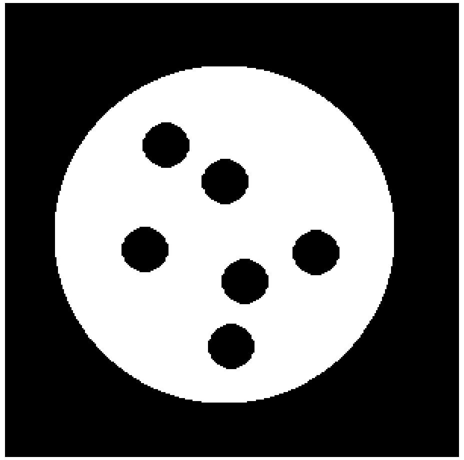
Thresholded



Closing by a round structuring element

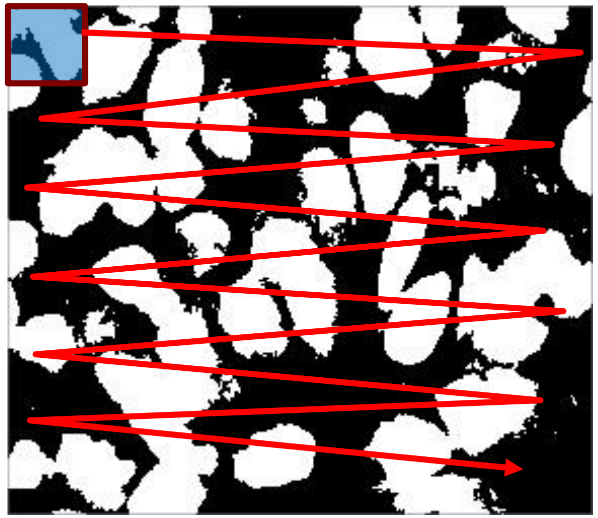
The size of structuring element determines the maximal size of holes that will be filled.





Example: opening + closing

Morphological operators in OpenCV

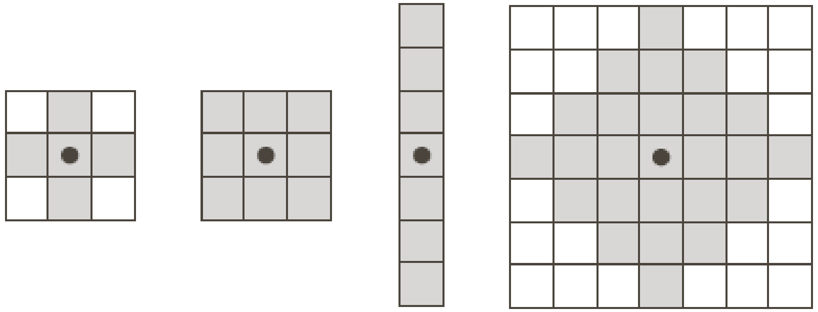


- Main operations
  - Dilation (OpenCV: `cv2.dilate`)
  - Erosion (OpenCV: `cv2.erode`)

- Several important combinations
  - Opening (OpenCV: `cv2.morphologyEx(img,cv2.MORPH_OPEN, kernel)`)
  - Closing (OpenCV: `cv2.morphologyEx(img,cv2.MORPH_CLOSE, kernel)`)
  - Boundary extraction

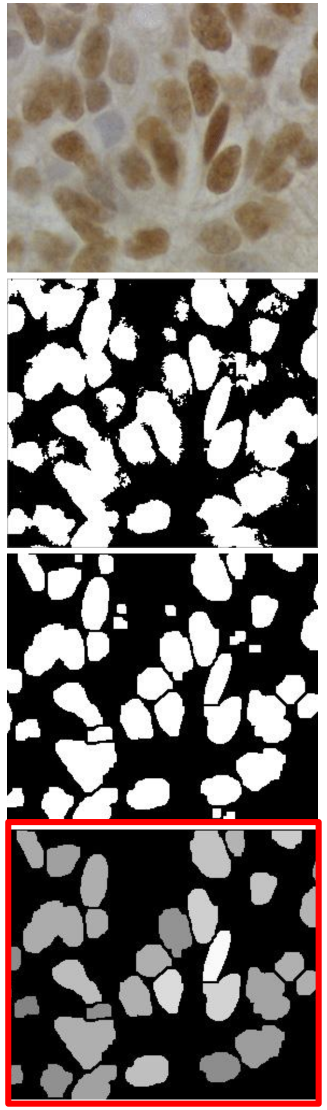
- Much more available (see help)

Examples of structuring elements:



Machine perception

LABELLING REGIONS

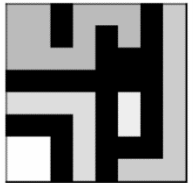


Connected components for labeling

- Goal: find separate connected regions

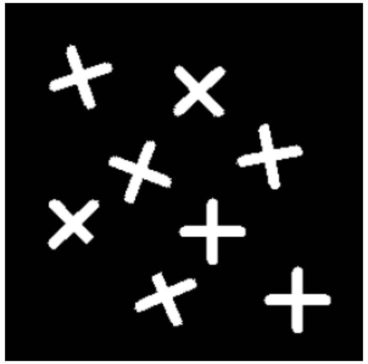
1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

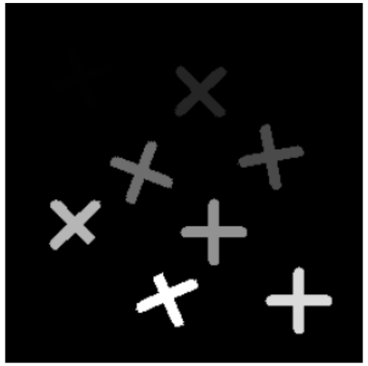
Binary image



1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

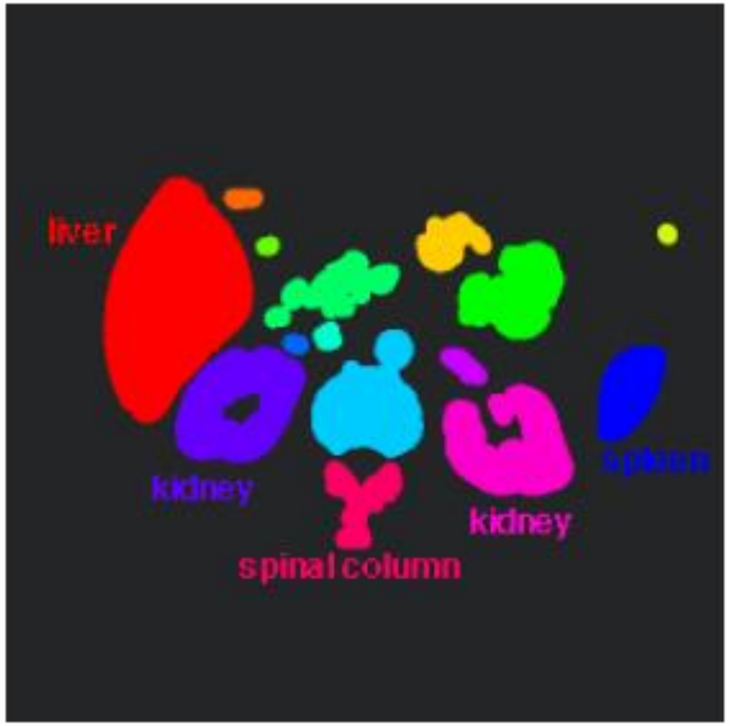
connected components





Sources: Shapiro & Stockman, Chandra

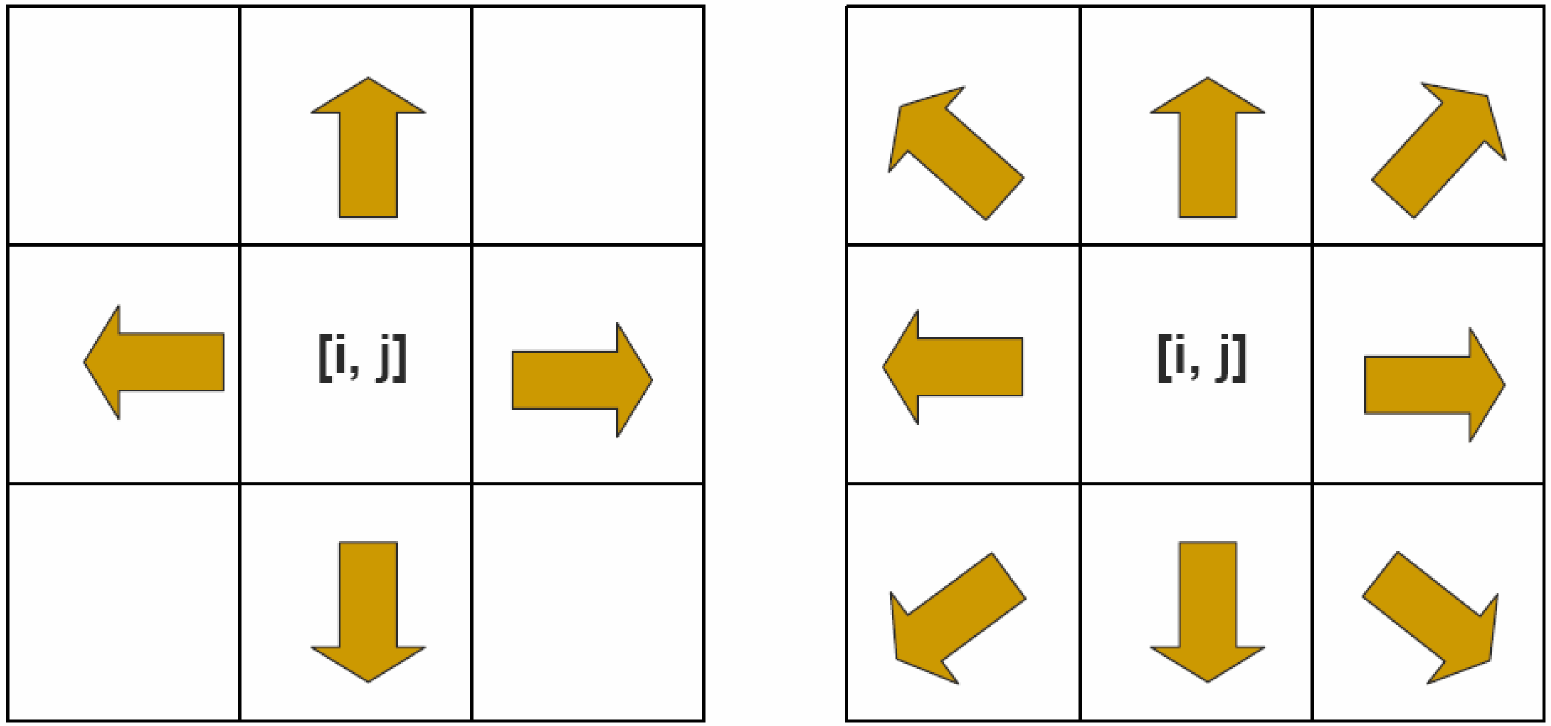
Examples of connected components



connected components of 1's from thresholded image

Connectivity

- Determines which pixels are considered neighbors.



4-neighborhood

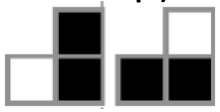
8-neighborhood

Sequential connected components

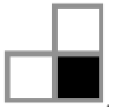
- Process image from left to right, from top to bottom:

  1.) If the current pixel value is 1

    i.) If only one neighbor (left or top) is 1, copy its label.



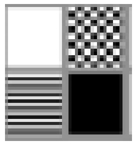
    ii.) If both neighbors are 1 and have same label, copy that label.



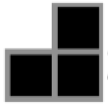
    iii.) If they have different labels

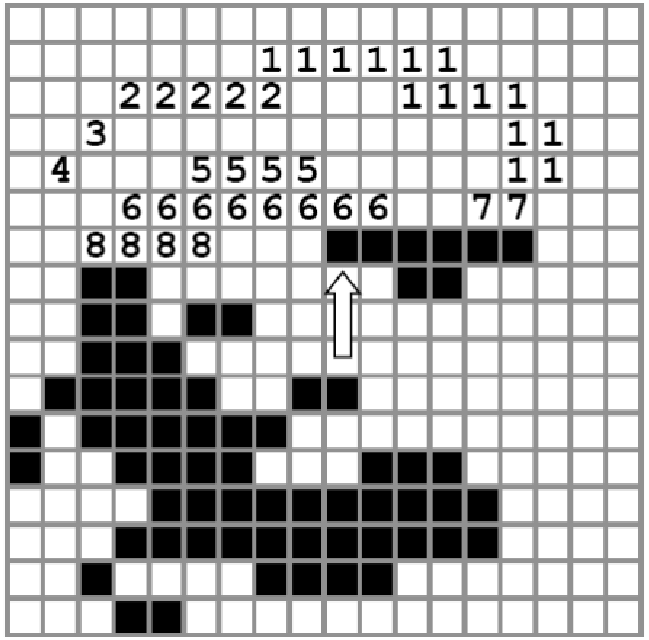
      – Copy label from the left.

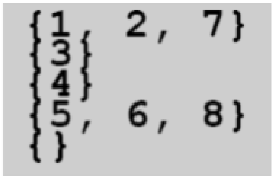
      – Update the table of equivalent labels.



    iv.) Otherwise form a new label.







- Relabel with the smallest equivalent labels.

Example SCC: 8-connectivity

(a)

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0
0	1	1	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0 Background

1 Foreground

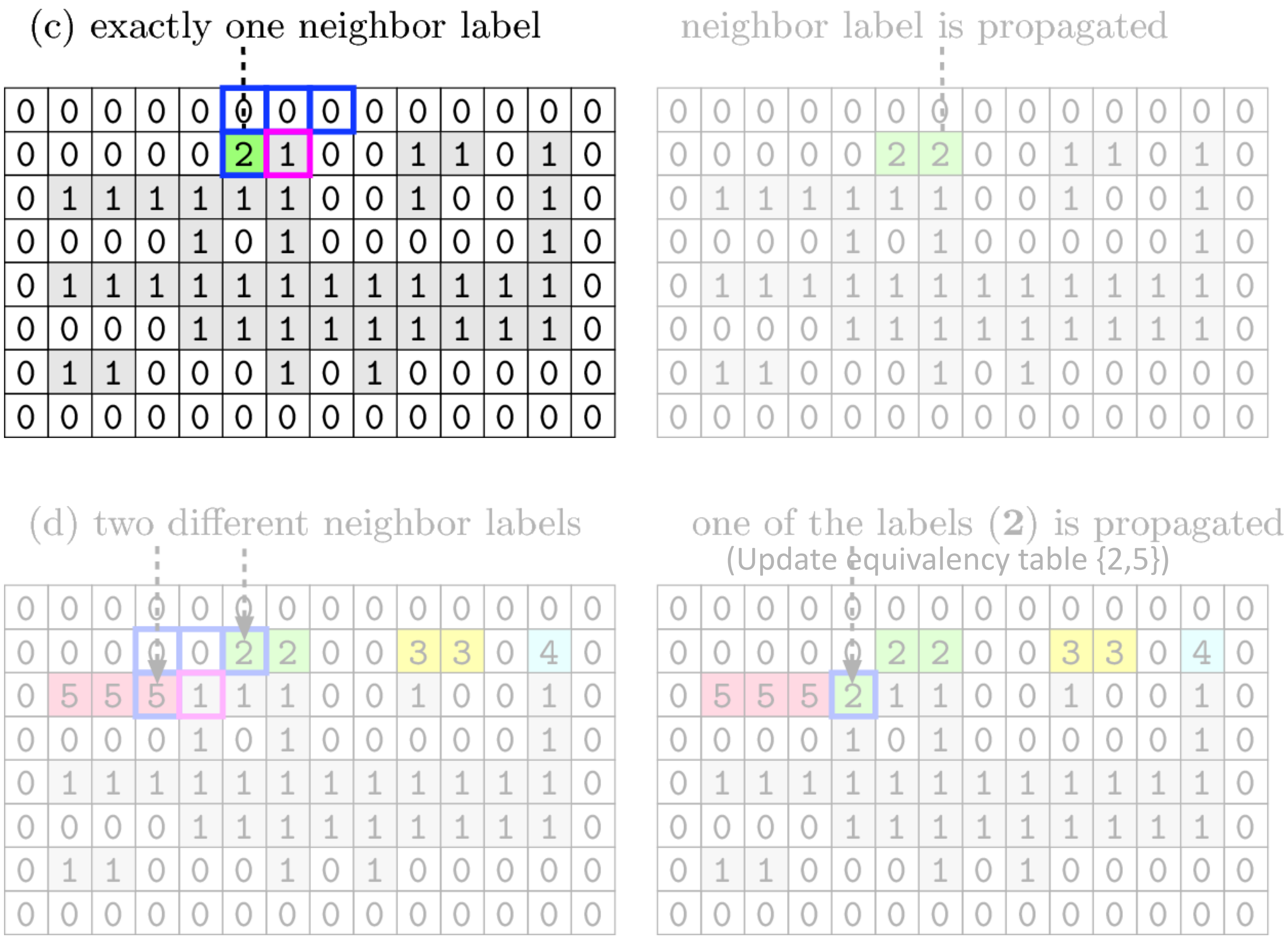
(b) only background neighbors

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0
0	1	1	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

new label (**2**)

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	0
0	1	1	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Example SCC: 8-connectivity

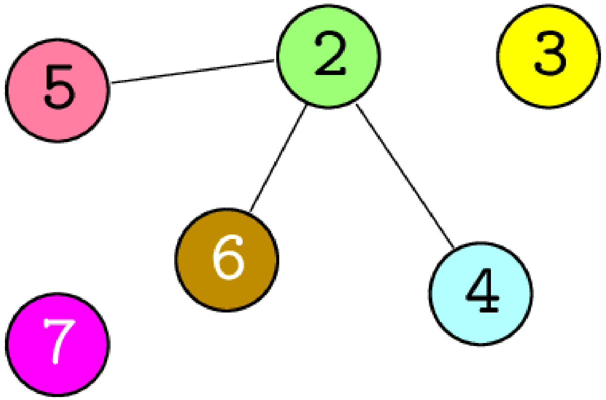


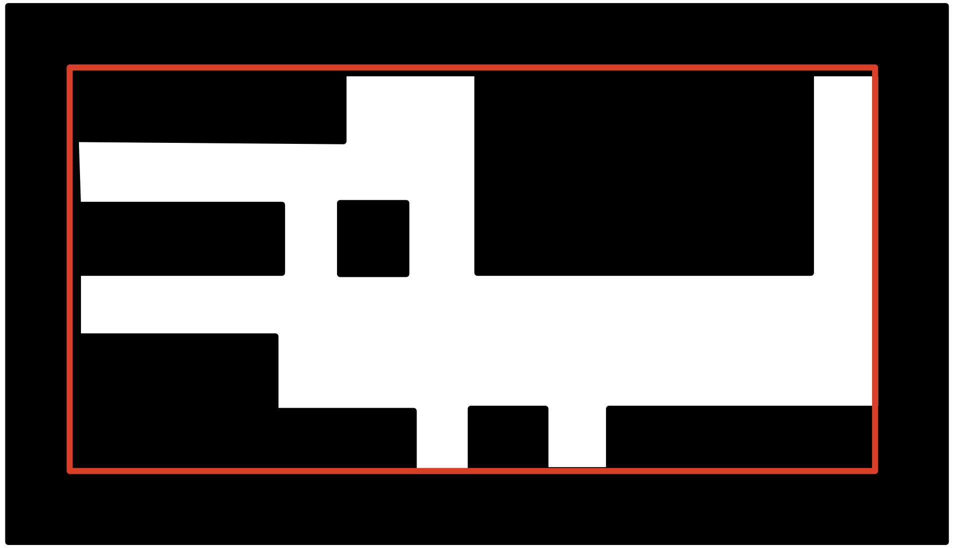
Example SCC: 8-connectivity

First pass: label

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	2	0	0	3	3	0	4	0
0	5	5	5	2	2	2	0	0	3	0	0	4	0
0	0	0	0	2	0	2	0	0	0	0	0	4	0
0	6	6	2	2	2	2	2	2	2	2	2	2	0
0	0	0	0	2	2	2	2	2	2	2	2	2	0
0	7	7	0	0	0	2	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Equivalency table

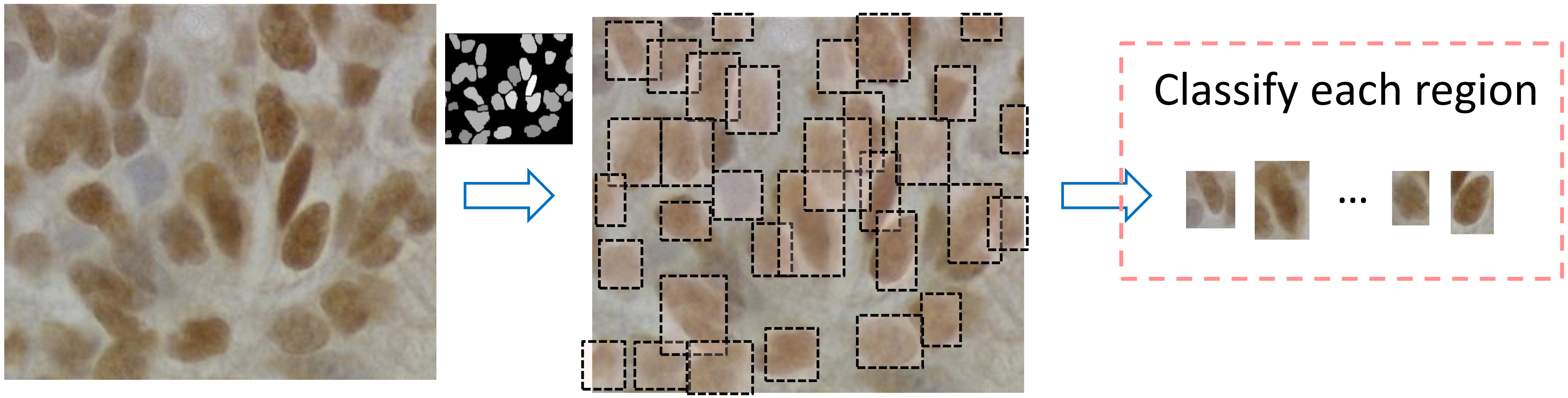




l==2

Second pass: apply equivalences

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	2	0	0	3	3	0	2	0
0	2	2	2	2	2	2	0	0	3	0	0	2	0
0	0	0	0	2	0	2	0	0	0	0	0	2	0
0	2	2	2	2	2	2	2	2	2	2	2	2	0
0	0	0	0	2	2	2	2	2	2	2	2	2	0
0	7	7	0	0	0	2	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0



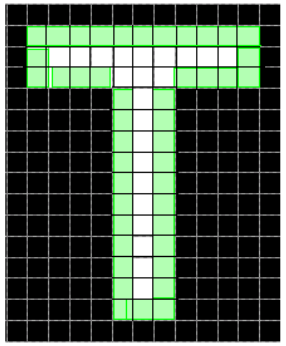
Machine perception

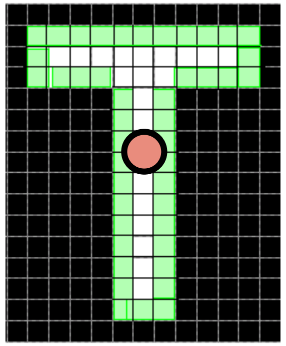
REGION DESCRIPTORS

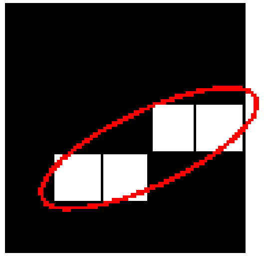
Simple region descriptors

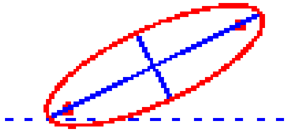
- A region can be detected using the connected components.
- How to describe it?
- Some examples:
  - Area A
  - Perimeter l
  - Compactness $c=l^2/(4\pi A)$
  - Circularity, roundness $1/c$
  - Centroid (center of mass)
  - Major and minor axes λ_1, λ_2
  - Eccentricity $||\lambda_1||/||\lambda_2||$
  - Minimal bounding box area $A_m = h\,b$
  - Rectangularity A/A_m

Matlab: *regionprops*





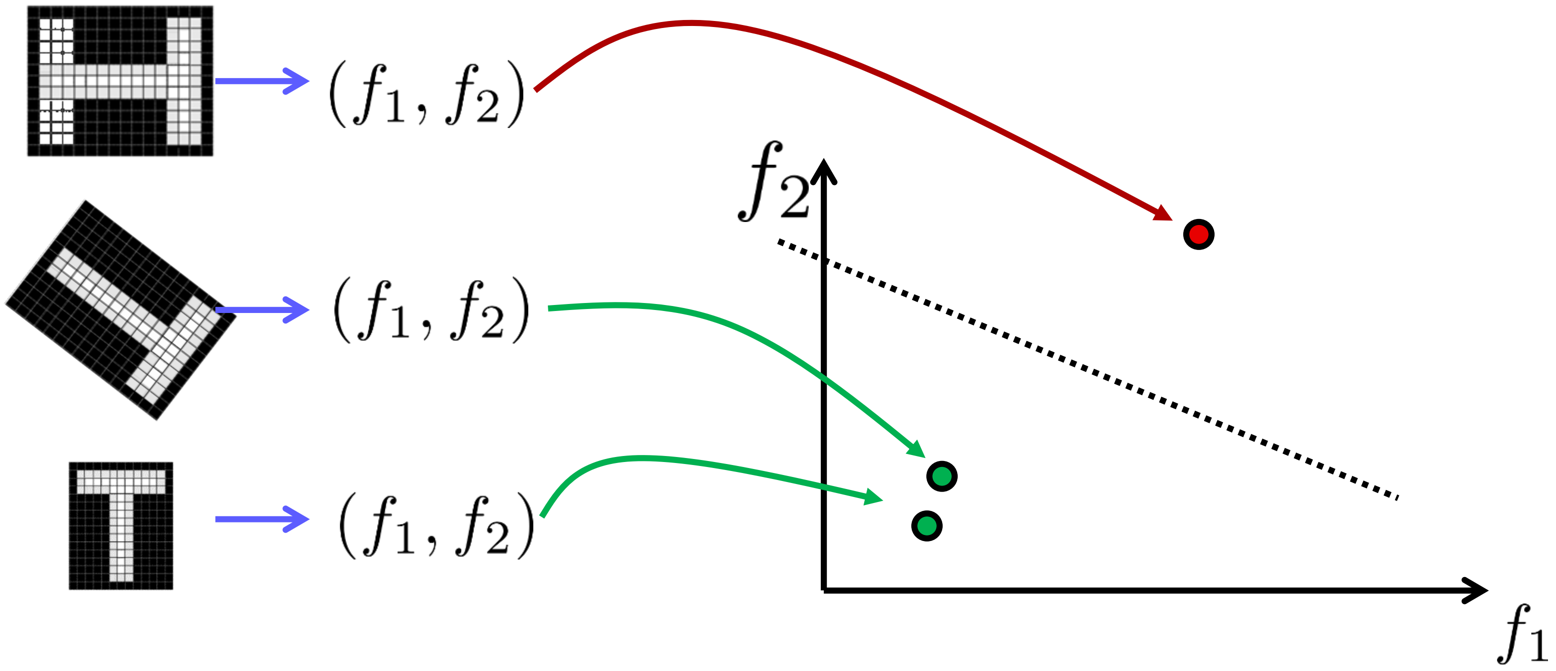




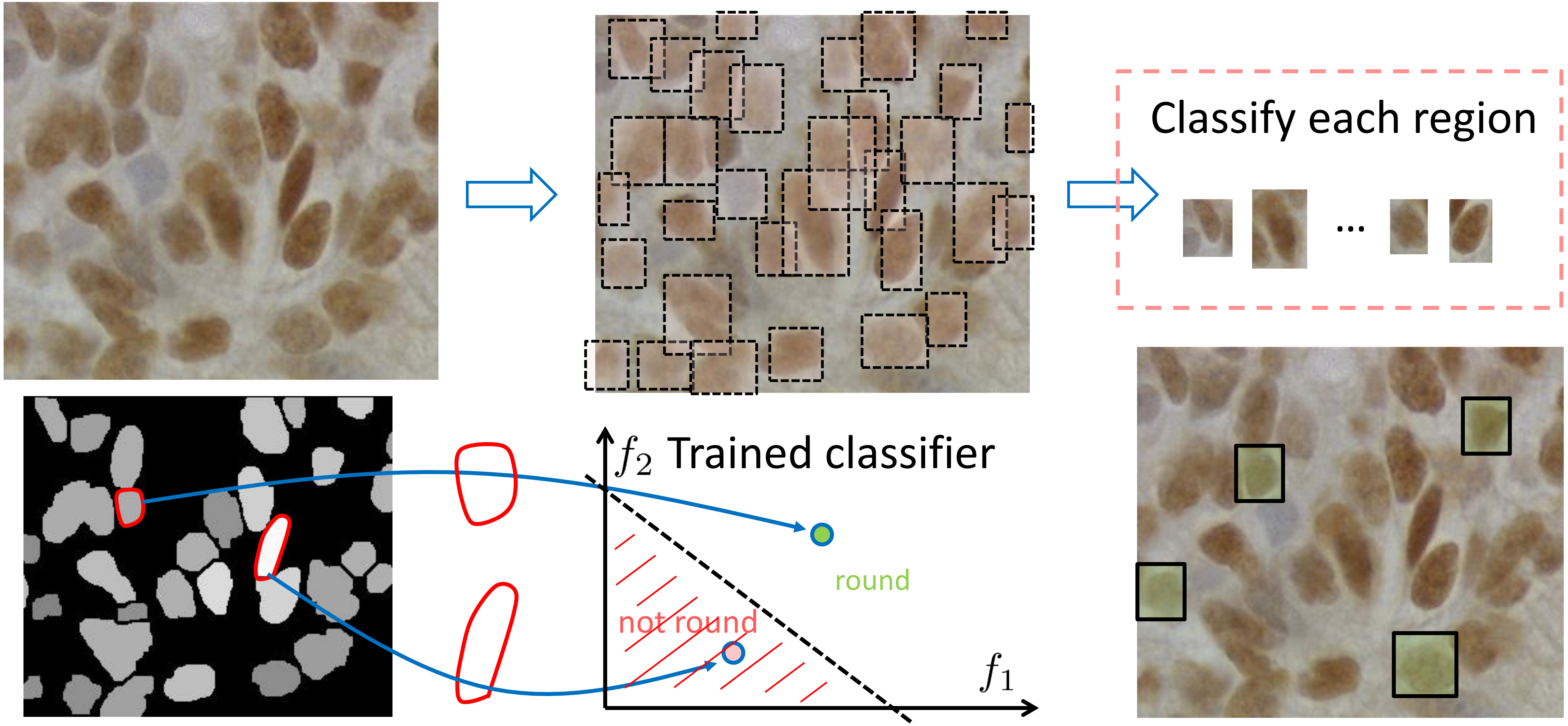
(Easy to come up with your own)

Require a level of invariance (App dependent)

- Ideal descriptor will map:
  - Two images of the same object close-by in feature space.
  - Two images of different objects to points far between each other.



Task: Detect round cells



Summary: Binarization

- Pros
  - Fast, simple to store
  - Simple techniques
  - Works in constrained setups
- Cons
  - Difficult to get „clean“ shapes
  - Many real-world scenarios contain noise
  - Often too coarse representation
  - Not robust in changes of 3D view changes

Python code

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

img_bgr = cv2.imread('C:/Users/matej/Documents/Articles/Lectures/Machine Perception/1_Image processing 1/code/matlab/coins.jpg')
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

plt.figure(1)
plt.imshow(img_rgb)
plt.show()

a_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY)
plt.imshow(a_gray, cmap='gray')
plt.show()

a_bin = a_gray<170
a_bin = a_bin.astype(np.uint8)

plt.imshow(a_bin, cmap='gray')
plt.show()

kernel = np.ones((5,5),np.uint8)

a_close = cv2.morphologyEx(a_bin, cv2.MORPH_CLOSE, kernel)

plt.imshow(a_close, cmap='gray')
plt.show()

ret, labels = cv2.connectedComponents(a_close)

plt.imshow(labels==1, cmap='gray')
plt.show()

mask = labels==1
mask = mask.astype(np.uint8)

img_masked = cv2.bitwise_and(img_rgb, img_rgb, mask=mask)

plt.imshow(img_masked, cmap='gray')
plt.show()
```

References

- R.C. Gonzales, R.E. Woods, *Digital Image Processing*. Prentice Hall, 2001
- Morfology – Spletni tutorial: http://homepages.inf.ed.ac.uk/rbf/HIPR2/
- David A. Forsyth, Jean Ponce, Computer Vision: A Modern Approach (2nd Edition), (prva izdaja dostopna na spletu)
- R. Szeliski,Computer Vision: Algorithms and Applications, 2010
- R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, 2004
- Rob Fegus, „Computer Vision“, lectures
- Bastian Leibe „Computer Vision“, lectures
- Kristen Grauman „Computer Vision“, Lectures
- Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.
- Barron, J.T., A Generalization of Otsu’s Method and Minimum Error Thresholding, CVPR2020