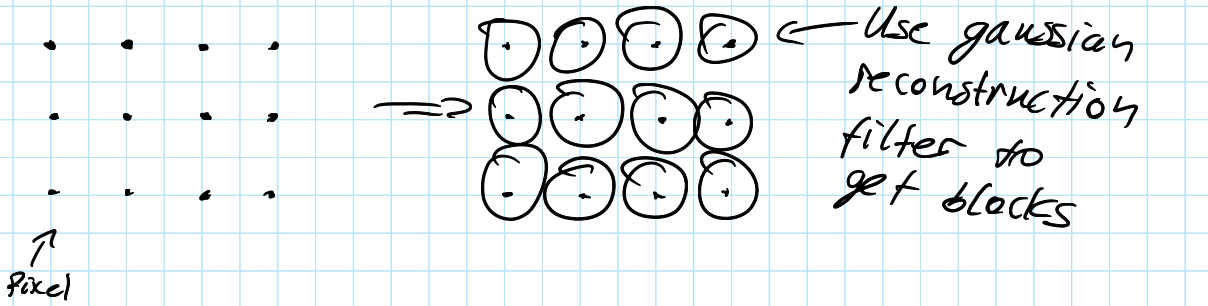
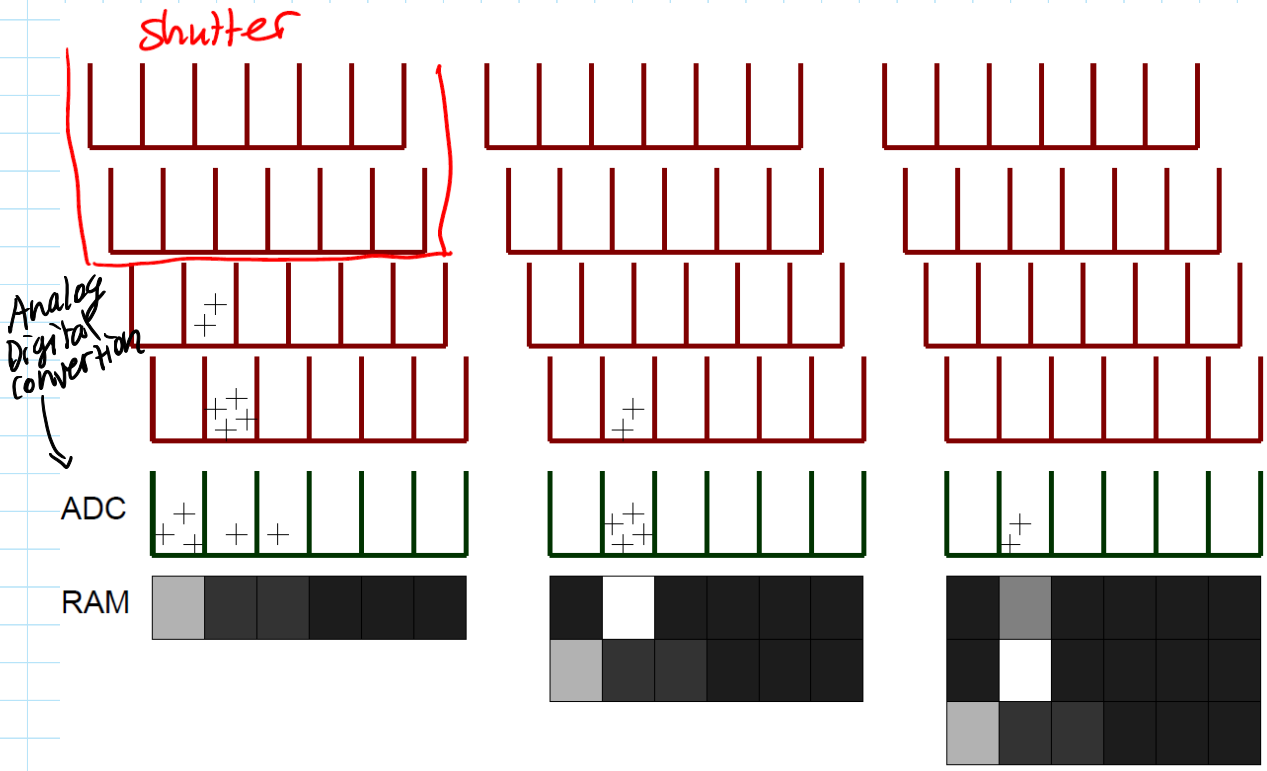
Visual Computing

- Problems with Pictures
  - A lot of light can overwelm the sensor
  - JPEG compression artefacts
  - Sensor noise
  - Bad contrast
- What is an Image
  - Function with (x, y) or (x, y, z) inputs (z: time for video)
  - Enter (x, y) values and get brightness
- Pixel
  - Not a square!
  - A measurement of a real-world continuous function



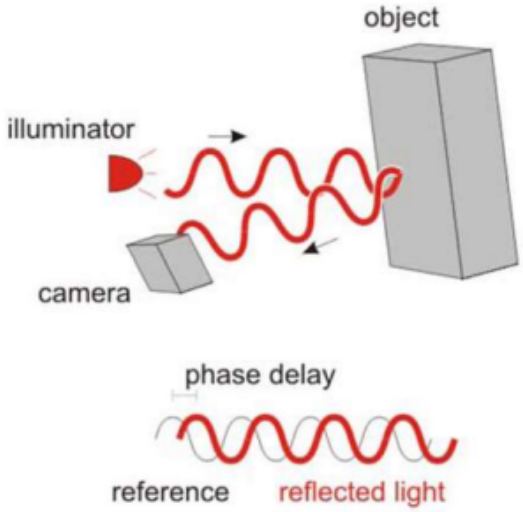
  - Real pixels can be seen in newspaper-print
  - Human has low-pass filter because from a distance we don't see little dots
- Charge Coupled Device (CCD) Sensor
  - Sensor array, each element has an overlay filter for each color
  - Each element is like a water-bucket that contains charge proportional to the right-intensity during exposure



- Convertion happens line by line
- Blooming
  - Each bucket has finite capacity
  - "Ausbrennen einer Stelle"
- Dark Current
  - Each sensor senses some current even if there isn't any because of thermals

- CMOS:
  - Cheaper
  - Better
  - Each photo sensor has own amplifier
  - Video sensor issues
    - Each line has a little offset because of the shutter-line-time-emulation
    - Read one line after the other gives rolling shutter

- Depth sensing cameras



3 sensors

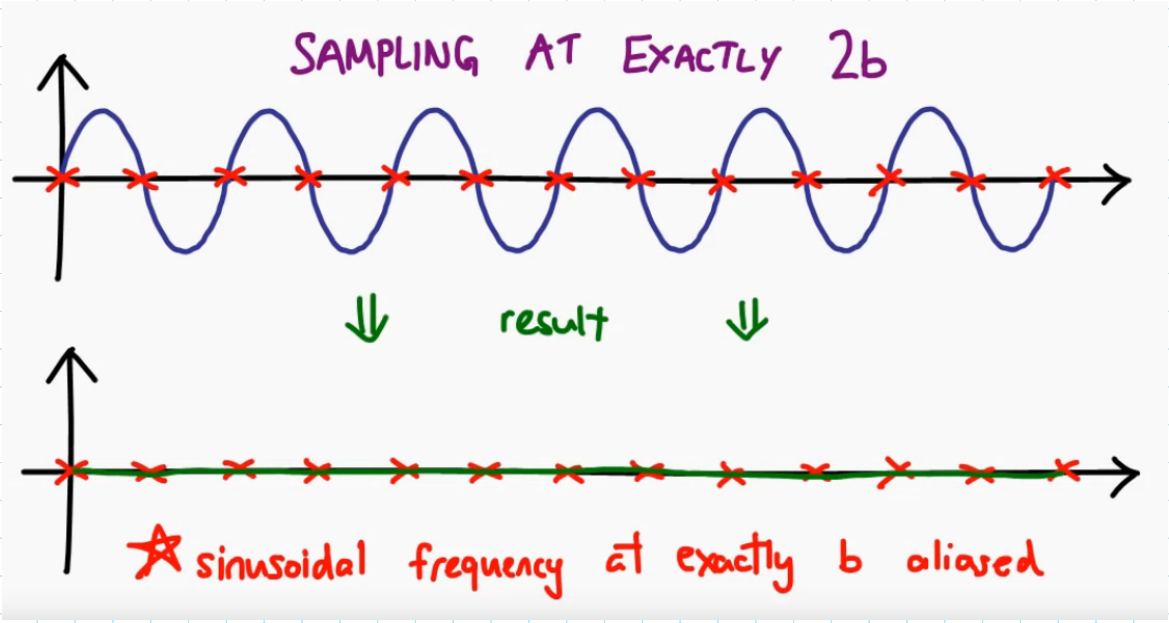
- Prism: High end cameras
- Mosaic: CMOS
- Wheel: scientific

Sampling

- Measure some discrete points of the continous function to interpolate it back, later

Aliasing

- We took too less measurements to represent the given scene
- High frequency signal appearing low frequency after sampling at a too low sampling rate
- Nyquist-Shannon sampling theorem
  - Sampling rate should be more than double maximum frequency you want to retain



- High frequencies aren't sampled well and now show up as low frequencies

Reconstructing continous signal

- Sample signal => continous signal
- Bilinear interpolation: look at four nearest neighboors and do linear interpolation

Quantization

- Represent brightness 0-255
- Lossy
- Grayscale image: 8 bit per pixel : $2^8 = 256$ grayvalues
- RGB: 8 bit/channel : $2^{24} = 16,7M$ colors

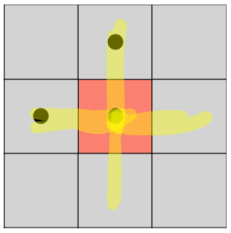
Image properties

- Image resolution: How many pixels
- Geometric resolution: How many pixels per area
- Radiometric resolution: How many bits we spend per pixel

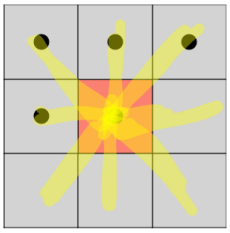
Image Segmentation

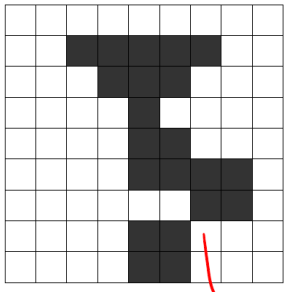
- only works with grayscale-images
- connectivity

4-connectivity

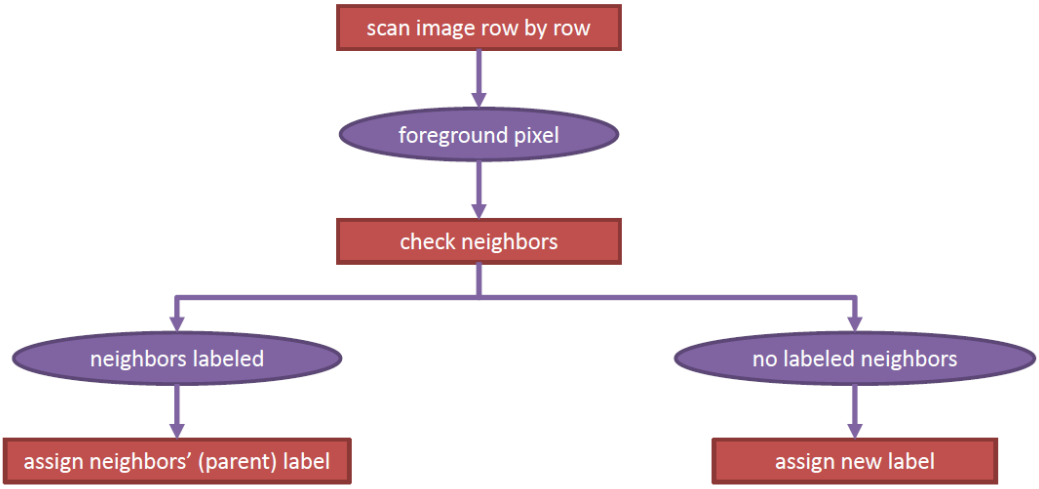


8-connectivity

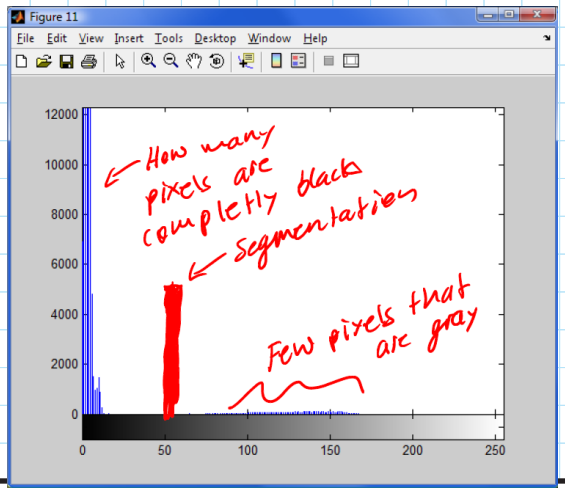


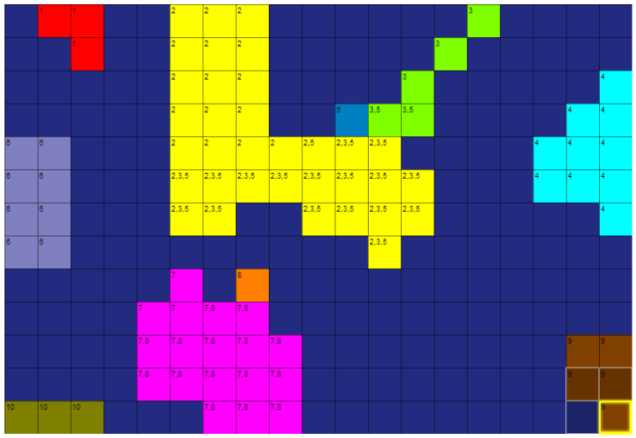


- Connected components raster scanning



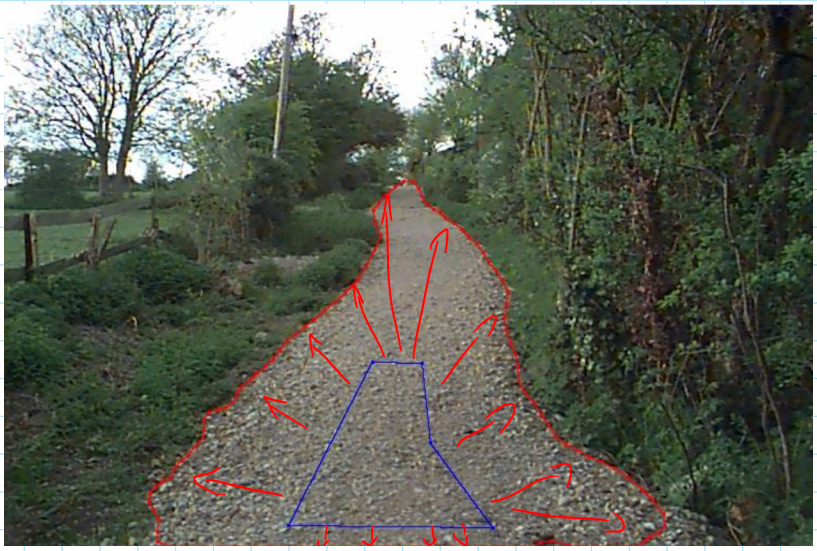
- Second stage: union sets, if they are connected





- Region growing: start from point and add neighboring pixels that satisfy threshold
  - "Zauberstabwerkzeug"
- Thresholding:
  - Produces a binary grid with 1/0
  - Label each pixel in or out
  - $B(x,y) = 1$ if $|I(x,y) - \text{reference}| \geq T$
    0 if $|I(x,y) - \text{reference}| < T$

- Snakes:



  - Don't grow by a point, rather by a polygon

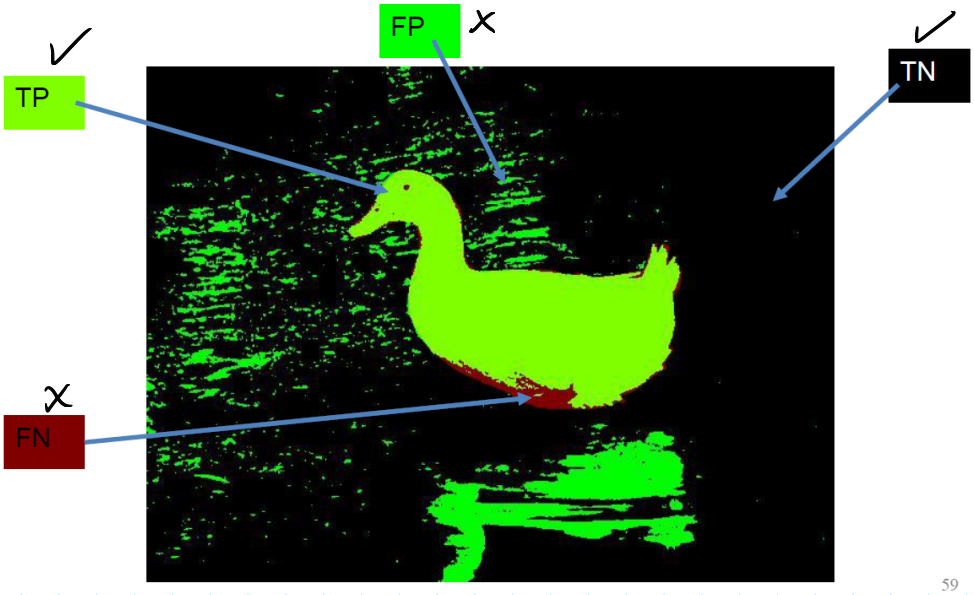
Image Noise

- You have different noise-sources with different distribution => In the limit we have normal-distrib.
- Salt and pepper: Black pixels in bright regions
  white pixels in dark regions
- Signal to noise ratio is an index for quality

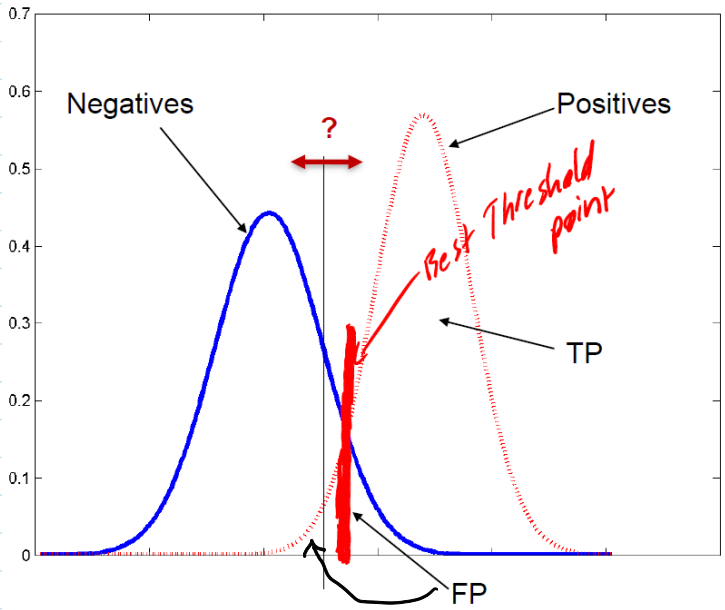
ROC Analysis

- Classification for each pixel

True positive	False negative	} P: Total # positives
True negative	False positive	} N: Total # negatives
Classified:	correctly	incorrectly

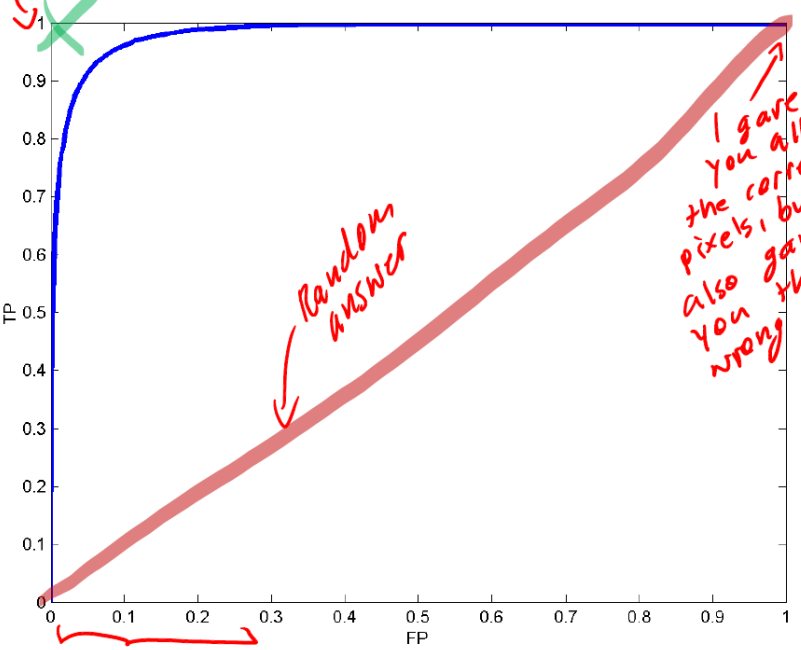


59



The ROC Curve

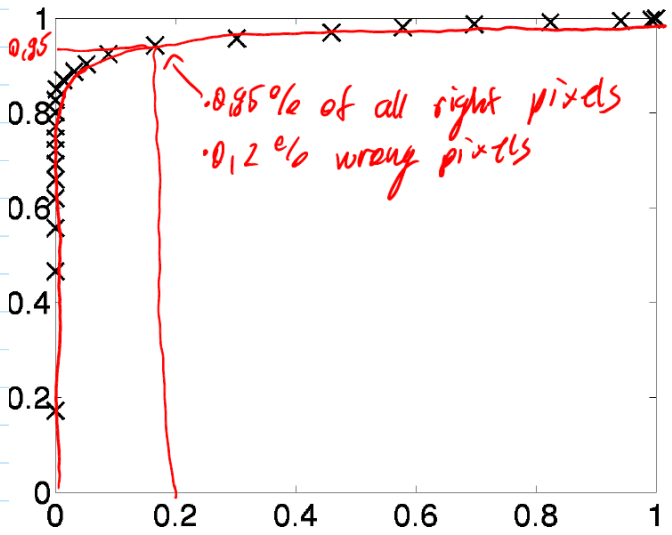
If we get to 1 we have all pixels that are true but introduce some false positives



$$\frac{\text{True positive count}}{P = TP + FN}$$

Conservative high treshold (e.g. 200)

$$\frac{\text{False positive count}}{N = FP + TN}$$

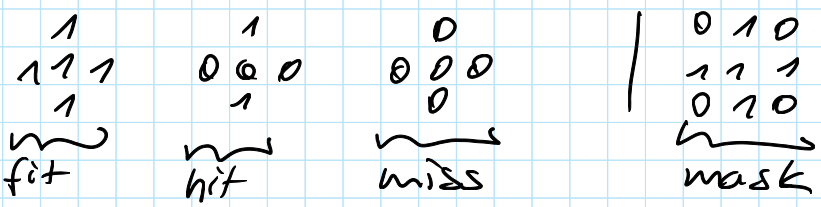


Morphological Operations

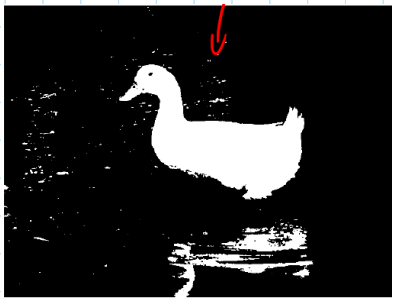
- Local pixel transformations for processing region shapes
- Eight-neighbor erode (= Minkowsky subtraction)
  - Way to shrink your region
  - Smooth out boundaries
  - Irreversible, can't be undone
- Eight-neighbor dilate (= Minkowsky addition)
  - Add pixels to edge

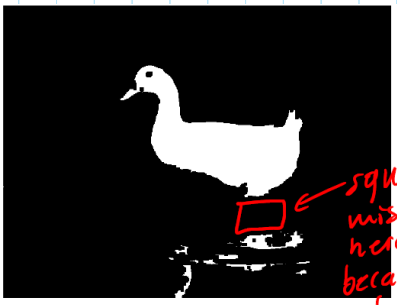
=> Remove noise and artefacts from segmentation

- Operation takes two inputs: - Binary segment. image
  - Structuring element
- Structuring element (= filter array / structure array)
  - Binary grid S
  - S fits I at x: every 1 overlaps with mask
  - S hits I at x: partial overlap
  - S misses I at x: No overlap to filter



- Erosion $\ominus$
  - Erodiert einige Pixel weg, die nicht mit maske übereinstimmen
  - $E(x) = \begin{cases} 1 & \text{if } S \text{ fits at } x \\ 0 & \text{otherwise} \end{cases}$



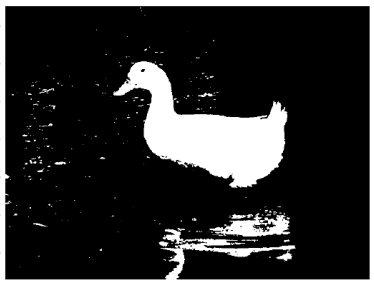


Structuring element

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

68

- Dilation $\oplus$
  - smoothed binien aus
  - Inverse of erosion
  - $D(x) = \begin{cases} 1 & \text{if } S \text{ hits } x \\ 0 & \text{otherwise} \end{cases}$



Structuring element

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

71

- Opening: $I \circ S = (I \ominus S) \oplus S$ (= grow region)
- Closing: $I \bullet S = (I \oplus S) \ominus S$ (= shrink region)

↳ To remove holes in the foreground and islands in the background, repeatedly do opening/closing

Hit and miss transform ⊗

- Search for a specific structure (e.g. corner)

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

⊗

1	0
*	1

r

corner

=

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

- Erosion (=Thinning) and Dilation (=Thickening) can be done via hit-and-miss and specific structuring element
  - Thinning: $I \oslash S = I \setminus (I \otimes S)$
  - Thickening: $I \oplus S = I \cup (I \otimes S)$

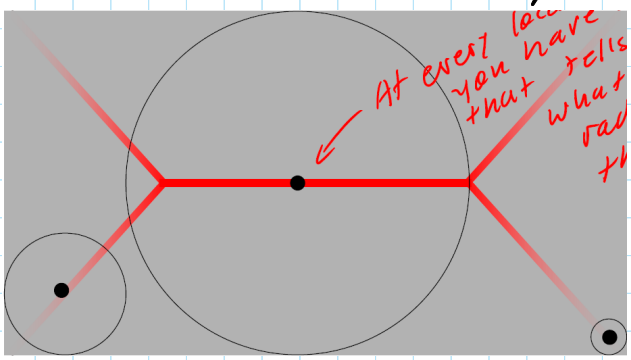
Skeletonization

- Look at the shape via a skeleton
- "Start a grassfire at the boundary of the region"
- The skeleton is the set of points at which two fire fronts meet

---

Medial axis transform

- At every location on the skeleton you have a value that tells you, what radius the circle around that point is



We want to represent gray box with MAT

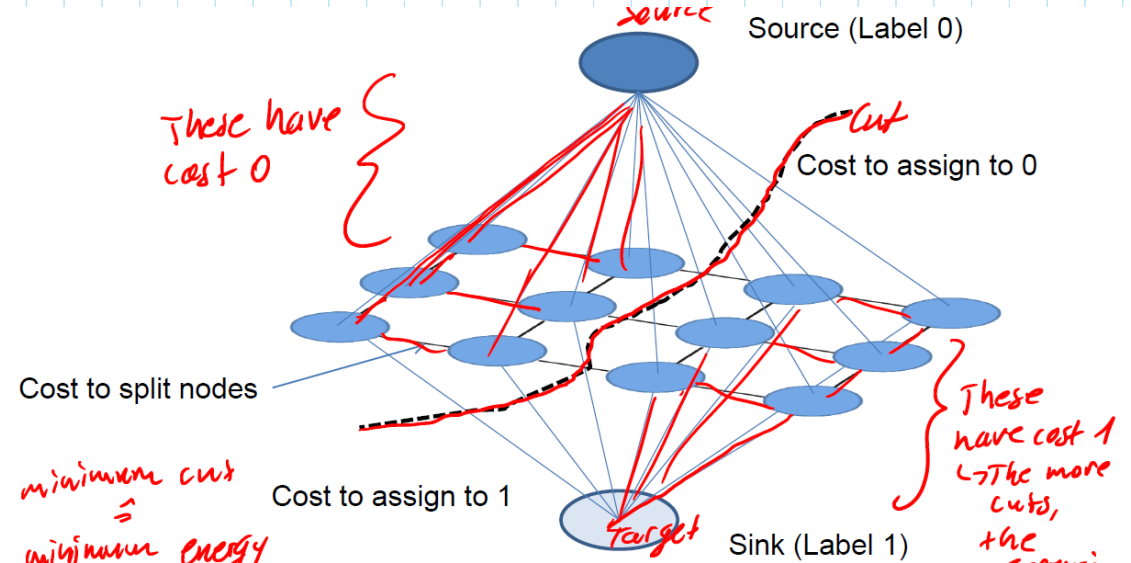
Skeletonization using morphology

- Instead of the radius, we count how many times the structuring element was applied

---

Markov Random fields (=MRF) (=Graph cut)

- Smooth segmentation
- Cost if neighbors have different labels
- Minimize energy with minimum cut



$$Energy(\mathbf{y};\theta,data) = \sum_i \psi_1(y_i;\theta,data) + \sum_{i,j\in edges} \psi_2(y_i,y_j;\theta,data)$$