Basic Image Processing Algorithms

PPKE-ITK, 2016

Lecture 10.

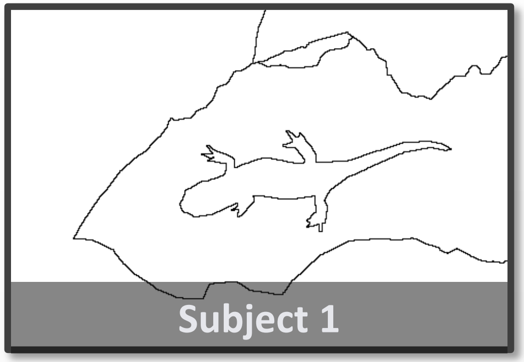
Image and Video Segmentation

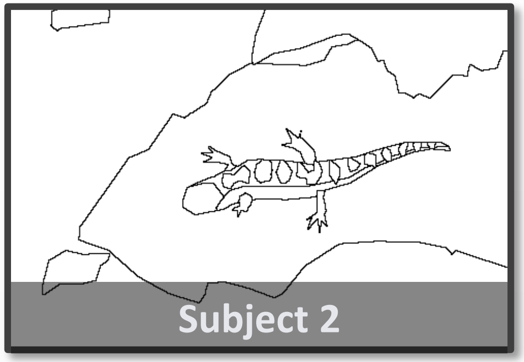
Image Segmentation

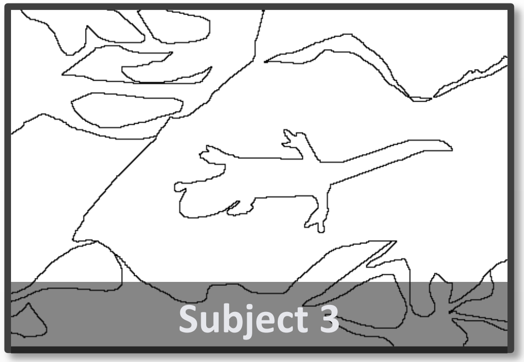
- What is on the image? – This is maybe the most important question we want to answer about an image.
- For a human observer it is a trivial task, for a machine it is still an unsolved problem.
- An important step toward our goal is to segment the image into meaningful parts.
- The objective is to group pixels together based on some common characteristics:
  - they belong to the same physical object
  - they have the same intensity level/color/texture
  - they belong to the background/foreground
  - …

Image Segmentation

- Sometimes even humans cannot agree on a unique solution!







Sample from BSDS500 (Berkeley Segmentation Data Set and Benchmarks 500):
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

Image Segmentation

- The segmentation can be **knowledge-driven** (top-down) or **data-driven** (bottom-up).
- Knowledge driven segmentation methods builds prior knowledge into the segmentation algorithm:
  - Hard to implement
  - Cannot stand alone: need cues from bottom-up segmentation
- Data-driven methods builds on the raw pixel data:
  - they are easier to implement
  - they often fail on real life images
- There is the so-called **semantic gap** between the two approach.
- The complex, high level definitions of top-down methods are hard to embed efficiently into low level algorithms.

Image Segmentation

- Intensity Level Based Segmentation
  - Otsu's Method
- Region-based Segmentation
  - Region growing
  - Region Splitting and Merging
- Clustering in the Feature Space
- Video Segmentation:
  - Foreground/Background Segmentation

Intensity Level Based Segmentation

- Thresholding
  - **Assumption**: the image parts (e.g. object and background) can be separated based on their intensity level.

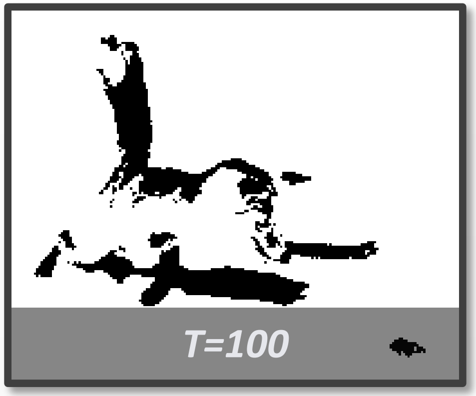
$$s(n_1, n_2) = \begin{cases} \text{object} & x(n_1, n_2) < T \\ \text{background} & x(n_1, n_2) \geq T \end{cases}$$

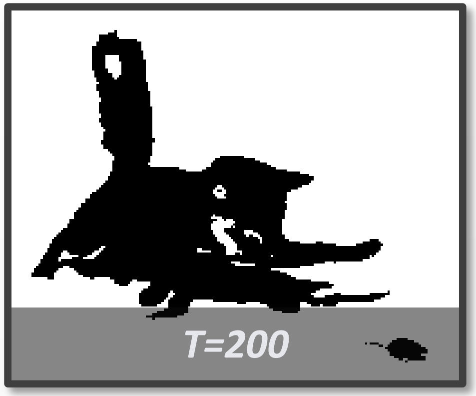
…where $s(n_1,n_2)$ is the cluster of the (n_1,n_2) pixel of the x image and T is a threshold.

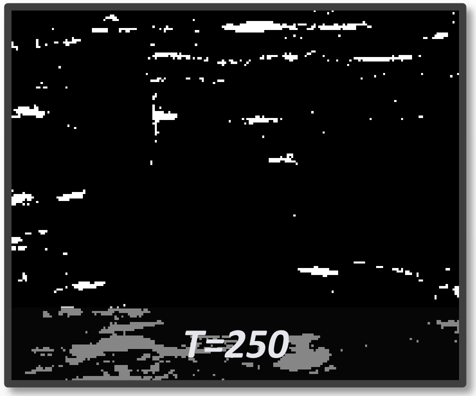
  - The main question is how to determine the threshold?

Intensity Level Based Segmentation

- Thresholding:
  - The main question is how to find the optimal threshold?







Intensity Level Based Segmentation

- Otsu's method:
  - Automatically determines the optimal global threshold by minimizing the intra-class variance.
  - The intra-class variance is defined as follows:

$$\sigma_w^2(k) = \omega_1(k)\sigma_1^2(k) + \omega_2(k)\sigma_2^2(k)$$

where ω_i and σ_i are the probability and the variance of the two classes separated by the threshold k.

  - Otsu showed that ***minimizing the intra-class variance is the same as maximizing inter-class variance***:

$$\sigma_b^2(k) = \sigma^2 - \sigma_w^2(k) = \omega_1(k)\omega_2(k)\left(\mu_1(k) - \mu_2(k)\right)^2$$

where μ_i are the means of the two classes separated by threshold k.

Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys., Man., Cyber.* **9** (1): 62–66.

Intensity Level Based Segmentation

- Otsu's method:

$$\sigma_b^2(k) = \sigma^2 - \sigma_w^2(k) = \omega_1(k)\omega_2(k)\left(\mu_1(k) - \mu_2(k)\right)^2$$

  - To calculate ω_i and μ_i the normalized histogram of the image is used:

$$\omega_1(k) = \sum_{i=0}^{k} p_i \qquad \omega_2(k) = \sum_{i=k+1}^{L-1} p_i$$

$$\mu_1(k) = \left(\sum_{i=0}^{k} i p_i\right) \Big/ \omega_1 \qquad \mu_2(k) = \left(\sum_{i=k+1}^{L-1} i p_i\right) \Big/ \omega_2$$

where p_i is the *i*-th entry in the normalized histogram of the image (probability of the *i*-th intensity level).

The Otsu threshold is the value that maximizes the inter-class variance.

* Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys., Man., Cyber.* **9** (1): 62–66.

Intensity Level Based Segmentation

- Otsu's method:

Region-Based Segmentation Methods

- Let R be the entire image region, and $R_1,\ldots,R_n$ are subregions.
- We want to find a segmentation that is..
  - Complete: $\bigcup_{i=1}^{n} R_i = R$
  - Points in the region R_i (i=1,…,n) are connected
  - The regions are disjoint: $R_i \cap R_j = \phi \quad \text{for } \forall i \neq j$
  - All the pixels in a region has common properties…
  - …that they don't share with pixels from other regions.

Region-Based Segmentation Methods

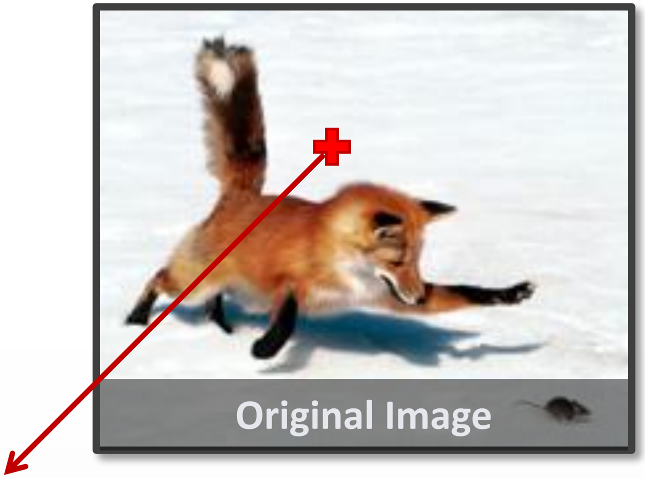
- Region growing:
  - The method is initialized with a set of **seed points** as regions
  - We start growing the regions by adding neighboring pixels to the region if they has **similar predefined properties** as the seed points.

  - The seeds can be selected based on prior information, or evenly, or random...
  - The similarity criteria is usually depending on the segmentation result we want. (Commonly used properties are the intensity level, color, texture, motion,…)

  - **Pros:** simple, works well on images with clear edges, prior knowledge can be easily utilized, robust to noise…
  - **Cons:** time consuming

Region-Based Segmentation Methods

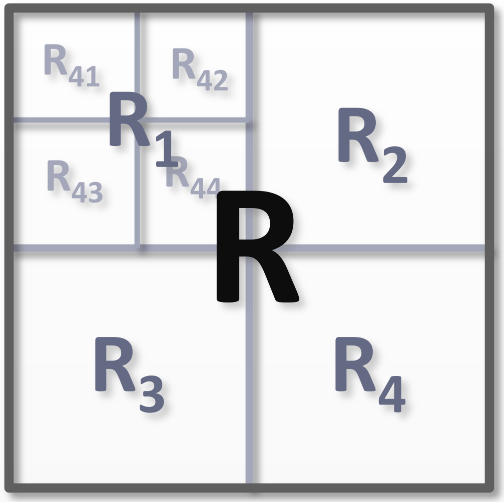
- Region growing:



seed point

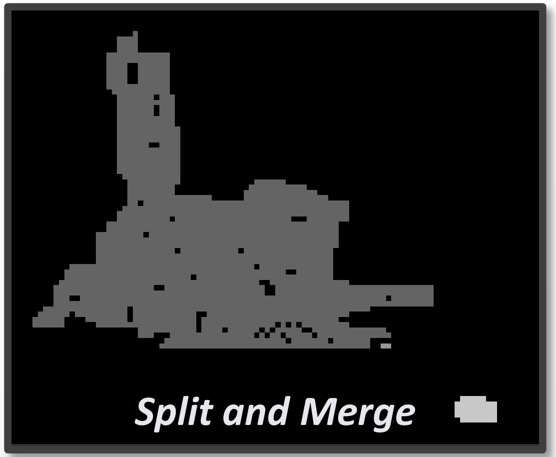
Region-Based Segmentation Methods

- Region splitting and merging:
  - Let ***R*** represent the entire image region and ***P*** be a predicate.
  - The splitting and merging steps are alternating:
    - We split the region R_i into 4 sub regions if $P(R_i)$ = false
    - We merge 2 neighboring regions R_i and R_j if $P(R_i \cup R_j)$ = true
  - The minimum region size has to be selected.



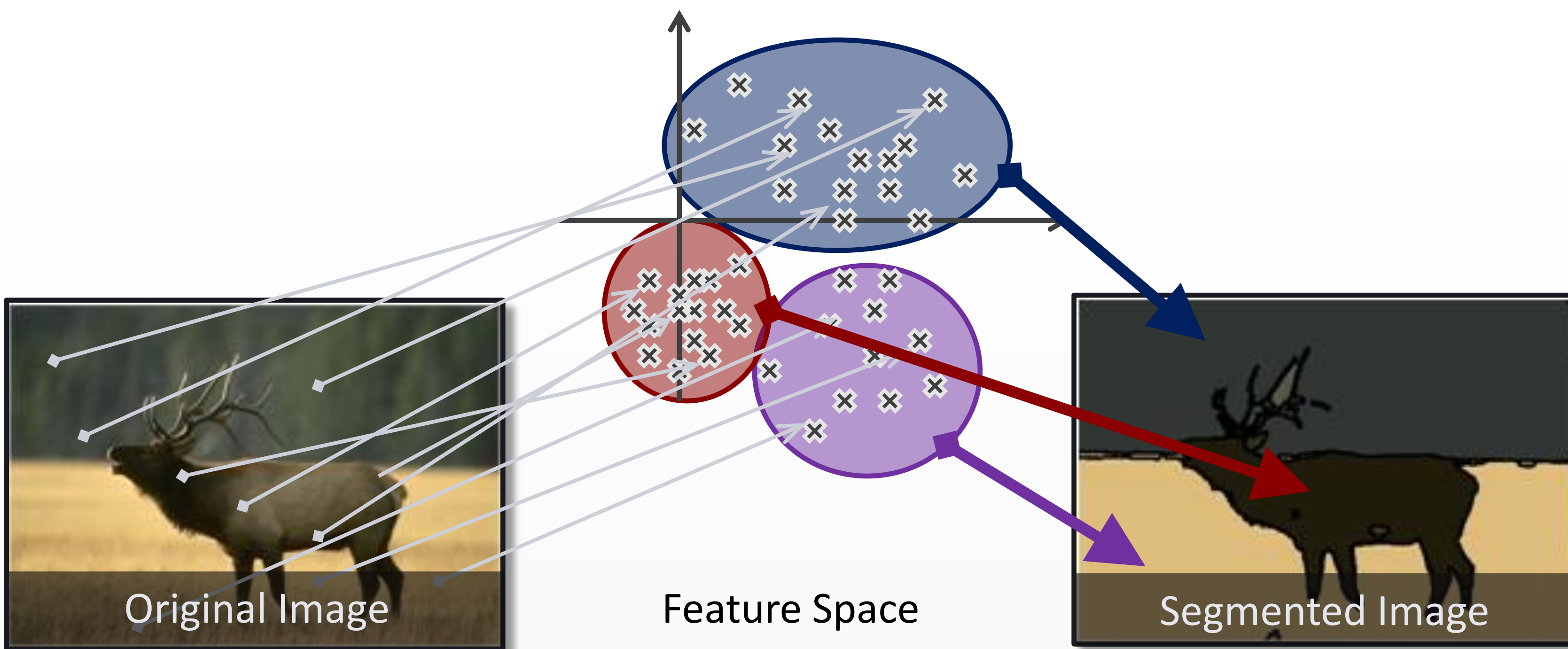
Region-Based Segmentation Methods

- Region splitting and merging:



Clustering in the Feature Space

- A clustering algorithm is used to find structure in the data.
- The pixels are represented in the feature space.
- Usual features: colors, pixel coordinates, texture descriptors,..



Source of the Images: http://ivrgwww.epfl.ch/supplementary_material/RK_CVPR09/

Video Segmentation

Video Segmentation

- In bottom-up segmentation we want to ***group pixels with similar properties together***.
- In case of video segmentation ***motion is an important feature*** to extract object of interest from the irrelevant background.
- It is an important task in…
  - Robotics
  - Object detection and tracking
  - Surveillance
  - Video manipulation
  - Video compression
  - Scene reconstruction
  - …

Video Segmentation: Frame Difference

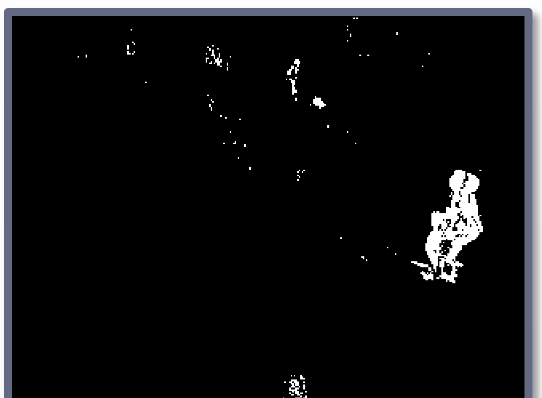
- Frame Difference:
  - We detect motion based on the difference of two consecutive frames:

$$d_{i,j}(x,y)=\begin{cases}1, & \text{if } \left|f(x,y,i)-f(x,y,j)\right|>T\\ 0, & \text{otherwise}\end{cases}$$

where T is a predefined threshold.

Frame ***i***

Frame ***j***



Motion Mask (*T=15*)

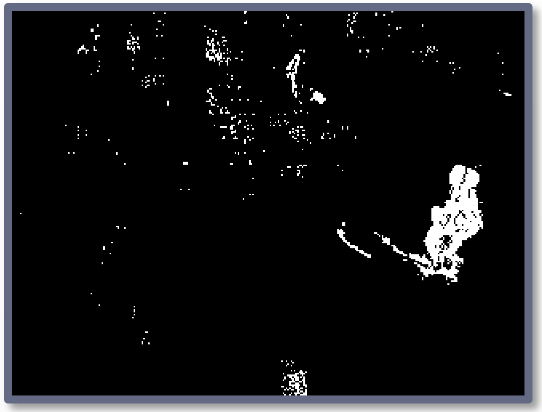
Video Segmentation: Frame Difference

Video Segmentation: Frame Difference

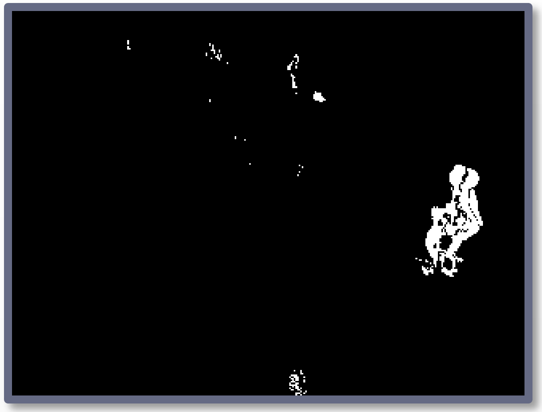
Frame ***i***

Frame ***j***

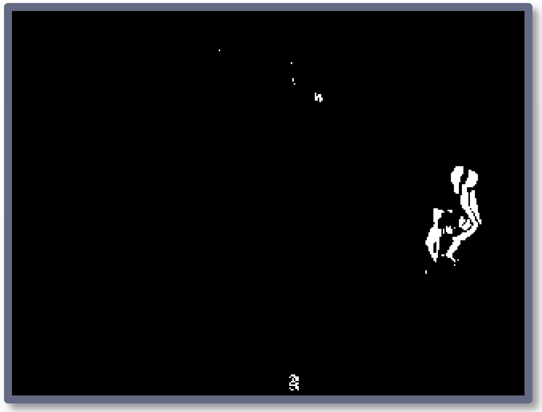
Difference Frame



Motion Mask (*T=10*)



Motion Mask (*T=20*)



Motion Mask (*T=50*)

Video Segmentation: Background Subtraction

- Problems with frame difference:
  - We detect only the contour of the moving object
  - It is sensitive to the speed of the motion

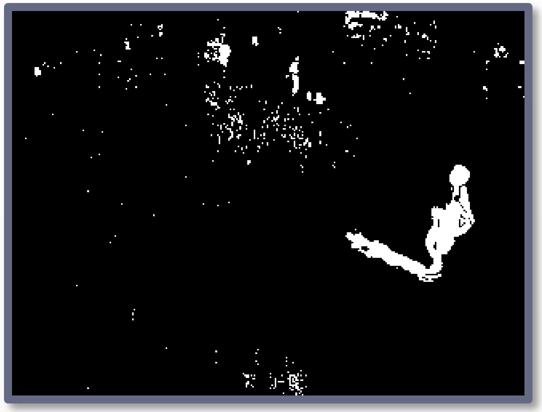
- Background Subtraction:
  - Main steps:
    - Model the background and create an image with only the background on it
    - Subtract the background from the current frame.
    - Threshold the difference frame to get the foreground mask.

Video Segmentation: Background Subtraction

- Simplest solution: use a single frame as background
  - Assumption: we have a frame which does not contain foreground objects.
  - Cannot handle changes of the background (illumination changes, object becomes part of the background, …)
  - Because of the above constraints: Cannot be used in a real life scenario!

Background Frame

Current Frame


Foreground Mask (*T=20*)

Video Segmentation: Background Subtraction

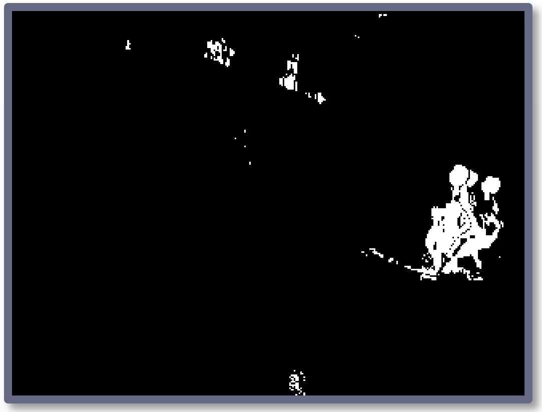
- Running Average:
  - The following equation is used to calculate the background (B) for each time instance:

$$B(x, y, i+1) = \alpha \cdot B(x, y, i) + (1-\alpha) \cdot f(x, y, i)$$

where α is a predefined constant, which defines the adaptivity of the background.

Background Frame (α=0.7)

Current Frame


Foreground Mask (*T=20*)

Video Segmentation: Background Subtraction

- Running Average with α=0.7:

Video Segmentation: Background Subtraction

- Running Average with α=0.98:

Video Segmentation: Background Subtraction

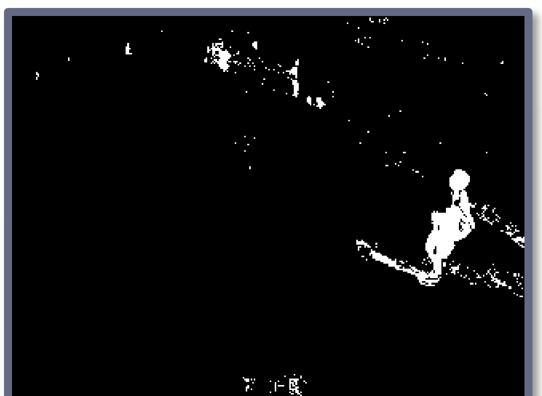
- Running Average with foreground masking:
  - The following equation is used to calculate the background (B) for each time instance:

$$B(x, y, i+1) = \begin{cases} B(x, y, i) & \text{if } (x,y) \text{ is a foreground pixel} \\ \alpha B(x, y, i) + (1-\alpha) f(x, y, i) & \text{otherwise} \end{cases}$$

where α is a predefined constant, which defines the adaptivity of the background.

Background Frame (α=0.7)

Current Frame

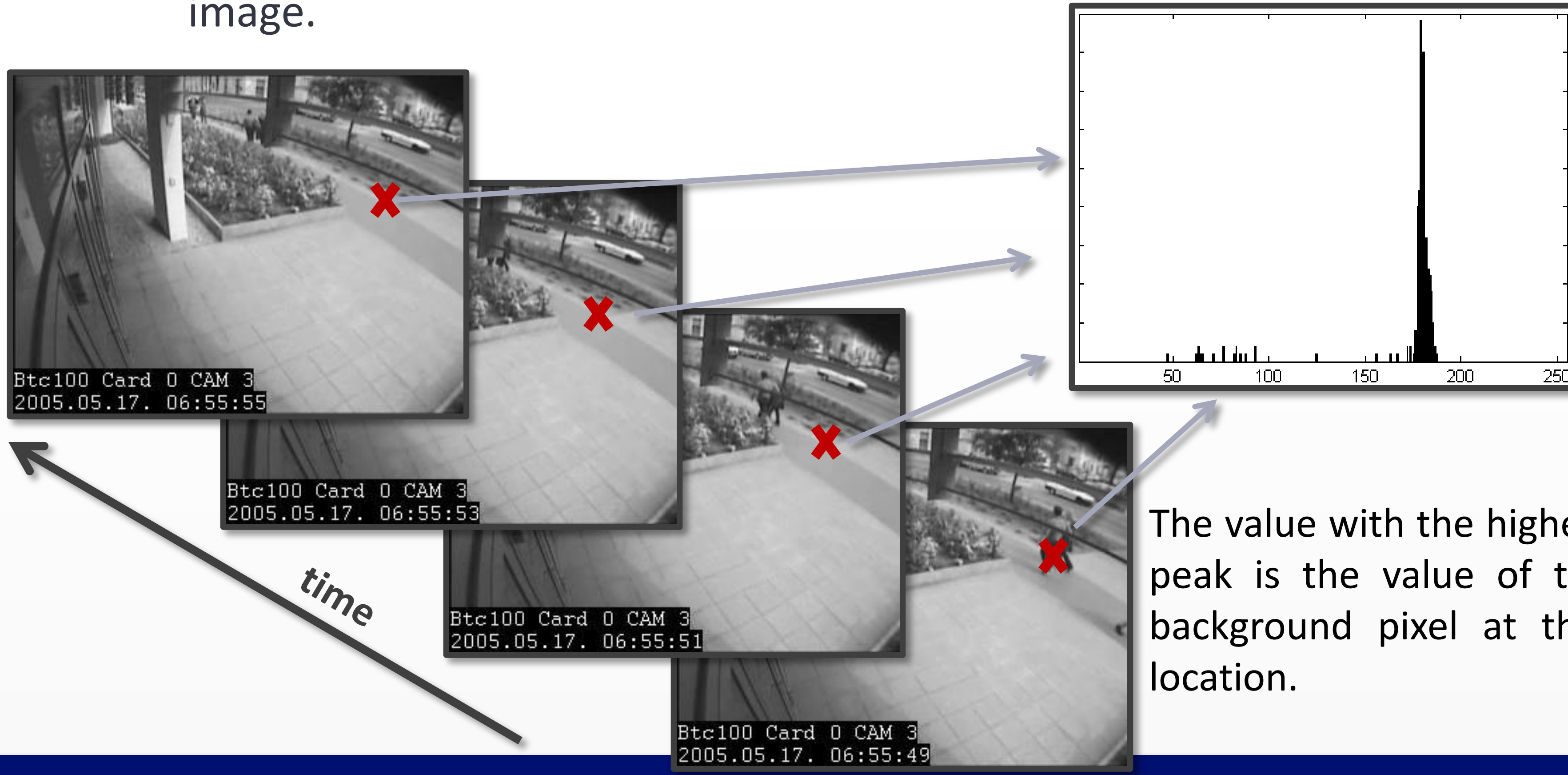

Foreground Mask (*T=20*)

Video Segmentation: Background Subtraction

- Running Average with foreground masking (α=0.7):

Video Segmentation: Background Subtraction

- Statistical background models – Temporal Histogram :
  - The temporal histogram of each pixel is used to generate the background image.



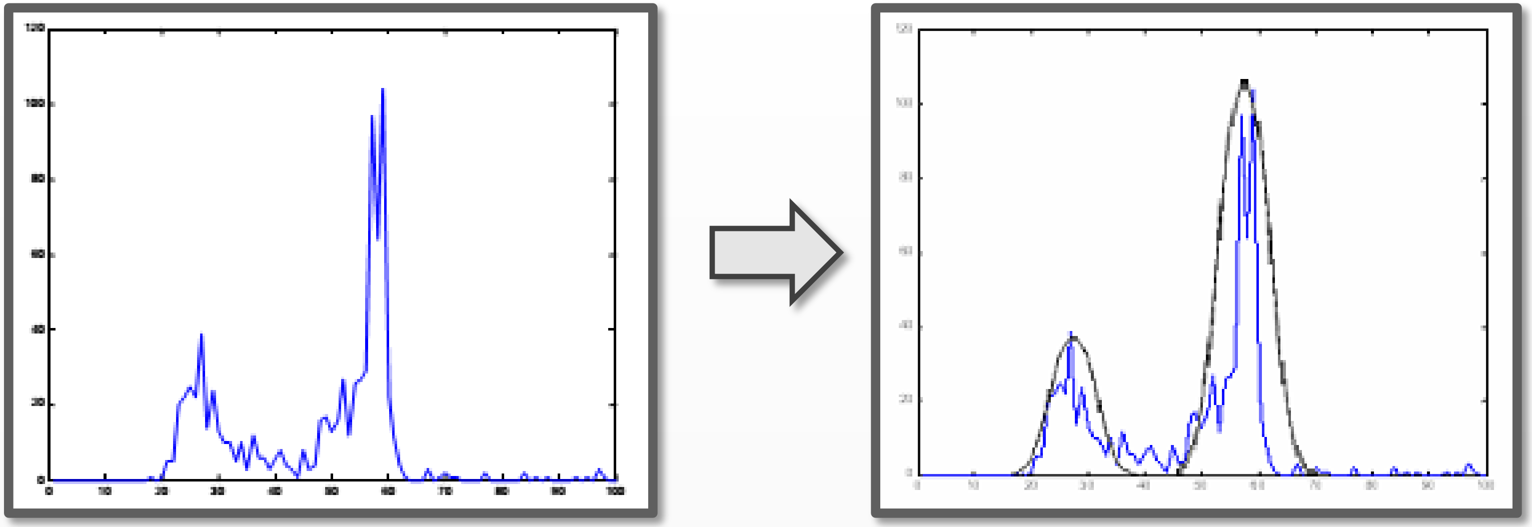
The value with the highest peak is the value of the background pixel at that location.

Video Segmentation: Background Subtraction

- Histogram based background model using the last 50 frames:

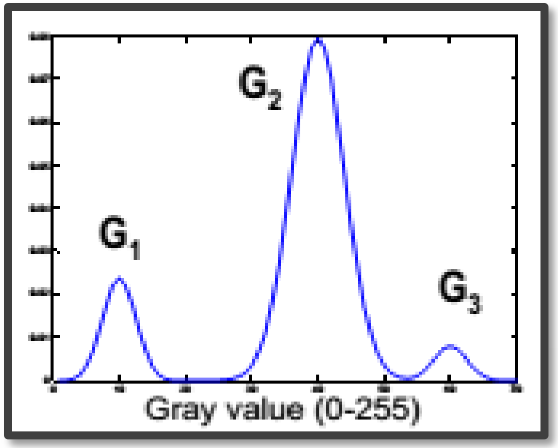
Video Segmentation: Background Subtraction

- Statistical background models – Mixture of Gaussians:
  - Using the temporal histogram of each pixel means we have to store the histograms and has to find its maximum for each pixel for each frame. This means high memory and high computational power requirements!
  - Better idea is to use Mixture of Gaussians to estimate the background.



Video Segmentation: Background Subtraction

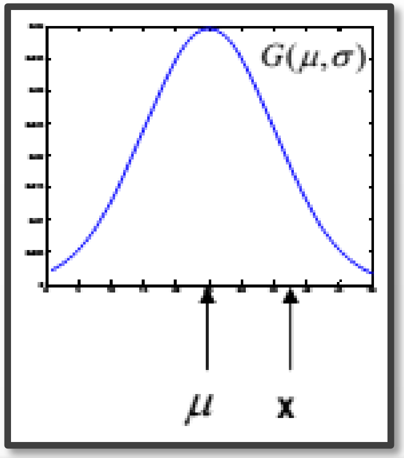
- Statistical background models – Mixture of Gaussians:
  - The main parameters of the model:
    - K: The number of Gaussians
    - The mean(μ_i), the variance(σ_i) and the weight (w_i) of each Gaussian (i=1,..,K)



  - A pixel is considered background, if it belongs to the Gaussian with the highest weight:

$$M(x,G)=\begin{cases}1 & \text{if } \left|\frac{x-\mu}{\sigma}\right| < T \\ 0 & \text{otherwise}\end{cases}$$

where T is a predefined constant.



Video Segmentation: Background Subtraction

- Statistical background models – Mixture of Gaussians:
  - Parameter updating:
    - Weight updating:

$$w_i(t+1) = (1-\alpha)\cdot w_i(t) + \alpha \cdot M(x, G_i)$$

$$w_i = \frac{w_i}{\sum_{j=1\ldots K} w_j}$$

    - Gaussian parameter updating:
      - If there is a matching Gaussian (*M(x, G_i)=1*) we update its parameter as follows:

$$\mu_i(t+1) = (1-\rho)\cdot \mu_i(t) + \rho \cdot x$$

$$\sigma_i^2(t+1) = (1-\rho)\cdot \sigma_i^2(t) + \rho \cdot (x-\mu_i)^2$$

$$0 < \rho < 1$$

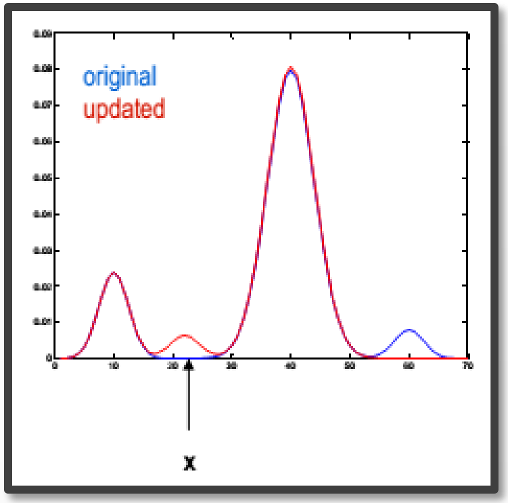
Video Segmentation: Background Subtraction

- Statistical background models – Mixture of Gaussians:
  - Parameter updating:
    - Gaussian parameter updating:
      - If there is no matching Gaussian we replace the Gaussian with the lowest weight with a new Gaussian with the following parameters:

$$\mu_i(t+1) = x$$

$$\sigma_i^2(t+1) = \sigma_{init}^2$$

$$w_i(t+1) = w_{init}$$



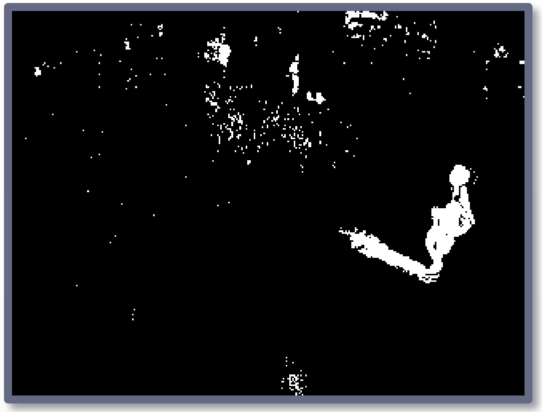
      - After all the Gaussians are updated the weights are normalized.

Video Segmentation: Background Subtraction

- Statistical background models – Mixture of Gaussians:

Background Frame

Current Frame



Foreground Mask (*T=20*)

Video Segmentation: Background Subtraction

- Statistical background models – Mixture of Gaussians:

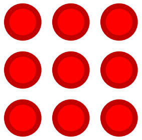
Morphological Operations

- Morphological operations are affecting the form, structure or shape of an object.
- They are used in pre- or postprocessing (filtering, thinning, and pruning) or for getting a representation or description of the shape of objects/regions (boundaries, skeletons convex hulls).
- Two basic operations:
  - **Dilation**: expands the object, fills in small holes and connects disjoint objects.
  - **Erosion**: shrinks objects by removing (eroding) their boundaries.
- The basic idea in binary morphology is to probe an image with a ***structuring element*** (a simple, pre-defined shape), drawing conclusions on how this shape fits or misses the shapes in the image.

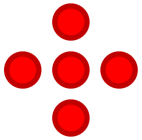
Morphological Operations

- Structuring element:

  e.g.:



8 neighbors



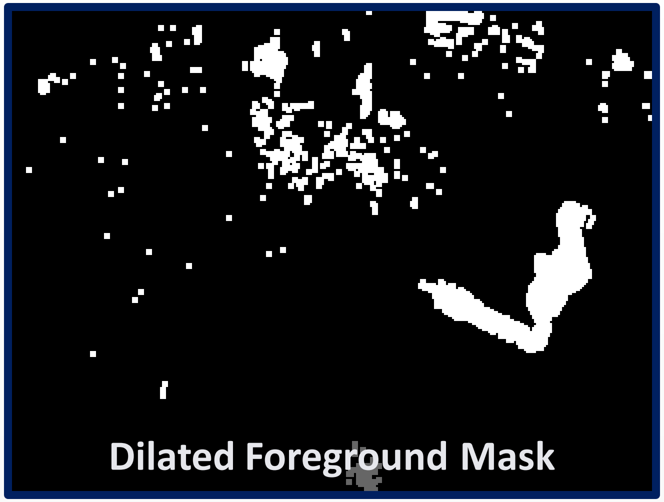
4 neighbors

- Dilation:
  - A shift-invariant operator, that expands the object, fills in small holes and connects disjoint objects.
  - Steps:
    - The structuring element is placed on each pixel on the image
    - If the pixel belongs to the foreground pixel, we do nothing
    - If the pixel belongs to the background, we change it to a foreground pixel if any pixel covered by the structuring element is a foreground pixel.

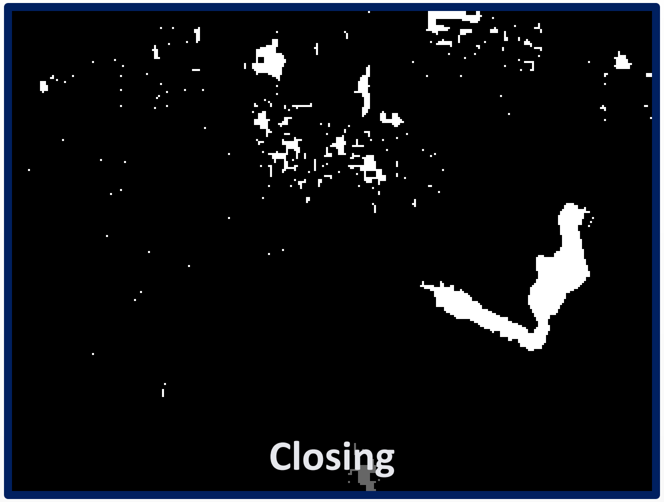
Morphological Operations

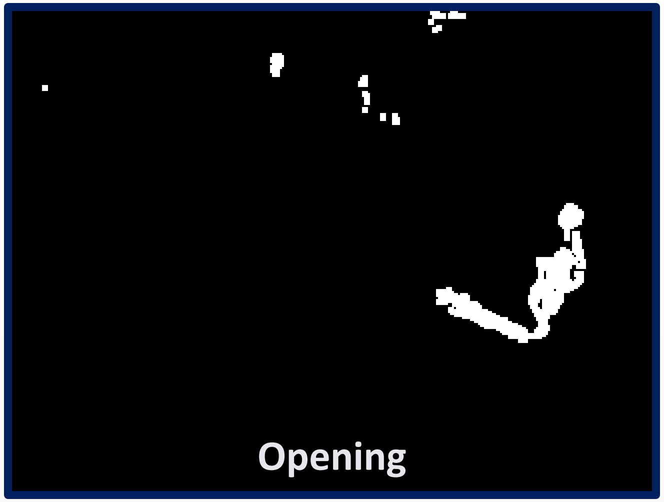
- Erosion:
  - A shift-invariant operator, that erodes away the boundaries of regions of foreground pixels. Thus areas of foreground pixels shrink in size, and holes within those areas become larger.
  - Steps:
    - The structuring element is placed on each pixel on the image
    - If the pixel is a background pixel, we do nothing
    - If the pixel is a foreground pixel, we change this pixel to a background if any pixel covered by the structuring element is a background pixel.
- Erosion on the image has the same effect as dilatation on the inverse image.
- **Opening**: Erosion + Dilation
- **Closing**: Dilation + Erosion

Morphological Operations

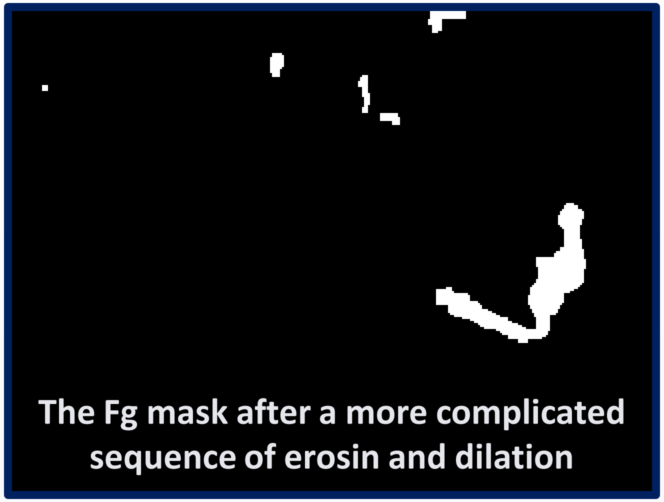


Morphological Operations





Morphological Operations



Main Sources and Further Readings

- Fundamentals of Digital Image and Video Processing lectures by Aggelos K. Katsaggelos
- Image Processing slides of Csaba Benedek: ”Bevezetés és a programozási környezet bemutatása”, 2008
- Yu-Hsiang Wang: „Tutorial: Image Segmentation” (http://disp.ee.ntu.edu.tw/meeting/%E6%98%B1%E7%BF%94/Segmentation%20tutorial.pdf)