Lab 10

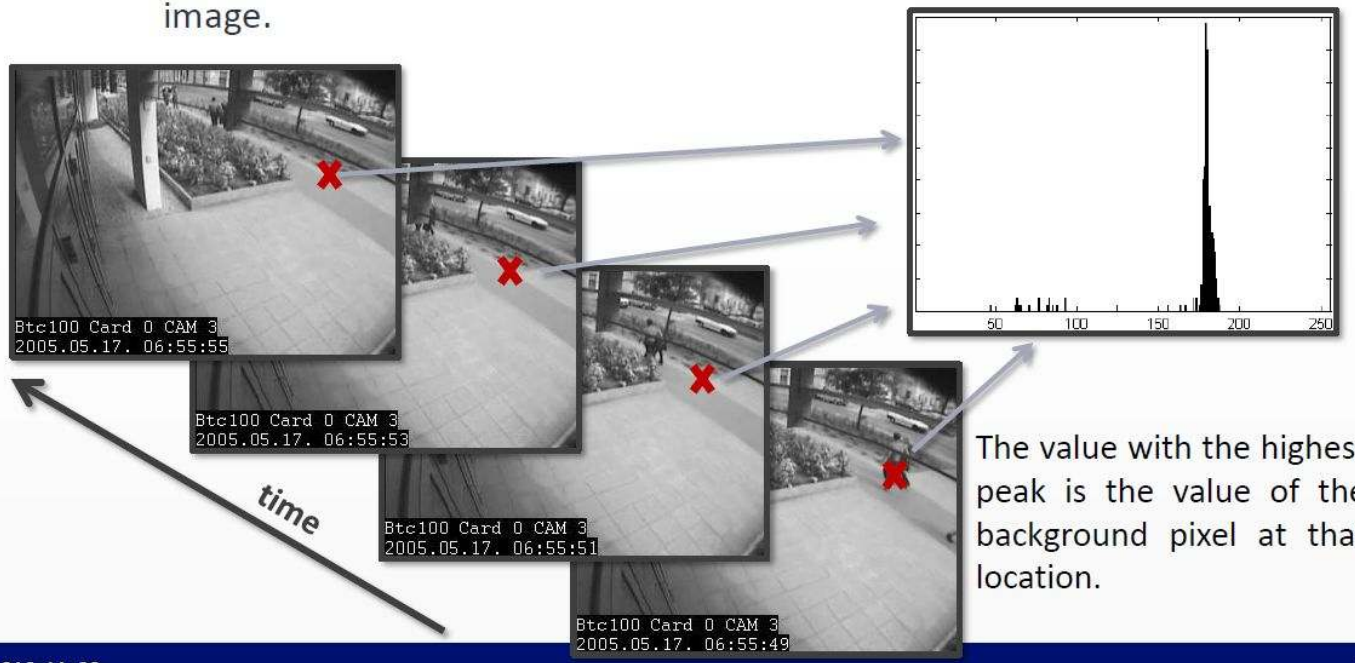
Basic Image Processing Algorithms
Fall 2016

Lab 10: video segmentation with temporal histogram

- deadline (if you do not finish until the end of the lab): 23:59, 30/11/2016;
  please send it to: bipa2016fall@gmail.com
  with the subject: LAB10
  please send only a script

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.

Exercise - create a script

- load the input video file
  - in MATLAB, you will need a **`VideoReader`** object:

    `video_obj = VideoReader('video1.avi');` % under MAC/linux: *video1.ogv* suggested
  - the **`read`** operation on this object returns you the image-sequence as a 4D array (please check the **`size`** of it, which should be equal to *[240x320x3x1001]*)
- open the output video file
  - in MATLAB, you will need a **`VideoWriter`** object:

    `video_out = VideoWriter('video1_your_output.avi');`

    (if you can't see the output later, try to set the format to **`'Uncompressed AVI'`** but be careful, it will result you a huge file!)
  - in the case of your video writer: you can specify the framerate only *before* the opening:

    `video_out.FrameRate = video_obj.FrameRate;`
  - the **`open`** operation on this object opens you the writer
  - do not forget to call close operation on this object *at the end of your script*:

    `close(video_out);`

Exercise - continued

- convert your 4D color image array to a 3D grayscale image array
  you can do it only frame-by-frame
- the way you can create *the next frame* of your output video:
  - update the **`subplot`**s of your existing figure
  - call the **`writeVideo`** method on your output video object, with the current frame as a second parameter:
    ```
    writeVideo(video_out, getframe(gcf));
    ```
  - be careful with the length of the processed video: experiment only with a shorter sequence (eq. length of 10 or 20 frames)

Exercise - continued

- every output frame of your video should contain 5 subplots:
  1. the actual grayscale frame from the input sequence
  2. the statistical background in a predefined time-window (eg. T=100)

     with the built-in **`mode`** function you can compute the *mode* of an array along a specified dimension, eg.

     ```
     temporal_background = mode(grayscale_video(:, :, idx-T:idx-1), 3);
     ```
  3. the difference (positive number) between the *current frame* and the *temporal background*
  4. the b&w version of the difference, computed with a predefined threshold (eg. 50)

     think about the particularly excellent array indexing capabilities of MATLAB, like

     ```
     data_array = [1, 2, 3; 4, 5, 6];
     threshold_value = 3;
     data_array_copy = 1.* data_array;
     data_array_copy(data_array_copy > threshold_value) = 6;
     data_array_copy(data_array_copy <= threshold_value) = 1;
     ```
  5. apply the morphological opening (erosion + dilation) on your threshold image, useful MATLAB commands: **`imerode`**, **`imdilate`**

     both of them will need a structuring element (**`strel`**) as a second parameter (it can be *disk-shaped* with *radius 1*, eg **`strel('disk', 1)`** --- *theory recap on the next slides*

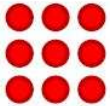
Morphological Operations

- 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

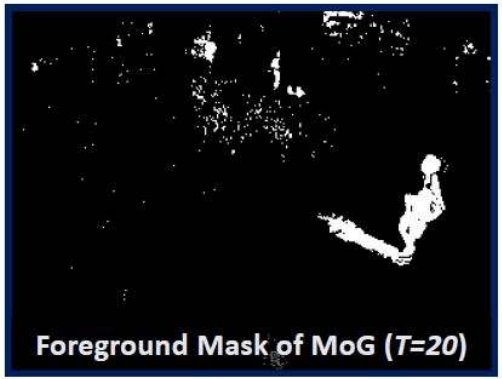
4 neighbors

- 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.

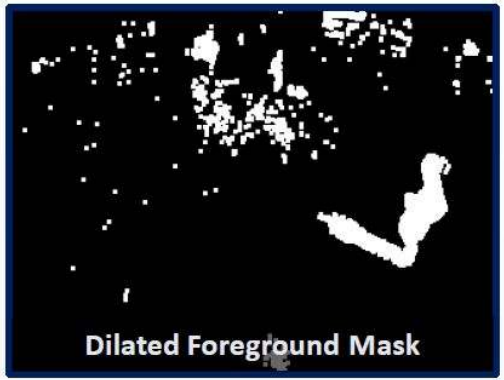
Morphological Operations

- 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



Foreground Mask of MoG (*T=20*)



Dilated Foreground Mask

Eroded Foreground Mask

Morphological Operations