Motion-Based Segmentation

Object segmentation is an important subject in computer vision. It is the basis of numerous applications such as object recognition and event detection. In surveillance videos, the motion of an object can be used to separate the object from its background (which is assumed static). There are several methods to exploit motion, one or two of which you will explore in this exercise. The first one is background subtraction, where you model the background by 'averaging' pixel values over time, so that outliers (passing objects) are removed. By measuring the difference between a frame and the modeled background, changed pixels are assumed an object in motion. A more sophisticated method is based on motion vectors, where you first perform a block-based motion estimation and then use the vector field to cluster areas in a frame that move at the same speed in the same direction. In this exercise, you will start by doing the background subtraction and then compare that to the possibilities of the motion estimation approach.

1 Basic - Background Subtraction

You will evaluate your algorithm(s) using provided test video frames from a surveillance video and the corresponding groundtruth/foreground masks. They are available as separate bitmap files. In Matlab, you can get a file list using *listOfFrames = dir([yourFramePath '*.bmp'])*. Where yourFramePath will be something like './GroundtruthSeq/RawImages/'. Next, you can iteratively read the images using *imread([yourFramePath listOfFrames(1).name])*;.

Starting with the basic background subtraction approach, implement the following tasks:

1. First, write a temporal median filter to generate the background image. (Hint: you can process the R, G and B color planes separately.)
2. Then, perform the actual foreground segmentation for each frame:
   (a) Compute a difference metric between the input frame and the background model.
   (b) Segment the foreground by setting a threshold on that metric. Compare two different thresholding approaches: a single-pixel test and a multi-pixel test (in the multi-pixel test, consider the pixel as background only when all pixels in a small neighborhood appear to be

background, as described in the course slides). (Hint: note that you need to evaluate different thresholds for question 3, so it is convenient to store the real-valued differences with the background model per frame in a big matrix for each method.)

(c) It is important to evaluate your performance quantitatively and objectively. To this extend, compare the results obtained by each method with the provided groundtruth data; measure how well they match. (Hint: note that there are a lot of different metrics for such an evaluation (Accuracy, Precision, Sensitivity, Matthew's Correlation Coefficient, etc. Think about which is most applicable in this context.)

3. Change your threshold that distinguishes the background label and foreground label, and draw Receiver Operating Characteristic (ROC) curves to compare your different approaches.

2 Evaluation

For this project you must write a short report (6 pages single column maximum) preferably in LaTeX or in other word processing software such as Microsoft Word addressing at least the following points:

2.1 Basic

- How did you implement each step?
- Comment on the qualitative results of your foreground segmentation algorithm
- Indicate which of your implemented foreground measures performs better and why it does..
- Discuss what quantitative metric you use to compare your foreground segmentation with the groundtruth. Why is this one the most appropriate? What can you conclude from your ROC plots?
- In which situations would your algorithm definitely fail? How do you think your current segmentation pipeline could be improved or extended?
- Provide the source code. This can be delivered electronically to t.alkanat@tue.nl.

Additionally, a brief demonstration of your code is necessary for evaluation. During this demonstration you will run your code live while showing some intermediate results and explaining them.