INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

ÁREA DEPARTAMENTAL DE ENGENHARIA DE ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

ELECTRONIC, TELECOMMUNICATIONS AND COMPUTER ENGINEERING DEPARTMENT

INFORMATICS AND MULTIMEDIA ENGINEERING

Image Processing

2nd Laboratory Project – Motion Detection for Object Tracking

1. Goal

a. Develop a computer vision application able to detect, classify and track objects that appear in a video sequence typical from surveillance environment;

b. Familiarization with the OpenCV (Open Source Computer Vision) library to develop real-time computer vision applications (for the Python programming language).

2. Description

a. It is intended to develop an algorithm, useful to integrate an automatic surveillance system, capable of detecting active regions (areas of movement) in a sequence of images. The objects of interest are people and cars. The detected regions should be classified, instantly, into three classes: PEOPLE, CAR, OTHER.

b. The active regions detected in consecutive images should be consistently matched to compute the overall object trajectories (only the regions of the same class should be matched each other);

c. The code should be developed in Python/OpenCV and encapsulated in a single function with the following interface: *outVideo* = ***MotionDetection****(inVideo, firstFrame, lastFrame)*, where *inVideo* is a string with a file name containing the input video stream, *firstFrame* and *lastFrame* are the indices of the first and last image to be processed, respectively, and *outVideo* is a string with the file name of the output video containing the results of the algorithm. The detected regions should be marked with a colored bounding box according to the results of the classification procedure (for example, PEOPLE – green, CAR – blue, OTHER - red) and associated with an identifier (number), corresponding to the detected object (figure 1 shows an example).

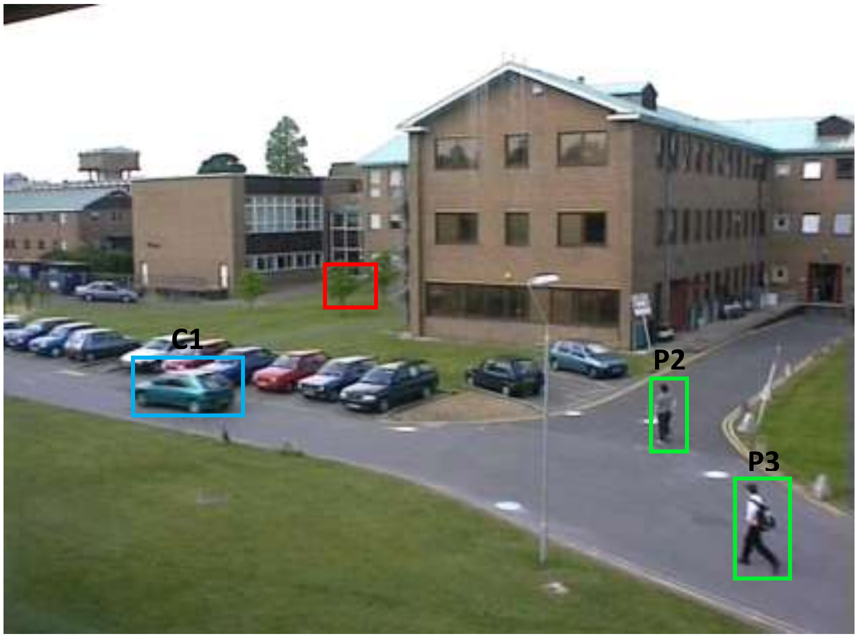

Figure 1

A typical sequence of tasks and related operations

1. Estimation of background image (Hint: use the temporal median filter);
2. Detection of active pixels;
3. Application of morphological operators;
4. Detection of active regions;
5. Extraction of region properties;
6. Classification of the detected regions;
7. Matching regions detected in consecutive frames;