Development of a computer vision system aimed at intrusion detection

Students should develop a software system that, based on automatic video analysis, can detect objects (*intruders*) that do not belong to a static reference scene (*background*) and establish which of such objects are persons.

• **Characteristics of the video sequence**

• *Technical characteristics*: 12 frame/s (total duration is about 41 *secs*), 320x240 pixels, 8 bit/pixel (256 gray levels). The sequence is compressed by the Radius Cinepak CODEC.

• *Semantic characteristics*: a person (hereinafter also "subject") moves continuously (i.e., there exists no time interval showing the background only) within an indoor environment (i.e., an office) according to a wide range of velocities (from almost still to relatively fast). For quite a long time the subject is shown in close-up view, so that he covers a large portion of the entire video frame. The first part of the sequence shows the subject moving within the environment while lighting changes slowly (darkening). In the second part, the subject first removes an object from the scene then abandons another object which was not part of the scene previously.

Therefore, the blobs achievable by labeling the output of a change detection algorithm may correspond to either a person or not. Accordingly, each blob must be properly classified as "person" or "other object". This simple classification task may be accomplished based on basic blob features, such as e.g. those related to the size of the blob.

<u>First task (*mandatory*)</u>

• **Functional specifications**

Based on a change detection algorithm and additional suitable processing steps, the system should provide the following outputs:

• *Graphical Output*: in each frame (but, possibly, those deployed for initialization purposes) the system should show either the labeled blobs (as illustrated in the left picture below) or the labeled contours (right picture below) corresponding to detected objects.

• *Text Output*: the system should create an output text file reporting, for each frame, the number of detected objects, the values of the associated blob features and the classification into either "person" or "other". The table below shows an example of how such a kind of output might be organized.

frame index — *number of detected objects*

300	3			
1	3582	310	…	person
2	785	152	…	other
3	628	126	…	other

object identifier — *Area* — *Perimeter* — *other blob feaures* — *classification*

Students willing to implement different ways to deliver the outputs may contact the teacher or the tutors of the course to discuss their proposals.

• Suggestions

Based on the material presented in the "Change Datection" lecture and following visual analysis of the video sequence, the students may implement the change detection algorithm deemed more suitable to solve the problem. Once the algorithm has been selected and implemented, the parameters, such as the detection threshold, T, and the adaptation rate, α, may be tuned so to obtain the best possible output. Moreover, suitable binary morphology operators may be applied to improve the change detection output. A possible sequence of operations to improve the change detection output is illustrated below;

- Area-opening (i.e. deletion of small-size blobs);
- closing (by a suitable structuring element);
- opening (by a suitable structuring element);

Should a background subtraction approach be applied, the background image should be generated using as few initial frames as possible, so to maximize the number of frames where the actual intrusion detection process may be carried out. Students may choose whether to apply a

"blind" or "selective" background updating method. Should the latter option be preferred, for the purpose of background updating it is advisable to rely on a morphologically improved (and perhaps further dilated) mask rather than on the "raw" change detection output.

Second task (optional)

In the second part of the sequence, after the subject removes one object and abandons another object (more precisely after he separates himself from the abandoned object) the change detection algorithm would find 3 blobs, which, according to the specifications of the first task, will be classified as one person and two "other" objects. However, it is clear that of the two blobs classified as "other", only one is indeed a "true" object, namely that abandoned by the subject, whilst the other corresponds to a "false" object detected by the algorithm due to the removal of one "true" object by the subject. Students should develop a suitable algorithm to distinguish between such "true" and "false" objects.

- **Suggestion**

  Reason about the contour of a blob found by the change detection algorithm: in case of "true" objects this contour .....