GUI and Integration Development for an Activity Recognition and Object Tracking Video Processing Toolbox

Author: Afonso Teodoro

Supervisors: João Miranda Lemos
            Jorge Salvador Marques

This work has been performed in the framework of project:
ARGUS – Activity Recognition and Object Tracking Based on Multiple Models,
supported by FCT under contract: PTDC/EEA-CRO/098550/2008.

July 4, 2013
Contents

1 Objectives 2

2 Functionalities 3
  2.1 Module Description .................................................. 3
    2.1.1 Video Acquisition .............................................. 3
    2.1.2 Background Model Estimation .................................. 3
    2.1.3 Active Region Detection ....................................... 4
    2.1.4 Active Region Tracking ......................................... 4
    2.1.5 Velocity Fields and Switching Matrices Estimation ............. 5
    2.1.6 Anomalous Trajectory Detection .................................. 7
  2.2 System Outputs ......................................................... 7

3 Tests 8

4 Conclusions 9
1 Objectives

The main goal of this work is the development of a functional Graphical User Interface (GUI) based on the activity recognition model developed in the scope of the ARGUS project. The interface allows the user to learn, from a video sequence, a model for the human behaviour based on multiple velocity fields and then use the model to automatically identifying anomalous trajectories in a given scenario. In order to achieve this, several modules had to be implemented and/or put together. A brief description of each module is presented in the section below.

Figure 1: Graphical User Interface.
2 Functionalities

The decision process can be divided in two steps: training and testing. Both of them have some aspects in common such as reading the input image sequence and detecting and tracking active regions. The input data is, however, different in each step. For training one should use a long video sequence or a set of video sequences with a significant number of trajectories so that there is enough data to estimate a set of velocity fields and switching matrices. These velocity fields and switching matrices explain the typical pedestrian motion patterns in the scene and are used in the testing step to evaluate the likelihood of a new set of trajectories.

2.1 Module Description

In this section, a brief description of the implemented modules is provided. Note that the inputs and outputs mentioned below are indicative of the steps taken to reach the objective. Additional arguments were used to help the software implementation.

2.1.1 Video Acquisition

The information about the scenario is provided by a video sequence. Up to date, the software only supports video input from the hard drive, either in *.avi or *.mat formats. The former is faster to start the processing, since only one frame is read at a time, but is less fluid, in the display, than the latter.

2.1.2 Background Model Estimation

This step is made by selecting a number of samples from the video sequence, spaced in time, and computing the median of those samples. For simplicity, the model was considered to be static over all the video although this solution is not very adequate, mainly due to illumination changes. This module receives, as input, the image sequence and returns a single image with the background model (figure 2).

Figure 2: Estimated Background Model.
2.1.3 Active Region Detection

After estimating a model for the background, the active regions are detected by computing the difference between the current video frame and the background and applying a threshold. Afterwards, to remove spurious regions, morphological filtering techniques are employed (figure 3). The detected active regions are characterized by the coordinates of their centroids. The inputs to this module are the current video frame and the background model and the output is a matrix with the centroids’ coordinates.

![Figure 3: Active Region Detection.](image)

(a) Binary image after threshold.  
(b) Intermediate step.  
(c) Binary image after filtering.

2.1.4 Active Region Tracking

Having determined the coordinates of the centroids in frame $t$ it is necessary to match them with the centroids in frame $t - 1$. This is done using the Hungarian method ([1] [2]). This block receives the coordinates from both frames and a structure with all the trajectories up to frame $t - 1$ and returns that same structure updated with the new centroids. After matching the centroids the trajectories are plotted over the current frame (figure 4).
Figure 4: Active Region Tracking Sequence.

2.1.5 Velocity Fields and Switching Matrices Estimation

Afterwards, the detected trajectories are filtered (low-pass), making the evolution of the centroids’ coordinates along each trajectory smoother, and used to estimate the velocity fields and switching matrices. The estimation is done using the Expectation-Maximization algorithm. This module receives all the filtered trajectories and returns a set of parameters that characterize the velocity fields and switching matrices (for more details see [3] [4]). The results are illustrated in figures 5 and 6.
Figure 5: Estimated velocity fields from detected trajectories.

Figure 6: Switching Matrices.
2.1.6 Anomalous Trajectory Detection

The last step is computing the likelihood of each new trajectory being explained by the estimated velocity fields and switching matrices. The Forward algorithm, [5], was used here. If the computed likelihood is below a given threshold the trajectory is considered anomalous. Sometimes the computation of the likelihood leads to underflow problems. In these situations, the sum of the squared error is used instead. This evaluation is performed every $k$ frames and over the last $K$ frames, both defined by the user. The results for a test video sequence are illustrated below (figures 7 and 8).

![Figure 7: Anomalous (red); Regular (green); n/a (yellow).](image1)

![Figure 8: Test trajectories.](image2)

2.2 System Outputs

Besides the graphical output, the system also saves some data to be reused either on the software described here or on any other, provided that additional processing may be needed to convert the data to the desired format. The following content is stored:

- **background.mat** - RGB image with the estimated background model.
- **trajectories.mat** - Includes a cell array with the detected trajectories (filtered) where each cell is a 2D array with a sequence of centroid coordinates. This file also includes the height ($h$) and width ($v$) of the processed video.
- **fields.mat** - Includes the parameters (number of fields, field model, grid size), the estimated velocity fields and switching matrices, as well as the normalized trajectories used in the estimation.
- **trajectories_warp.mat** - Cell array with the detected trajectories after applying the homography.
- **homography.mat** - Includes the necessary parameters to compute the homography.
3 Tests

To test the implemented software, several image sequences from the IST campus were processed. From the previously described modules, the one responsible for the detection of the active regions seems to be the weakest link, due to its simplicity. Most of the trajectories were detected although some outliers are also present and breaks in the trajectories are common namely because of shadows. These two factors may lead to a poor estimation of the velocity fields. Another difficulty concerns setting the threshold value for the detection of the anomalous trajectories. Experimentally, this method leads to a solution that has either several false alarms or several misses. Nevertheless, the implemented software performs reasonably well for the tested scenarios and serves as a proof of concept for a real-world application.

In terms of computational time, using the profile tool from MATLAB, we get the following results for two example video sequences:

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Dimensions</td>
<td>720x250</td>
<td>720x250</td>
</tr>
<tr>
<td>Number of Frames</td>
<td>~ 3400</td>
<td>~ 300</td>
</tr>
<tr>
<td>Background Estimation</td>
<td>~ 3 s</td>
<td>~ 3 s</td>
</tr>
<tr>
<td>Read from file</td>
<td>~ 0.2 s/frame</td>
<td>~ 0.16 s/frame</td>
</tr>
<tr>
<td>Detection</td>
<td>~ 0.03 s/frame</td>
<td>~ 0.03 s/frame</td>
</tr>
<tr>
<td>Tracking</td>
<td>~ 0.005 s/frame</td>
<td>~ 0.008 s/frame</td>
</tr>
<tr>
<td>Display</td>
<td>~ 0.06 s/frame</td>
<td>~ 0.07 s/frame</td>
</tr>
<tr>
<td>Number of fields</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Grid Size</td>
<td>11x11</td>
<td>-</td>
</tr>
<tr>
<td>Field Estimation</td>
<td>~ 18 s/iter</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Computational time spent.

Note: The elapsed time indicated above is merely indicative since it depends strongly on the video (number of detected active regions, trajectory length). The reported processing time was measured on a 2.5GHz, dual core computer.
4 Conclusions

As mentioned in the previous section the implemented software is functional although there are still a few modules that should be improved, namely the active region detection module and the anomalous trajectory detection module. The former is expected to perform better if invariant to illumination changes so a more adequate solution would be, for example, to estimate the background model using a moving median or average or, more complex, a Gaussian Mixture model ([6]). Moreover, a shadow removal algorithm could be useful as well since it would be easier to avoid clustering situations. With a better detection module, the module for the identification of anomalous trajectories is expected to perform better as well, requiring only the adjustment of the threshold. Also, to be able to use this software in real-time it is also necessary to reduce the computational time spent, eventually by porting the developed code to another programming language. Nonetheless this GUI serves as a proof of concept for other applications, as well as a basis for an automatic activity recognition system.
References


