Human activity analysis using pedestrian's trajectories performed in open spaces

I. Introduction

One of the cardinal topics in computer vision and image processing, is to perform the trajectory analysis of moving objects in a given environment. Such analysis is important, since it provides motion information of the moving objects in the scene. Also, it is possible to collect all the trajectories performed over time so as to obtain a statistical description of the activities performed. For instance, considering the vehicles, it is possible to collect information regarding the traffic and to broadcast information regarding where high traffic occurs. Once this information is available, it is possible avoid the congestions areas. If the context of application is related with human activities, trajectory can also be a valuable information. Specifically, typical paths that are performed can be collected, and also, abnormal or suspicious behaviour can be detected. This can be interpreted as the less typical paths taken that most deviate from traditional trajectories.

Another class of applications can be considered. For instance, in the fields of mobile robotics to inform Human-Robot Interaction systems, or automotive providing input to Advanced Driver Assistance Systems.

However, obtaining such algorithm for trajectory collection poses several challenges. For instance, the high variability that characterizes the pedestrians; the appearance of a pedestrian on the image that is affected by the person's pose; clothing; the atmospheric conditions that influence the illumination changes; the background clutter and occlusion. All the above issues play a role in making pedestrian detection a challenging problem to be solved.

The goal of this work is to develop an algorithm capable detect the location of the pedestrians as a way to obtain the performed trajectories. Conventional handcrafted features will be used for this purpose.

As a final remark, please bear in mind that, **it is expected that the output of the algorithm should be able to provide enriched visual information as much as possible.**

II. DATASETS

For this work we will use the publicly benchmark datasets. Among several datasets, we will use the PETS family dataset. The dataset is available in [Dataset]. In this link, we can see that several datasets are available, including:

1) Dataset S0: includes the subsets (i) *background*, (ii) *city center*, and (iii) *regular flow*.
2) Dataset S1: includes two *walking* and one *running* sequences.
3) Dataset S2: includes three *walking* sequences.
4) Dataset S3: includes the subsets *multiple flows* and *event recognition*.

In the dataset S2, there exists three subsets, denoted as **S2.L1**, **S2.L2** and **S2.L3**. We will focus only on the subset **S2.L1**. This subset is concerned with *sparse crowd*. This means that isolated pedestrians are considered with little interactions between them. This sequence has a level of difficulty of 1 (in a range of 1-3 complexity levels).

Considering the **S2.L1**, there are several acquisitions, each containing a different view. The views are numbered as follows: View001, View002, ..., View008.

In this work we will concentrate in the **View001** sequence that contains 795 frames.

The students, however, are free and welcome to use more difficult and challenging views sequences if they feel like. Of course, a reward will be granted.

Fig. 1 shows some images samples belonging to the dataset **S2.L1** in the **View001** sequence.

Figura 1. Frame samples from the dataset **S2.L1** in the **View001** sequence.

III. Ground Truth data to measure the Algorithm performance

To measure the detection performance of the algorithm, we need to have some gold standard, or ground truth (GT) of the pedestrians position. Basically, the idea is to compare the algorithm's predicted output with the GT locations. One way to perform this task is to use the bounding boxes to represent the location of the pedestrian in the image domain. Thus, comparing the bounding boxes of the GT against the ones obtained with the algorithm it is possible to ascertain if the estimated bounding boxes are close or not comparing to the GT positions.

The GT information is available in the GT.txt file available in GitHub [Ground Truth]. The contents of the file above follows the same structure as in [1] (see also Table 2 in [1]). Each line of the file contains:

1) *Frame number*: Indicate at which frame the object is present
2) *Identity number*: Each pedestrian trajectory is identified by a unique ID
3) *Bounding box left*: Coordinate of the top-left corner of the pedestrian bounding box
4) *Bounding box top*: Coordinate of the top-left corner of the pedestrian bounding box
5) *Bounding box width*: Width in pixels of the pedestrian bounding box
6) *Bounding box height*: Height in pixels of the pedestrian bounding box
7) *Confidence score*: Indicates how confident the detector is that this instance is a pedestrian. For the ground truth and results, it acts as a flag whether the entry is to be considered.
8) *x*: 3D x position of the pedestrian in real-world coordinates (-1 if not available)
9) *y*: 3D y position of the pedestrian in real-world coordinates (-1 if not available)
10) *z*: 3D z position of the pedestrian in real-world coordinates (-1 if not available)

Fig. 2 shows an example of an image sample (left) and the same image with the corresponding GT detections represented in bounding boxes (right).

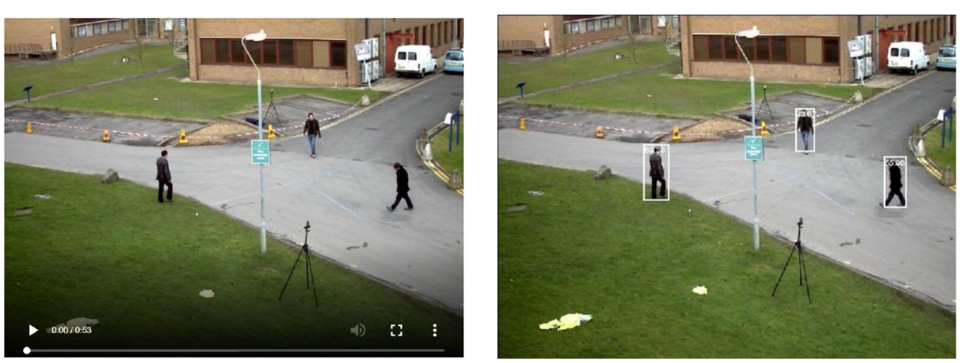

Figura 2. One frame sample from the sequence $\mathbf{S2.L1}$ in the $View001$ (left) and the same frame with the ground truth in bounding boxes (right).

The work will have several goals, the majority intend to enrich the *visual information* that can be extracted from the image sequence. Thus, the students are welcome to fulfill the following challenges:

1) Plot the GT (readable from the GT.txt file) and draw the bounding boxes in each frame in the sequence, (see Fig. 2 right). **(3.0v)**
2) Now, using your detector algorithm, perform the tracking of pedestrians. The predicted bounding boxes should be visible for each detection. Assign a label (*i.e.*, a number) for each detected bounding box. At this stage is not required to have the same label assigned for a given pedestrian. Label switching can occur. **(4.0v)**
3) Plot the performed trajectories. To avoid a possible excess of the information visualisation, you can plot the trajectories dynamically. **(4.0v)**
4) Provide consistent labels through time. This means that a given pedestrian should be assigned to the same label through the sequence. **(2.0v)**
5) Provide to the user, the information regarding the map (*i.e.*, occupancy) of the trajectories performed in the video. Specifically
   - Provide a heatmap, using a Gaussian distance metric (or other), where the color is assigned to the number of occurrences in a given position (region) of the image. Concerning this regard different heatmaps can be generated, this can include (i) static heatmap, (ii) dynamic heatmap. **(2.0v)**
6) Using the Expectation-Maximization (EM) algorithm, provide a statistical analysis concerning the trjectories performed by pedestrians. **(1.5v)**
7) Provide an evaluation performance of the algorithm. Specifically provide: (i) the success plot (see Sec. IV for details), and (ii) the percentage of False Negatives (FN) or misdetections and False Positives (FP). Please provide figures illustrating the success plot, FPs and FNs, and also some frames illustrating the FPs and FNs. **(3.5v)**

IV. Evaluation metrics

One important issue to be considered is that every algorithm hat its own limitations. This means that, no matter the approach is adopted, there is always some failures regarding the true location of the pedestrian. For instance, a merge or split in a given bounding box that can occur. Also, some misdetections may occur as well. Thus, one way to evaluate the algorithm is to use evaluation metrics. An evaluation strategy can be done as follows:

1) The first step is to built the ground truth as already mentioned above.
2) After this stage, the students are in conditions to show both the ground truth and the estimated bounding boxes provided by the algorithm.
3) Now, evaluation must be done. To accomplish this, the following metric is suggested:
   - Provide the *success* plot using the Intersection over union measure (IoU) that is defined as follows:

     $$IoU = \frac{R_d \cap R_{gt}}{R_d \cup R_{gt}} \tag{1}$$

     where R_d is the detected region estimated by the algorithm and R_{gt} is the ground truth (manual labeled) region. Basically, the IoU provides a measure of the overlap (or match) between the R_d and R_{gt}. A score of $\mathrm{IoU} = 1$ means a perfect match is obtained, and $\mathrm{IoU} = 0$, means that the target is loss.

     The success plot shows the percentage of frames whose bounding box overlap ratio is higher than a given threshold. For threshold, it can be considered the values ranging from 0 to 1, with step of, *e.g.* 0.1.

Deadline: The students must upload the projects to my e-mail jan@ist.tecnico.ulisboa.pt until, April 14th, 23h59m. when submitting your project name your zip file simply with your group number, *e.g.* **14.zip**.

V. Reading material

The students are welcome to read the following paper:

[1] L. Leal-Taixe, A. Milan, I. Reid, S. Roth, and K. Schindler "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking", arXiv 2015.