2004 IEEE Intelligent Vehicles Symposium
University of Parma
Parma, Italy • June 14-17, 2004

A tool for vision based pedestrian detection performance evaluation*

M. Bertozzi, A. Broggi, P. Grisleri, and A. Tibaldi
Dipartimento di Ingegneria dell'Informazione
Università di Parma
Parma, I-43100, Italy
{bertozzi,broggi,grisleri,tibaldi}@ce.unipr.it

M. Del Rose
Vetronics Research Center
U.S.Army TARDEC
Warren, MI, U.S.A.
DelRoseM@tacom.army.mil

Abstract

This paper describes a system for evaluating pedestrian detection algorithm results.

The developed tool allows a human operator to annotate on a file all pedestrians in a previously acquired video sequence. A similar file is produced by the algorithm being tested using the same annotation engine. A matching rule has been established to validate the association between items of the two files. For each frame a statistical analyzer extracts the number of mis-detections, both positive and negative, and correct detections. Using these data, statistics about the algorithm behavior are computed with the aim of tuning parameters and pointing out recongnition weaknesses in particular situations.

I. Introduction

The detection of human shapes is one of the most active research objective in the field of artificial vision. Various approaches have recently been proposed (many applications rely on such detectors, like automotive precrash, security and surveillance systems) [1], [2], [3], [4]. An important issue at the basis of the design of a human shape detector is the availability of a tool for performance evaluation. Working on real images, because of the intrinsic problem complexity, some kind of external information is necessary in order to validate the algorithm results.

A system for performance evaluation needs to know the "ground truth", this can be obtained using two different approaches: recording additional data together with processed images data and using the annotation based approach, presented in this paper. The former collects information about the pedestrians position using sensors different from vision, such as radio transmitters. The correctness of the algorithm can be evaluated in realtime but some problems may occur in cases, for example, where a pedestrian is partially occluded but the radio transmitter (or other) is anyway sensed by the detector. The other approach, the one dealt with in this paper, relies on a frame by frame manual annotation, by a human operator, of all pedestrians appearing in each frame of a video sequence. This is a post processing operation, thus images must be acquired and saved on a storage device. Subsequently, the images are annotated in laboratory: a human operator, using a GUI, defines the position and size of pedestrians in each frame and produces a file containing the description of all pedestrian in the image sequence. A similar file of the same format is created by the pedestrian detection algorithm which is under test. Finally the two files are compared and statistics are extracted. Parameters and thresholds can be adjusted and their effect on the algorithm behaviour highlighted.

The outline of this paper is as follows. Section 2 briefly reviews the state of the art in performance evaluation tools for vision algorithms. Section 3 describes the annotation tool composed by: engine, GUI, and performance analyzer. In section 4, an evaluation method for algorithms is proposed along with a case study. Finally, the paper is concluded with a discussion describing results about the optimization of the case study and future work.

II. Performance evaluation

This section describes the state of the art in performance evaluation for vision algorithms and in particular for pedestrian detection.

Vision applications proved their efficiency and usefulness in many fields but current research practices, and in particular system-building techniques, are inadequate especially for fine tuning and filter combination testing. One key aspect of this problem is the inability to conduct adequate performance characterization of new technologies (like pedestrian detectors). Reasons of this fact are due to the complexity of real scenes, sometimes pedestrians are occluded by other obstacles, sometimes parts of framed obstacles looks like pedestrians (even for humans observers).

The main purpose of a general approach to Performance Characterization of Computer Vision Systems [5] is the statistical testing, tuning, algorithmic combination and algorithmic re-use in order to improve algorithms reliability and robustness.

The work presented in [6] uses ground truth automatically extracted from pseudo synthetic video to perform evaluation of a pedestrian tracker on typical surveillance images taken from a fixed camera. However when dealing with real images taken from moving a cameras installed

*The research described in this work was sponsored by US Army.

on a vehicle, the manual ground-truth generation approach seems to be more robust.

ViPER (Video Performance Evaluation Resource) described in [7] and [8] is a Java integrated tool for authoring ground truth meta-data in image sequences and evaluate performance of algorithms.

A similar system is proposed in this paper. A key advantage of this tool is it's integration in a complete environment for the development of vision algorithms. This simplify design and tuning of parameters allowing to directly check the impact, of their variation, on performances.

The study presented in [3] points out the importance of a good performance evaluation method for the actual deployment of systems on board of vehicles. This study was based on a large number of tests. Results were analyzed using ROC curves to highlight the impact of parameters variations on the algorithm performance in terms of detection rate and false positive rate.

III. The annotation tool

A. Description

Performance evaluation using the annotation tool takes place in three steps: supervised sequence annotation by a human operator, automatic sequence annotation by the algorithm being tested, annotations comparison and analysis. Thanks to a common annotation engine, the tool allows the human operator and the algorithm to extract, into separate files, pedestrian information relative to the same image sequence. Pedestrians are described by means of the bounding boxes (BBs) framing their shape. The two files are compared and statistical information about the algorithm behavior are extracted. Each step is described below in detail. The performance evaluation tool structure is shown in figure 1.

1) Supervised sequence annotation by a human operator: During this step a human operator analyzes every frame of a pre-recorded video sequence. For each frame in the sequence the operator manually draws the BB around the pedestrian using a graphical user interface described in section III-B. For each BB, the operator also locates the region containing the pedestrian's head. The head is an important human shape feature that can easily be found. The existence of informations describing heads allows to profile recognition performances of the algorithms that looks for this feature.

It is also possible to classify the pedestrian as completely visible or partially occluded by an obstacle. This description of the sequence ground truth is stored in a file named H shown in figure 1. An example reporting a frame during the annotation process is reported in figure 2.a.

2) Automatic sequence annotation by the algorithm under test: The output of a pedestrian detection algorithm can be described in terms of a list of BBs for each frame. Optionally, the algorithm can also produce information regarding the position of the head or some other interesting feature related to the pedestrian.

If needed an additional block can be added to the algorithm output stage in order to translate its results in a format compatible with the annotation engine input. For example if an algorithm extracts the human shapes from source images its output can be converted in a list of BBs each one defining the pixel area in the source image occupied by the pedestrian's shape. The description of the sequence produced by the algorithm is saved in a file named A also shown in figure 1. An example of BB generated by the algorithm under test is reported in figure 2.b.

3) Annotations Comparison and Analysis: This is the last step of the performance evaluation process. It takes as input the two previously created files and compares them frame by frame, extracting statistical information about the algorithm behavior. Three values are calculated for each frame: false positives (FP), false negatives (FN), and correct detections (CD).

In order to distinguish if a BB generated by the algorithm represents a correct detection (CD) it is necessary to match it to all the BBs annotated by the human operator.

Two BBs, p and q, of area Z_p and Z_q respectively, are defined as *matching* if $Z_{pq} \doteq W_{pq}^2/Z_pZ_q$ is greater than Z_{Th}, were W_{pq} is the overlapped area between p and q and Z_{Th} is a threshold adjustable by the user (a good value may be $Z_{Th} = 0.7$).

This relation embodies the following property: well overlapped BBs generate high values of Z_{pq}, but as the overlapping area decreases linearly, the value of Z_{pq} decreases at higher rate (square).

Every frame of the sequence analyzed through a particular algorithm can be modeled with the following two sets:

$$H_n \doteq \{\text{BBs annotated by a human operator}\}$$
$$A_n \doteq \{\text{BBs annotated by the examined algorithm}\}$$

where n is the frame number of a specific video sequence N frames long.

Let the symbol $|X|$ represents the cardinality of X, namely the number of elements in the set.

It is possible to define the matching operator $\oslash$ between a BB annotated by the operator and one annotated by the algorithm under testing in this way:

let $a_i \in A_n$ and $h_j \in H_n$:

$$a_i \oslash h_j \quad \doteq \quad \begin{cases} 1 & Z_{a_ih_j} = max\{Z_{a_ih_k} | Z_{a_ih_k} > Z_{Th}\} \\ 0 & otherwise \end{cases}$$

Based on this definition $\forall a_i \in A_n$ exists at most one j such that $a_i \oslash h_j = 1$. In fact, given a_i, $Z_{a_ih_j} > Z_{th}$ for different values of j. This ambiguity is resolved by the $max()$ function.

A graphic representation of the matching process result is reported in figure 2.c.

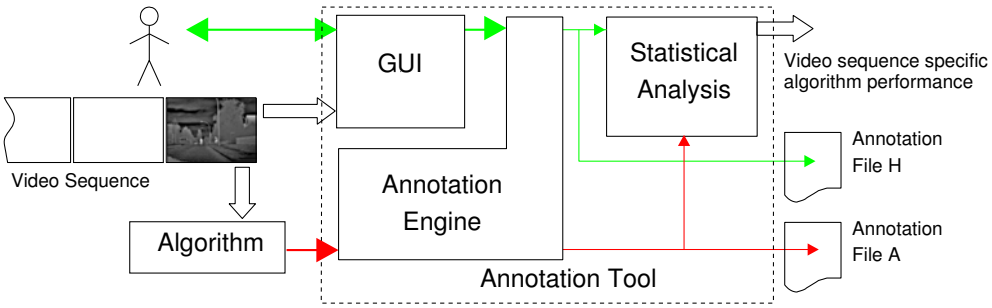

Fig. 1. Block diagram of the algorithm: human and algorithm process the same video sequence producing annotation files H and A. The two files are compared producing statistics about algorithm performance

(a) (b) (c)

Fig. 2. (*a*) Annotation window during the human supervised annotation process: the currently selected BB is cyan filled (it can be resized or moved), non-filled green BBs are non selected BB, the yellow one is marked as occluded, the green-filled one is currently being drawn by the operator. Each BB is composed of two rectangles framing the pedestrian shape and head. (*b*) BBs generated by the algorithm. The red numbers indicate the distance of the pedestrian, the violet area represents the 3D space where stereo vision can be applied. (*c*) Matching phase result: BBs found by the pedestrian detector are presented in red, annotated BBs are in blue and yellow (if occluded), matched BBs are in green.

Using these definitions, three different values are defined:

$$\begin{aligned} CD_n &= \sum_{i=0}^{|A_n|} \sum_{j=i}^{|H_n|} a_i \oslash h_j \\ FP_n &= |A_n| - CD_n \\ FN_n &= |H_n| - CD_n \end{aligned}$$

These values represent respectively the number of correct detections, false positives, and false negatives for the n-th frame of the sequence. In this way it is possible to identify frames in which the algorithm works fine and situations related to algorithm weaknesses.

Now a set of global values is defined referred to the whole sequence in order to compare different algorithms working on that sequence.

$$CDR = \frac{\sum_{n=0}^{N-1} CD_n}{|H_n|} \quad (1)$$

$$FPR = \frac{\sum_{n=0}^{N-1} FP_n}{N} \quad (2)$$

These values are sequence specific and measure respectively: the correct detection rate and the false positive rate. FPR cannot be normalized because false positives have no upper limit.

B. User Interface

The input interface has been designed with the objective of reducing, as much as possible, the workload for frame annotation. An example of the annotation window during the drawing process of a new pedestrian is presented in figures 2.a. In figure 6.b is reported the annotation panel during the annotation process. The key points for reducing the annotation time are the following:

Similarity between consecutive frames. Usually a frame in a real-time sequence contains little differences from the previous one. For this reason it can be assumed that a BB containing a pedestrian will have a similar position and size in the subsequent frame, possibly with some little corrections. To remove the need for a complete redrawing of BBs on every frame of a sequence, the GUI copies all BBs of a certain frame to the following one, leaving to the operator the task of adjusting size and position, as well as adding and deleting new and disappeared BBs.

Easy input method. The interface has been studied keeping ergonomics in mind: the operator uses one hand to

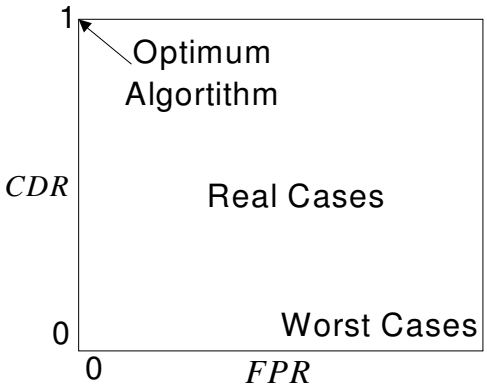

Fig. 3. Vision based pedestrian detection normalized evaluation space.

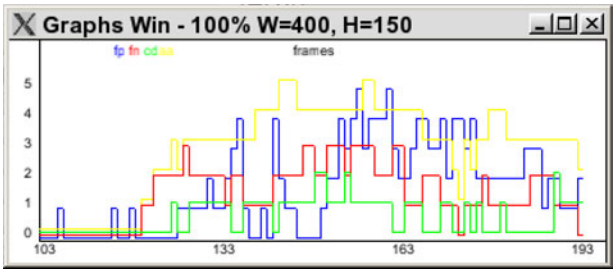

Fig. 5. Statistics extraction screen-shot: for each frame values of CD, FP, FN and human annotated (in yellow) are computed.

command the mouse and the other one for the keyboard. In this way all important commands such as tracing, resizing and repositioning of BBs are directly available to the user. Moreover using the mouse wheel the operator can select the target BB to modify. Some additional keyboard commands allow to speed up common operations such as deleting all BBs in the frame.

It has been proven that this kind of interface is user friendly. This GUI allows a human operator to annotate about 100 frames/h. This number is the average speed obtained from 10 different users who never used the tool before, each annotating 200 frames. However it is reasonable to assume this speed will increases with experience. A more accurate drawing of BBs can be obtained magnifying the tracing area.

IV. Algorithms Evaluation Space

This section contains some considerations regarding algorithms evaluation.

The values CDR and FPR introduced in the previous section were referred to a single sequence. Indeed in order to have a more general and robust statistical description of the algorithm these values must be computed on a sufficiently large sequence including a wide variety of different scenarios.

The 2D space $< FPR, CDR >$ is defined as shown in figure 3; the optimal algorithm is placed in the point $(0, 1)$. Namely, all algorithms that do not give any false positive should stay in the segment $[0, a]$ with $a \in [0, 1]$. Algorithms with bad performance fall in the right bottom part of the space while real cases fall inside the central area.

This kind of evaluation allows to determine if an algorithm modification improves (even slightly) the recognition performance. For example this system is useful to fine tune parameters and thresholds. It is possible to evaluate the impact that a parameter modification has on the algorithm performance observing the movement of the point representing the algorithm in the evaluation space. Moreover the inclusion of new filters can be evaluated measuring performance variations. Indeed the time consuming annotation process that requires the human supervision is performed only once for every sequence.

It is also possible to define a metric (for example the euclidean distance from the [0,1] point) to asses improvements. It is necessary to underline that the specific optimality criterion is strictly dependent on the application. In some applications, such as quality control for example, it may be desirable to avoid false positives and disregard false negatives. In other applications, such as automotive precrash systems, some false positives may be acceptable, even if an excessive number of FP reduces the user confidence in the recognition system. In these two cases the distance from the optimum point should be defined in different ways.

V. Preliminary Results and Conclusions

The performance evaluation tool described in this paper has been used to evaluate pedestrian detection algorithms. In particular, the algorithm presented in [9] has been chosen as a case study. A number of sequences for nearly 1500 images have been manually annotated. The sequences were taken in different scenarios (parking lot, open field, and downtown), under different illumination and weather conditions, and framed different subjects at different distances. The aim of this step was to create a test set describing many of the the possible cases that the algorithm can deal with. In the test sequence, composed of 1500 images, 1897 human shapes were annotated as completely visible while 361 were marked as partially occluded. Figure 4 shows a number of different situations for the test sequence.

The overall system performance shows that the correct detection rate is about 83% (1572/1897=82.9%). The high sensitivity of the algorithm (83%) comes along with an appreciable false positives rate 0.46 FP per frame. Indeed, a set of higher thresholds in the algorithm would decrease the number of false positives, but, at the same time would reduce the correct detection rate. The false negatives rate (426/1897=26.1%) summed up with the correct detection rate exceeds 100% (82.9%+26.1%=109%) since the latter also includes occasional detections of occluded human shapes.

The main result of these tests were the statistical characterization of the detector behaviour and the precise identification of particularly challenging segments of a large

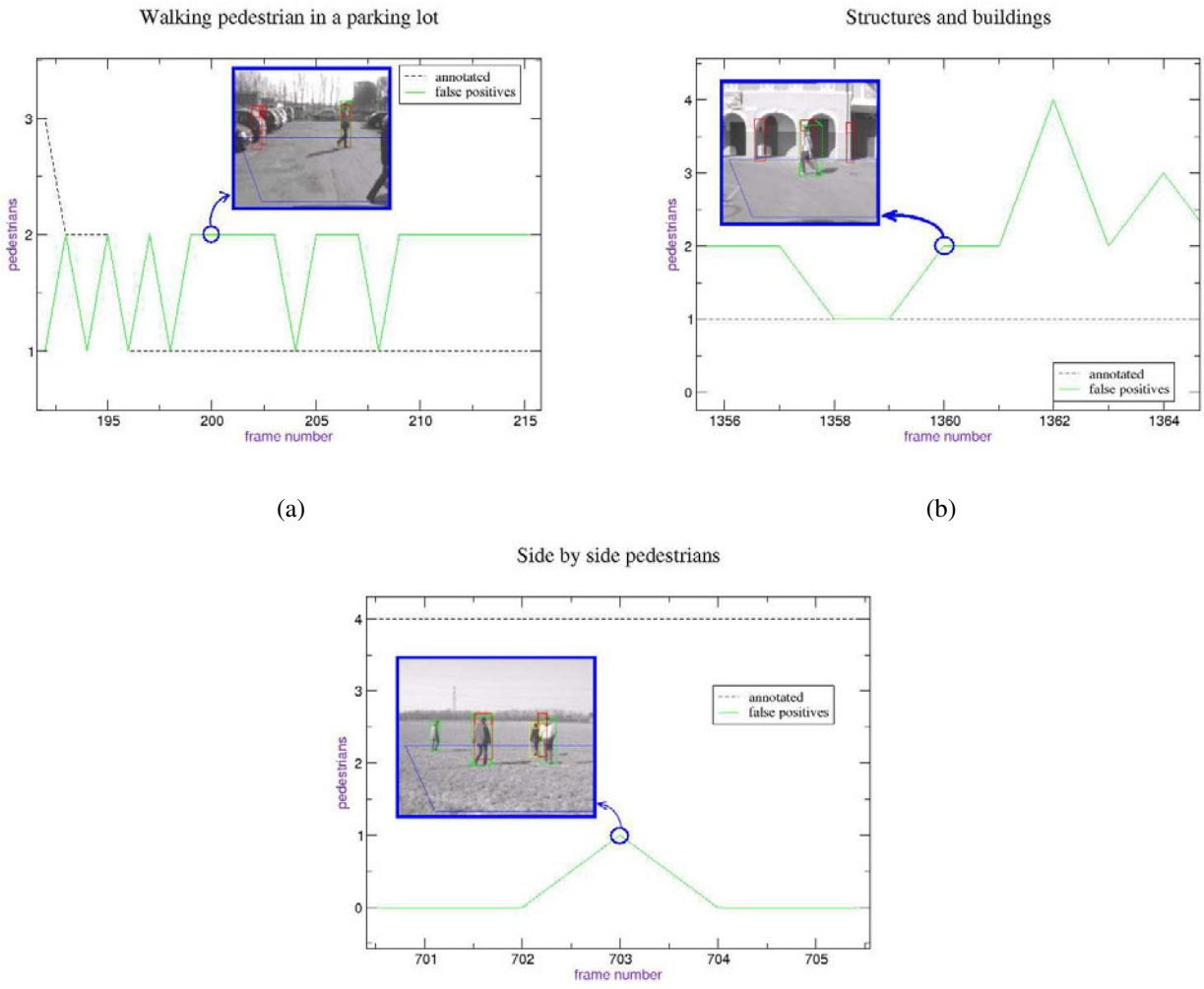

Fig. 4. Examples of situations in which the human shape detection algorithms partially fails: (*a*) background noise generated by parked vehicles introduces false positives; (*b*) columns generate false positives due to their symmetry; (*c*) two pedestrians walking side by side mislead symmetry evaluation.

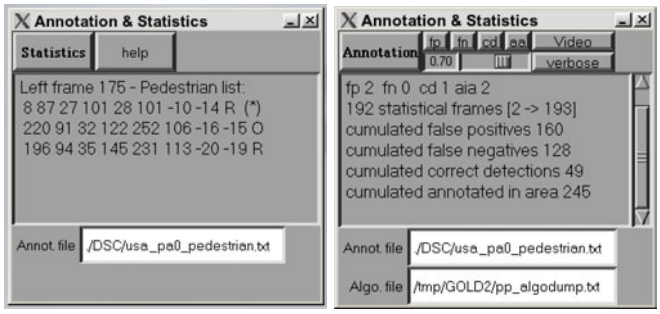

Fig. 6. (*a*) Statistics extraction screen-shot: for each frame values of CD, FP, FN are computed and cumulated up to the current frame. (*b*) Annotation panel during the human supervised annotation process: information about current operation are displayed

sequence. The program output while extracting statistics from the algorithm is shown in figures 5 and 6.b.

The presented performance evaluation tool has been proven to be effective though it requires a very expensive annotation process. The time required to annotate one frame is a value than can be reduced only by either modifying the input method (finding more efficient shortcuts for frequent operations) or trying to detect off-line the BB modifications. Thus, improvements should be in GUI refining and in automatic resize/reposition of the bounding boxes using motion detection techniques whenever possible. GUI refinement can be done following impressions of user that performs longs annotations. Motion detection techniques such as correlation analysis between frames and optical flow can be used to determine the new coarse position and size of BBs in new frame starting from those in the previous frames. It is necessary to consider that the time

spent to perform such detection can't be too hight in order to maintain the number of annotated frames/h comparable with the human one. Relying on an accurate prediction of the new BBs position, operator's task would be reduced to a mere supervision activity reducing the time spent to annotate each single frame.

This study shows a tool to detect positive aspects and weakness points of a pedestrian detector working on a given video sequence. The tool can also serve as a performance comparison method between different algorithms.

The system was implemented using the C language and included in the GOLD software, a framework for vision applications development implemented at the University of Parma.

REFERENCES

[1] M. Bertozzi, A. Broggi, T. Graf, P. Grisleri, and M. Meinecke, "Pedestrian Detection in Infrared Images," in *Procs. IEEE Intelligent Vehicles Symposium 2003*, (Columbus, USA), pp. 662–667, June 2003.

[2] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, "Shape-based pedestrian detection and localization," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2003*, (Shangai, China), pp. 328–333, Oct. 2003.

[3] D. M. Gavrila and J. Geibel, "Shape-Based Pedestrian Detection and Tracking," in *Procs. IEEE Intelligent Vehicles Symposium 2002*, (Paris, France), June 2002.

[4] H. Elzein, S. Lakshmanan, and P. Watta, "A Motion and Shape-Based Pedestrian Detection Algorithm," in *Procs. IEEE Intelligent Vehicles Symposium 2003*, (Columbus, USA), pp. 500–504, June 2003.

[5] N. Thacker, "Performance characterization in computer vision," tech. rep., PCCV project of the EU-IST programme, July 2003.

[6] J. Black, T. Ellis, and P. Rosin, "A Novel Method for Video Tracking Performance Evaluation," in *Procs. Joint IEEE Intl. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, (Nice, France), Oct. 2003.

[7] C. Jaynes, S. Weeb, R. M. Steele, and Q. Xiong, "Development Environment for Evaluation of Video Surveillance Systems," in *Procs. IEEE Intl. Workshop on Performance Analysis of Video Surveillance and Tracking*, (Copenhagen, Denmark), June 2002.

[8] D. Doermann and D. Mihalcik, "Tools and Techniques for Video Performance Evaluation," in *Procs. IEEE Intl. Conf. on Pattern Recognition*, vol. 4, (Barcelona, Spain), pp. 167–170, Sept. 2000.

[9] A. Broggi, M. D. Rose, A. Fascioli, I. Fedriga, and A. Tibaldi, "Stereo-based Preprocessing for Human Shape Localization in Unstructured Environments," in *Procs. IEEE Intelligent Vehicles Symposium 2003*, (Columbus, USA), pp. 410–415, June 2003.