Extraction of traffic information from images at DEIS

Enrico Viarani
DEIS Department of Electronics, Computer Science and Systems
University of Bologna
Via Risorgimento,2 - 40126Bologna, Italy
eviarani@deis.unibo.it

Abstract

This paper reports about the research work currently going on at DEIS, University of Bologna, on the use of Computer Vision for extraction of information from urban traffic images. Two algorithms are described: The first, named "Virtual Inductive Loop" (VIL), computes image differences on small windows of the traffic scene between to an updated background and the actual window and is specific for vehicle counting and detection. The second algorithm, based on Block Matching Algorithm (BMA), is particularly suitable for vehicle motion detection and tracking. It consists of a BMA implementation (improved with respect to the standard version) followed by a regularisation step based on a custom median filter, of a labelling algorithm and of a tracking step. Finally, we present our future plan for the research developments concerning these topics.

1. Introduction

Nowadays the citizen has to deal with worsening traffic conditions in urban areas. Congestion, road accidents and environmental impact are among the serious problems caused by traffic which severely affects the quality of life of the citizen. In this framework, a strategic role is played by Intelligent Transportation Systems (ITS), which, thanks to the combining of information, communications and transport technologies, allow the development of advanced traffic-related applications on top of the existing urban infrastructure. Among ITS information technologies, computer vision is naturally suited to the development of surveillance-monitoring-enforcement traffic-related applications.

In the early 1998 the group at Vision Lab., DEIS, University of Bologna, started a novel research activity on the use of machine vision techniques in the ITS domain, with specific emphasis on traffic monitoring and control within urban areas. At this early stage, the aim of our work is to develop a set of general computer vision techniques and tools to be deployed in the near future in several traffic-related applications, which we are currently defining in co-operation with a number of partners. These include companies involved in the development of traffic control and video-enforcement systems as well as public bodies and public-transport companies. We list here some possible applications which are currently under consideration: *monitoring of reserved lanes* (detecting abuse situations for prompting enforcement by authorised patrols as well as statistical measurement of abuse) *monitoring of unauthorised parking* (aimed at prompting vehicle removal and/or fining), *monitoring of traffic flow at intersections* (including inductive loops emulation, queue length measurement and evaluation of turning rates), *assessment of traffic situation like congestion-detection or incident-detection.*

In this work we report on two topics which are currently under investigation in the context of the above mentioned research activity.

The first topic, referred to as VIL (Virtual Inductive Loop) consists in the development of "tripwire" vehicle detection algorithms for loop-emulation. Tripwire detection algorithms are based on the definition of small active sub-images which are repeatedly compared against a known background in order to detect the presence of vehicles. One major issue concerns the development of effective strategies for updating the reference image according to the scene light changes. Clearly, tripwire algorithms are the very natural approach for the implementation of video detectors capable of emulating the behaviour of inductive loops. The final goal of our activity is the development of a low-cost vision-based device to be interfaced directly with existing traffic-control systems (which so far accept inductive loops outputs).

The second topic concerns effective and reliable vehicle detection and tracking using the Block Matching Algorithm. These techniques should form the very basic tools in the context of applications such as evaluation of turning rates at intersections, queue and congestion detection, incident detection, queue length measurement, traffic parameters estimation, vehicle classification, assessment of traffic situations.

2. Virtual Inductive Loop (VIL)

Inductive-loop sensors are the most widespread technology for the existing traffic control systems [1]. These systems are able to detect and count vehicles passing over loop sensors. These are posed into the asphalt, normal to the lane direction, close to intersections [2]. With inductive loops, traffic flow at intersections cannot be monitored directly, but only on the basis of loop counting on the different lanes insisting on the intersection and a-priori estimated turning rates. The main problems with these sensors are due to installation and maintenance. Inductive loops are rather cheap, and this is certainly an important factor in the context of traffic lights control since a typical medium size city may require hundreds of these sensors. Yet, loop installation requires road works, with the lane being closed for some hours; moreover the elasticity of the road mantle, the ordinary maintenance road works and the normal road wear are factors that shorten loops lifetime. Thus, despite their low unit cost, inductive loops are characterised by high installation and maintenance costs.

Given the aforementioned problems, an alternative approach to vehicle may be implemented through video cameras together with suitable machine vision algorithms. They can be installed on top of pre-existing lights pole and is often possible to replace several inductive loops with a single camera. With video detection the installation cost can be kept low and installation does not require lane closure. In addition, maintenance costs are very low.

The VIL is thought to provide a signal totally compatible with the one provided by inductive-loops sensors.

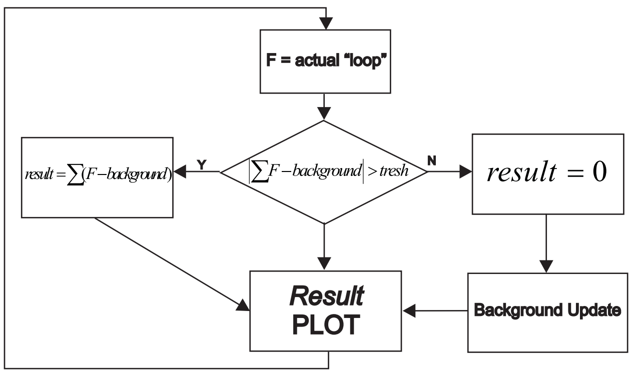

Figure 1. The VIL flow-chart.

Moreover, camera installation and maintenance does not require road works and lane closures. The flow chart of Figure 1 summarises the VIL algorithm which we have implemented. The VIL needs the definition of small rectangular windows in the video image which (for dimensions and position) looks like true inductive loops. For each window, a background reference image, *background*, is updated, during time, in absence of variations due to vehicle presence. When a vehicle rolls over the window it produces a variation of the pixel grey tone levels. Per every cycle, a difference between the background reference window and the current values of the same window F is estimated pixel-by-pixel. At every step, the VIL output *result* correspond to the sum of these differences. If this sum is higher than a small threshold *thresh* the counter is triggered and the sum is plotted in the display; if not, the sum is set to zero in order to clean up noise effects.

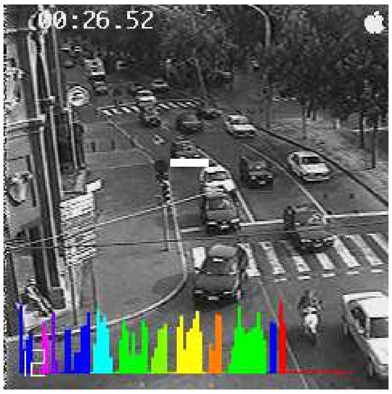

Figure 2. VIL output display (img. source courtesy of the Traffic Control Centre of Bologna.)

The Figure 2 shows a frame of the VIL output display: the white area shows the location of the VIL detector. The signal displayed in the bottom represents its output during a given time interval and different tones are used to identify different vehicles. This signal emulates the output which would be generated by an inductive loop buried in the road in correspondence of the VIL window. In the sequence showed in the figure the VIL counts 21 vehicles versus 21 true passed ones, over a 46” sequence duration. This is a good result, but we have found that in night images the precision severely decreases due to totally different light conditions.

3. BMA-based vehicle detection and tracking

We have developed a vehicle detection technique based on the Block Matching Algorithm (BMA), which is the algorithm used for motion estimation in the MPEG image compression standard [3]. This algorithm provides the motion estimation through the Cross Correlation function [4]. It consists in partitioning a frame of a sequence into square blocks of fixed size (e.g. 6x6, 8x8,...) and detecting blocks displacement between the actual frame (at time t) and the previous one (at time t-1), searching inside a given scan area. The BMA output is a Displacement Vector Field (DVF) where each vector represents the displacement between two successive of the associated block. In order to reduce the computational load, we have chosen to scan the whole area only if the correlation of the block in the same position of the previous frame is not high enough. A Vector Median Filter (VMF) [5] is the applied to the DVF in order to reduce noise and restore wrong matches. We have improved the basic VMF in order to avoid deletion of motion vectors in quite sparse DVFs. Our modified VMF works only on non-null motion vectors and takes into account only non-null neighbours when computing the median. Thus, it may be seen as an adaptive rank filter [6]. A labelling algorithm then identifies vehicle by grouping blocks on the basis of closeness (8-connection) and motion (i.e. DVF) similarity. It is basically a sequential iterative connected components labelling algorithm [7] in which the blocks having non-null displacement vectors are considered as "foreground points" and two blocks are considered neighbours if they are 8-connected and the L_2-norm of their displacement vectors is less than a threshold. The labelling output is a map having the same size as the original image and containing "objects" (i.e. vehicles) made of out of 8-connected blocks. In this image called "labels-map" each vehicle is marked with a different label value.

Very recently we have added a tracking step to this vehicle detection technique. This is based on the idea that BMA already tracks single blocks. By merging the block-level tracking provide by BMA with the grouping provided by the labelling step, we can track entire moving objects, which indeed are vehicles.

To track an object, both, the BMA and the labelling steps, needs to be done for the actual and the previous frame. We consider the set of the blocks belonging to a labelled object. Then, as showed in the example of Figure 3 we shift back the object from its actual position by the reversed associated displacement vector and we seek for what label value is more overlapped in the previous frame: this will be "5" in the example of Figure 3a because there are more pixels in the "five gray-area" than in the "six gray-area". To do this computation we calculate the number of the entries for each different pixel-label value. After repeating these operations for each block belonging to the label in the actual frame, we finally obtain an array of label values where each cell contains the number of the entries for that label value. The number of the cell containing the maximum number of pixel (cell number "5" in the example of Figure 3a) will be the new label for all the blocs of the actual object as shown in Figure 4 which is the tracking output the of the object of Figure 3b.

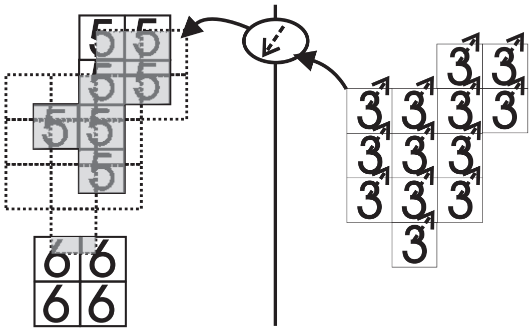

Figure 3. a) The label overlap match. 3b) The actual object shift-back.

If any labelled pixel in the old frame is overlapped, a new unused label value will be attributed to our actual object (it is a "new born").

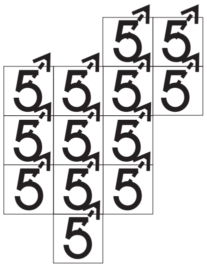

Figure 4. Tracked object of Figure (3b.

During the sequence, due to noise and to the moving object shot change, the blocks of same labelled object may build different shapes, and different detection of moving blocs may cause small area variations (look at label differences in Figure 5a-b). For these reasons the overlap can result incomplete and we might find both overlap some labelled pixels and some unlabelled ones. Consequently, the value for the maximum number of overlapped pixels might result as "unlabelled". If we don't take care of the presence of "labelled" pixels in the set of the overlapped ones it happens that we do label as "new born" an object which conversely exists in the previous frame and should be tracked.

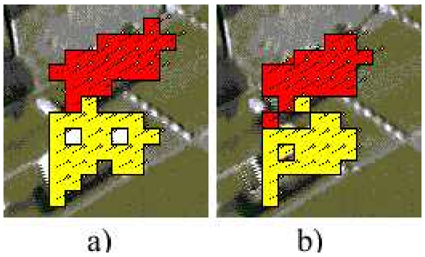

Figure 5. Two labelled vehicles of the (5a are tracked with the same tones in (5b.

In order to avoid mismatches, if the greatest number is for unlabelled pixels, we search for the presence labelled ones and we choose, as the new label for our object, the value with the maximum number of pixel entries. For instance, in a extreme case, if we have overlapped at least one labelled pixel and others are unlabelled ones, the labelled pixel value will be the new value for the whole actual object.
We proceed in the same way for all the objects of the actual frame obtaining the tracking of the previous image objects and the detection of "new-born" and "dead" ones.

4. Conclusion

The results we have presented in this paper encourage us in planning future developments of the two approaches. Our efforts are currently spent on the development of fast and robust algorithms for the background updating in the VIL system, with respect to illumination changes and the whole spectrum dynamic of possible vehicle gray-tones. We are currently developing a slow/weighted update method for the current value of background matrix *background(t)*, depending on two coefficients w_1 and w_2:

$$background(t) = w_1 \cdot background(t-1) + w_2 \cdot window(t)$$

where w_1+w_2=*1*, *window(t)* is the actual window state and *background(t-1)* is the previous updated background. Concerning the BMA-based approach we are implementing a perspective correction algorithm in order to estimate motion correctly in case of far and near vehicles. The worst problem for BMA seems to be its computational load. For this reason, in order to fasten the process, we plan to develop in the a new BMA approach where "interest blocks" are pre-selected on the basis of a rough motion estimation provided by frame differencing. We are also working on improvements of the tracking step based on exploiting the feedback coming from several high-level rules.

References

[1] P. Briquet. Video Processing Applied To Road and Urban Traffic Monitoring. *6th IEEE Int'l Conf. on Road Traffic Monitoring and Control*, 92(335):153-157.

[2] P. G. Michalopulos. Vehicle Detection Video Through Image Processing: The Autoscope System. *IEEE Transactions on vehicular technology*, 91(40):21-29, February 1991.

[3] P. Baglietto, M. Maresca, M. Migliardi, N. Zingirian. Customizing MPEG Video Compression Algorithm to Specific Application Domains: the Case of Highway Monitoring. *Lectures Notes in Computer Science*, 97(1311):46-53, Springer Verlag, 1997.

[4] H.G. Musmann, P. Pirsch, H.J. Grallert. Advances in Picture Coding. *Proceedings of the IEEE*, 94(73):523-546, No. 4, April 1994.

[5] J. Astola, P. Haavisto, Y. Neuvo. Vector Median Filters. *Proceedings of IEEE*, 99(78), No. 4, April 1990.

[6] R. J. Shalkoff. *Digital Image Processing and Computer Vision*, Wiley, 137-144, 1989.

[7] R. M. Haralik, L. G. Shapiro. *Computer And Robot Vision*, Addison-Wesley, (1):28-33, 1992.