Evaluation of Inductive-Loop Emulation Algorithms for UTC Systems

Luigi Di Stefano, Igor Milani, Enrico Viarani
University of Bologna
DEIS-Department of Electronics Computer Science and Systems
Via Risorgimento 2, 40136-Bologna, Italy - http://labvisione.deis.unibo.it
ldistefano, imilani, eviarani@deis.unibo.it

Abstract

The paper presents the recent results of a research project active at the University of Bologna and aimed at the emulation by means of computer vision algorithms of the inductive sensors used for vehicle detection in Urban Traffic Control (UTC) systems. Emulation of inductive sensors using computer vision has traditionally been based on Sum of Absolute Differences (SAD) algorithms. In this paper we propose an approach based on the Normalized Cross Correlation function (NCC) and present experimental results aimed at comparing SAD and NCC based vehicle detection. Our results show that the NCC-based algorithm significantly outperforms the SAD-based one.

1. Introduction

In recent years most cities all over the world have installed UTC systems to improve traffic fluency and reduce pollution. These systems [6] need the vehicle flows at intersections and along road segments as input data. Traffic data can be collected either by inductive loops [5], [4] or by suitable computer vision algorithms applied to traffic images [10], [8], [7], [3]. The latter rely on the analysis of sensitive windows positioned on the image by an operator during set-up operations. Once the windows has been defined, vehicles are detected within each window by "comparing" [13] the current gray-level distribution with a reference background distribution acquired when no vehicle is running over the window. However, though the use of vision-based vehicle detection systems is notably increasing in highways environments, so far UTC systems carry on using inductive loops. The reason of this is to be found in the shared perception among UTC producers of an insufficient level of reliability of computer vision solutions. Moreover, even though computer vision based sensors require low installation and maintenance costs, they still sell at quite high prices, definitely higher than inductive-loops.

In this context, our project's goal is to develop a very reliable, low-cost and compact computer vision-based sensor capable of emulating the behaviour of an inductive loop. In this paper we address the issue of the method used to compare current image and background. While most computer vision systems for loop emulation rely on the SAD (Sum of Absolute Differences), we propose the use of the NCC function and show experimental results aimed at evaluating these two approaches.

2. Similarity Measures

As mentioned before, in computer vision based loop-emulation the analysis of the sensitive window consists in comparing a reference image (the background taken without vehicles) with the current image [15]. To the best of our knowledge the Sum of Absolute Differences (SAD) function has been the typical choice for the execution of this comparison [12], [11], [16], [2], [14].

Due to the extreme variability of the environment conditions in urban areas, the background changes frequently its aspect and a suitable background updating stage must be incorporated into the standard detection algorithm. Shadows, clouds and artificial-related light changes, together with devices "auto-gain" (or "auto-iris") are among the most common noise sources for background. In this paper we propose to carry out the image-background comparison in the sensitive windows by means of the Normalized Cross Correlation function (NCC). Looking at Figure 2, if we consider the pixels of an $M \times N$ sensitive window as the components of an $(M \times N)-$dimensional vector, the SAD corresponds to the L_1 norm of the difference $\vec{D}$ between the two vectors representing respectively the current sensitive window $\vec{C}$ and the background $\vec{B}$. This is significantly affected by both image brightness and image texture, with the former typically changing very frequently as a result of the previously mentioned noise sources ($\vec{C}$ goes in $\vec{C}'$ and the difference $\vec{D}$ goes in $\vec{D}'$). Conversely, the NCC function correspond to the angle between the two vectors represent-

Figure 1. The VIL output display.

ing the image and the background, and therefore is much more robust with respect to noisy brightness variations (i.e. it is potentially sensitive only to the true texture variations).

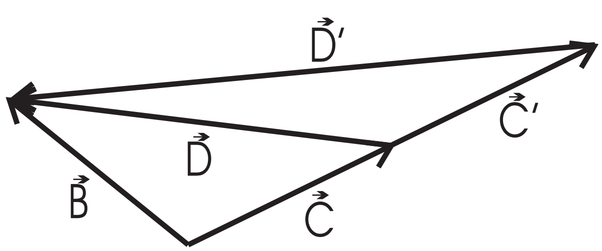

Figure 2. SAD comparison representation under variations.

3. Implementation

We have implemented both SAD and NCC-based vehicle detection in a software application running on a personal computer equipped with a frame grabber board which can be connected to a VCR or a video-camera. The frame grabber used to process the image flow is a low cost one: it is based on a common chip (the Brookthree “BT-848”) that integrates both the ADC video sampler and the PCI bus interface. The software development kit for interfacing the grabber drivers with the operative system is the Microsoft “Vision SDK” [9]. This tool allows to access to the image pointers so as to get images from the frame grabber as matrices at video-rate speed and display-it on the application window. The image processing stage is performed by Visual C++ code. The program, referred to as “VIL Analyser” (where VIL is an abbreviation of Virtual Inductive Loop), allows the user to position the sensitive windows in the im-

	Seq1	Seq2	Seq3
Light trends	Constant light on whole duration	A smooth increase of lighting bring noise over SAD thresholds on 4% of true events	A fast light increase happens after 43% of true events; the light stays high until the end
Duration	6’ 05”	7’ 06”	6’ 49”
True events	99	93	74
Occlusions on true events	15%	4%	22%

Table 1. Features of benchmark sequences

age and then evaluate the performances of the algorithms in the vehicle counting task. The VIL output display is shown in Figure 1: the virtual loop defined by the user is indicated by the black arrow while the vehicle counter appears in the application state-bar; the output, “loop-like” signal produced by the VIL comparing the current image against the background is displayed in the bottom-left area of the application-window. The vehicle counter is incremented by thresholding the output signal by means of a two-valued, hysteresis process: to count a vehicle the signal must first overtake the high threshold and then fall below the low one.

The VIL Analyser program allows also the user to choose between fixed background (typically acquired when no vehicles are in the loop-window) and “weighted background updating”, in which the background is continuously updated by the algorithm according to two weights: $New\_bkg = Old\_bkg \cdot w_1 + Actual\_img \cdot w_2$, with $w_1 + w_2 = 1$.

The two weights control the speed with which the background is updated by the algorithm: as w_1 gets closer to 1 the updating process gets slower.

4. Experimental results

To perform the evaluation of SAD and NCC based vehicle detection we have defined benchmark sequences characterised by a specific light trend. We have scanned a four hours video-tape, finding the three sequences defined in Table 1. “*True events*” is the number of vehicles that should be detected by an ideal virtual loop. “*Occlusions on true events%*” is the percentage of true events in which the background does not appear between successive vehicles. In Table 2 we report the experimental results of the evaluation process. The continuous background updating is done with weights $w_1 = 0.95$, $w_2 = 0.05$. Virtual loop size are 7 pixels height, 26 pixels width. Relative thresholds for hysteresis has been kept the same over the whole measurement set. The basic performance measure is defined as:

$$Accuracy\% = \left(1 - \left|\frac{TrueEv - MeasEv}{TrueEv}\right|\right) \cdot 100.$$

The estimation of $accuracy\%$ does not takes into account the possible over/under-counts compensation. This is a common way to estimate the accuracy of these systems [1] since vehicle detection sensors for UTC systems must keep counting under several light conditions rather than being punctually exact in counts.

Over-counts often happens due to the presence of the windscreen (or by multiple edges on long vehicles) that causes more than one peak in the output signal. This problem can be alleviated by increasing the size of the virtual loop, but unfortunately this increases also the under-counts since it reduces the resolution when two vehicles are queued very close each other. Indeed, the major cause of under-counts are vehicle occlusions, i.e. “train” of vehicles riding very close each other and hiding the road background in between. This drawback increases when images are taken under strong perspective distortion.

Looking at Table 2 from left to right, starting from $Seq1$ we can observe, in column 2, that NCC performs 18% better than SAD. This is due to SAD’s higher sensitivity to vehicles gray-tone: if vehicles are similar to the road background, SAD cannot detect enough difference. Conversely NCC can detect vehicles much more independently of their gray-tone. In column 3, both SAD and NCC improve their accuracy under the effect of the weighted background updating stage. A detection improvement due to background updating might be surprising in a sequence without light changes. Yet, this can be understood if we consider long occlusions due to “train” of vehicles. Under continuous updating the background tends to be turned to the texture of the vehicle actually running in the window, so that the detector can recognise differences between different vehicles. Indeed, $Seq1$ is a sequence with a remarkable percentage of occlusions. In other words, the weighted updating stage improves the robustness of VIL with respect to vehicle occlusions.

In $Seq2$, the 4% of true events is affected by a light change that makes SAD overtaking the lower threshold also without vehicles. When this happens the SAD-based VIL stops counting. In Table 2, column 4, we find that the performance difference between SAD and NCC is 22%, i.e. exactly a 4% worse than the difference in the case of no light changes (column 2). If we start the weighted background updating, the difference between SAD and NCC decreases to 20% (column 5) because SAD can

	Seq1: constant light		Seq2: smooth light change that impacts 4% of total events		Seq3: light discontinuity after 43% of events	
	Bkg acquired once: Accuracy%	Bkg updated continuously: Accuracy%	Bkg acquired once: Accuracy%	Bkg updated continuously: Accuracy%	Bkg acquired once: Accuracy%	Bkg updated continuously: Accuracy%
SAD	72.7%	81.5%	78.5%	79.3%	36.5%	70.3%
NCC	90.9%	98.7%	100.0%	98.9%	86.5%	89.2%
(Δ%)	(18%)	(16%)	(22%)	(20%)	(50%)	(19%)

Table 2. SAD and NCC performances under different lighting situations

count (after a while, depending on the weight values) even during the light variation. Please note that the 100% accuracy of NCC in column 4 is due to the compensation of under/overcounts. Finally, in $Seq3$, after 43% of the events, the SAD value goes over the threshold due to the sharp lighting change and therefore the SAD-based VIL stops counting. For this reason the SAD accuracy is quite low in column 6. Even the NCC performs worse in the same column, but this is due to the higher percentage of occlusion (22%) in this sequence. If we turn on the weighted background updating (column 7) the NCC improves as usual and SAD improves quite a lot (from 36.5% to 70.3%) because it can now count vehicle also after the light discontinuity.

5. Conclusion

In this paper we have proposed the use of the NCC function as a comparison criteria for vehicle detection in vision-based sensors for UTC systems. We have compared an NCC-based loop-emulation algorithm with a SAD-based one, since SAD is the comparison criteria traditionally employed in vision systems for loop-emulation. Our experimental results show that NCC significantly outperforms SAD (the accuracy is always better of, at least, a 16%). Our results for SAD are comparable to others found in literature [1]. Using weighted background updating both SAD and NCC grant better performance, under either constant or varying light. The improvement in the constant light situation is due to the better ability to detect “train” of vehicles. The NCC is insensitive to most common background noise sources (generally, changes of light) and allow detection systems to work even without a background noise filter. The NCC used as a “change detection function” over small windows does not add significant computational complexity and results as a fast image processing elaboration. Running our software on a 200 MHz Pentium processor, we always processed 25 images per second, corresponding to the PAL video standard frame-rate.

As consequence of these experimental results, we will use the NCC function as the comparison criteria in the implementation of the compact, low-cost sensor mentioned in the introduction which is the final goal of our work.

References

[1] B. Cottrell jr. Evaluation of a video image detection system (vids) - final report. http://www.bts.gov/NTL/DOCS/cottre.html, may 1994.

[2] E. Hilbert et. al. Wide area detection system-conceptual design study. JPL Interim Rep. for FHWA, Feb. 1978.

[3] http://www.citilog.fr/Citilog us/Company.htm. Traffic data measurement system, 2000. WEB site.

[4] http://www.elsema.com/ld30.htm. Inductive loop producer, 2000. WEB site.

[5] http://www.icon.co.za/ gold/loop.html. Inductive loop producer, 2000. WEB site.

[6] http://www.miz.it/Products/utopia.html. Utc traffic control system, 2000. WEB site.

[7] http://www.odetics its.com/. Traffic data measurement system, 2000. WEB site.

[8] http://www.peek traffic.com/start.html. Traffic data measurement system, 2000. WEB site.

[9] http://www.research.microsoft.com/research/vision/. Vision software development kit version 1.1, 2000. downloadable via WEB.

[10] http://www.traficon.be/. Traffic data measurement system, 2000. WEB site.

[11] R. Iñigo. Traffic monitoring and control using machine vision: a survey. *IEEE Transactions on Industrial Electronics*, IE-32(3):177–185, august 1985.

[12] R. Iñigo. Applications of machine vision to traffic monitoring and control. *IEEE Transactions on Vehicular Technology*, 38(3):112–122, 1989.

[13] P. Michalopulos. Vehicle detection video through image processing. *IEEE Transactions on Vehicular Technology*, 40(1):21–29, 1991.

[14] E. Viarani. Panoramica sulle tecniche di controllo del traffico urbano utilizzanti la visione artificiale. Technical Report CSITE-004-98, CSITE-CNR, CSITE-CNR c/o DEIS, Via Risorgimento,2 - 40136 Bologna ITALY, May 1998. Also on Http://www-labvisione.deis.unibo.it/~eviarani/research/publications/csite/csite.htm.

[15] E. Viarani. Extraction of traffic informations from images at deis. In *Proc. 10th Intl. Conf. on Image Analysis and Processing*, pages 1073–1077, Venice, september 1999. IEEE Computer Society.

[16] R. Waterfall and K. Dickinson. Image processing applied to traffic, 2.practical experience. pages 60–67, Feb. 1994.