Workshop on Computer Vision System Control Architectures
VSCA 2003

Context analysis to optimize resources in a vision system

Paolo Lombardi[1,2], Bertrand Zavidovique[2] and Virginio Cantoni[1]

[1] Dip. Informatica e Sistemistica, Università di Pavia, Italy
[2] Institut d'Electronique Fondamentale, Université de Paris Sud, France
email: lombardi@vision.unipv.it

1. Introduction and motivations

A most challenging subject in real scene visual analysis is coping with an abrupt change in the environmental conditions. For instance, if the light decreases just when an intruder comes, any surveillance vision system based on background modelling would fail. This kind of failures can be avoided and theoretically the solution is well known: it consists in delivering more information to the system, either with a pre-recorded background model in the second condition or with a concurrent approach insensitive to light changes. In any case, the system needs a run-time analysis of the environmental conditions and their temporal alteration in order to control the active modules.

Increasing the number of modules in a system introduces new problems: how to organise the overall processing, how to integrate data coming from different knowledge sources (e.g. visual modules), how to inhibit the activity of redundant modules in order to optimize the limited computational resources for real-time. In this paper, we regard a vision system as a multi logical-sensor system, each visual operator acting as an independent data source.

Here we advocate for the replacement of image processing with less computationally expensive control, based on information stemming from context. Our aims are: i) to reduce the number of cases of critical failure, ii) to increase adaptability to changing conditions, iii) to pursue an economy of means.

By *context of a scene* we mean the set of assumptions to be made or denied on a scene. We distinguish between *environmental* ("present") and *temporal context* ("past"). Environmental covers scene illumination, discrimination between a static and a dynamic scene, etc. Temporal covers inefficiency due to progressive overload, tracking triggered after detection, etc. We further develop hereafter a computation saving sub-sampled analysis of the video stream in image regions that keep steady over a period of time.

Basic ingredients for the exploiting *context* in that sense are i) calculations to assess reliability of each visual operator in the current situation, ii) an evolving model of the world, and iii) a back-projection of the model onto the present frame. Research has been performed in controlling the structure of a vision system or in monitoring the algorithm at run, both by distributing the control separately in every module [1][2], and by centralising it in a single fusion module [3]. Here we investigate the latter option.

2. Contextual switching in detection modules

We are currently working on some improvements to the traditional outline of a vision system shown in Fig.1: a detector block generates candidate regions of the image to be further processed, while a validation block classifies the target. Each block may contain more than one module, either in a parallel configuration, or in a mutually exclusive relation (a sort of switch between concurrent modules). At the end of the detection/validation chain lays a module dedicated to updating the world model, based on tracking, given the past inform ation and results from the current frame.

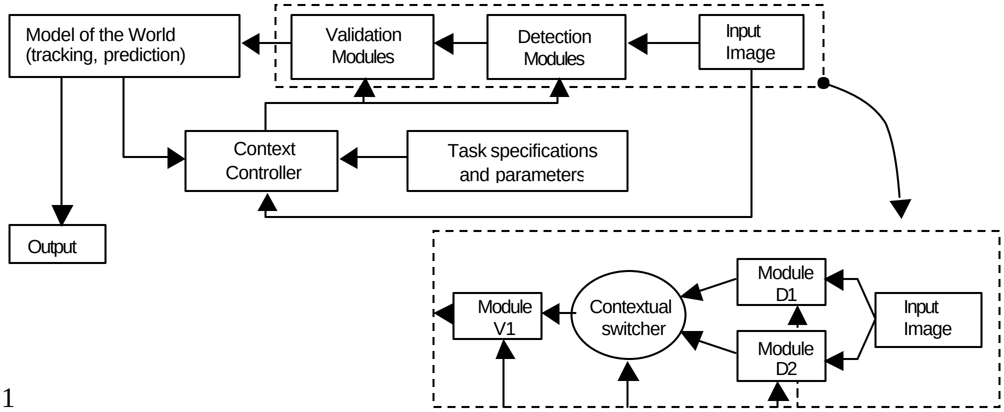


Fig.1

A first improvement consists in controlling the structure of the signal analysis chain by *switching* between two different detection operators according to the environmental situation. This strategy stems from the consideration that any vision operator depends on some hypothesis on the scene to be fully effective. If these assumptions fall, the module critically fails. Traditional fusion schemes (e.g. consensus voting, Bayesian reasoning, etc) could be used, but they would be valid only if the majority of the integrated modules do not fail. On the other hand, the robustness of a system can be increased by providing means of detecting a critical failure of a module, and by excluding it until its normal working conditions are restored. The check may be done either on some parameters of a module's output after elaboration, or on the prior hypothesis mentioned above, or both. In our work, we aim at introducing a logical block called *context controller.* The controller block checks the operative conditions of the main detection module D1, and if they are not correct, switches to the second detector D2.

To investigate the problem, we treat two simplified applications. The first one is the visual surveillance of an indoor scene with a stationary colour camera. The aim here is detecting human intruders. We use a background subtraction operator as the main detection module D1 and a frame difference operator as D2 (refer to figure 2). Background subtraction is more reliable and extracts denser motion information. On the other hand, frame differencing always detects movement and does not need any initialisation or background model. In the validation step, we currently implement just a loose skin colour sensor (V1) that seeks for circular skin regions on top of the detected candidate. Contextual switching takes place whenever the controller marks a global change in the scene. At that point, the stored background model is acknowledged as inadequate and the control of the detection step is passed onto the frame difference module, which needs just one frame from the temporal context to function correctly. When enough static frames have been acquired to instantiate a new background model, control switches back from D2 to D1. Context analysis is performed by detecting changes in a pre-attentive filter computing the mean RGB value over a 10x10 pixels square in a regular grid. The system switches from D1 to D2 if the total number K of altered elements in this "*alert grid*" breaks a threshold K_{th}, and if a fraction B_{th} of K presents an unbalanced shift in the normalised luminance component (R+G+B)/(255*3). Thus, the detection of a contextual switch in this application depends on three thresholds (one to detect an alteration in a single element, one is K_{th}, one is B_{th}). The optimal threshold values strictly depend again on the *environmental context*: camera position, maximum apparent target size, entity of lighting change that invalidates the background model in D1, total area of image regions that can be interested by such a change are all elements playing a role in determining these values. By specifying them by hand at system set up, a human operator practically transfers its capability of perceiving a change into a numerical representation that can be also discriminated by a computer.

The second sample application concerns detecting pedestrians crossing the street from a moving grey-level camera mounted on-board a car. Here, D1 is an optical flow operator selecting image regions that present a distinct horizontal component. The module dramatically fails when the car turns, for then all the image moves horizontally. D2 is a filter similar to a Moravec interest operator, which in practice shows a high response when many vertical edges are present in a region. Certainly this feature is less characteristic of a crossing pedestrian than a marked horizontal movement, but as it works under no particular assumption, it can perform the detection step even if the optical flow fails. V1 is a vertical symmetry module [4] that checks the candidate areas. In a complete robot system, if the vehicle turns, the controller could receive this contextual information from ego-sensors like wheel inclination and vehicle velocity. In our case, no such sensors are available, so we rely only on image analysis to infer module failure. The context controller integrates a pre-attentive filter organised in a regular grid as mentioned above. It stores the mode of the grey-level histogram. The mode is less sensitive to region variations than the mean value. This helps in understanding if the car is following the road or is performing a turn: in the latter case, a number of squares *N* on one image side move their mode from a value typical of the street colour to a different one. Moreover, the configuration of the active squares becomes asymmetric with respect to the image central axis. The switching mechanism is alerted if *N* and a parameter estimating the asymmetry break a threshold. Compared to the application in video surveillance, here the optimal thresholds are more sensitive to the different test sequences and must be set by hand each time. An ego-sensor would probably prove a better choice.

At present, only the surveillance system updates a tracker (acting as model of the world). A simple frame-to-frame one-step matching follows the target and feeds its predictions back to the controller in the form of a binary *prediction grid* where elements are set to 1 if changes are expected in their region. A Boolean subtraction between the *prediction grid* and the *alert grid* detailed above actually accounts for global change detection to contextual switching.

3. Activation map

A second improvement is the introduction of an *activation map* in every detection module. The image is divided into a square grid, and operators perform computations only in those elements marked as *active* in the map. This mechanism allows for a distributed sub-sampling of the video sequence. The idea is that if the context in a sub-region of the scene does not change over time, no computational effort is likely to be spent on it because the contained information is already available from previous analysis.

Each element is given two integers describing its state: an *attention threshold* and a *charge*. Moreover, the last computed value of the feature in that element is stored. The *charge* decays from frame to frame, for instance linearly, until it falls under the *attention threshold* for that element. The operator then extracts a new value of its corresponding feature in the element region and checks the distance according to a suitable metric from the value in the memory. The currently tested control policy is as follows: if a change in the feature value is detected, the attention threshold is raised to the top level, otherwise it decreases. And in both cases, the charge is raised back to the top level (see Fig.2).

Every element thus updates its values asynchronously from others, maybe in different frames. Of course, this sub-sampling lowers the computational burden at a given frame and speeds up the process. Conversely, it also ties the system reaction to incoming information available only during the next activity cycle of the element. The system may even loose this piece of information if it disappears before onset (see Fig.3). In order to avoid such limitations, the pre-attentive filter run by the context controller (see section 3) acts as a *guardian*: as soon as a change is detected, the controller raises the attention thresholds of the corresponding elements in the activation maps of the detectors so that the new frame be immediately processed locally.

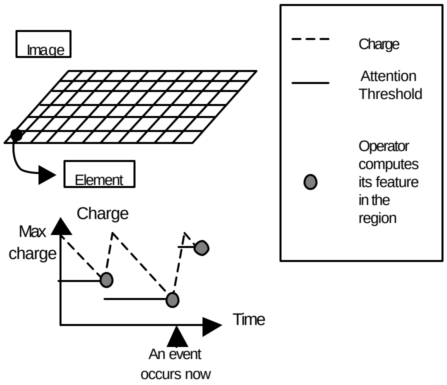


Fig. 2

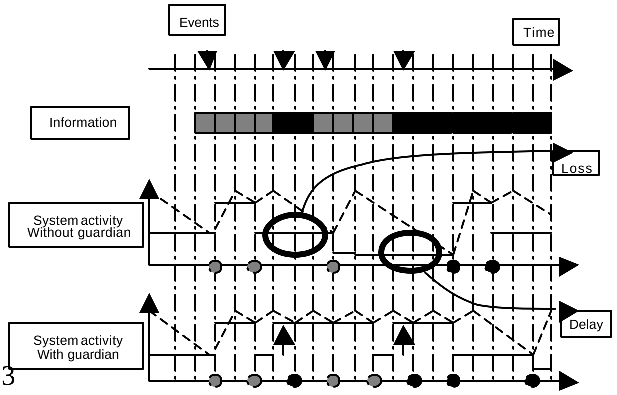
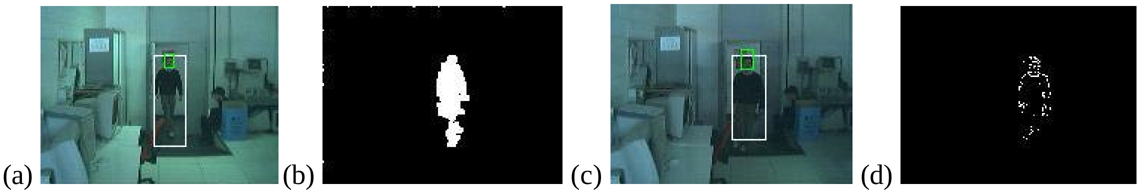

Events
Time
Information
Loss
System activity Without guardian
Delay
System activity With guardian

Fig.3

4. Preliminary results

As to video surveillance, the output of the background subtraction D1 in normal operation conditions is shown in Fig. 4(a) and 4(b). Fig. 4(c) and 4(c) display the system output when D2 is operative after a lighting change (room light switched off).

Fig.4



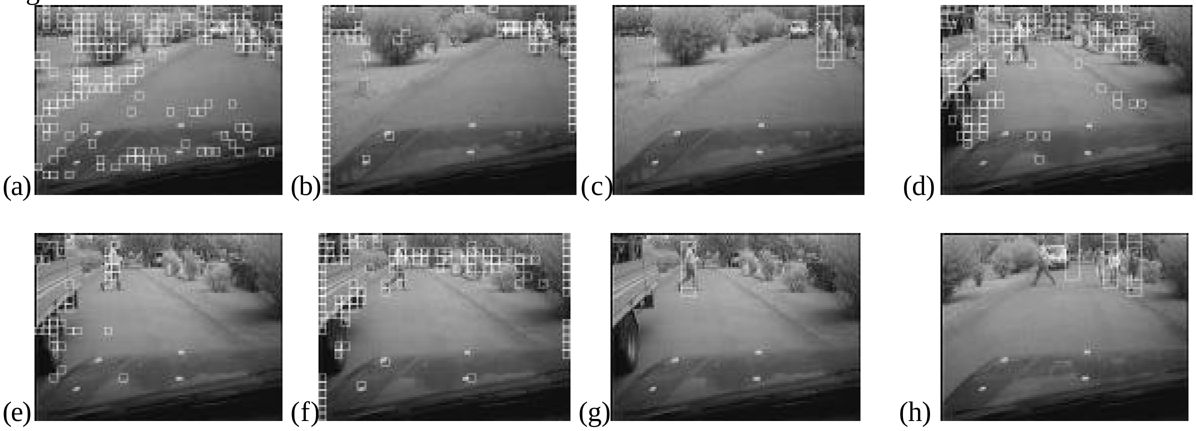
(a) (b) (c) (d)

Fig. 5(a) and 5(b) show the response of module D1 (optical flow) and D2 (Moravec interest), respectively, from a frame of a turning car in the experiment of pedestrian detection mentioned in section 2. As the vehicle is turning, D1 is inactive and the output of the detection step is the one of Fig. 5(b). Fig. 5(c) displays the system output after validation.

The same responses are shown in figures 5(e)-(g) as an example of the vehicle proceeding straight. Here candidate generation is based on the output of D1, displayed in Fig. 5(e).

Some problems are shown below. Fig. 5(d) depicts the output of D1 in a frame where small lateral movements were not detected by the context controller and switch to D2 was not performed. The vertical symmetry module, used alone in the validation step, produces errors like in Fig. 5(h).

Fig. 5



(a) (b) (c) (d)

(e) (f) (g) (h)

References

[1] K.Toyama, G.Hager, “Incremental Focus of Attention for Robust Vision-based Tracking”, *Int. J. Computer Vision*, v.35(1), pp.45-63, 1999.

[2] K.Toyama, E.Horvitz, “Bayesian Modality Fusion: Probabilistic Integration of Multiple Vision Algorithms for Head Tracking”, *Proc. ACCV’00, Fourth Asian Conference on Computer Vision*, Tapei, Taiwan, 2000.

[3] M. Spengler, B.Schiele, “Towards Robust Multi-cue Integration for Visual Tracking”, *Lecture Notes in Computer Science*, v.2095, pp.93-??, 2001.

[4] M.Bertozzi, A.Broggi, A.Fascioli, P.Lombardi, "Vision-based pedestrian detection: will ants help?", in *Proc. of IEEE Intelligent Vehilces 2002*, Versailles, June 2002.