TIRANO SYSTEM WITH LEARNING CAPACITY FOR THE DETECTION OF MOVING TARGETS

Miguel A. Fernandez[1], A. Fernandez Caballero[1], IEEE Member, Juan Moreno[2] & Gabriel Sebastian[2]

Departamento de Informatica
Escuela Universitaria Politécnica de Albacete
Universidad de Castilla-La Mancha, 02071 - Albacete, Spain
email: [1]{miki, caballer}@info-ab.uclm.es
[2]{jmoreno, gsebas}@sancho.info-ab.uclm.es

Abstract. A system based on the charge-discharge characteristics of the neural synapses of the visual path, is shortly introduced. The proposed system uses the LSR (length/speed ratio) descriptor. After image segmentation, the LSR characteristic is used to detect and classify motion in the scene. This paper mainly focuses on its simulator named TIRANO and in particular on its learning sessions and results. That's the reason why TIRANO learning capacity is demonstrated for the detection of moving targets in the following cases: (a) recognition of a family of synthetic moving elements, (b) recognition of real elements, (c) recognition of several positions of the same object, and, (d) recognition of characteristic motion situations.

1. INTRODUCTION

Detection of moving elements has been largely studied over the past decades. An excellent classification of computational models in pattern motion can be found in [Ser93]. Models of local motion detection are the gradient models ([Fen79] [Hor81] [Mar81] [Law89]), the correlation models ([Has56] [Bar65] [Ade85] [Wat85]) and the image difference models ([Fu88] [Sch89]), whereas models of pattern motion measurement can be divided into models that incorporate multiple motion constraints ([Fen79]), matching models ([Tho81] [Pra83]), and models that use a smoothness constraint ([Hor81] [Hil84] [Koc86] [Yui88]). We propose in this paper our particular solution to the detection of moving elements.

The developed system is based on the charge-discharge characteristics of the neural synapses of the visual path ([Fer92] [Fer95]). By analysing the problem of the detection of moving targets in image sequences, we found that one relevant point was the time of presence of a specific grey level in the same coordinate of the scene. A fast target is present during a shorter time than a slower one, and the greater the size of the target the longer time it stays on each pixel of the sensor. The descriptor `length/speed ratio´ (LSR) was therefore studied.

We define the LSR of a moving zone as the ratio between its length L in the direction of movement and its speed S. This descriptor can be interpreted as a measure of the permanence of a given zone over the sampling point. In other words, the LSR value measures the time that a certain element of the image activates a particular coordinate in the array of sensors. In the case of points at which the image is static, there is no substantial modification of grey levels, so a saturation value is generated for the LSR. So, this value is not significant.

The LSR characteristic is invariant under changes in the distance between the sensor and the moving element. It is also invariant under changes in the direction of movement, but it is not invariant under rotations of the moving element with respect to the direction of movement.

This LSR relation can be locally computed, it permits multilayer processing and presents good discriminating features, it is able to learn, and thus it means it is neural.

2. TIRANO SYSTEM

In order to solve this problem, a network made up of three layers has been used:

1. *Segmentation.* The variations in the image are detected.

2. *Selection and extraction of characteristics.* The LSR characteristics set used to classify motion in the scene is detected and selected.

3. *Classification and association*. The classes linking the elements of the learning family are differentiated.

TIRANO is the simulator for the described problem. This system has learning capacity through a training period in which the system is shown, beside the video signal, an assistance signal indicating the elements appearing in each scene. The system learns to classify motion situations or elements and is able to differentiate them later [Fer97].

3. SIMULATION

The simulation is designed with the possibility to control and debug learning and computational processes. The simulator uses an undefined sequence of photograms as input and performs the same process as the proposed network. The simulator processes the information frame by frame allowing the input of a new frame when the previous one has already been processed.

The working method can be divided into several steps:

1. *Design and assembly of the training, test and operation sequences*. The scene backgrounds and moving elements are designed by means of different simulator tools.

2. *Design of the assistance values*. Assistance values are generated simultaneously using the same routines.

3. *Preparation and definition of the state*. The simulator defines the values of all parameters of the system. The system provides a full set of predefined values for all parameters, so you can properly work with them. Nevertheless, they can be modified for specific applications before starting the learning sessions or at any point of the process.

4. *Trace and analysis of the learning process and behaviour*. The simulator allows the analysis of the evolution of variables behaviour and the system modules' state of activation.

The management of image sequences presented to the system during the learning session is essential. A learning process can imply several sessions, and some of them can even be substituted by any other during the process. Modifications of system parameters or alterations of system age during the learning process can also be accomplished. This permits to analyse the influence of these modifications in the process, or to introduce changes into the learning process or into the running process in a controlled manner. The final state of the system will depend upon all these processes.

The simulation trace management is directed to the control of the trace, allowing the definition of the level of information supplied to the user. Some possibilities are allowed trough this option. The definition of the simulation stops after the process of the layers 1 (segmentation), 2 (characteristic extraction) or 3 (classification), or when the number of processed photograms is multiple of a constant; or the possibility to save the state of the system when the number of processed photograms is multiple of a specific constant.

The management of system parameters allows, beside other modifications, to change the parameters of the three layers and those of the system. In the first layer, operations such as segmentation activation on the digitising card can be performed to see in the monitor the binary image. For the case of layer 1, the simulator allows to show all data related to the realised process. Monitoring of the state of layer 2 consists in showing state and activation values of the modules of this second layer as well as the value of the state variables. Therefore all data related to the LSR characteristic are shown. Data associated to the monitoring of the layer 3 state will be all those regarding associative and class modules. The working stage parameters of the system are shown in each layer, as well as the number of sequence within the present session, number of photogram in sequence, number of repetition of the sequence, number of photograms processed and age of the system.

4. LEARNING SESSIONS AND RESULTS

We next describe several examples of learning sessions. Consider the fact that the system used for all examples is exactly the same. Its learning capacity allows the system to adapt itself to the environment it is shown with no intervention at all.

4.1. Recognition of a family of synthetic moving elements.

The aim of this learning exercise is to achieve that the system learns to recognise a set of moving elements, being able to detect them and differentiate them even if they are defective or accompanied by noise. Here we choose a family of elements with very well known characteristics (see figure 2a), allowing thus to predict and check the system performance.

During the learning process a scene composed of a static transparent background (figure 1a) and the set of moving elements (figure 2a), all of them at same speed, is shown to the system accompanied by their respective assistance values. This will cause the LSR characteristic to appear in the scene.

After the network training process, different types of sequences are offered. In first place a complete sequence with all the elements of the learning family are shown, with no noise, in perfect state and with no assistance values (figure 2a). The network recognises all moving elements (see table 1). In second place, in a non-transparent background (figure 1c), the detection of one of the elements is accomplished defectively (table 2). When a zone of a mobile overlaps with a zone of the background with the same grey level (and there does not exist any intermediate grey level variation), the LSR value is lost, due to the fact that the charge values on the static background were previously saturated. In order to check up to what level the network is able to differentiate satisfactorily the elements in the scene, a set of defective elements has been generated (see figure 2b), both on a transparent background (figure 1a) and on a non-transparent background (figure 1c), yielding highly satisfactory results (see tables 3 and 4, respectively).

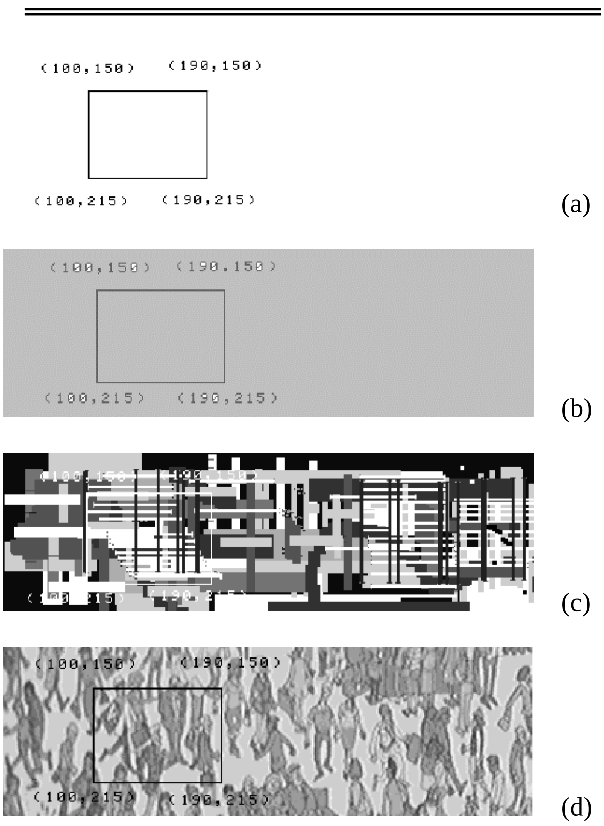

Figure 1. Different backgrounds. (a) Transparent. (b) Uniform. (c) Synthetic. (d) Wally.

In all presented tables *Shown Element* is the icon moving through the scene, *Expected Response* should be the correct answer of the system, while *System Response* corresponds to the element that has really been recognised by the system. We offer one more of all available system's variables. *Certainty* shows the certainty of the system's decision on the recognition of the element.

Table 1: Results obtained for perfect synthetic icons on a transparent background

Shown Element	Expected Response	System Response	Certainty
S0	0	0	100
S1	1	1	100
S2	2	2	100
S3	3	3	100
S4	4	4	100
S5	5	5	100
S6	6	6	100
S7	7	7	100

Table 2: Results obtained for perfect synthetic icons on a non-transparent background

Shown Element	Expected Response	System Response	Certainty
S0	0	0	100
S1	1	1	100
S2	2	2 / 7	46
S3	3	3	100
S4	4	4	100
S5	5	5	100
S6	6	6	100
S7	7	7	100

Table 3: Results obtained for non-perfect synthetic icons on a transparent background

Shown Element	Expected Response	System Response	Certainty
DS0	0	0	81
DS1	0	6	100
DS2	2	2	81
DS3	2	2 / 7	40
DS4	2	5	62
DS5	5	5	81
RS0	1	1	100
RS1	1	1	100

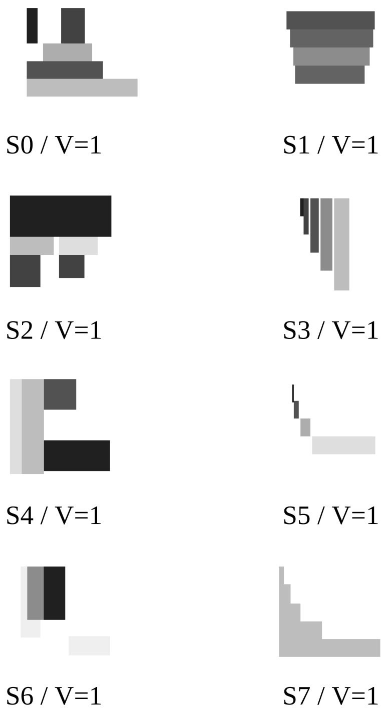

Figure 2a. Moving elements. Perfect synthetic.

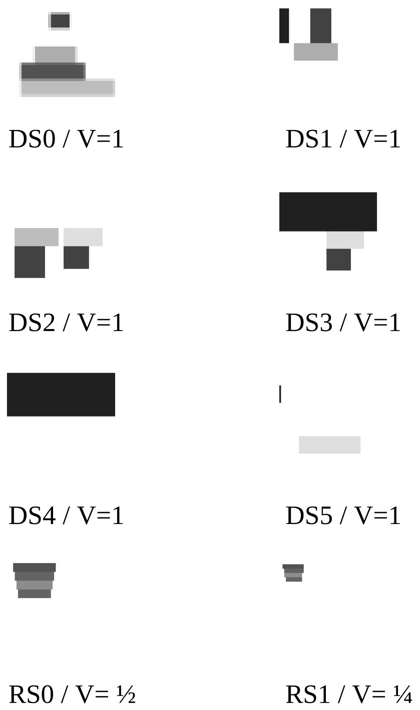

Figure 2b. Moving elements. Non-perfect synthetic.

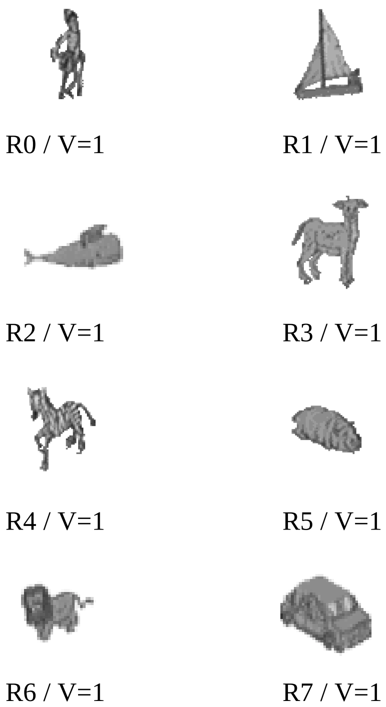

Figure 2c. Moving elements. Perfect real.

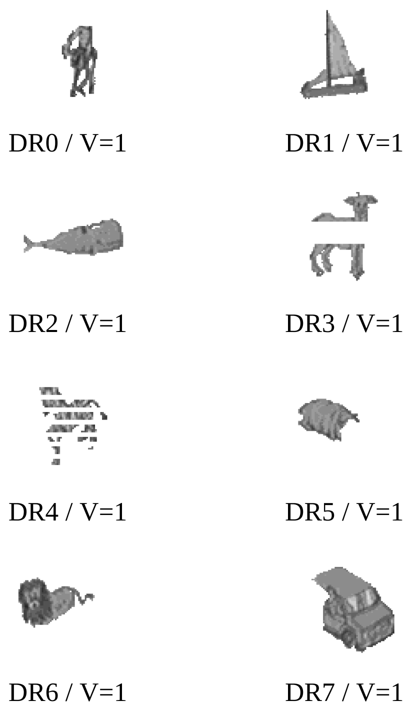

Figure 2d. Moving elements. Non-perfect real.

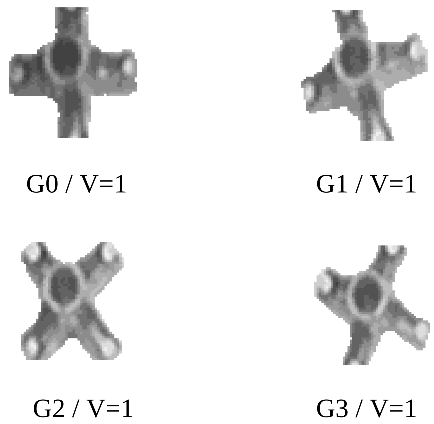

Figure 2e. Moving elements.
Several positions of a same element.

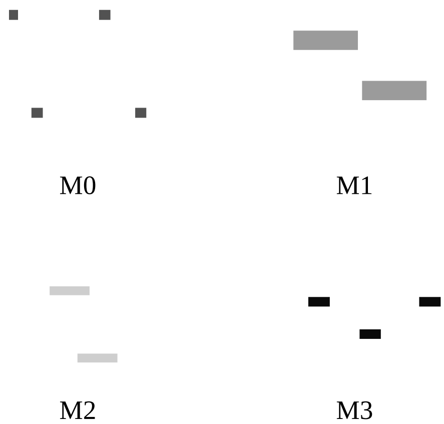

Figure 2f. Moving elements.
Characteristic motion situations.

Table 4: Results obtained for non-perfect synthetic icons on a non-transparent background			
Shown Element	Expected Response	System Response	Certainty
DS0	0	0	81
DS1	0	6	100
DS2	2	6	99
DS3	2	2 / 7	40
DS4	2	5	62
DS5	5	5	81
RS0	1	1	100
RS1	1	1	81

4.2. Recognition of real elements.

This time the moving elements are extracted from real images taken by a TV camera (figure 2c). So their characteristics are unknown. Following the steps of the previous example –this time, the background is a uniform non-transparent one (see figure 1b)-, the results obtained confirm that the network is not only effective for synthetic working environments but also for real environments.

We offer in tables 5 through 7, the results obtained using the real perfect elements of figure 2c on the uniform non-transparent background of figure 1b (table 5), those of the real perfect elements of figure 2c on the non-uniform non-transparent background of figure 1d (table 6) and , finally, those of the non-perfect elements of figure 2c on the non-uniform and non-transparent background of figure 1d (see table 7).

This offers the possibility to use the network for numerous applications where it is usual to modify the elements to be detected. Our system contains the necessary operative and evolutionary resources to require just a single learning session with a predetermined number of photograms properly accompanied by assistance.

Table 5: Results obtained for perfect real icons on an uniform background			
Shown Element	Expected Response	System Response	Certainty
R0	0	0	95
R1	1	1	97
R2	2	2	98
R3	3	3	97
R4	4	4	96
R5	5	5	95
R6	6	6	93
R7	7	7	100

Table 6: Results obtained for perfect real icons on a non-transparent non-uniform background			
Shown Element	Expected Response	System Response	Certainty
R0	0	0	95
R1	1	1	97
R2	2	2	98
R3	3	3	97
R4	4	4	96
R5	5	5	95
R6	6	6	93
R7	7	7	100

Table 7: Results obtained for non-perfect real icons on a non-transparent non-uniform background			
Shown Element	Expected Response	System Response	Certainty
DR0	0	0	95
DR1	1	1	97
DR2	2	2	98
DR3	3	4	97
DR4	4	4	96
DR5	5	5	95
DR6	6	6	93
DR7	7	7	100

4.3. Recognition of several positions of the same object.

In this session we intend to illustrate how the lack of invariance of the LSR characteristic can be used in situations of orientation change of the mobile with respect to motion direction.

During the learning process a scene composed of a static background (figure 1a) and the same moving element which appears with different orientations with respect to the direction of the speed vector (figure 2e), but always at the same speed, is presented to the system. For this session, we extract a set of elements from real images obtained with a black and white TV camera.

During the stage of operation, the network recognises all orientations of the moving element, thus proving the starting hypothesis (see results on table 8).

Table 8: Results obtained for the detection of various orientations of the same element			
Shown Element	Expected Response	System Response	Certainty
G0	0	0	100
G1	1	1	100
G2	2	2	100
G3	3	3	100

4.4. Recognition of characteristic motion situations.

In this process we want the system to detect characteristic motion situations. The system should also be able to associate them to the corresponding assistance values.

For this aim a set of simple moving elements has been designed (see figure 2f). Each mobile may have a different speed (in pixels per frame), as can be observed in table 9. We assume that each combination of motion is characteristic of a system's state and our goal is that it differentiates at each moment the state of the scene being observed. An example of such a system may be a traffic control system where the fluidity of vehicles actually represents the characteristic of the state of the scene. Results are presented on table 10.

Table 9				
Situations	M0 speed	M1 speed	M2 speed	M3 speed
SM0	1	1	1	1
SM1	½	½	1	1
SM2	1	1	½	½
SM3	1	½	½	1
SM4	2	1	1	2
SM5	1	2	2	1
SM6	½	1	2	2
SM7	2	2	1	½

Table 10: Results obtained for the detection of characteristic motion situations			
Shown Element	Expected Response	System Response	Certainty
SM0	0	0	100
SM1	1	1	100
SM2	2	2	100
SM3	3	3	100
SM4	4	4	80
SM5	5	5	80
SM6	6	6	80
SM7	7	7	80

5. CONCLUSIONS

After considering the results obtained for all of these learning sessions it can be stated that all our proposals are valid. We have demonstrated through the shown examples the versatility of our system in adapting to a great number of different training sets with no change in the basic structure.

Some examples of this type of applications can be flow analysis or traffic control, moving elements detection and classification on a conveying belt or on real extern backgrounds, or other applications with similar characteristics.

Our research group actively goes on working on this field.

Acknowledgements

We are thankful to Professor Jose Mira in his personal and financial support.

Bibliography

[Ade85] E.H. Adelson & J.R. Bergen. (1985). Spatiotemporal energy models for the perception of motion. Journal of the Optical Society of America A, 2, 284-299.

[Bar65] H.B. Barlow & R.W. Levick. (1965). The mechanism of directional selectivity in the rabbit's retina. Journal of Physiology 173, 477-504.

[Fen79] C.L. Fennema & W.B. Thompson. (1979). Velocity determination in scenes containing several multiple moving objects. Computer Graphics and Image Processing 9, 301-315.

[Fer92] M.A. Fernandez & J. Mira. (1992). Permanence memory: A system for real time motion analysis in image sequences. IAPR Workshop on Machine Vision Applications, 249-252. MVA'92, Tokyo.

[Fer95] M.A. Fernandez, J. Mira, M.T. López , J.R. Alvarez, A. Manjarrés & S. Barro. (1995). Local accumulation of persistent activity at synaptic level: Application to motion analysis. From Natural to Artificial Neural Computation, IWANN'95, 137-143. Springer-Verlag, Germany.

[Fer97] M.A. Fernandez. (1997). Una arquitectura modular de inspiración biológica con capacidad de aprendizaje para el análisis de movimiento en secuencias de imagen en tiempo real. Tesis Doctorales, 48. Universidad de Castilla-La Mancha. Ph. D. thesis.

[Fu88] K.S. Fu, R.C. González & C.G.S. Lee. (1988). Robótica: Control, detección, visión e inteligencia. McGraw-Hill, Madrid.

[Has56] B. Hassenstein & W.E. Reichardt. (1956). Functional structure of a mechanism of perception of optical movement. Proceedings of the 1st International Congress of Cybernetics in Namar, 797-801.

[Hil84] E.C. Hildreth. (1984). The measurement of visual motion. The MIT Press.

[Hor81] B.K.P. Horn & B.G. Schunck. (1981). Determining optical flow. Artificial Intelligence 17, 185-203.

[Koc86] C. Koch, J. Marroquin & A. Yuille. (1986). Analog "neuronal" networks in early vision. Proceedings of the National Academy of Sciences USA 83, 4263-4267.

[Law89] T.B. Lawton. (1989). Outputs of paired Gabor filters summed across the background frame of reference predict the direction of movement. IEEE Transactions of Biomedical Engineering, 36, 130-139.

[Mar81] D. Marr & S. Ullman. (1981). Directional selectivity and its use in early visual processing. Proceedings of the Royal Society (London) B, 211, 151-180.

[Pra83] J.M. Prager & M.A. Arbib. (1983). Computing the optic flow: The MATCH algorithm and prediction. Computer Vision, Graphics and Image Processing, 24, 271-304.

[Ser93] M.E. Sereno (1993). Neural Computation of Pattern Motion. The MIT Press.

[Sch89] R.J. Schalkoff. (1989). Digital image processing and computer vision. John Wiley and Sons, Canada.

[Tho81] W.B. Thompson & S.T. Barnard. (1981). Lower-level estimation and interpretation of visual motion. IEEE Computer 14, 20-28.

[Wat85] A.B. Watson & A.J. Ahumada. (1985). Model of visual-motion sensing. Journal of the Optical Society of America A, 2, 322-341.

[Yui88] A.L. Yuille & N. Grzywacz. (1988). A computational theory for the perception of coherent visual motion. Nature 333, 71-74.