A REAL TIME SIMULATED HUMAN VISION SYSTEM USING CONNECTIONIST MODELS APPLIED TO TARGET TRACKING

Hidenori Inouchi, Niall Mc Loughlin
Hitachi Dublin Laboratory, Dublin 2, Ireland

David Vernon
Computer Science Department, Trinity College, Dublin 2, Ireland

ABSTRACT

This paper describes an algorithm and an architecture of a simulated human vision system using connectionist models. An original approach to solve two major problems, occlusion and collision, is proposed for a multiple target tracking system. This approach uses the recall properties of various neural network models. Currently, the system has been built on 64 node Transputer system with 2 MB memory on each Transputer. This system employs both image-intensity-based optical flow methods to compute motions of multiple targets, and token-based methods to track a designated target. Preliminary results using simple multiple objects in an unclutterd scene are also presented.

1. INTRODUCTION

The mechanism of the human visual system to perceive motion has been investigated in the domains of medicine and biology. The human visual system may be viewed as a truly massive parallel system of simple compute elements. The two basic methods used at present in machine vision for motion processing are the token based methods and the optical flow based methods. There is psychological evidence that both methods are used. The neurobiological study of the retina and visual cortex has produced a number of massively parallel neural models.

Artificial neural networks have been studied for many years in the hope of improving machine performance in a number of complex domains. All the computational elements or nodes operate in parallel and are interconnected via weights that are typically adapted during use to improve performance.

An enormous quantity of neural models exist at present - some computationally very efficient at certain tasks and others which follow neurobiological data closely. A considerable amount of research is being performed on using neural models as classification networks . However the neural models in our system perform the various tasks of optical flow calculation (*MOC* filter), centroid location (instar-competitive, outstar-competitive neural network), and the calculation of desired camera motion (adaptive novelty filter).

In this paper, a hybrid system is described which makes use of a number of adaptive neural network models to track multiple targets. The paper is split up into five sections. First, an overview of the system is presented. Second, the motion computation of the system and a new method to partially overcome the occlusion problem is described. Third, the current design of our motion analysis system which will help reduce the problems associated with partial occlusion and collision is presented. Fourth, results from the testing are presented to show the feasibility of the proposed approach. Finally, the work to date is summarised.

2. SYSTEM OVERVIEW

The system is split up into two separate co-operating subsystems (FIGURE 1). Both the Interesting Point Locator Subsystem (*IPLS*) and the Motion Tracking Subsystem (*MTS*) receive a series of "snapshots" of the moving objects. The *IPLS* calculates the approximate location of the centroids of all objects in motion from the snapshots presented. This information is passed to the *MTS*. The *MTS* uses this information to analyse the object motions and to drive the eye motor which in turn drives the camera in the direction of target motion.

3. MOTION COMPUTATION

A simplified version of motion oriented contrast-sensitive filter (*MOC* filter) [1] was used to compute local motion of moving edges in the image. This filter is composed of four levels of computation. The main components of this filter are Sustained Response Cells, Transient Response Cells, and Local Motion Detectors. These components are used to produce optical flow measurements between two successive frames ($I_{ij}(t_1)$, $I_{ij}(t_2)$) which are sensitive to both direction of contrast and direction of motion. Our implementation has eight directional resolutions.

The local motion signals of $U_{ij}(k, t)$ at each position (i, j) in direction k at time t are spatially filtered by a Gaussian filter and normalized by the following equation :

$$\varepsilon \frac{d}{dt} V_{ij}(k, t) = -\alpha V_{ij}(k, t) + (A - V_{ij}(k, t))\mathcal{NE}_{ij} - V_{ij}(k, t)\,\mathcal{NI}_{ij} \qquad (1)$$

Where $\mathcal{NE}_{ij} = U_{ij}(k, t)$

$$\mathcal{NI}_{ij} = \sum_{(u,v)\neq(0,0)} G_{\sigma}(u,v)\,U_{(i+u)(j+v)}(k, t)$$

$$G\sigma(\mu, \nu) = \frac{1}{2\pi\sigma} Exp\left(-\frac{u^2+v^2}{2\sigma^2}\right)$$

We call $V_{ij}(k, t)$ spatio-temporally normalized velocity. This corrected velocity is used to integrate the local motion signals to increase the reliability derived from *MOC* filter and to a suppress the temporary influence due

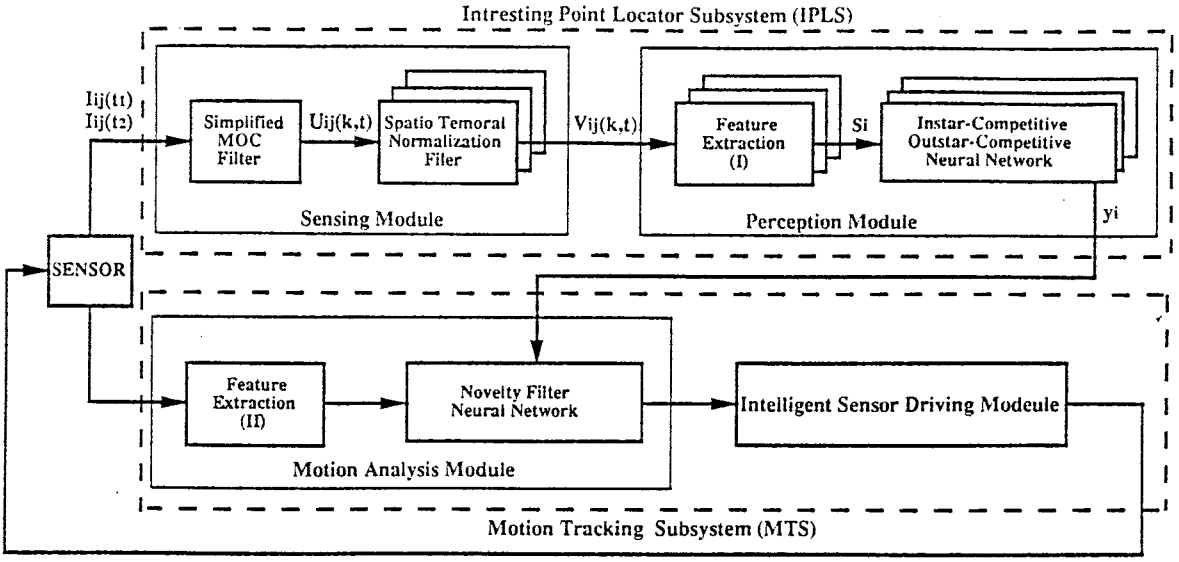

FIGURE 1 Target Tracking System by Neural Network

to passing objects which occupy an extremely high velocity field compared to others. $V_{ij}(k, t)$ is initially projected, by summation, to both the X and Y axis to produce $1D$ characteristic feature of the corrected velocity field in the feature extraction module (see FIGURE 2), and then passed to the Perception module to obtain a global representation of the moving objects. To make explanation simple, we assume that each object moves in the same direction, and therefore only two waveforms are produced. These projected $1D$ waveforms are then input into each instar-competitive, outstar-competitive neural network (see FIGURE 3) to locate the centroid(s) of the moving object(s).

Each waveform, which is analog by nature, is presented to the *F1 and F3* layer in parallel. The instar (fan-in)-competitive neural network is used to adaptively code the input waveforms. The *F2* layer is an on-center/off-surround competitive network. This performs contrast enhancement and noise suppression automatically. The activation of each node in *F2* layer is governed by the following cell-membrane equation :

$$\frac{dx_j}{dt} = -\alpha x_j + (A - x_j)[\sum_{i=0}^{n} w_{ji} s_i + f(x_j)] - x_j \sum_{i \neq j}^{n} f(x_i) \qquad (2)$$

where x_j is the activation of node j, w_{ji} is the weight from node i in *F1* layer to node j in *F2* layer, s_i is the external input to node i in *F1* layer, A is a constant which limits the maximum activation, α is a decay constant and f is chosen to be a sigmoidal function. This instar-competitive network extracts local maximum values from the input waveform and keeps them in a short term memory (*STM*) created in *F2* layer.

The learning rule for instar coding is given by

$$\frac{dw_{ji}}{dt} = \varepsilon(t)\,[\, s_i - D w_{ji}\,]\, x_j \qquad (3)$$

where x_j and s_i are as before. $\varepsilon(t)$ is a gain control function. D is a decay constant. Equation (3) means that the reverberated signals in *F2* layer are stored in a Long Term Memory (*LTM*) denoted by w_{ji}.

The outstar (fan-out)-competitive neural network is used to perform spatial pattern learning. The node activation is controlled by the following cell-membrane equation:

$$\frac{dy_i}{dt} = -\alpha y_i + (A - y_i)[\sum_{j=0}^{n} z_{ij} x_j + g(y_i) + s_i\,] - y_i \sum_{j \neq i}^{n} g(y_j) \qquad (4)$$

where y_j is the activation of node j. z_{ij} is the weight from node j in *F2* layer to node i in *F3* layer. s_i is the external input to node i in *F3* layer. g is chosen to be any linearly continuous function. A, α , and x_j are as in Equation (2). Note that Equation (4) slightly differs from Equation (2). Additional term s_i was added for inherent outstar learning scheme.

The learning rule for outstar coding is given by

$$\frac{dz_{ij}}{dt} = \varepsilon(t)\,[\, y_i - D z_{ij}\,]\, x_j \qquad (5)$$

where x_j is the activation of node j in *F2* layer. $\varepsilon(t)$ and D are as in Equation (3). When psychological analogies are applied, x_j is interpreted as a conditional stimulus and is differentiated from the external stimulus s_i which is interpreted as an unconditional input [2][3][4].

Once this network is trained, it can recall learned patterns in the *F2* layer, even if the external unconditional stimulus s_i is absent. This property is used to overcome some of the problems of occlusion. When occlusion occurs, the input waveform corresponding to the occluded target decreases gradually owing to Equation (1). However, the instar-competitive network memorizes the last local maximum values in its *STM* for a short while and thus the outstar-competitive network automatically recalls the last known position of the

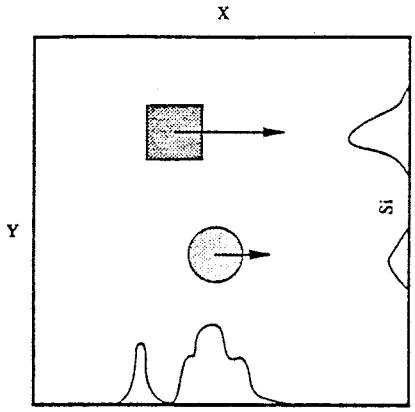

FIGURE 2 Moving Objects and anticipated projected profile of the velocity field due to vertical edges (k=0)

tracked (occluded) object. This information can be used to predict future position of the occluded objects in a limited way, if the object is moving in the same direction as before. Centroidal information can be reliably extracted from the nodes in the *F3* layer by simple thresholding.

To consider instar-competitive network as peak hold circuit and outstar-competitive as non-volatile memory facilitates the understanding of these networks.

4. MOTION TRACKING

Once centroidal information of each object is obtained by the Perception module mentioned above, then this information is extensively used to focus the attention on tracked objects in the motion analysis module.

This module has been designed to overcome some of the problems of object collisions. The motion analysis module is built around an adaptive novelty filter [5][6]. The novelty filter is an adaptive filter which is based on mechanisms close to associative memories. The filter is "taught" a series of image features. The training of the network is controlled by the following equation :

$$\frac{dw_{ij}}{dt} = -\varepsilon(t)\,\alpha\, x_i\, x_j \qquad (6)$$

Where x_i, x_j are position, rotation, and scale independent feature vector elements such as log-polar-fourier representation of image features, $\varepsilon(t)$ is a gain control function, and α, which may vary between 1.0 and 0.0, can be used to control the functionality of the filter as follows. When α is close to 0.0 the network converges completely, and if the same pattern is presented to the network later no novelties will be detected - the pattern is learned. If the pattern is presented and is slightly different the second time around, the novelty filter outputs these differences or novelties. However, if α is selected close to 1.0 and the network is addressed with an obscured pattern (eg from partial occlusion) the complete learned pattern is recalled. In this case, the novelty filter acts as an associative memory, which completes its recall in one step.

The novelty filter also distinguishes novelties related to added parts from novelties related to missing parts. This very useful property is exploited in our system design to quantify the amount of desired motion required to cope with collisions and partial occlusions.

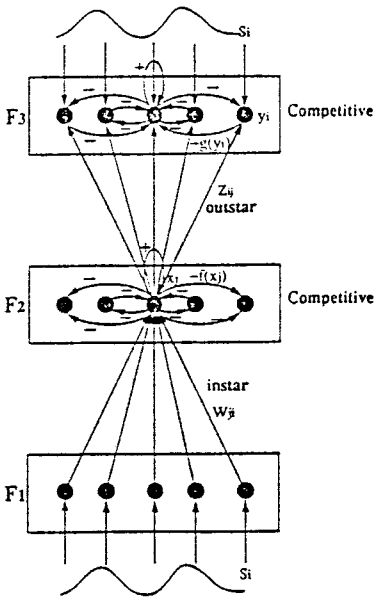

FIGURE 3 Instar-Competitive, Outstar-Competitive Neural Network for computing single centroid coordinate in single direction

The output of novelty filter is sent to an intelligent sensor driving module and used to drive a camera in the direction of the desired objects motion.

5. EXPERIMENTAL RESULTS

We have tested the network's ability to resolve the occlusion and collision problem in a rather artificial scene. FIGURE 4 shows the snapshots of the activities in the Sensing Module. FIGURE 5 and 6 show the time evolution of the activities of nodes in the Perception Module. Since it is almost impossible to show all the activities in each network, only the partial results are shown here. In these results, typical parameters used were: ε = 1000.0, σ = 3.0, α = 0.1, A = 1.0, D = 1.0, n = 50. The initial value of node and weight was 0.1, and 0.0 respectively.

We initially carried out the experiments on Sun4 in C with X-window interface. Currently the system is fully implemented on a 64 nodes Meiko Transputer System with Sun4 as a front-end machine. Our current implementation is a mixture of OCCAM2 and C. The native parallelism of neural networks is fully described in OCCAM2. Each process has been implemented to run in parallel with some spreading over a large number of transputers.

6. CONCLUSION

A model of multiple target tracking system has been developed . This model relies on the existing neural models of human vision. There are a lot of biological details which could not be incorporated into our system, such as speed selectivity for the direction selective cells. However, our system is fully adaptive and it can be trained either on-line or off-line, and either supervised or nonsupervised. It can be concluded that our system can work effectively in real time applications.

Acknowledgements

The authors would like to thank Nobuo Hataoka of Hitachi Dublin Lab for his continual support of our research, to Dr. Masakazu Ejiri of Hitachi Central

Research Lab for his useful suggestion and encouragement, and Ronan Waldron and Massimo Tistarelli of Trinity College for their useful advice.

References

[1] S.Grossberg and E.Mingolla, "Neural Dynamics of Motion Segmentation: direction fields, apertures and resonant grouping", *in Proc. of IJCNN*, pp. 11-14, 1990.
[2] G.A.Carpenter, and S.Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.
[3] S.Grossberg, editor, "The adaptive Brain", North-Holland, Amsterdam, 1988.
[4] S.Grossberg and G.Carpenter, "Art2: Self-organization of stable category recognition codes for analog input patterns", *Applied Optics*, vol. 26(23), pp. 4919-4930, 1987.
[5] T.Kohonen, "Self-Organization and Associative Memory", Springer-Verlag, Berlin, 1988.
[6] E.Ardizzone, A.Chella, and F.Sorbello, "Application of the novelty filter to the motion analysis", in *Proc. of INCC*, pp. 46-49, July 1990.

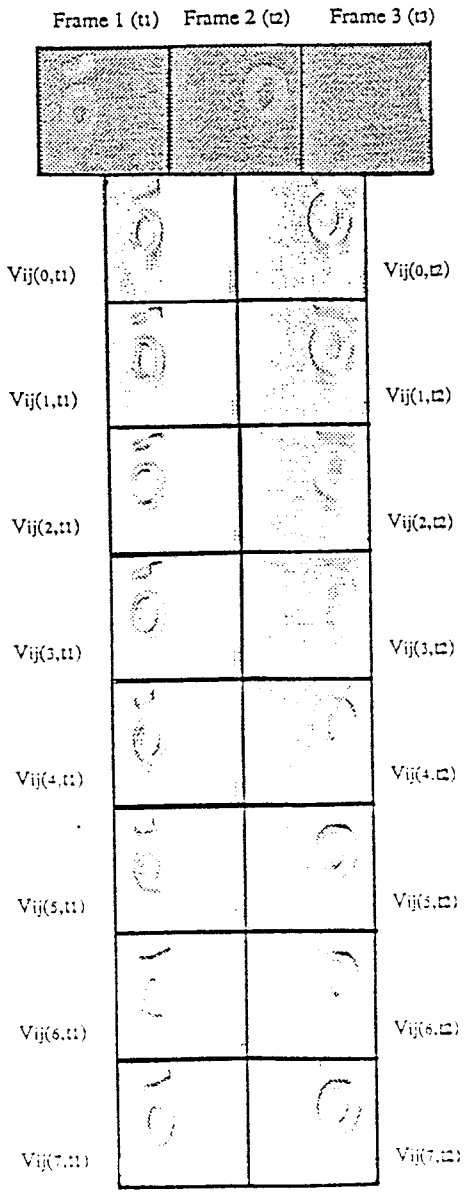

FIGURE 4 The normalised optical flow $V_{ij}(k, t)$ produced by a sequence of three images. 200x160 pixels bitmaps were used. $V_{ij}(k, t)$ carries limited global motion information better than $U_{ij}(k, t)$.

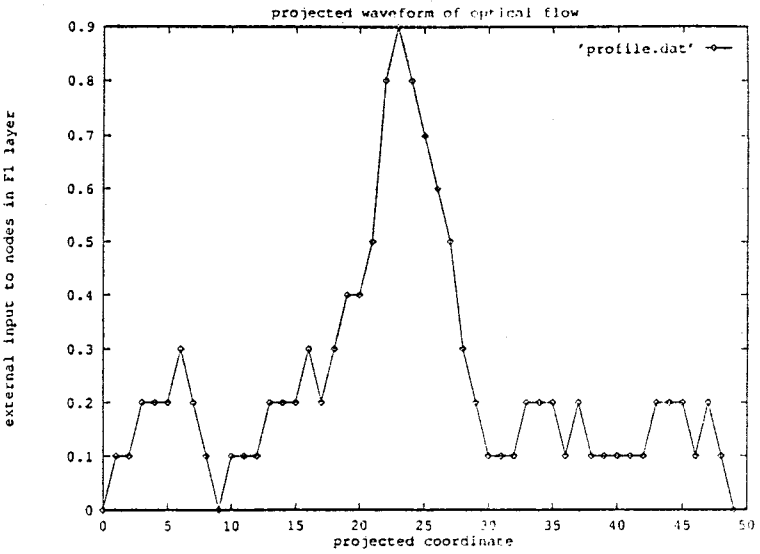

FIGURE 5 External Input to the Instar-Competitive, Outstar-Competitive Neural Network

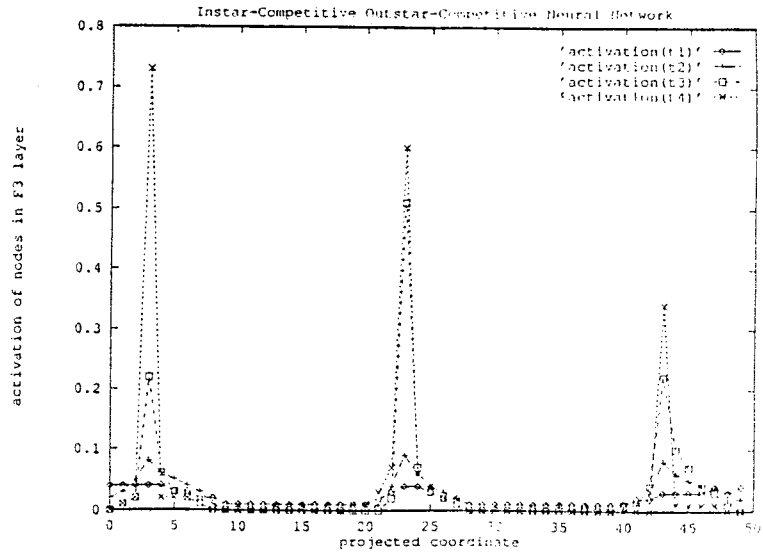

FIGURE 6 Time Evolution of the nodes' Activations in the Instar-Competitive, Outstar-Competitive Neural Network. It can been seen that contrast enhancement is carried out in the network