Accumulative Computation Method for Motion Features Extraction in Active Selective Visual Attention

Antonio Fernández-Caballero[1], María T. López[1], Miguel A. Fernández[1], José Mira[2], Ana E. Delgado[2], and José M. López-Valles[3]

[1] Universidad de Castilla-La Mancha, E.P.S.A., 02071 - Albacete, Spain
caballer@info-ab.uclm.es
[2] Universidad Nacional de Educación a Distancia, E.T.S.I. Informática, 28040 - Madrid, Spain
jmira@dia.uned.es
[3] Universidad de Castilla-La Mancha, E.U.P.C., 13071 - Cuenca, Spain
JoseMaria.Lopez@uclm.es

Abstract. A new method for active visual attention is briefly introduced in this paper. The method extracts motion and shape features from indefinite image sequences, and integrates these features to segment the input scene. The aim of this paper is to highlight the importance of the accumulative computation method for motion features extraction in the active selective visual attention model proposed. We calculate motion presence and velocity at each pixel of the input image by means of accumulative computation. The paper shows an example of how to use motion features to enhance scene segmentation in this active visual attention method.

1 Introduction

Findings in psychology and brain imaging have increasingly suggested that it is better to view visual attention not as a unitary faculty of the mind but as a complex organ system sub-served by multiple interacting neuronal networks in the brain [1]. At least three such attentional networks, for alerting, orienting, and executive control have been identified. The images are built habitually as from the entries of parallel ways that process distinct features: motion, solidity, shape, colour, location [2]. Vecera [3] introduced a model to obtain objects separated from the background in static images by combing bottom-up (scene-based) and top-down (task-based) processes. The bottom-up process gets the borders to form the objects, whereas the top-down process uses known shapes stored in a database to be compared to the shapes previously obtained in the bottom-up process. One of the most influential theories about the relation between attention and vision is the Feature Integration Theory [4]. They hypothesized that simple features were represented in parallel across the field, but that their conjunctions could only be recognized after attention had been focused on particular locations.

Recognition occurs when the more salient features of the distinct feature maps of features are integrated.

The first neurally plausible architecture of selective visual attention was proposed by Koch and Ullman [5], and is closely related to the Feature Integration Theory. A visual attention system inspired by the behaviour and the neural architecture of the early primate visual system is presented in [6]. Multiscale image features are combined into a single saliency map. The model of Guided-Search (GS) [7] uses the idea of saliency map to realize the search in scenes. GS assumes a two-stage model of visual selection. The first, pre-attentive stage of processing has great spatial parallelism and realizes the computation of the visual simple features. The second stage is spatially serial and it enables more complex visual representations to be computed, involving combinations of features.

Recently, a neural network (connectionist) model called the Selective Attention for Identification Model (SAIM) has been introduced [8]. The function of the suggested attention mechanism is to allow translation-invariant shape-based object recognition. Also a system of interconnected modules consisting of populations of neurons for modelling the underlying mechanisms involved in selective visual attention is proposed [9]. The dynamics of the system can be interpreted as a mechanism for routing information from the sensory input. A very recent model of attention for active vision has been introduced by Backer and Mertshing [10]. In this model there are two selection phases. Previous to the first selection a saliency map is obtained as the result of integrating the different features extracted. Concretely the features extracted are symmetry, eccentricity, colour contrast, and depth. The first selection stage selects a small number of items according to their saliency integrated over space and time. These items correspond to areas of maximum saliency and are obtained by means of active neural fields. The second selection phase has top-down influences and depends on the system's aim. Some implemented systems based on selective attention have up to date covered up several of the following categories: recognition (e.g. [11]), teleconferencing [12], tracking of multiple objects (e.g. [13]), and mobile robot navigation (e.g. [14]).

In this paper, we briefly describe our approach to selective visual attention [15]. But our intention is to highlight the benefits of using accumulative computation as a method for motion features extraction, as one of the most important contributions to general feature extraction step.

2 Selective visual attention model

The layout of the Selective Visual Attention model developed in our research team is depicted in Figure 1(a). Next a brief description of all tasks involved in our model is offered. The aim of task Attention Construction is to select zones (blobs) of those objects (figures) where attention is to be focused. Notice that after Attention Construction complete figures will not be classified, but all blobs configuring the figures will have been labelled. Blob has to be understood as a homogeneous zone of connected pixels. Therefore, blobs are constructed from

image pixels that fulfil a series of predefined requisites (interest points). Task Motion Features Extraction is justified by the need to acquire active features of the image pixels. Concretely, features extracted are "motion presence" and "velocity". Now, task Form Features Extraction computes the values of various shape properties of the objects to be selected. The input to this task is stored as blobs in the Working Memory and as figures in the Attention Focus. Features extracted for the blobs are the size, width and height. As figures stored in the Attention Focus are approximations to complete objects, the features extracted are the same ones than for the blobs, plus features width-height ratio and compactness. The output of task Features Integration is the Interest Map, produced from an integration of motion and form features. Task Attention Reinforcement is dedicated to the final construction of figures and the persistence of attention on some figures (or objects) of interest in the image sequence.

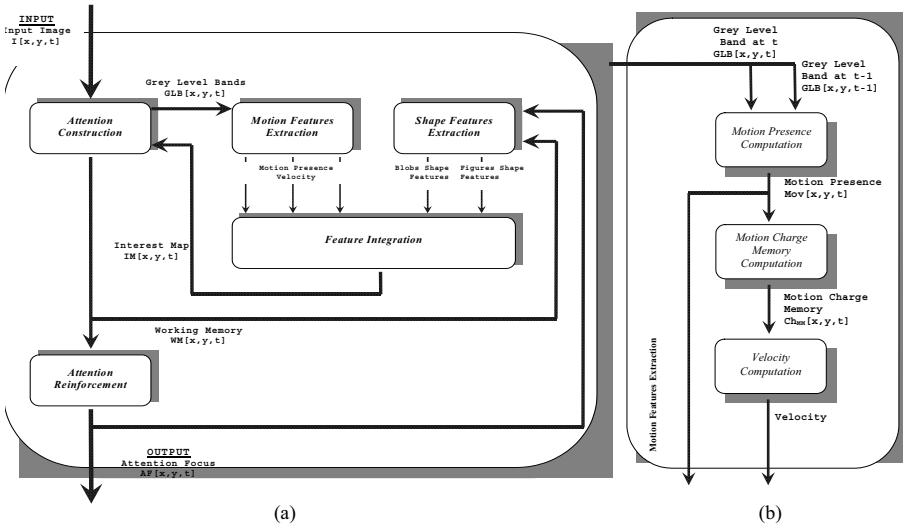


Fig. 1. (a) Selective Visual Attention architecture. (b) Layout of Motion Features Extraction

3 Accumulative computation

Accumulative computation has now been largely applied to moving objects detection, classification and tracking in indefinite sequences of images (e.g. [16], [17], [18], [19]). The more general modality of accumulative computation is the charge/discharge mode, which may be described by means of the following generic formula:

$$Ch[x,y,t] = \begin{cases} \min(Ch[x,y,t-\Delta t] + C, Ch_{max}), \text{ if "}property P[x,y,t]\text{"} \\ \max(Ch[x,y,t-\Delta t] - D, Ch_{min}), \text{otherwise} \end{cases} \quad (1)$$

The temporal accumulation of the persistency of the binary property $P[x,y,t]$ measured at each time instant t at each pixel $[x,y]$ of the data field is calculated.

Generally, if the *property* is fulfilled at pixel $[x, y]$, the charge value at that pixel $Ch[x, y, t]$ goes incrementing by increment charge value C up to reaching Ch_{max}, whilst, if *propertyP* is not fulfilled, the charge value $Ch[x, y, t]$ goes decrementing by decrement charge value D down to Ch_{min}. All pixels of the data field have charge values between the minimum charge, Ch_{min}, and the maximum charge, Ch_{max}. Obviously, values C, D, Ch_{min} and Ch_{max} are configurable depending on the different kinds of applications, giving raise to all different operating modes of the accumulative computation. Values of parameters C, D, Ch_{max} and Ch_{min} have to be fixed according to the applications characteristics. Concretely, values Ch_{max} and Ch_{min} have to be chosen by taking into account that charge values will always be between them. The value of C defines the charge increment interval between time instants $t-1$ and t. Greater values of C allow arriving in a quicker way to saturation. On the other hand, D defines the charge decrement interval between time instants $t-1$ and t. Thus, notice that the charge stores motion information as a quantified value, which may be used for several classification purposes. In [20] the architecture of the accumulative computation module is shown. Some of the operating modes may be appreciated there, demonstrating their versatility and their computational power.

4 Motion features extraction by accumulative computation

As told before, the main objective of this paper is to highlight the importance of the accumulative computation method for motion features extraction in the active selective visual attention model proposed. The aim of task Motion Feature Extraction is to calculate the active (motion) features of the image pixels, that is to say, in our case, the presence of motion and the velocity. Due to our experience (e.g. [21]) we know some methods to get that information.

Firstly, in order to diminish the effects of noise due to the changes in illumination in motion detection, variation in grey level bands at each image pixel is treated. We work with 256 grey level input images and transform them to a lower number of levels n. In concrete, good results use to be obtained with 8 levels. These 8 level images are called images segmented into 8 grey level bands and are stored in the Grey Level Bands Map [16], [18], as stated in Equation 2:

$$GLB[x, y, t] = \frac{GL[x, y, t] \cdot n}{GL_{max} - GL_{min} + 1} + 1 \tag{2}$$

where $GLB[x, y, t]$ is the grey level band of pixel $[x, y]$ at t, GL stands for grey level and n is the total number of grey level bands defined.

In Figure 1(b) you may observe the layout of task Motion Features Extraction. The values computed are Motion Presence, Motion Charge Memory and Velocity. Motion Charge Memory is obtained by means of accumulative computation on the negation of property Motion Presence. Velocity is computed from values stored in Motion Charge Memory. By Velocity we mean the module and angle of vector velocity.

4.1 Motion Presence computation

The first motion feature calculated is Motion Presence, $Mov[x, y, t]$, which is easily obtained as a variation in grey level band between two consecutive time instants t and $t-1$:

$$Mov[x, y, t] = \begin{cases} 0, \text{ if } GLB[x, y, t] = GLB[x, y, t-1] \\ 1, \text{ if } GLB[x, y, t] \neq GLB[x, y, t-1] \end{cases} \quad (3)$$

4.2 Motion Charge Memory computation

As we already stated before, Motion Charge Memory is calculated by means of accumulative computation on the negative of property Motion Presence. The accumulative computation operation mode used in this case is the LSR (length-speed ratio) mode [22]. The property measured in this case is equivalent to "no motion" at pixel of co-ordinates $[x, y]$ at instant t.

In this mode C_{MM} (formerly C in Equation 1 is now the charge increment value on Motion Charge Memory. Notice that D_{MM}, (formerly D) the decrement charge value does not appear explicitly, as we consider that $D_{MM} = Ch_{max}$. The idea behind the LSR is that if there is no motion on pixel $[x, y]$, charge value $Ch_{MM}[x, y, t]$ goes incrementing up to Ch_{max}, and if there exists motion, there is a complete discharge (the charge value is given value Ch_{min}). Thus, charge value $Ch_{MM}[x, y, t]$ represents a measure of time elapsed since the last significant variation in brightness on image pixel $[x, y]$.

$$Ch_{MM}[x, y, t] = \begin{cases} Ch_{min}, & \text{if } Mov[x, y, t] = 1 \\ \min(Ch_{MM}[x, y, t-1] + C_{MM}, & \\ Ch_{max}), & \text{if } Mov[x, y, t] = 0 \end{cases} \quad (4)$$

Equation 4 shows how charge at pixel $[x, y]$ gradually increases through time (frame to frame) in a quantity C_{MM} (charge constant due to motion) up to a maximum charge or saturation Ch_{max}, while motion is not detected. At the opposite, charge falls down to a minimum of charge Ch_{min}, when motion is detected at pixel $[x, y]$.

4.3 Velocity computation

Calculation of velocity is performed starting from the values stored in the Motion Charge Memory, as explained in Table 1. It is important to highlight that velocity obtained from Motion Charge Memory is not the velocity of an object point that occupies pixel $[x, y]$ in time t, but rather the velocity of an object point that caused motion presence detection when it passed over pixel $[x, y]$ a number $k = \frac{C_{MM}[x,y,t] - Ch_{min}}{C_{MM}}$ time units ago. Thus, notice that Motion Charge Memory shows the same value for all those pixels where a simultaneous motion occurred at a given time. Now, in order to perform Velocity Computation we calculate the velocity in x-axis, v_x, as well as in y-axis, v_y. Once values v_x and v_y, have been obtained, the module and the angle of vector velocity are gotten. Firstly,

Table 1. Description of values stored in Motion Charge Memory

Value in Motion Charge Memory	Explanation
$C_{MM}[x,y,t] = Ch_{min}$	Motion is detected at pixel $[x,y]$ in t. Value in memory is the minimum charge value.
$C_{MM}[x,y,t] = Ch_{min} + k \cdot C_{MM} < Ch_{max}$	No motion is detected at pixel $[x,y]$ in t. Motion was detected for the last time in $t - k \cdot \Delta t$. After k charge increments the maximum charge has not yet been reached.
$C_{MM}[x,y,t] = Ch_{max}$	No motion is detected at pixel $[x,y]$ in t. We do not know when motion was detected for the last time. Value in memory is the maximum charge value.

to calculate velocity in x-axis, charge value in $[x,y]$, where an object is currently passing, is compared to charge value in another co-ordinate of the same row $[x+l,y]$, where the same object is passing. In the best case, that is to say, when both values are different from Ch_{max}, the time elapsed since motion was lastly detected in instant $t - k_{[x,y]} \cdot \Delta t$ at $[x,y]$ up to the time when motion was detected in instant $t - k_{[x+l,y]} \cdot \Delta t$ in $[x+l,y]$ may be calculated as:

$$\begin{aligned} &Ch_{MM}[x,y,t] - Ch_{MM}[x+l,y,t] = \\ &= (Ch_{min} + k_{[x,y]} \cdot C_{MM}) - (Ch_{min} + k_{[x+l,y]} \cdot C_{MM}) = \\ &= (k_{[x,y]} - k_{[x+l,y]}) \cdot C_{MM} \end{aligned} \tag{5}$$

This computation can obviously not be performed if any of both values are Ch_{max}, as we do not know how many time intervals have elapsed since last movement. Hence, for valid charge values, we have:

$$\Delta t = \frac{(k_{[x,y]} - k_{[x+l,y]}) \cdot C_{MM}}{C_{MM}} = k_{[x,y]} - k_{[x+l,y]} \tag{6}$$

From Equation 5 and Equation 6:

$$\Delta t = \frac{Ch_{MM}[x,y,t] - Ch_{MM}[x+l,y,t]}{C_{MM}} \tag{7}$$

And, as $v_x[x,y,t] = \frac{\delta x}{\delta t} = \frac{l}{\Delta t}$, finally:

$$v_x[x,y,t] = \frac{C_{MM} \cdot l}{Ch_{MM}[x,y,t] - Ch_{MM}[x+l,y,t]} \tag{8}$$

The same way, velocity in y-axis is calculated from the values stored in the Motion Charge Memory, as:

$$v_y[x,y,t] = \frac{C_{MM} \cdot l}{Ch_{MM}[x,y,t] - Ch_{MM}[x,y+l,t]} \tag{9}$$

Now, it is the turn to calculate the module $|\overrightarrow{v}[x,y,t]|$ and the angle $\beta[x,y,t]$ of the velocity.

$$\beta[x,y,t] = \arctan \frac{v_y[x,y,t]}{v_x[x,y,t]} \tag{10}$$

$$\overrightarrow{v}[x,y,t]| = (v_x[x,y,t]^2 + v_y[x,y,t]^2)^{0.5} \tag{11}$$

5 Data and results

In order to evaluate the performance of our active visual attention method, and particularly in relation to the motion features described, we have tested the algorithms on the famous Hamburg Taxi motion sequence from the University of Hamburg, usually accepted as an excellent benchmark in optic flow algorithms implementations.

The sequence may be downloaded via ftp://ftp.csd.uwo.ca/pub/vision/, and contains 20 190x256 pixel image frames. Notice that our algorithms only segment moving objects. The sequence contains a movement of four objects: a pedestrian near to the upper left corner and the three cars.

Our intention is to focus only on cars. Thus, we have to parameterize the system in order to capture attention on elements with a series of shape features. These shape features are described in Tables 2 and 3, and are thought to capture all moving cars in the scene. Table 1 shows the parameters used (as well as their values) to get the blobs in the Working Memory. Similarly, in Table 2 we show the parameters and values for the figures in the Attention Focus.

Firstly, results are shown in Figure 2 (upper images) when no predefined velocity is given to the system. In this figure you may appreciate some images of the sequence of selective attention on moving cars. In (a) an input image of the Hamburg Taxi sequence is shown, namely at time instant $t = 9$. In (b) we show in white color the pixels where motion has been detected. Remember that this is equivalent to the result of calculating the presence of motion in the example. Notice that, in the output of this task, a pixel drawn in white color means that there has been variation in the grey level band of the pixel in instant t with respect to the previous instant $t - 1$. There are pixels belonging to the desired objects, as well as to other parts of the image due to some variations in illumination in the scene. In (c) see the contents of the Attention Focus. In this figure, pixels drawn in white color on black background represent image elements where attention has been captured and reinforced through time.

In this example we may appreciate that the attention focus really corresponds to moving cars. But, although all moving cars are initially detected - through motion presence feature-, only two of the three cars in movement are segmented. This is due to the fact that the segmentation in grey level bands (as explained in Motion Features Extraction task) unites the moving car to a tree. This union affects our algorithms in a negative way, as the so formed object does not fit into the shape features given in Tables 2 and 3.

Table 2. Blob shape features and values

Feature	Value (number of pixels)
Spot maximum size	6000
Spot maximum width	85
Spot maximum height	65

Table 3. Figures shape features and values

Feature	Value (pixels)	Value (ratio)
Object size range	400 - 6000	
Object width range	20 - 85	
Object height range	20 - 65	
Object width-height ratio range		0.05 - 2.50
Object compactness range		0.40 - 1.00

This example is very helpful to highlight some pros and contras of our method described. Firstly, it is able to discriminate moving objects in an indefinite sequence into different classes of objects. This has been shown by the elimination of the pedestrian in the scene through shape features parameterization. But, clearly, some problems related to temporal overlaps affect our method. Now, consider the lower images at Figure 2, where the attention focus selection has been changed to incorporate velocity parameters. In this case, we are interested in using more motion features to enhance segmentation. Our intention is now to obtain cars that move to the right. This has been accomplished by looking for an angle in vector velocity in the range -22.5 to +22.5, that is to say:

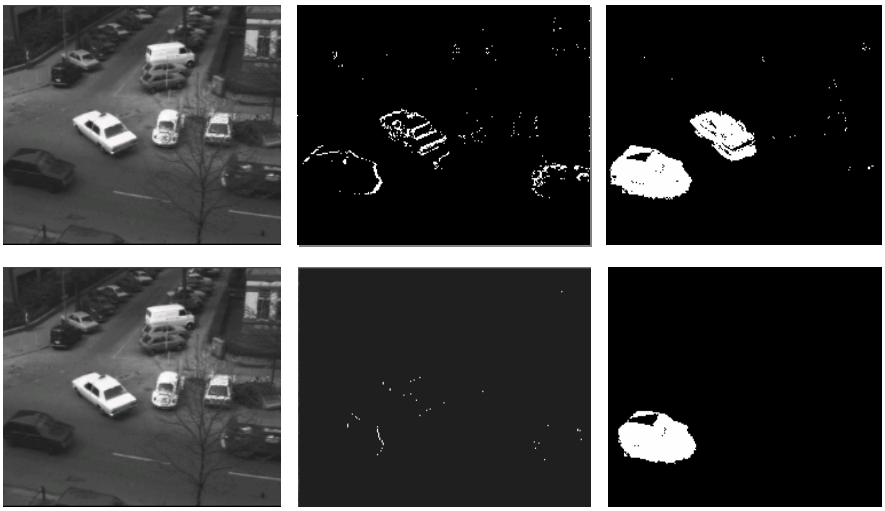
$$-22.5 \leq \beta[x, y, t] = \arctan \frac{v_y[x, y, t]}{v_x[x, y, t]} \leq +22.5$$

In the results offered in Figure 2 (second row) you may observe that white pixels in (b) image have greatly decreased respect to the results in Figure 2 (first row). That is because only pixels moving with a given velocity angle are filtered. This example shows the importance of motion features to enhance the segmentation in our active visual attention system whilst shape features are maintained constant.

6 Conclusions

A model of dynamic visual attention capable of segmenting objects in a real scene has been briefly described in this paper. The model enables focusing the attention at each moment at shapes that possess certain features and eliminating those that are of no interest. The features used are related to motion and shape of the elements present in the grey level images dynamic scene. The model may be used to observe real environments indefinitely in time.

The principal aim of this paper has been to highlight the importance of the accumulative computation method for motion features extraction in the dynamic selective visual attention model proposed. This is true, because we calculate



(a) (b) (c)

Fig. 2. Sequence of selective attention on moving cars (upper row), and on cars moving to the right (lower row). From top to bottom: (a) Input image. (b) Motion Presence (c) Attention Focus

motion presence and velocity at each pixel of the input image by means of accumulative computation.

Apart from this, our paper highlights the importance of motion features - motion presence and velocity - to enhance the segmentation and classification of objects in real scenes. An example has been offered where, by incrementing the number of motion features, whilst maintaining the shape features constant, the attention focus is changed to the user's interest.

7 Acknowledgements

This work is supported in part by the Spanish CICYT TIN2004-07661-C01-01 and TIN2004-07661-C02-02 grants.

References

1. Posner, M.I., Raichle, M.E.: Images of Mind. Scientific American Library, NY (1994)
2. Desimone, R., Ungerleider, L.G.: Neural mechanisms of visual perception in monkeys. In: Handbook of Neuropsychology, Elsevier (1989) 267–299
3. Vecera, S.P.: Toward a biased competition account of object-based segregation and attention. In: Brain and Mind, Kluwer Academic Publishers (2000) 353–384
4. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology **12** (1980) 97–136

5. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiology **4** (1985) 219–227
6. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1254–1259
7. Wolfe, J.M.: Guided Search 2.0. A revised model of visual search. Psychonomic Bulletin & Review **1** (1994) 202–238
8. Heinke, D., Humphreys, G.W., diVirgilo, G.: Modeling visual search experiments: Selective Attention for Identification Model (SAIM). Neurocomputing **44** (2002) 817–822
9. Deco, G., Zihl, J.: Top-down selective visual attention: A neurodynamical approach. Visual Cognition **8**:1 (2001) 119–140
10. Backer, G., Mertsching, B.: Two selection stages provide efficient object-based attentional control for active vision. Proceedings of the International Workshop on Attention and Performance in Computer Vision (2003) 9–16
11. Paletta, L., Pinz, A.: Active object recognition by view integration and reinforcement learning. Robotics and Autonomous Systems **31**:1-2 (2000) 71–86
12. Herpers, R., Derpanis, K., MacLean, W.J., Verghese, G., Jenkin, M., Milios, E., Jepson, A., Tsotsos, J.K.: SAVI: An actively controlled teleconferencing system. Image and Vision Computing **19** (2001) 793–804
13. Wada, T., Matsuyama, T.: Multiobject behavior recognition by event driven selective attention method. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**:8 (2000) 873–887
14. Ye, Y., Tsotsos, J.K.: Sensor planning for 3D object search. Computer Vision and Image Understanding **73**:2 (1999) 145–168
15. López, M.T., Fernández, M.A., Fernández-Caballero, A., Delgado, A.E.: Neurally inspired mechanisms for the active visual attention map generation task. Computational Methods in Modeling Computation, Springer-Verlag (2003) 694–701
16. Fernández-Caballero, A., Mira, J., Fernández, M.A., López, M.T.: Segmentation from motion of non-rigid objects by neuronal lateral interaction. Pattern Recognition Letters **22**:14 (2001) 1517–1524
17. Fernández-Caballero, A., Mira, J., Delgado, A.E., Fernández, M.A.: Lateral interaction in accumulative computation: A model for motion detection. Neurocomputing **50** (2003) 341–364
18. Fernández-Caballero, A., Fernández, M.A., Mira, J., Delgado, A.E.: Spatio-temporal shape building from image sequences using lateral interaction in accumulative computation. Pattern Recognition **36**:5 (2003) 1131–1142
19. Fernández-Caballero, A., Mira, J., Fernández, M.A., Delgado, A.E.: On motion detection through a multi-layer neural network architecture. Neural Networks **16**:2 (2003) 205–222
20. Mira, J., Fernández, M.A., López, M.T., Delgado, A.E., Fernández-Caballero, A.: A model of neural inspiration for local accumulative computation. 9th International Conference on Computer Aided Systems Theory, Springer-Verlag (2003) 427–435
21. Fernández, M.A., Mira, J.: Permanence memory: A system for real time motion analysis in image sequences. IAPR Workshop on Machine Vision Applications (1992) 249–252
22. Fernández, M.A., Fernández-Caballero, A., López, M.T., Mira, J.: Length-speed ratio (LSR) as a characteristic for moving elements real-time classification. Real-Time Imaging **9** (2003) 49–59