Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 469–478
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Motion features to enhance scene segmentation in active visual attention

María T. López [a], Antonio Fernández-Caballero [a,*], Miguel A. Fernández [a], José Mira [b], Ana E. Delgado [b]

[a] Departamento de Informática, Escuela Politécnica Superior, Universidad de Castilla-La Mancha, 02071 Albacete, Spain
[b] Departamento de Inteligencia Artificial, Facultad de Ciencias and E.T.S.I. Informática, Universidad Nacional de Educación a Distancia, 28040 Madrid, Spain

Received 12 November 2003; received in revised form 25 July 2005
Available online 20 October 2005

Communicated by T.K. Ho

Abstract

A new computational model for active visual attention is introduced in this paper. The method extracts motion and shape features from video image sequences, and integrates these features to segment the input scene. The aim of this paper is to highlight the importance of the motion features present in our algorithms in the task of refining and/or enhancing scene segmentation in the method proposed. The estimation of these motion parameters is performed at each pixel of the input image by means of the accumulative computation method, using the so-called permanency memories. The paper shows some examples of how to use the ''motion presence'', ''module of the velocity'' and ''angle of the velocity'' motion features, all obtained from accumulative computation method, to adjust different scene segmentation outputs in this dynamic visual attention method.

Keywords: Active visual attention; Permanency memories; Segmentation; Feature extraction; Motion; Image sequences

1. Introduction

Motion is a major information source for segmenting objects perceived in dynamic scenes. Therefore, techniques for estimating the velocity field (optical flow field) are of great interest for enhancing scene segmentation (Gautama and Van Hulle, 2002). In previous works, our research team has taken advantage of motion information in segmenting (Fernández-Caballero et al., 2001, 2003) and classifying moving objects (Fernández et al., 2003) through accumulative computation (Fernández et al., 1995) and algorithmic lateral inhibition (Mira et al., 2004) by means of a series of charge maps related to pixel-wise ''motion presence'' information. But, up to this moment, we had not directly used calculated motion parameters—e.g. velocity parameters, such as the ''module of the velocity'' and the ''angle of the velocity''—to perform better and more robust segmentation and tracking in video sequences. Our latest research deals with dynamic visual attention systems, where parameters of this kind have turned out to be necessary.

In this paper, we introduce a new active visual attention method for scene segmentation. Although the whole structure of the system is described briefly, we highlight the importance of motion-related parameters. In this sense, a great emphasis is placed on the *Motion Features Extraction* task within the overall *Active Visual Attention* proposed method. The layout of the paper is as follows. In Section 2, the proposed active visual attention method is described. In Section 3, we offer a series of data and results, including a performance evaluation on the famous Hamburg Taxi

* Corresponding author. Tel.: +34 967 599 200; fax: +34 967 599 224.
E-mail address: caballer@info-ab.uclm.es (A. Fernández-Caballero).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.09.010

sequence. Section 4 shows the main conclusions for this article.

2. Active visual attention method

Our approach defines a method for the generation of an *Active Attention Focus* on a dynamic scene to obtain the objects that keep the user's attention in accordance with a set of predefined features, including motion and shape features (Chella et al., 2000). In Fig. 1 the general layout of the proposed solution is shown. As you may notice the *Active Visual Attention* method is decomposed into three general tasks, namely, *Feature Extraction and Integration*, *Attention Capture* and *Attention Reinforcement*. The rest of this section explains each of these tasks. *Motion Features Extraction*, the center of this paper belongs to the *Feature Extraction and Integration* task.

2.1. Feature extraction and integration

The *Feature Extraction and Integration* task is made up of two broad blocks: the first one is related to feature extraction (*Motion Features Extraction and Spot and Object Shape Features Extraction*), whilst the second one is feature integration.

2.1.1. Motion features extraction

Motion Features Extraction task calculates the dynamic (motion) features of the image pixels; in our case, the "presence of motion" as a Boolean value and the "velocity" as a vector. This task has been inherited from previous works of our research team (Fernández et al., 1995, 2003). Velocity, as told in the introduction section, is a new feature incorporated in the current work.

Firstly, and 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 performed. We work with 256 grey-level input images and transform them into a lower number of levels n. In particular, good results are obtained with eight levels in normal illumination indoor and outdoor scenes (Fernández-Caballero et al., 2001, 2003). A higher value rarely gives better results, whilst lower values (say, 2 or 4) may be used for night vision. A higher value of number of grey-level bands usually enables to better discriminate the whole shapes of the moving non-rigid objects. Nevertheless, a too high value of this parameter may include some image background into the shapes. This may even lead to fuse more than one different shape into one single silhouette (Fernández-Caballero et al., 2003). The eight level images are called images segmented into eight grey level bands and are stored in the *Grey-Level Bands Map*.

The first motion feature to be calculated is motion presence, $\mathrm{Mov}[x,y,t]$, obtained as a variation in grey-level band between two consecutive time instants t and $t-1$

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

where $\mathrm{GLB}[x,y,t]$ is the grey-level band of pixel (x,y) at t.

Then, velocity is obtained by calculating its module and angle. But, in first place, we start from the memorization along time (accumulation) (Fernández et al., 1995) of charge $\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]$ at each image pixel (x,y). This memorization has been called permanency memories effect (Fernández et al., 2003). In general, permanency memories work on binary images (1-bit digitized) according to grey-level thresholds. The permanence memories define a map of data items for each frame t.

In Fig. 2 the permanency memories model's behavior is shown in one-dimensional and very easy situations. Let us suppose that input values correspond to an indefinite sequence of images where several objects are moving. Let

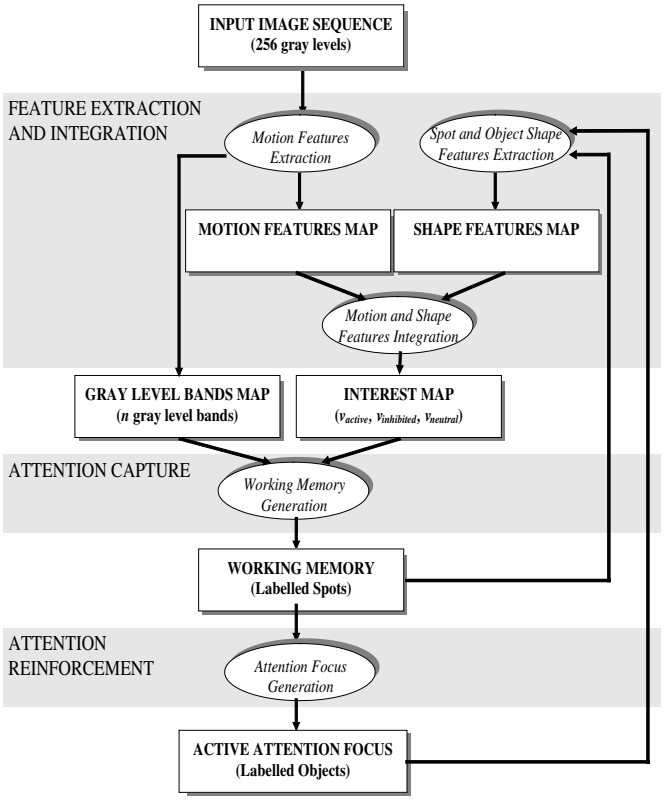


Fig. 1. Layout of the "Active Visual Attention" method.

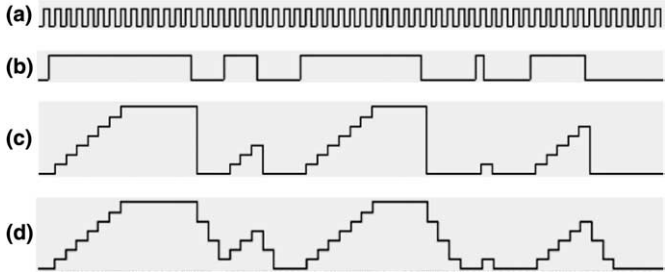


Fig. 2. Illustration of the permanency effect: (a) clock t, (b) property p, (c) LSR modality and (d) general charge/discharge modality.

us also suppose that a measured property, $p[x,y,t]$ is the result of detecting motion on pixel (x,y) at time instant t. Then, values of $p[x,y,t]$ in two successive instants are used, interpreting that $p[x,y,t]=1$ means that motion has been detected over pixel (x,y) at t and that $p[x,y,t]=0$ means there is no motion. For this property, the evolution of charge and discharge of its persistency is shown in Fig. 2 for some modalities. Fig. 2c shows the behavior of the accumulative computation model in a modality called LSR (length speed relation) (Fernández et al., 2003). This modality has been used for the purpose of classification of moving objects. The more general charge/discharge modality is also shown (Fig. 2d).

Now, in this way, the value in frame t of the permanency memory $\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]$, associated to pixel (x,y) might be defined in terms of its value at time $t-1$ and the binary input $\mathrm{Mov}[x,y,t]$. The accumulative computation operation mode used in this case is the LSR mode applied on the inverse of the property described. Thus, the property measured in this case is equivalent to "no motion presence" at pixel of co-ordinates (x,y) at instant t

$$p[x,y,t] = 1 - \mathrm{Mov}[x,y,t] \tag{2}$$

Thus, in this work, the formula used to represent the charge due to accumulative computation by permanency memories is the following one:

$$\mathrm{Ch}_{\mathrm{Mov}}[x,y,0] = \mathrm{Ch}_{\max}$$
$$\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] = \begin{cases} \mathrm{Ch}_{\min}, & \text{if } \mathrm{Mov}[x,y,t]=1 \\ \min(\mathrm{Ch}_{\mathrm{Mov}}[x,y,t-1] + C_{\mathrm{motion}}, \mathrm{Ch}_{\max}), & \text{if } \mathrm{Mov}[x,y,t]=0 \end{cases} \tag{3}$$

In LSR mode C_{motion} is called the charge increment value. The idea is that if there is no motion on pixel (x,y), charge value $\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]$ goes incrementing up to $\mathrm{Ch}_{\max}$, and if there is motion, there is a complete discharge (the charge value is given value $\mathrm{Ch}_{\min}$). Usually, $\mathrm{Ch}_{\max}$ and $\mathrm{Ch}_{\min}$ are chosen to be 255 and 0, respectively, that is to say, the maximum and minimum value of any possible grey level. This range has been kept as such since our first works on motion analysis as it fits in one single byte and consumes little memory when dealing with large image sequences. Thus, notice that charge value $\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]$ represents a measure of time elapsed since the last significant variation in brightness on image pixel (x,y).

As an example let us consider the motion situation, where an object initially moves horizontally with two velocities (first, 1 pixel each two frames; and then, 1 pixel each four frames), and then continues moving vertically with a velocity of 1 pixel per frame in a 20×20 pixel image. The basic parameters for the accumulative computation are: $\mathrm{Ch}_{\max}=255$, $\mathrm{Ch}_{\min}=0$, and $C_{\mathrm{motion}}=5$. Fig. 3 graphically shows the values for the permanency memory at $t=25$ (with a scale change $\mathrm{Ch}_{\max}-\mathrm{Ch}_{\mathrm{Mov}}[x,y,25]$).

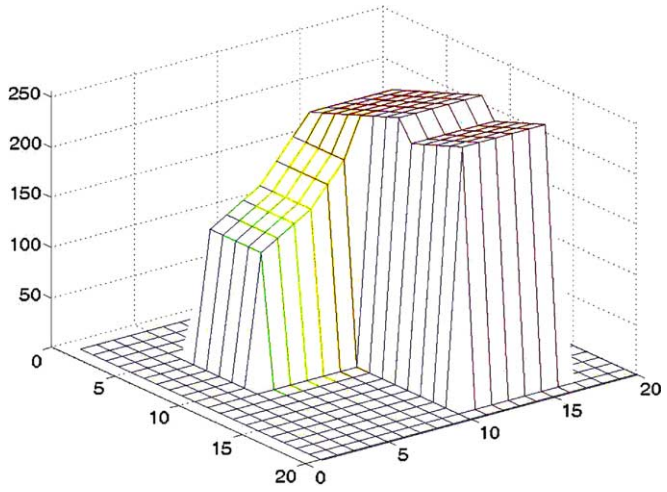


Fig. 3. A graphical view of the contents of the permanency memory after 25 image frames.

Table 1
Description of the relationship between permanence values and motion detection

Value in motion charge memory	Explanation
$\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]=\mathrm{Ch}_{\min}$	Motion is detected at pixel $[x,y]$ in t. Value in memory is the minimum charge value
$\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]=\mathrm{Ch}_{\min}+k\cdot C_{\mathrm{motion}}<\mathrm{Ch}_{\max}$	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
$\mathrm{Ch}_{\mathrm{Mov}}[x,y,t]=\mathrm{Ch}_{\max}$	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

You may observe that the slope of the permanency remains constant while the velocity is constant at 1 pixel each frame, and then augments when the velocity decrements (1 pixel each four frames). When considering the motion in the vertical axis, the slope is the lowest as the velocity is only 1 pixel per frame.

Once the charge map (or permanency memory) has been obtained, velocity may be calculated starting from these charges stored, as explained in Table 1.

It is important to highlight that the velocity obtained in this way 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

$$k = \frac{\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\min}}{C_{\mathrm{motion}}} \tag{4}$$

of time units before. Thus, a given charge shows the same value for all those pixels where a simultaneous motion occurred at a given time. Now, in order to get the velocity we calculate the velocity in x-axis, v_x, as well as in y-axis, v_y. Once values v_x y v_y, have been obtained, the module and the angle of vector velocity are also got.

Firstly, 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 $\mathrm{Ch}_{\max}$, 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

$$\begin{aligned}\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\mathrm{Mov}}[x+l,y,t] &= (\mathrm{Ch}_{\min} + k_{[x,y]} \cdot C_{\mathrm{motion}}) - (\mathrm{Ch}_{\min} + k_{[x+l,y]} \cdot C_{\mathrm{motion}}) \\ &= (k_{[x,y]} - k_{[x+l,y]}) \cdot C_{\mathrm{motion}}\end{aligned} \tag{5}$$

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

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

From Eqs. (3.1) and (3.2)

$$\Delta t = \frac{\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\mathrm{Mov}}[x+l,y,t]}{C_{\mathrm{motion}}} \tag{7}$$

And, as $v_x = \frac{\partial x}{\partial t} = \frac{l}{\Delta t}$, finally

$$v_x[x,y,t] = \frac{C_{\mathrm{motion}} \cdot l}{\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\mathrm{Mov}}[x+l,y,t]} \tag{8}$$

In the same way, velocity in y-axis is calculated from the values stored as charges, as

$$v_y[x,y,t] = \frac{C_{\mathrm{motion}} \cdot l}{\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\mathrm{Mov}}[x,y+l,t]} \tag{9}$$

In Fig. 4 the velocities in x for the same example are provided after applying Eq. (8) for $l = 1$ on the permanency values at $t = 25$. See, for instance, the value for coordinate $(x, y) = (5, 5)$ at the velocity in x-axis table (Fig. 4). The value 0.5 means that there has been motion detected at coordinate $(5, 5)$ 25 frames (or time units) ago. Indeed, we have, according to the values provided at Fig. 3, and using formula (4), that $k = \frac{\mathrm{Ch}_{\mathrm{Mov}}[5,5,25] - \mathrm{Ch}_{\min}}{C_{\mathrm{motion}}} = \frac{125-0}{5} = 25$. And lastly, according to formula (8), the value 0.5 is the result of calculating $v_x[x,y,t] = \frac{C_{\mathrm{motion}} \cdot l}{\mathrm{Ch}_{\mathrm{Mov}}[x,y,t] - \mathrm{Ch}_{\mathrm{Mov}}[x+l,y,t]} = \frac{5 \cdot 1}{125-115} = 0.5$.

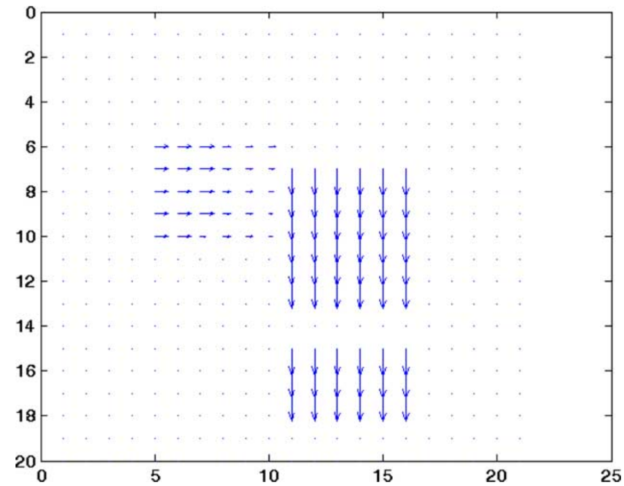


Fig. 5. Velocity vectors for the example.

Fig. 5 represents the velocity vectors for the example.

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

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

$$|\vec{v}[x,y,t]| = \sqrt{v_x[x,y,t]^2 + v_y[x,y,t]^2} \tag{11}$$

The output of the *Motion Features Extraction* subtask is the *Motion Features Map*, which includes motion presence detection and velocity at each pixel (x, y).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0.5	0.25	0.25	0.25	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0.5	0.25	0.25	0.2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0.5	0.25	0.25	0.166	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0.5	0.25	0.25	0.14	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.5	0.5	0.5	0.25	0.25	0.125	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4. Velocities in x-axis for the example.

2.1.2. Spot and object shape features extraction

The *Spot and Object Shape Features Extraction* task incorporates the extraction of features related to spots and to objects in relation to their shapes. Firstly, it extracts different shape features of the labeled elements stored in the *Working Memory*—obtained in the *Attention Capture* subtask—(the size, the width and the height). Notice that the labels in the *Working Memory* are also obtained by grey-level bands, just as a moving object is formed by a set of spots with different labels. In a similar way the features of the objects stored in the *Active Attention Focus*—see *Attention Reinforcement* task—, are obtained (the size, the width, the height, the width-height ratio and the compactness). These are now complete objects united by a common identifying label. All these features are stored in the *Shape Features Map*.

2.1.3. Motion and shape features integration

The output of task *Feature Integration* is the *Interest Map*, obtained by integrating the *Motion Features Map* (our motion features) with the *Shape Features Map* (our shape features). The *Interest Map* stores for each image pixel one of three possible classes: "active", "inhibited" and "neutral". The states of "active" or "inhibited" are reserved for those pixels where motion presence has been detected at current time t (information available in *Motion Presence Map*), or for pixels belonging to an object—or object spot—of interest at time instant $t-1$ (information found in *Shape Features Map*). Now, "neutral" pixels are the rest of the image pixels. "Active" pixels are those that fulfill the requirements imposed by the user, whilst "inhibited" pixels do not fulfill the requirements.

2.2. Attention capture

The objective of task *Attention Capture*, which only incorporates task *Working Memory Generation*, is to label image zones (or patches) included in objects of interest. *Attention Capture* is the central segmentation task in our model, and, as in many other approaches (e.g. Wu et al., 1996), represents a partition of each frame of the sequence into a set of regions which are homogeneously merged through time with regard to the motion criterion used. The output of this task has been called *Working Memory*. In our case, only those patches which appear in the *Working Memory* will potentially convert into the system's attention focus.

Some research lines to solve the problem of defining what are the elements which decompose the scene are based on border extraction, and obtain complex objects from more simple ones by looking for families of shapes. Our approach starts obtaining the object's parts from their grey level bands. Later on these objects parts (also called zones, patches or spots) will be treated as whole objects incorporating lateral interaction methods (Fernández-Caballero et al., 2001, 2003; López et al., 2003). In this proposal, the patches present in the *Working Memory* are constructed from the *Interest Map* compared with the *Grey Level Bands Map*. Firstly, only those connected regions that include an "active" pixel in the *Interest Map* are selected. Each one of these regions (or silhouettes) of a uniform grey-level band is defined as a scene spot belonging to a potentially interesting object. As the model works with n grey-level bands, the value at each pixel of the *Working Memory*, $\mathrm{WM}[x,y,t]$, will be the maximum value of the *Working Memory* calculated at each grey-level band

$$\mathrm{WM}[x,y,t] = \arg\max_i \mathrm{WM}_i[x,y,t], \quad \forall i \in [1,\ldots,n] \tag{12}$$

Next the way the *Working Memory* is obtained for each grey-level band is explained. The initial value (patch label) for each pixel (x,y) at grey-level band i is the pixel's position within the image ($\mathrm{label}_{(x,y)} = 1 + \text{coordinate } x$ multiplied by the number of image columns + coordinate y) whenever the pixel is in state "active" in the *Interest Map*. A maximum value ($\mathrm{label}_{\max} =$ number of columns × number of rows + 1) is assigned if the pixel is labeled as "neutral" and a minimum value ($\mathrm{label}_{\min} = 0$) if the pixel is "inhibited". Notice that a computation is only performed on "active" pixels. In this way, performance is enhanced in our motion-based segmentation and tracking system. We concentrate only on regions of interest that contain moving objects instead of the whole image, as in EMBOT (Zaki et al., 2004).

This initial value is compared to the neighbors' values that are at the same grey-level band i in an iterative way up to reaching a common value (common label) for all the pixels of a same element. Finally the value obtained by consensus is assigned to the *Working Memory* at each grey-level band.

2.3. Attention reinforcement

In the *Working Memory* scene object patches whose shape features do not correspond to those defined by the observer may appear at a time instant t. But, if these spots shape features really do not seem to be interesting for the observer, they will appear as "inhibited" in $t+1$ in the *Interest Map* (now, in $t+1$, their shape features will have been obtained). And, this means that in $t+1$ they will disappear from the *Working Memory*. In order to obtain only objects with the desired features at each frame, we have to provide *Attention Reinforcement* by means of accumulative mechanism followed by a threshold. Accumulation is performed on pixels that have a value different from $\mathrm{label}_{\min}$ (pixels that do not belong to labeled zones) in the *Working Memory*. The result of this process offers as output the *Active Attention Focus*, $\mathrm{AF}[x,y,t]$ by means of the *Active Attention Focus Generation* task. Moreover, to obtain the *Active Attention Focus*, an intermediate memory called *Attention Map*, $\mathrm{AM}[x,y,t]$, is used. In particular, pixels that appear with a value different from $\mathrm{label}_{\min}$ in the *Working Memory* reinforce attention in the *Attention Map*, whilst those that appear as a $\mathrm{label}_{\min}$ decrement the attention value. This accumulative effect followed by a

threshold θ maintains "active" a set of pixels that belong to a group of scene object of interest to the observer. Hence, this is a charge/discharge process (permanency effect) similar to the one explained in motion detection

$$\mathrm{AM}[x,y,t] = \begin{cases} \max(\mathrm{Ch}_{\mathrm{AM}}[x,y,t-1] - D_{\mathrm{AM}}, \mathrm{Ch}_{\min}), \\ \quad \text{if } \mathrm{WM}[x,y,t] = \mathrm{label}_{\min} \\ \min(\mathrm{Ch}_{\mathrm{AM}}[x,y,t-1] + C_{\mathrm{AM}}, \mathrm{Ch}_{\max}), \\ \quad \text{otherwise} \end{cases} \tag{13}$$

Now, based on the information provided by the *Attention Map*, objects need to be labeled in the *Active Attention Focus*. This is performed by using an initial value at each pixel of the *Active Attention Focus* as seen in *Attention Capture*.

$$\mathrm{AF}[x,y,t] = \begin{cases} 0, & \text{if } \mathrm{AM}[x,y,t] < \theta \\ \mathrm{label}_{(x,y)}, & \text{otherwise} \end{cases} \tag{14}$$

This initial value is contrasted with the values of the neighbors until a common value for all pixels of a same moving object is reached. Finally, the value obtained by consensus is assigned to the *Active Attention Focus*.

3. 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 real video sequences. We show in this paper the results of our algorithms on a couple of traffic sequences. In the first example we will show the enhancement obtained by establishing the angle of the velocity, whereas in the second example by fixing the module of the velocity it will be possible to obtain only some of the moving objects.

3.1. Hamburg taxi motion sequence

The first example uses the famous *Hamburg Taxi* motion sequence from the University of Hamburg. The sequence contains 20 190×256 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 three vehicles. As our intention is to focus only on the cars, we have to parameterize the system in order to capture attention on elements which fulfill a car's shape features. These shape features are described in Tables 2–4, and are thought to capture all moving cars in the scene, eliminating other moving elements by their size.

Table 2
Spot Shape Features used in *Working Memory*

Parameter	Value (number of pixels)
Spot maximum size	5525
Spot maximum width	85
Spot maximum height	65

Table 3
Object Shape Features used in *Active Attention Focus*

Parameter	Value (in pixels)	Value (ratios)
Object size range	400–5525	
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

Table 4
Parameters of the *Attention Map*

Parameter	Values
Charge constant: C_{AM}	50
Discharge constant: D_{AM}	250
Threshold: θ	100

Table 2 shows the parameters used (as well as their values) to get the patches' shapes in the *Working Memory*. Similarly, in Table 3 we show the parameters and values for the object's shapes in the *Active Attention Focus*. Evidently these are parameters which depend on the scene and the situation of the camera and which have to be adjusted at an initial parameter establishment phase. Lastly, the parameters used to calculate the *Attention Map* are offered in Table 4. Values offered at Tables 2 and 3 are dependent on the values of the parameters of the objects of interest. Table 2 shows the values starting from the maximum sizes of the grey-level spots in a same grey-level band. As in this case the elements of interest may have the spots in one single grey level band, these values correspond exactly to the maximum values of the objects of interest (see Table 3). On the other hand, parameters offered in Table 4 indicate that it is necessary that pixels appear in the *Working Memory* during 3 consecutive time instants to be able to configure elements of the Attention Focus, because the charge value is 50 and the threshold is 100. On the other side, as the maximum charge value is 255 and the discharge value is 250, an element present in the Attention Focus at time instant $t-1$ will immediately disappear at t if it vanishes from the *Working Memory* at that instant t.

Firstly, results are shown in Fig. 6 when no predefined velocity is given to the system. In this figure you may see some images of the sequence of selective attention on moving cars in different time instants. In column (a) some input images of the *Hamburg Taxi* sequence are shown, namely at time instants $t=1$, $t=2$, $t=3$, $t=9$, and $t=18$. Column (b) shows the "active" pixels of the *Interest Map*. This is the result of calculating the presence of motion in the example. Remember that, in the output of this subtask, 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 the same figure, we have drawn in black color

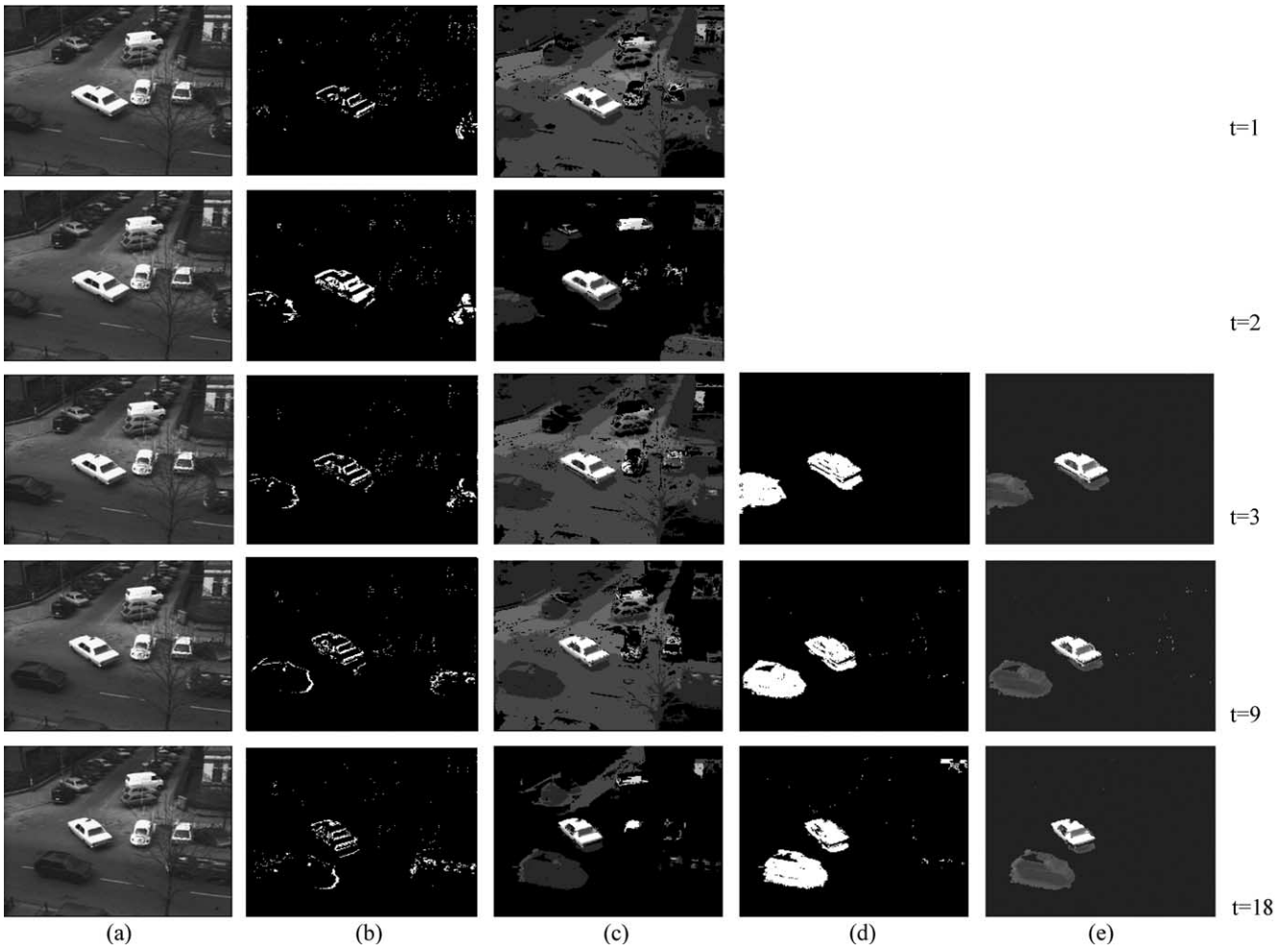


Fig. 6. Sequence of selective attention on moving cars in different time instants: (a) input image, (b) "active" pixels of the *Interest Map*, (c) *Working Memory*, (d) *Active Attention Focus* and (e) *Active Attention Focus* overlapped with input image.

the "inhibited" pixels as well as the "neutral" pixels. In column (c) see the contents of the *Working Memory*, and in column (d) the *Active Attention Focus*. Lastly, on column (e) the *Active Attention Focus* has been overlapped with the input image. Fig. 6d shows the result of the accumulative computation on the *Active Attention Focus* and the later threshold. In this figure, pixels drawn in white color on black background represent image elements where attention has been focused and reinforced through time.

In this example we may notice that the attention focus really corresponds to moving cars. But, although all moving cars are initially detected, only two of the three cars in movement are segmented, because segmentation in grey-level bands (as explained in the *Motion Features Extraction* subtask) unites in this precise case the moving car (indeed, a van) to the tree visible at right lower corner of the image sequence. 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. This example is very helpful to highlight some pros and cons of the described method. Firstly, it is able to discriminate moving objects of a video sequence into different classes of objects. Indeed, in this case the moving pedestrian, belonging to a different class than cars, has been eliminated. This has been shown by the elimination of the pedestrian in the scene through shape features parameterization. But some problems related to partial occlusions affect our method.

Table 5
Performance results

Frame number	Hits (%)	False positives (%)	False negatives (%)	Global failure (%)
Respect to left car ground truth				
3	78.79	14.43	23.04	17.96
9	90.74	28.88	7.26	20.26
18	92.09	22.46	6.66	15.81
Respect to central car ground truth				
3	78.69	7.80	24.96	15.09
9	89.40	9.22	10.77	9.92
18	88.92	13.53	10.78	12.33

Nonetheless we also offer in Table 5 a performance evaluation of our algorithm for the correctly detected cars. In this table we have defined hits, false positives, false negatives, and global failure as

$$hits = \frac{\text{pixels that belong to the attention focus and to the ground truth}}{\text{number of pixels of the ground truth}} \times 100$$

$$false\ positives = \frac{\text{pixels that belong to the attention focus, but not to the ground truth}}{\text{number of pixels of the attention focus}} \times 100$$

$$false\ negatives = \frac{\text{pixels that belong to the ground truth, but not to the attention focus}}{\text{number of pixels of the ground truth}} \times 100$$

$$global\ failure = \frac{false\ negatives + false\ positives}{\text{number of pixels of the ground truth} + \text{number of pixels of the attention focus}} \times 100$$

When looking at the results table (Table 5), we may observe that the highest value for false negatives is got at frame number 3. This is the first frame where it is possible to get elements in the Attention Focus, according to our algorithms with values provided at Table 4. From that moment on, you may note how the percentage of false negatives drops rapidly. It is also important to highlight that the percentage of false positives is substantially lower for the central car than for the taxi, whilst false negatives are higher for the taxi than for the central car.

Now, consider Fig. 7 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, looking for more refined features of the cars present in the traffic scene. Our intention now is to obtain cars that are driving to the right. This has been accomplished by establishing an angle in the range $-22.5°$ to $+22.5°$, that is to say, $-22.5° \leqslant \beta[x, y, t] =\leqslant +22.5°$.

In the results offered in Fig. 7 you may observe that "active" pixels in the *Interest Map* have greatly decreased respect to the results in Fig. 6. This is because 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 when shape features are maintained constant.

3.2. dt_passat motion sequence

The second example uses the *dt_passat* motion sequence, which contains 576×768 pixel image frames. The sequence is a typical traffic sequence, namely a traffic sequence showing the intersection Karl–Wilhelm/Berthold–Straße in Karlsruhe, recorded by a stationary camera from the Institut für Algorithmen und Kognitive Systeme. In this case there are a lot of vehicles present in the scene, but only five cars are moving in the frames shown. Again, just as in example 1, our intention is to focus only on moving cars. Thus, we again 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 6–8.

The huge range covered between the minimum and maximum values for the parameters in Table 7 is due to the

Table 6
Spot Shape Features used in *Working Memory*

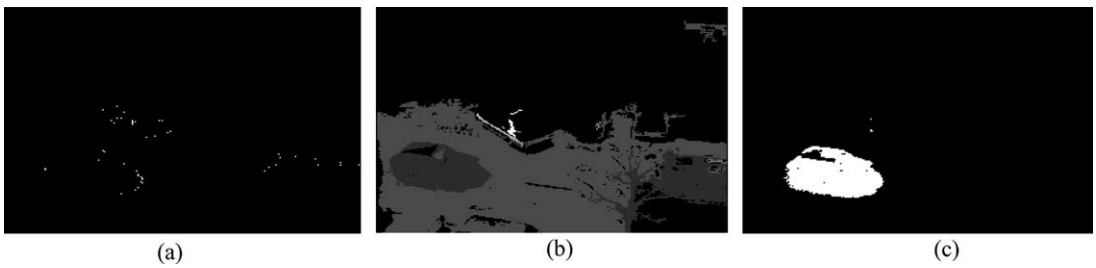
Parameter	Value (number of pixels)
Spot maximum size	4000
Spot maximum width	100
Spot maximum height	100

Table 7
Object Shape Features used in *Active Attention Focus*

Parameter	Value (in pixels)	Value (ratios)
Object size range	200–4000	
Object width range	10–100	
Object height range	10–100	
Object width–height ratio range		0.10–2.00
Object compactness range		0.40–1.00

Table 8
Parameters of the *Attention Map*

Parameter	Values
Charge constant: C_{AM}	200
Discharge constant: D_{AM}	205
Threshold: θ	205



(a) (b) (c)

Fig. 7. Sequence of selective attention on the car moving to the right at time instant $t = 18$. From left to right: (a) "Active" pixels of the *Interest Map*, (b) *Working Memory*, (c) *Active Attention Focus*.

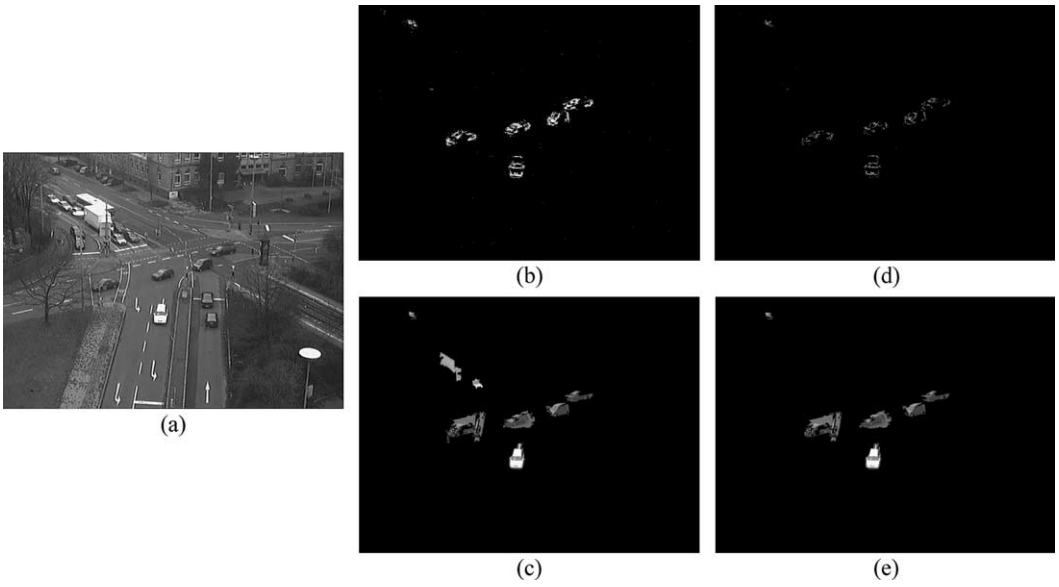


Fig. 8. Selective attention at frame #599 of the sequence: (a) input image, (b) "Active" pixels of the *Interest Map* for all moving cars, (c) *Active Attention Focus* overlapped with input image for all moving cars and (d) "Active" pixels of the *Interest Map* for cars at a velocity of 1 pixel per frame and (e) *Active Attention Focus* overlapped with input image for cars at a velocity of 1 pixel per frame.

very varying values of the proper parameters of the elements of interest (moving cars). The size observed for a same car is very different depending of its proximity to the camera. The values of width and height (and hence, the width/height ratio) are also different depending on the orientation of the cars. Furthermore, the parameters used in Table 8 indicate that it is necessary that pixels appear in two consecutive instants for elements to appear in the *Working Memory* (charge value of 200, and threshold of 205).

In Fig. 8a there is input image # 599 of the sequence of selective attention. Results are shown in Fig. 8b and c when no predefined velocity is given to the system. Row (b) shows the "active" pixels of the *Interest Map.* In row (c) you can see the contents of the *Active Attention Focus* that has been overlapped with the input image. As the system looks for any moving elements in the scene, apparently there are seven or eight moving vehicles, because of illumination changes or little motions of the proper camera. By introducing a concrete velocity module to the system—in this case, 1 pixel per frame— only the real objects of interest are segmented in the scene, as shown in Fig. 8d and e. In this particular example, the importance of introducing motion related parameters to obtain accurate segmentation results has been demonstrated.

4. Conclusions

A model of dynamic visual attention capable of segmenting objects in a real scene has been introduced in this paper. The model proposed enables focusing the attention at each moment on objects that possess certain features and eliminating objects 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. Thus, our proposal follows an attentional-scene-segmentation-integrating approach (Maki et al., 2000), where shape and motion are integrated. The model may be used to observe real environments indefinitely in time with the purpose of tracking a wide variety of objects, including, among others, people, animals, and vehicles. This paper highlights the importance of motion features—motion presence and velocity—to enhance the segmentation and classification of objects in real scenes, providing a way to discriminate the objects of real interest in certain applications.

In relation to the most common motion suppositions—the objects stay in the scene, null or constant motion of the camera, one single object in the scene, no occlusions, slow and continuous motion (Moeslund and Granum, 2001)—the major problem of our approach is related to occlusions, which affect negatively our model. On the contrary, our model is able to classify more than one moving object with no difficulty. In relation to environmental suppositions, that is to say, constant illumination, static image background, uniform background, we can state, without any doubt, that our model is a good one. In relation to suppositions concerning the segmented objects—known starting situations, known objects—let us say that we are able to segment moving objects by introducing just a few simple features of the objects.

When comparing our results of the *Hamburg Taxi* sequence, for instance, with the results of an optic-flow-based approach as the Gautama and Van Hulle (2002) one, we conclude: (a) The intentions of both approaches are different. Gautama and Van Hulle follow the optical-flow approach of Fleet and Jepson (1990), and they are interested in obtaining general motion parameters, whilst our interest is in segmenting a class of objects—in this case,

we are only interested in cars. That is why our method does not obtain the pedestrian as Gautama and Van Hulle do. (b) Gautama and Van Hulle have to label the objects manually after the calculus of direction, speed and spatial location. Our algorithm labels automatically the objects segmented as an important part of visual attention methodology. (c) Our algorithm has problems with occlusions, whilst the optical-flow approach (whose interest is not in labeling objects) apparently does not suffer this limitation in the *Hamburg Taxi* sequence.

In relation to probabilistic approaches to motion grouping and segmentation, e.g. (Robles-Kelly and Hancock, 2004), and also comparing the results with the *Hamburg Taxi* sequence, we observe that again our algorithm loses one car due to the imposed shape size criteria (leading to the union of the tree with the car). Following the terminology of Robles–Kelly and Hancock, our algorithm does not detect one of the three clusters. Nevertheless, when comparing the result of *Active Attention Focus* overlapped with input image of our approach with the final result of Robles–Kelly and Hancock's probabilistic grouping and segmentation method in relation to ground truth, our segmented objects are much closer to the ground truth, as it may be noticed having a quick look.

Summing it up, the main contribution in our approach is the incorporation of the ''motion presence'', ''angle of the velocity'' and ''module of the velocity'' motion features at pixel level. The last two features are rarely used in object segmentation and tracking (Fennema and Thompson, 1979), but have proven to be good discriminants in the examples offered in this paper. Indeed, a couple of examples have been offered where, by incrementing the number of motion features, whilst maintaining the shape features constant, the attention focus is enhanced.

Acknowledgements

This work is partially supported by the Spanish CICYT TIN2004-07661-C02-01 and TIN2004-07661-C02-02 grants. The authors are thankful to the anonymous reviewers for their very helpful comments.

References

Chella, A., Frixione, M., Gaglio, S., 2000. Understanding dynamic scenes. Artificial Intell. 123 (1–2), 89–132.

Fennema, C.L., Thompson, W.B., 1979. Velocity determination in scenes containing several moving objects. Comput. Graphics Image Process. 9, 301–315.

Fernández, M.A., Fernández-Caballero, A., López, M.T., Mira, J., 2003. Length-speed ratio (LSR) as a characteristic for moving elements real-time classification. Real-Time Imaging 9, 49–59.

Fernández-Caballero, A., Fernández, M.A., Mira, J., Delgado, A.E., 2003. Spatio-temporal shape building from image sequences using lateral interaction in accumulative computation. Pattern Recognition 36 (5), 1131–1142.

Fernández, M.A., Mira, J., López, M.T., Álvarez, J.R., Manjarrés, A., Barro, S., 1995. Local accumulation of persistent activity at synaptic level: application to motion analysis. From Natural to Artificial Neural Computation. Springer-Verlag, pp. 137–143.

Fernández-Caballero, A., Mira, J., Fernández, M.A., López, M.T., 2001. Segmentation from motion of non-rigid objects by neuronal lateral interaction. Pattern Recognition Lett. 22 (14), 1517–1524.

Fleet, D.J., Jepson, A.D., 1990. Computation of component image velocity from local phase information. Internat. J. Comput. Vision 5 (1), 77–104.

Gautama, T., Van Hulle, M.M., 2002. A phase-based approach to the estimation of the optical flow field using spatial filtering. IEEE Trans. Neural Networks 13 (5), 1127–1136.

López, M.T., Fernández, M.A., Fernández-Caballero, A., Delgado, A.E., 2003. Neurally inspired mechanisms for the dynamic visual attention map generation task. Computational Methods in Modeling Computation. Springer-Verlag, pp. 694–701.

Maki, A., Nordlund, P., Eklundh, J.-O., 2000. Attentional scene segmentation: integrating depth and motion. Comput. Vision Image Understanding 78 (3), 351–373.

Mira, J., Delgado, A.E., Fernández-Caballero, A., Fernández, M.A., 2004. Knowledge modelling for the motion detection task: the algorithmic lateral inhibition method. Expert Systems with Appl. 27 (2), 169–185.

Moeslund, T.B., Granum, E., 2001. A survey of computer vision-based human motion capture. Comput. Vision Image Understanding 81, 231–268.

Robles-Kelly, A., Hancock, E.R., 2004. A probabilistic spectral framework for grouping and segmentation. Pattern Recognition 37 (7), 1387–1405.

Wu, L., Benois-Pineau, J., Delagnes, P., Barba, D., 1996. Spatio-temporal segmentation of image sequences for object-oriented low bit-rate image coding. Signal Process.: Image Comm. 8 (6), 513–543.

Zaki, M., El Nahas, M.Y., Youssef, M., 2004. EMBOT: an enhanced motion-based object tracker. J. Systems Software 69 (1–2), 149–158.