Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Visual perception based on salient features

Nicolas Courty, Éric Marchand
IRISA - INRIA Rennes
Campus de Beaulieu,
35042 Rennes Cedex, France
Email Nicolas.Courty@irisa.fr, Eric.Marchand@irisa.fr

***Abstract*— We present in this paper an original model to simulate visual perception based on the detection of *salient features*. Salient features correspond to the maximum of conspicuity in a static image or in a sequence of images. We intend to use this information to provide visually interesting targets that can be used in multiple contexts. The extraction process of such an information is performed through multiple steps that correspond to different features locally encoding the conspicuity of a spatial location. We mainly use for those features spatial orientation and motion information, but other types of feature could be used as well. Two kinds of application are presented in this paper: video surveillance application and simulation of the visual perception of a synthetic actor.**

I. Introduction

Detecting visually interesting features in robotics is part of the attention selection process. Such detection can be useful for remote detection, electronic surveillance processes [1] or to simulate typical behaviors, like social interactions, or perception behaviors [2]. In this paper we proposed a bottom-up attention selection mechanism based on static spatial information and motion estimation. We demonstrate the capacities of our system in two different contexts: control of a pan/tilt camera, and simulation of a virtual humanoid perception process.

Our approach is closely related to Wolfe's model of visual attention and visual search [3] which have also been previously used in other models [4], [5], [2]. Those models are based on the construction of a *saliency map*. Saliency maps have been widely used in the computer vision field to solve the attention selection problem [6], [7], [4]. The purpose of the saliency map is to represent the conspicuity (as a scalar value) of each locations of the visual field. Building such maps relies on the broadly tuning of visual stimuli into channels which are also named *feature maps*. Our system aims at simulating the early human vision process in the sense that the feature maps are extracted in a biologically plausible way. We mainly use two types of information: spatial orientation and motion estimation. For orientation information we use a set of Gabor filters, which have the particularity to approximate the receptive field sensitivity of the orientation-sensitive cells of the retina [8], [9]. Motion information has been rarely used as feature map since the construction of saliency maps is usually performed on static images. Since the eyes, or the camera that simulates the visual perception system, are constantly in motion, such an assumption is not realistic. To get such a map with a mobile perception system, we use a robust motion estimator [10] which will be described further in this paper. Conversely, the motion detector used in [2], is based on the absolute difference between two consecutive images, which does not allow to get any interesting results if the camera is moving. If available we can also add a third feature map based on depth information. Such a map has proved to give some important information in the attention selection process [3]. Once the final saliency map has been built, a visual servoing process is used to control the pan/tilt camera or the eyes of the virtual humanoid.

The structure of the paper is the following: first we describe the extraction of the different feature maps, then the control of the robot through the final saliency map is described and illustrated with some examples and results.

II. Extracting feature maps

This section describes the extraction processes of the different feature maps. Those maps give information about spatial frequencies and motion of objects. It is possible to enhance this system with other feature maps (such as color or depth maps). Figure 1 displays the combination of those feature maps and the construction of the saliency map. These different steps are described below.

A. Spatial salient features: 2D Gabor filtering.

Gabor filtering allows to get information about local orientation in the image. Bidimensional Gabor filters are part of a family of bidimensional Gaussian functions modulated by a complex exponential:

$$G_{f,\theta,\sigma}(x,y) = \frac{1}{2\pi\sigma^2} e^{(-\frac{x^2}{2\sigma^2}-\frac{y^2}{2\sigma^2})} e^{(j2\pi f(x\cos(\theta)+y\sin(\theta)))} \quad (1)$$

where f is the frequency, θ the orientation and σ the scale. These filters have the particularity to approximate the receptive field sensitivity of orientation-sensitive cells of the retina [8], [9]. We convolute the original image with a bank of Gabor filters obtained from particular orientations ($\theta_i = \frac{i\pi}{n}$) and particular scales σ_i. If I_o is the resulting

0-7803-7860-1/03/$17.00 © 2003 IEEE

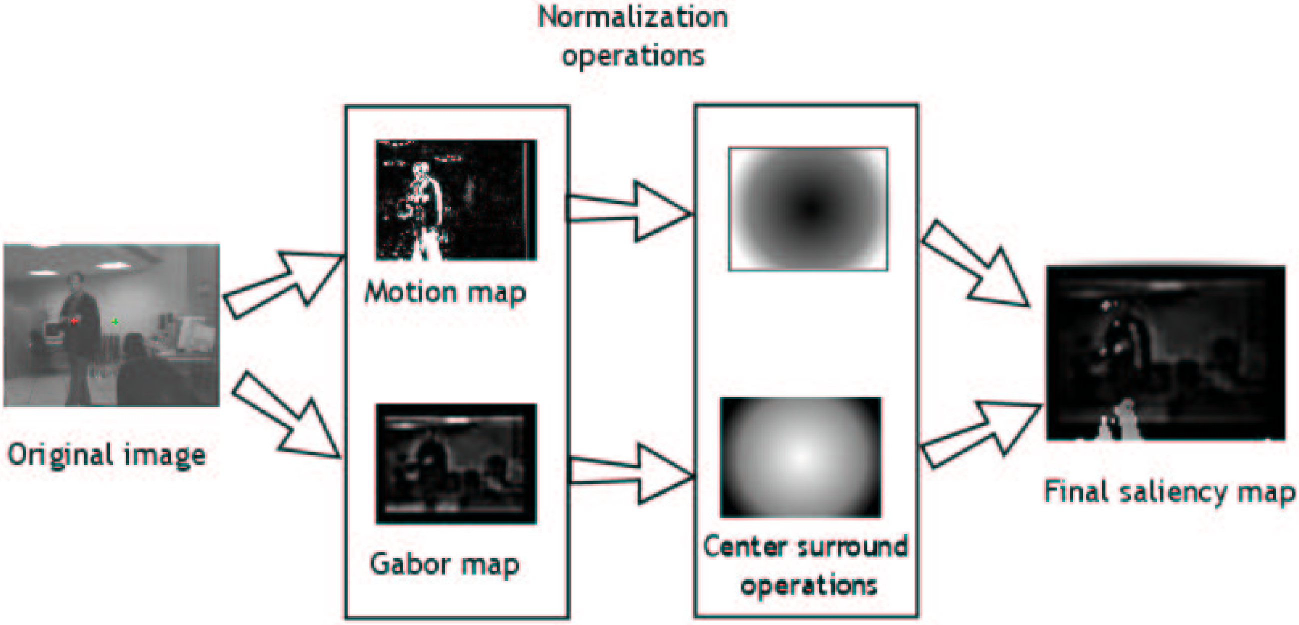

Fig. 1. Obtaining a saliency map based on spatial frequencies and motion information

orientation map, I_o is given by:

$$I_o = \sum_{i=1}^{S} \sum_{j=1}^{O} I \star G_{f,\theta_j,\sigma_i} \tag{2}$$

with S the number of scales, O the number of orientations and $\star$ the convolution operator. The orientation map is then normalized with a traditional normalization operator. Let us note that it exists some fast computation algorithms for Gabor filtering, allowing to run this process at high rates.

B. Spatio-temporal salient features: Robust motion estimator.

The goal here is to detect salient features with respect to their underlying motion. To achieve this goal, we present a technique entirely based on image motion analysis. We perform the robust estimation of the dominant image motion assumed to be due to the camera motion. Then, by considering the outliers to the estimated dominant motion, we can straightforwardly detect objects that are not motionless.

The estimation of the 2D parametric motion model accounting for the dominant image motion is achieved with a robust, multi-resolution, and incremental estimation method exploiting only the spatio-temporal derivatives of the intensity function [10].

1) Quadratic motion models: A 2D image motion model cannot account for the global 3D motion of the object within the scene. A good compromise is thus to consider a 2D quadratic motion model which corresponds to the projection of the rigid motion of a planar surface. This model involves eight free parameters. Let us denote $\Theta = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$, the velocity vector $\mathbf{W}_\Theta(P)$ at pixel $P = (u, v)$ corresponding to the quadratic motion is given by:

$$\mathbf{W}_\Theta(P) = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} a_6 & a_7 & 0 \\ 0 & a_6 & a_7 \end{bmatrix} \begin{bmatrix} u^2 \\ uv \\ v^2 \end{bmatrix} \tag{3}$$

Let us note that it is also possible to consider an affine model with only the 6 first parameters. Though less general it gives also good results in this application context.

2) Dominant image motion estimation: To estimate the dominant image motion between two successive images I_t and I_{t+1}, we use the gradient-based multiresolution robust estimation method described in [10]. To ensure robustness to the presence of independent motion, we minimize a M-estimator criterion with a hard-redescending function. The constraint is given by the usual assumption of brightness constancy of a projected surface element over its 2D trajectory. Thus, the motion model estimation is defined as:

$$\widehat{\Theta} = \operatorname*{argmin}_{\Theta} E(\Theta) = \operatorname*{argmin}_{\Theta} \sum_{P \in I_t} \rho\left(\mathrm{DFD}_\Theta(P)\right) \tag{4}$$

$$\text{with} \quad \mathrm{DFD}_\Theta(P) = I_{t+1}(P + \mathbf{W}_\Theta(P)) - I_t(P). \tag{5}$$

$\rho(x)$ is the Tukey's biweight robust function [11]. In practice, it will be restricted to a specific area of the image. The minimization is embedded in a multi-resolution framework and follows an incremental scheme. At each incremental step k, we can write: $\Theta = \widehat{\Theta}_k + \Delta\Theta_k$, where $\widehat{\Theta}_k$ is the current estimate of the parameter vector Θ. A linearization of $\mathrm{DFD}_\Theta(P)$ around $\widehat{\Theta}_k$ is performed, leading to a residual quantity $r_{\Delta\Theta_k}(P)$ linear with respect to $\Delta\Theta_k$:

$$\begin{aligned} r_{\Delta\Theta_k}(P) &= \nabla I_t(P + \mathbf{W}_{\widehat{\Theta}_k}(P)).\mathbf{W}_{\Delta\Theta_k}(P) \\ &\quad + I_{t+1}(P + \mathbf{W}_{\widehat{\Theta}_k}(P)) - I_t(P) \end{aligned}$$

where $\nabla I_t(P)$ denotes the spatial gradient of the intensity function. Then, we consider the minimization of the expression given by

$$E_a(\Delta\Theta_k) = \sum_P \rho\left(r_{\Delta\Theta_k}(P)\right). \qquad (6)$$

This function is minimized using an Iterative-Reweighted-Least-Squares procedure. It means that the expression (6) is replaced by:

$$E_a(\Delta\Theta_k) = \frac{1}{2}\sum_P \omega(P) r_{\Delta\Theta_k}(P)^2, \qquad (7)$$

and that we alternatively estimates $\Delta\Theta_k$ and update the weights $\omega(P)$ (whose initial values are 1).

This motion estimation algorithm supplies two sets of information:

- the motion parameters $\widehat{\Theta}$ corresponding to the dominant motion between two images (i.e due to the camera motion). This information itself is not really useful for the presented application;
- the map I_ω of the weights $\omega(P)$ used in (7) which account for the fact that pixel P is conforming or not to the computed dominant motion.

Mathematical morphology operators are applied to this outliers map I_ω in order to suppress noise. I_ω is used to detect features salient with respect to their motion in the image sequences. Indeed, each pixel that is not conform to the estimated motion may be considered as belonging to a moving object.

C. Features Integration: Simulating center-surround.

The combination of the two feature maps is performed through a center-surround operation, which globally promotes spatial orientation information in the center of the image, and motion in periphery [12]. Once the Gabor and motion maps have been computed, the two maps are normalized and combined into the final saliency map I_s:

$$I_s = N\left(N\left(I_o - (1 - I_{gauss})\right) + N\left(I_\omega - I_{gauss}\right)\right) \quad (8)$$

if $N()$ is the energy normalization operator and I_{gauss} an image of a Gaussian. This Gaussian performs an exponential decreasing (for the motion map) or increasing (for the Gabor map) of the local conspicuity in the image. This operation can also be performed in a linear way.

III. Application to visual servoing

As already stated we consider two different applications: an automatic video surveillance application and a simulation of a virtual humanoid perception process. Both applications aim at simulating the eyes saccade of the human visual system. In each case the camera (resp. the eyes of the virtual humanoid) are controlled using a classical visual servoing control law.

1) Visual servoing: The *image-based visual servoing* consists in specifying a task as the regulation in the image of a set of visual features[13][14]. Let us denote $\mathbf{p}$ the set of selected visual features used in the visual servoing task. To ensure the convergence of $\mathbf{p}$ to its desired value $\mathbf{p_d}$, we need to know the interaction matrix (or image Jacobian) $\mathbf{L_p}$ that links the motion of the object in the image to the camera motion. It is defined by the classical equation [13]:

$$\dot{\mathbf{p}} = \mathbf{L_p}\mathbf{T_c} \qquad (9)$$

where $\dot{\mathbf{p}}$ is the time variation of $\mathbf{p}$ (the motion of $\mathbf{p}$ in the image) due to the camera motion $\mathbf{T_c}$.

The control law that ensures an exponentially decrease of the error and that computes the velocity given as input to the camera is given by:

$$\mathbf{T_c} = -\lambda \mathbf{L_p^+}(\mathbf{p} - \mathbf{p_d}) \qquad (10)$$

In our focusing task process $\mathbf{p} = (x, y)$ is a point defined as the as a local maximum of energy in the saliency map. Since we specify a focusing task we got $\mathbf{p_d} = (0, 0)$ which correspond to a point at the center of the image. The interaction matrix for a pan/tilt system is then given by:

$$\mathbf{L_p} = \begin{pmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{pmatrix} \qquad (11)$$

2) Video surveillance application: The principle of this application is quite simple: each frame acquired by the camera is processed using the presented algorithm. A feature map that includes both spatial and spatio-temporal information is created. The global maximum of the map is determined through a simple scanning of the feature map and given as input of the visual servoing process. The pan/tilt camera is then focused at this point using the control law presented in equation (10).

Let us note that it is not always necessary to process all the images acquired by the camera. Indeed if, as reported, all the images are processed the camera behavior is similar to the eye saccades. A tracking process can also be considered. In that case the most salient point is detected using the presented approach and this point is then tracked using a SSD-based tracking algorithm [15].

When the tracking fails, after a given number of frames or when required by an operator, a new saliency map is computed and a new point is selected and tracked.

3) Virtual humanoid application: We present now a new and original context intending to simulate the visual perception of a synthetic actor. Within computer graphics field, simulating virtual humans has become a challenging task. Animating such an autonomous actor within a virtual environment requires most of time the modeling of a perception-decision-action cycle. To model a part of the perception process corresponding to spontaneous looking, we introduced the visual perception process described in this article. The initial image is an image rendered from the point of view of the humanoid. We use another feature map based on depth information, as far as it is possible to get this information easily through the graphical rendering pipeline (*Z-Buffer*). Our hypothesis is that close objects are more conspicuous than distant ones.

Once the saliency map has been built, it is possible to build a list of salient points (fixation points). The process used to build such a list is depicted in Figure 2. The saliency map is cut out of a square zone to prevent one zone to attract all the fixations (inhibition mechanism). The most salient zone is selected and the process is then repeated until the saliency goes under a given threshold. We can thus handle different gazing locations from one point of view, which allows to run computation of the saliency map at rather low rates.

The points are then given as inputs to our image-based animation engine [16], [17]. As in the previous case the motions of the torso, head and eyes of the virtual humanoid are controlled using a visual servoing control law similar to the one given in equation (10) and presented in [17].

IV. Experimental Results

4) Controlling camera orientation using feature maps: As described in the previous section, we proposed results in the target selection and tracking process. We servo a pan/tilt camera using the classical visual servoing control law given in equation (10). The feature maps and control law are computed on a PC Pentium III (1.2Ghz) at 5Hz.

Figure 3 shows the output of the various algorithm described in this paper. The initial image is shown in Figure 3.d. Figure 3.a shows the result of the filtering using Gabor filter (8 orientations and 4 scales were used). In this first feature map, the most conspicuous locations are the light spots in the foreground and some parts of the shelves in the middle of the image (part of the saliency map corresponding to the image borders are not consider in the features selection process afterward). Figure 3.b show "motion saliency map". As stated, this map shows the points (in white) that are not conform with the dominant image motion (outliers). The value in each point of the map reflects the confidence we have in the motion consistency. Most of these points belong to moving objects. However, let us note here, due to shallow effects, some of these points may belong to static objects whose 2D apparent motion is very different of the other object 2D motion (this is closely related to the objects depth). Most of the time, these points may also be considered as salient objects though for a different reason. Figure 3.c shows the result of the integration of the two maps. The "center-surround" attenuation is applied to simulate the importance of spatial information in the center of the visual field and of the motion in its periphery.

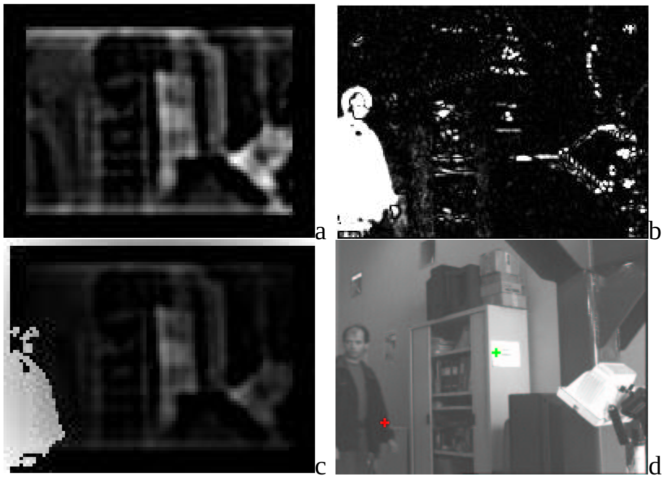

Fig. 3. Computing features maps: (a) spatial saliency map using 2D Gabor filters (b) motion feature map using motion estimation (c) Final features maps obtained as described in section II-C (d) focused point (red) on the initial image.

Figure 4 shows some snapshots of the complete selection and focusing process on a complete images sequence. The first row shows the computed feature maps and the second row shows the most salient point (red cross) and its desired location in the image (green cross). On Figure A1 a point on the shelves is detected as a salient point and the camera focuses on this point (Figure A2) using the visual servoing process. When a pedestrian enter the camera field of view (A3), according to an important motion, it is selected as a salient feature. When it stops moving (B1), it disappears from the feature map (only remain the face and hands that are spatially salient features on their own). We shall not describe the reminder of the sequence but it is important to note that, despite the camera egomotion, the moving object can be easily recovered and incorporated within the feature maps.

5) Virtual humanoid: We have tested our system on a virtual character wandering along the street into a virtual city. This animation was generated in real-time on SGI330 Linux PC, under our animation and simulation framework. Figure 5 shows different salient maps computed along its path. A special thread was designed to compute those

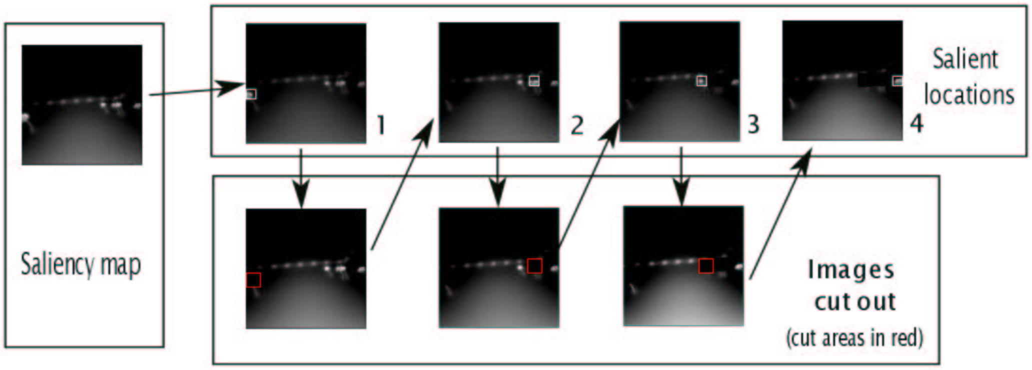

Fig. 2. Detecting a list of fixations; white squares show maximums of energy, red squares show cut out zones in the saliency map

maps, and the time elapsed during its execution was around 100 ms, which is less than the time needed to gaze at the different conspicuous zones. One can observe that most of the important objects are detected: traffic signs, houses, sidewalks. Hence, at the end of the animation, our humanoid had looked at many different locations. Conversely, a traditional animation system would have required to set, in the environment, some predefined targets which assumes an important knowledge about the environment. Meanwhile, the resulting animation is quite interesting in the sense that it gives to our virtual character a lively, human-like behavior (see Figure 6 for snapshots of the animation).

V. Conclusion

In this paper we have presented new and original saliency or feature maps. We have designed a simple and original model of saliency map that considers spatial, motion and when available depth information. We think that considering such information can be useful to improve attention selection models. Integration of the feature map within a simple visual servoing process allows the automatic selection of interesting features and their tracking. Such a system mimics the eye saccades as describe in [6]. Considering computer graphics applications, while other approaches didn't take into account the properties of the human visual perception system, we have proposed a way to simulate, in a more biologically plausible way, humanoids evolving in virtual environments.

Dealing with future work, one of the most important problems lies in the comparison of the output of our system with a real human visual system. Improvements can also be performed in the definition of the saliency map, notably thanks to the addition of other types of feature maps (e.g., color). Considering the temporal aspects, it seems to be of need to add a top-down attention selection process (based on memory) that can come up with the redundancy problem existing between two or more consecutive maps.

VI. References

[1] A. Crétual and F. Chaumette, "Application of motion-based visual servoing to target tracking," *Int. Journal of Robotics Research*, vol. 20, no. 11, pp. 878–890, November 2001.

[2] C. Breazeal and B. Scassellati, "A context-dependent attention system for a social robot," in *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99-Vol2)*, San Francisco, 1999, pp. 1146–1153.

[3] J.M. Wolfe and G. Gancarz, "Guided search 3.0," in *Basic and Clinical Applications of Vision Science*, Dordrecht, Netherlands, 1996, pp. 189–192, Kluwer Academic.

[4] L. Itti, J. Braun, D.. Lee, and C. Koch, "A model of early visual processing," in *Advances in Neural Information Processing Systems*. 1998, vol. 10, The MIT Press.

[5] J. Driscoll, R. Peters, and K. Cave, "A visual attention network for a humanoid robot," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS-98)*, Amsterdam, October 1998, pp. 1022–1028.

[6] A.L. Yarbus, *Eye Movements and vision*, Plenum Press, New York, 1967.

[7] S. Culhane and J.K. Tsotsos, "An attentional prototype for early vision," in *Proceedings of Computer Vision (ECCV '92)*, Berlin, Germany, May 1992, vol. 588 of *LNCS*, pp. 551–562.

[8] S. Marcelja, "Mathematical description of the responses of simple cortical cells," *Journal of Optical Society of America*, vol. 70, pp. 1297–1300, 1980.

[9] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, no. 7, pp. 1169–1179, July 1985.

[10] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models.," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, December 1995.

[11] P.-J. Huber, *Robust Statistics*, Wiler, New York, 1981.

[12] B. Ter Haar Romeny, *Front-End Vision and Multiscale Image Analysis: Introduction to Scale-Space Theory*, Kluwer Academic Publishers, 2002.

[13] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.

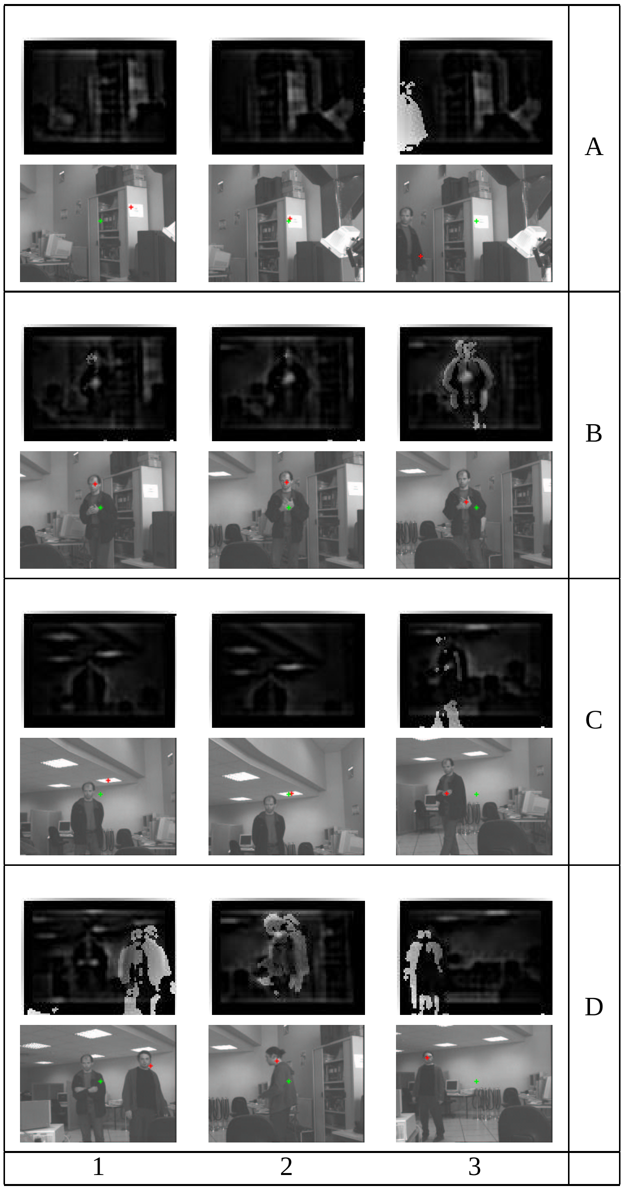

Fig. 4. Visual servoing using a combination of spatial and motion feature maps Each cell show the feature map and, bellow, the motion “salient” point (red cross) and its desired position (green cross).

[14] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[15] P. Hager, G. et Belhumeur, “Efficient region tracking with parametric models of geometry and illumination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, October 1998.

[16] N. Courty and E. Marchand, “Computer animation: a new application for image-based visual servoing,” in *IEEE Int. Conf. on Robotics and Automation*, Seoul, South Korea, May 2001, vol. 1, pp. 223–228.

[17] N. Courty, E. Marchand, and B. Arnaldi, “Through-the-eyes control of a virtual humanoïd.,” in *IEEE Computer Animation 2001*, H.-S. Ko, Ed., Seoul, South Korea, November 2001, pp. 74–83.

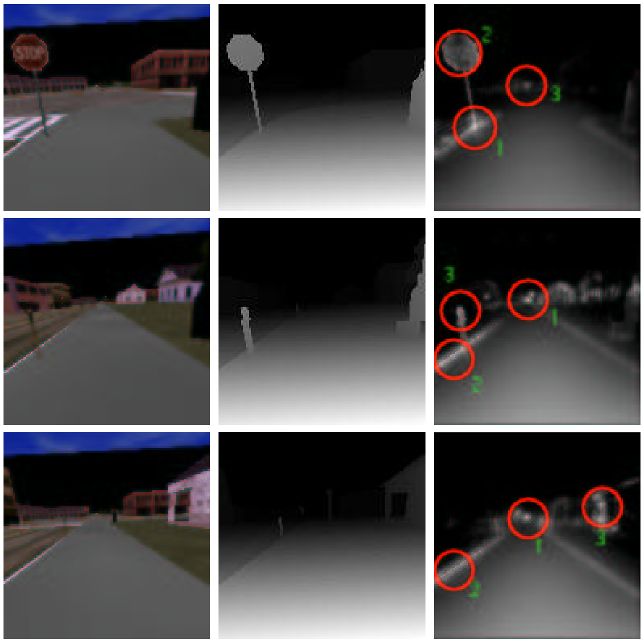

Fig. 5. Different saliency maps obtained during the wandering. First column shows the corresponding subjective view, second column shows the depth buffer and the last column is the saliency map with the location of three fixations

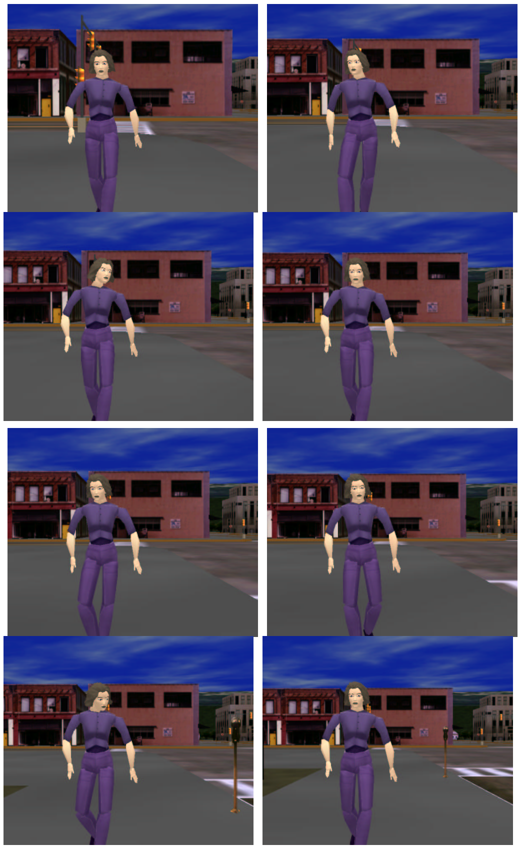

Fig. 6. Snapshots of the animation where our virtual characters walks along the sideway looking at some salient features computed from images “acquired” from its own viewpoint (see Figure 5).