Visual Perception System for a Social Robot (VISOR)

Deliverable 2

ATTENTION MECHANISM

Rebeca Marfil

Grupo de Ingeniería de Sistemas Integrados

Dpto. Tecnología Electrónica , Universidad de Málaga

Campus de Teatinos 29071 Málaga (Spain)

www.grupoisis.uma.es

Due date of deliverable: March 15th, 2005
Actual submission date: March 24th, 2005
Start date of project: September 15th, 2005
Duration: 12 months

Organizational name of responsible for this deliverable:
Grupo de Ingeniería de Sistemas Integrados
Revision: 0.2
Dissemination level: PU

Contents

List of Figures

1 Introduction

A social robot is a robot which is capable of communicate and interact with humans and other social robots. This interaction implies that the social robot must simultaneously perceive a great variety of natural social cues from visual and auditory channels, and to deliver social signals. This social behaviour can be evaluated in a more easy way if it is imposed that a socially interactive robot senses and interprets the same phenomena that humans observe. Besides, social robots must proficiently interpret human activity and behaviour.

If most human-oriented perception is based on passive sensing (artificial vision and auditory), the vision system is the responsible of solving the problems of identifying faces, measuring head and hands poses, capturing human motion, recognizing gestures and reading facial expressions to emulate human social perception. This information permits that the robot be able to identify who the human is, what the human is doing, how the human is doing it and even to imitate the human motion. Thus, the robot could treat the human as an individual, understand his/her surface behaviour, and potentially infer something about his/her internal states (e.g., the intent or the emotive state). On the other hand, these human-related tasks must be run in parallel with object-related ones, which permit the robot to recognize objects extracted from the environment. In order to achieve these goals, the visual perception system of the social robot should imitate the ability of natural vision systems to select the most salient information from the broad visual input. The use of attention to reduce the amount of input data has two main advantages: i) the computational load of the whole system is reduced, and ii) distracting information is suppressed. An attention mechanism is central to a system requiring a selection of the relevant information on which the system activities are based.

Probably one of the most influential theoretical models of visual attention is the spotlight metaphor [1], by which many concrete computational models have been inspired [2][3][4]. These approaches are related with the *feature integration theory*, a biologically plausible theory proposed to explain human visual search strategies [5]. All are organized into two main stages. First, in a preattentive task-independent stage, a number of parallel channels compute image features. The extracted features are integrated into a single saliency map which codes the saliency of each image region. The most salient regions are selected from this map. Second, in an attentive task-dependent stage, the

spotlight is moved to each salient region to analyze it in a sequential process. Analyzed regions are included in an inhibition map to avoid movement of the spotlight to an already visited region. Thus, while the second stage must be redefined for different systems, the preattentive stage is general for any application.

Although these models have good performance in static environments, they cannot in principle handle dynamic environments due to their impossibility to take into account the motion and the occlusions of the objects in the scene. In order to solve this problem, an attentional control mechanism must integrate depth and motion information to be able to track moving objects [6]. Thus, Maki et al. [7] propose an attention mechanism which incorporates depth and motion as features for the computation of saliency. Baker and Mertsching [6] also compute depth as a feature, but use dynamic neural fields to track the most salient regions of the saliency map in a semiattentive stage. The method is reported to take 30 seconds per frame, which makes its application to real-time, interactive systems unfeasible.

In this report a general purpose attentional mechanism based on the feature integration theory is presented. It is capable of handling dynamic environments, and detecting human faces or hands in a fast way. The proposed system integrates bottom-up (data-driven) and top-down (model-driven) processing. The bottom-up component determines and selects salient image regions by computing a number of different features. The top-down component makes use of object templates to filter out data and only track significant objects. Fig. 1.a shows the overview of the proposed architecture. The presented work is centered in the task-independent stage of a feature integration approach. Our method is related to the recent proposal of Backer and Mertsching [6] in several aspects. The first is the use of a preattentive stage in which parallel features are computed and integrated into a saliency map. However, in contrast with this and other attentional systems, we have introduced the skin colour as input feature in order to detect human faces or hands as possible regions of interest. Thus, in this work, skin colour is first detected using a chrominance distribution model [8] and then integrated as input feature in a saliency map. Other similarity is that this preattentive stage is followed by a semiattentive stage where a tracking process is performed. But, while Backer and Mertsching's approach performs the tracking over the saliency map by using dynamics neural fields, our method tracks the most salient regions over the input image with a hierarchical approach based on the Bounded Irregular Pyramid [9]. The output regions of the tracking

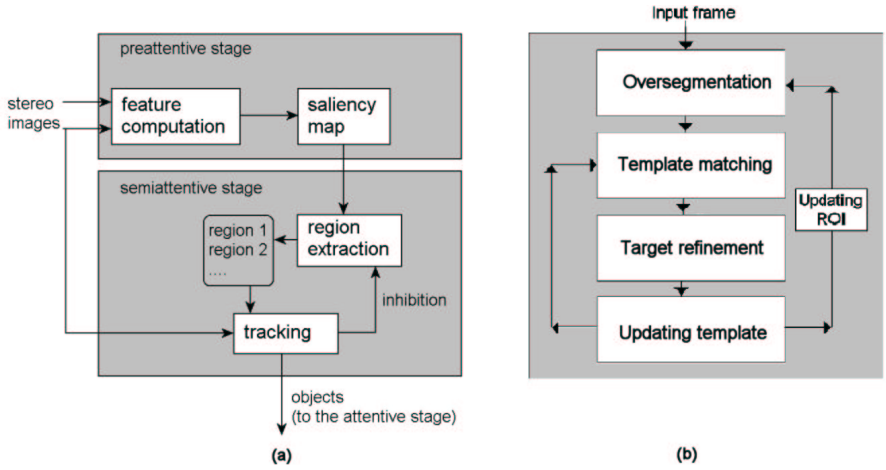

Figure 1: a) Overview of the proposed attentional mechanism and b) overview of the tracking algorithm

algorithm are used to implement the inhibition of return and avoid revisit or ignore objects. The main disadvantage of using dynamic neural fields for controlling behavior is the high computational cost for simulating the field dynamics by numerical methods. The Bounded Irregular Pyramid approach allows real time tracking of a non-rigid object without a previous learning of different objects views [10].

2 Preattentive stage

The proposed attentional mechanism uses a number of features computed from the available input image in order to determine how interesting a region is in relation to others. These features are independent of the task and they allow to extract the most interesting regions of the image. Besides, they allow to distinguish locations where a human may be placed. The chosen features are colour and intensity contrast, disparity and skin colour. Attractivity maps are computed from these features, containing high values for interesting regions and lower values for other regions in a range of [0...255]. The integration of these feature maps into a single saliency map allows to determine what regions of the input image are the most interesting. Other features can be easily added without changes in the following steps.

2.1 Computation of early features

2.1.1 Feature: colour contrast

Colour is employed for all attentional models because it can distinguish important aspects of the objects. The first step to compute colour contrast is to choose an adequate colour space. We have selected the HSV colour space due to its intuitive representation and the facility to separate the chrominance from the luminance information. Thus, the RGB colour information is firstly transformed into the HSV colour space. Second, the input image is segmented using a Bounded Irregular Pyramid (BIP) [9] in order to obtain homogeneous colour regions. And finally, in contrast with other methods which only compute the colour contrast for a set of colours [6], the proposed algorithm computes a colour contrast value for each homogeneous colour region of the input image independently of its colour. The colour contrast of a region i is calculated as the mean colour gradient MCG_i along its boundary to the neighbour regions:

$$MCG_i = \frac{S_i}{PL_i} \sum_{j \in N_i} pl_{ij} * d(< C_i >, < C_j >) \tag{1}$$

being PL_i the length of the perimeter of the region i, N_i the set of regions which are neighbours of i, pl_{ij} the length of the perimeter of the region i in contact with the region j, $d(< C_i >, < C_j >)$ the Euclidean distance between the colour mean values $< C >$ of the regions i and j and S_i the mean saturation value of the region i. Fig. 2.b shows the colour contrast saliency map associated to Fig. 2.a. It must be noted that the use of S_i in the MCG avoids that colour regions with low saturation (grey regions) obtain a higher value of colour contrast than pure colour regions. The problem is that white, black and pure grey regions are totally suppressed. To take into account these regions, the intensity contrast is computed.

2.1.2 Feature: intensity contrast

This feature map is computed in a similar way to the previous one. The intensity contrast of a region i is the mean intensity gradient MIG_i along its boundary to the neighbour regions:

$$MIG_i = \frac{1}{PL_i} \sum_{j \in N_i} pl_{ij} * d(< I_i >, < I_j >) \tag{2}$$

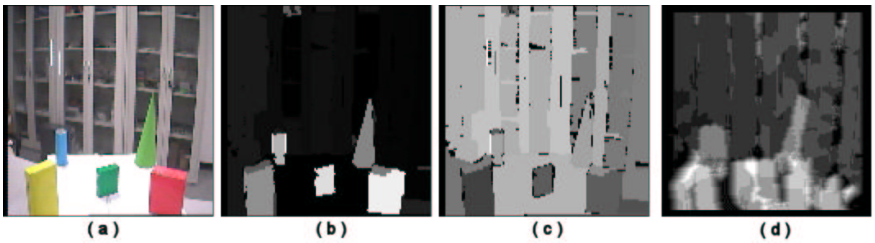

Figure 2: Colour and intensity contrast computation: a) Left input image; b) colour contrast saliency map; c) intensity contrast saliency map and d) disparity map

being $< I_i >$ the mean intensity value of the region i. Fig. 2.c shows the intensity contrast saliency map associated to Fig. 2.a.

2.1.3 Feature: skin colour

Skin colour is an important tool to distinguish locations in which a human is probably located. In order to segment skin colour regions from the input image, the skin colour segmentation method proposed in Deliverable 1.1 of VISOR project has been used. This method is briefly described below:

The first step to identify skin colour regions in an image is to compute an accurate skin chrominance model using a colour space. The used skin chrominance model has been built over the TSL colour space and it is based on the method proposed by Terrillon and Akamatsu [8]. Thus, the skin colour is modelled in the TSL colour space as an unimodal elliptical Gaussian joint probability density function computed on a set of 120 training images. This function is represented by its covariance matrix C_s and its mean vector m_s. The Mahalanobis metric is used to determine a threshold value T_s that efficiently discriminates between human skin and other objects.

Once the chrominance model has been established, the steps to segment skin regions from an image are the following: first, the RGB input image is transformed into a TSL image. Second, the Mahalanobis distance from each pixel (i, j) to the mean vector is computed. If this distance is less than T_s then the pixel (i, j) of the skin feature map is set to 255. In any other case, it is set to 0. Fig. 3.b shows the skin colour saliency map associated to Fig. 3.a.

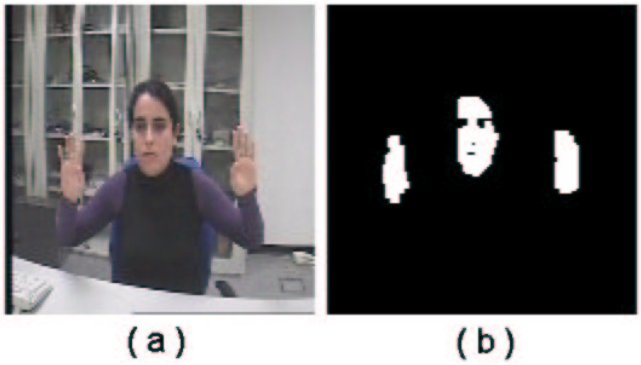

Figure 3: Skin colour computation: a) Left input image; and b) skin colour map

2.1.4 Feature: disparity

In our system, relative depth information is obtained from a dense disparity map which is scaled in the range [0 ... 255], being 255 the disparity value of the closest region. Thus, closed regions are considered more important. As disparity estimator we employ the zero-mean normalized cross-correlation measure. It is implemented using the box filtering technique. This allows to achieve fast computation speed [11].

Each computed zero-mean cross-correlation value is stored in a 3D disparity space with size MxNxD, where MxN is the image size and D the maximum disparity range. The disparity map is found in this space by obtaining the global 3D maximum surface which is computed using the two-stage dynamic programming technique proposed by Sun [11]. Fig. 2.d shows the disparity map associated to Fig. 2.a.

2.2 Saliency map computation

Similarly to other models [4][6], the saliency map is computed by combining the feature maps into a single representation. In our case, all the feature maps are normalized to the same dynamic range, in order to eliminate cross-modality amplitude differences due to dissimilar feature extraction mechanisms. A simple normalized summation has been used as feature combination strategy because, although this is the worst strategy when there are a big number of feature maps [12], it has been demonstrated that its performance

is good in systems with a small number of feature maps. Fig. 4.b shows the saliency map associated to Fig. 4.a.

3 Semiattentive stage

Once the saliency map is calculated, it is segmented in order to obtain regions with homogeneous saliency. Among the set of obtained regions, only big enough regions with a high saliency value are taken into account. In our experiments, a region has been considered as a salient one if its size is greater than the 0.2% of the input image size and its saliency is greater than the 60% of the saliency map maximum value. These threshold have been empirically obtained and work correctly in most cases.

A general problem in attentional mechanisms is to avoid revisiting or ignoring salient objects of the image when the system is working in a dynamic environment with moving objects. To solve this problem, it is necessary to include in the system a mechanism to avoid extracting the same objects in different frames, although they will be in different positions in the images. The attentional mechanism should be object-oriented and not region-oriented. The way to solve the problem of revisiting or ignoring objects is called "inhibition of return". The proposed attentional mechanism implements the inhibition of return by including a tracking process in the semiattentive stage to track the objects extracted from the scene. This tracking allows to know the position in the current frame of the previously extracted objects. It prevents the attentional mechanism from wrongly identify them as new objects.

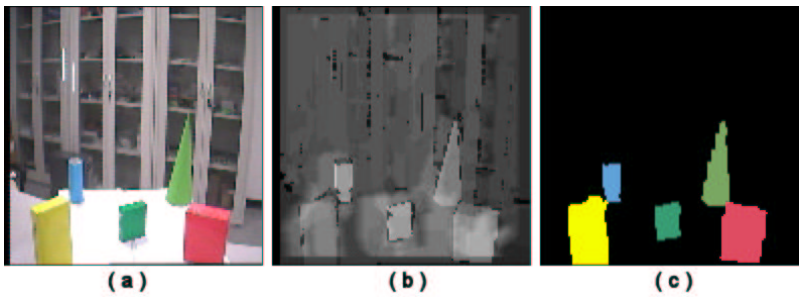

Figure 4: Saliency map computation and targets selection: a) Left input image; b) saliency map; and c) selected targets

The tracking algorithm is based on the Bounded Irregular Pyramid (BIP) [9]. It permits to track non-rigid objects without a previous learning of different object views in real time. To do that, the method uses weighted templates which follow up the viewpoint and appearance changes of the objects to track. The templates and the targets are represented using BIPs.

The most salient regions obtained by segmentation of the saliency map are directly related to homogeneous colour regions of the segmented left input image. These homogeneous colour regions are the targets to track. Fig. 4.c shows the selected targets associated to the saliency map in Fig. 4.b. It must be noted that targets are not necessary associated with homogeneous saliency regions, but with homogeneous colour ones. This mechanism provides better object candidates to the tracking stage. Once the targets are choosen, the algorithm extracts its hierarchical representations. Each hierarchical structure is the first template $M_r^{(0)}$ and its spatial position is the first region of interest $ROI_r^{(0)}$, where $r \in [1...N]$ and N is the number of salient regions to track.

Although in the following steps the general implementation of the tracking algorithm to work with colour objects is showed, it must be noted that when the target to track is a skin colour region the approach is slightly different. In the case of general colour objects the similarity criterium between nodes to build the BIP and to perform the tracking is to have similar colour. In the case of skin colour regions the employed similarity criterium is to be a skin or a non-skin node using the skin segmentation method explained in Section 2.1.3. The main steps of the proposed tracking algorithm (Fig. 1.b) are explained in the following subsections.

3.1 Over-segmentation

The first step is to represent hierarchically the regions of interest $ROI_r^{(t)}$, $\forall r \in [1...N]$, into the same hierarchical structure using the Bounded Irregular Pyramid. The BIP is a 4 to 1 structure where each level is generated by reducing the resolution of the previous one by a factor of four. Thus, a node of a new level l is generated by averaging the colour of the four nodes immediately below at level l-1. Contrary to other 4 to 1 structures, the BIP is an irregular structure in which not all sets of 4 nodes of a level originate a new node in the upper level. Thus, a new node (or valid node) is generated only when the four nodes below have similar colour. The resulting structure is an uncomplete regular pyramid. Each pyramidal node n is identified by

(i, j, l) where l represents the level and (i, j) are the (x, y) coordinates within the level. To build the different levels of the pyramid, each node has five parameters associated:

- Homogeneity, $Hom(i, j, l)$. $Hom(i, j, l)$ is set to 1 if the four nodes immediately underneath have colour difference values below a threshold T_C and their homogeneity values are equal to 1. Otherwise, it is set to 0. In the base or level 0, $Hom(i, j, 0) = 1$ if $(i, j) \in ROI_r^{(t)}$. Otherwise, $Hom(i, j, 0) = 0$.

- Chromatic phasor, $S_{\angle H}(i, j, l)$. The chromatic phasor is composed of the saturation (S) and the hue (H) values of the HSV colour space. If the cell is homogeneous, $S_{\angle H}(i, j, l)$ is equal to the average of the chromatic phasors of the four cells immediately underneath. If the cell is not homogeneous, $S_{\angle H}(i, j, l)$ is set to a null value.

- Intensity, $V(i, j, l)$. If the cell is homogeneous, $V(i, j, l)$ is equal to the average of the intensity values associated to the four nodes immediately underneath. Otherwise, it is set to a null value.

- Area, $A(i, j, l)$. It is equal to the sum of the areas of the four nodes immediately underneath.

- Parent link, $(X, Y)_{(i,j,l)}$. If $Hom(i, j, l)$ is equal to 1, the values of the parent link of the four cells immediately underneath are set to (i, j). Otherwise, these four parent links are set to a null value.

It must be noted that only nodes presenting a homogeneity value equal to 1 are valid nodes. Each valid node is linked to a homogeneous region at the base.

Each $ROI_r^{(t)}$ depends on the target position in the previous frame $T_r^{(t-1)}$, being updated as it is described in subsection 3.5. The hierarchical structure can be represented in each level as:

$$ROI^{(t)}(l) = \bigcup_{ij} p^{(t)}(i, j, l) \tag{3}$$

being p a node of the bounded irregular pyramid built over the ROI.

It must be noted that, once the structure is generated, valid nodes without parent are regarded as roots of trees defined by their links to lower level nodes. Thus, they perform an over-segmentation of the regions of interest by defining classes at the base of the structure.

3.2 Template Matching

Each template $M_r^{(t)}$ and target $T_r^{(t)}$ in every frame t are represented using BIP:

$$M_r^{(t)}(l) = \bigcup_{ij} m_r^{(t)}(i,j,l) \tag{4}$$

$$T_r^{(t)}(l) = \bigcup_{ij} q_r^{(t)}(i,j,l) \tag{5}$$

Fig. 5 presents an example of template representation using a 4-level pyramid. The base of the pyramid (level 0) contains 64x64 pixels. Fig. 5 shows how pixels at level l are arranged into sets of 2x2 elements to create a node at level l+1. It must be noted that nodes related to non homogeneous sets (white nodes in the figure) are removed from the structure and they are not taken into account in the tracking procedure.

In this step, the algorithm looks for the targets $T_r^{(t)}$ using a hierarchical template matching approach. Starting in the highest level, each template $M_r^{(t)}(l)$ is placed and shifted in its $ROI_r^{(t)}(l)$ until the target is found or until $ROI_r^{(t)}(l)$ is wholy covered. If a $ROI_r^{(t)}(l)$ was wholy covered and the target was not found, this target localization process would continue in the level below. When all the targets are searched in a level, the process continues in the level below looking for the targets which have not been previously found. The displacement of each template can be represented as $d_{r_k}^{(t)} = (d_{r_k}^{(t)}(i), d_{r_k}^{(t)}(j))$ in the range $[d_{r_0}^{(t)} d_{r_f}^{(t)}]$. $d_{r_f}^{(t)}$ is the displacement that situates the template in the position where the target is placed in the current frame. The algorithm chooses as initial displacement in the current frame $d_{r_0}^{(t)} = d_{r_f}^{(t-1)}$. In order to localize the target and obtain $d_{r_f}^{(t)}$, the overlap $O_{d_{r_k}^{(t)}}^{(t)}$ between $M_r^{(t)}(l)$

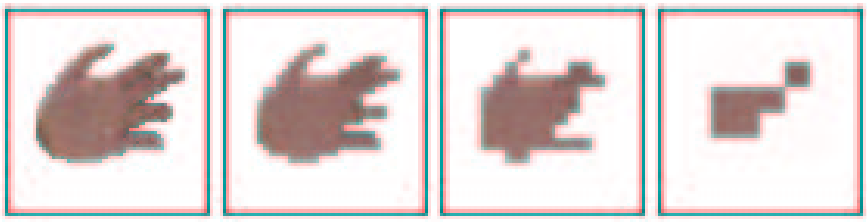

Figure 5: Template hierarchical representation

and $ROI_r^{(t)}(l)$ in each template displacement k is calculated as:

$$O_{d_{r_k}^{(t)}}^{(t)} = \sum_{ij \in \xi} w_r^{(t)}(m_r(i,j,l_w^{(t)})) \tag{6}$$

being $w_r^{(t)}(m_r(i,j,l))$ a weight associated to $m_r^{(t)}(i,j,l)$ in the current frame t, as explained in subsection 3.4. ξ is the subset of pixels that satisfy the following condition:

$$g(r,s) < T_C \tag{7}$$

with

$$r = f(m_r^{(t)}(i,j,l_w^{(t)}), a(t))$$
$$s = p^{(t)}(i + d_{r_k}^{(t)}(i), j + d_{r_k}^{(t)}(j), l_w^{(t)})$$

being $g(r,s)$ the colour distance between r and s and T_C the colour threshold employed in the pyramid generation. $f(m_r^{(t)}(i,j,l_w^{(t)}), a(t))$ is a coordinate transformation of $m_r^{(t)}(i,j,l_w^{(t)})$ that establishes the right correspondence between $m_r^{(t)}(i,j,l_w^{(t)})$ and $p^{(t)}(i + d_{r_k}^{(t)}(i), j + d_{r_k}^{(t)}(j), l_w^{(t)})$. $a(t)$ denotes the parameter vector of the transformation, which is specific for the current frame. Eq. (7) is satisfied when a match occurs.

We consider that a target has been found in a position if the overlap in that position is higher than 70%. All the ROI pixels that match with pixels of the template are marked as pixels of the target in the whole structure $ROI_r^{(t)}$. Thus, the hierarchical representation of the target $T_r^{(t)}$ is obtained.

3.3 Target Refinement

To achieve a more accurate appearance of the targets, each $T_r^{(t)}$ is rearranged level by level following a top-down scheme. From each node of $ROI_r^{(t)}$ that is not in $T_r^{(t)}$ a search is performed for all valid neighbour nodes in a 3x3 vicinity which belong to the target and have a similar colour to it. Among the set of candidates, the studied node is linked to the most similar one to it.

3.4 Updating Templates

The templates are updated in each frame in order to follow up varying appearances. To do that, we associate a probability value or weight $(w_r^{(t)}(m_r(i,j,l)))$ with each valid node of the template model. This value places more importance to more recent data and permits to forget older data in a linear and

smooth manner. Each template is updated as shown in equations (8) and (9):

$$m_r^{(t+1)}(i,j,l) = \begin{cases} m_r^{(t)}(i,j,l) & \text{if no match} \\ f^{-1}(q_r^{(t)}(i,j,l), a^{(t)}) & \text{if match} \end{cases} \quad (8)$$

$$w_r^{(t+1)}(m_r(i,j,l)) = \begin{cases} w_r^{(t)}(m_r(i,j,l)) - \alpha & \text{if no match} \\ 1 & \text{if match} \end{cases} \quad (9)$$

where the forgetting constant, α, is a predefined coefficient that belongs to the interval $[0, 1]$.

3.5 Updating Regions Of Interest

Once the targets have been found in the current frame t, each new $ROI_r^{(t+1)}$ is obtained. First, the level 0 of each new region of interest is computed. $ROI_r^{t+1}(0)$ is made of the pixels of the next frame $p^{(t+1)}(i,j,l)$ which are included in the bounding box of $T_r^{(t)}(0)$ plus the pixels included in an extra border ϵ of the bounding box. This extra border ensures that the target in the next frame will be placed in the new ROI. This step is performed at the end of the tracking process t. Second, at the beginning of the tracking process $t + 1$, the new regions of interest are oversegmented as it has been previously explained in subsection 3.1.

4 Results

The above described attentional scheme has been examined through experiments which include humans and objects in the scene. Fig. 6.a shows a sample image sequence seen by a stationary binocular camera head. Every 10th frame is shown. All salient regions are marked by black and white bounding boxes in the input frames. It must be noted that the activity follows the objects closely, mainly because the tracker works with the segmented input image instead of working with the saliency image. This approach has two main advantages: i) the regions of the segmented left image are more stable across time than the saliency maps regions, and ii) the regions of the segmented image represent real objects closer than saliency map regions. Furthermore, the tracking algorithm prevents the related object templates from being corrupted by occlusions. Backer and Mertsching [6] propose to

solve the occlusion problem with the inclusion of depth information. However, depth estimation is normally corrupted by noise and is often coarsely calculated in order to bound the computational complexity. In our approach, the tracker is capable of handling scale changes, object deformations, partial occlusions and changes of illumination. Fig. 6.b presents the saliency maps after inhibiting the regions which have been tracked in each frame. This inhibition avoids that the region extraction process extracts regions that have been already extracted in previous frames. In frame 1, the yellow box and the red extinguisher have been detected. The yellow box is tracked over the whole sequence because its saliency remains high. However, the saliency of the extinguisher goes down between frames 21 and 30 and therefore it is not tracked from frame 30 to the end of the sequence. In frame 11, a hand with a green cone is detected in the image. In frame 51, a red box is introduced in the scene. This box is not detected until frame 91, when it becomes located nearer to the cameras than the other objects. In frame 81, an occlusion of the green cone is correctly handled by the tracking algorithm, which is capable to recover the object before frame 91. It can also be observed how the mechanism follows appearance and view point changes of the salient objects.

The proposed method runs at 5 frames per second with 128x128 24-bit colour images, being faster than Backer's proposal [6] which is reported to take 30 seconds to process one frame. Beobot [4] runs a saliency mechanism at 30 frames per second with 160x120 images but, while we use a 850 MHz PC, Beobot uses two 1.26 GHz dual-CPU computer boards. Besides, Beobot does not include depth or movement information of the objects in its attentional mechanism.

References

[1] Eriksen, C., Yenh, Y.: Allocation of attention in the visual field. Journal of Experimental Psychology: Human Perception and Performance **11** (1985) 583–597

[2] Koch, C., Ullman, S.: Shifts selective visual attention: Towards the underlying neural circuitry. Human Neurobiology **4** (1985) 219–227

[3] Milanese, R.: Detecting salient regions in an image: from biological evidence to computer implementation. PhD Thesis, Univ. of Geneva (1993)

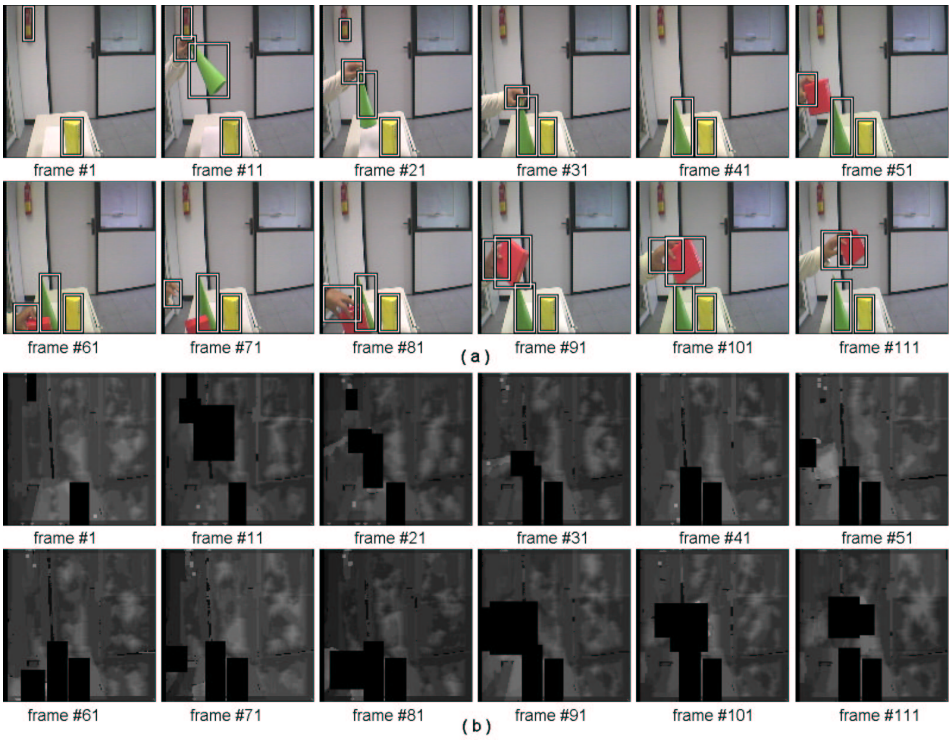

Figure 6: Example of selected targets: a) left input images; and b) saliency map associated to a)

[4] Itti, L.: Real-time high-performance attention focusing in outdoors color video streams. Proc. SPIE Human Vision and Electronic Imaging (HVEI'02) (2002) 235–243

[5] Treisman, A., Gelade, G.: A feature integration theory of attention. Cognitive Psychology **12** (1980) 97–136

[6] Backer, G., Mertsching, B.: Two selection stages provide efficient object-based attentional control for dynamic vision. Proc. Int. Workshop Attention and Performance in Computer Vision (2003) 9–16

[7] Maki, A., Nordlund, P., Eklundh, J.: Attentional scene segmentation: integrating depth and motion. Computer Vision and Image Understanding **78** (2000) 351–373

[8] Terrillon, J., Akamatsu, S.: Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. Proc. 12th Conf. on Vision Interface **2** (1999) 180–187

[9] Marfil, R., Rodriguez, J., Bandera, A., Sandoval, F.: Bounded irregular pyramid: a new structure for colour image segmentation. Pattern Recognition **37** (2004) 623–626

[10] Marfil, R., Rodriguez, J., Bandera, A., Sandoval, F.: Real-time template-based tracking of non-rigid objects using bounded irregular pyramids. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems **1** (2004) 301–306

[11] Sun, C.: Fast stereo matching using rectangular subregioning and 3d maximum-surface techniques. Int. Journal of Computer Vision **47** (2002) 99–117

[12] Itti, L., Koch, C.: Feature combination strategies for saliency-based visual attention systems. Journal of Electronic Imaging **10** (2001) 161–169