A Vision System for Symbolic Interpretation of Dynamic Scenes using ARSOM

Antonio Chella, Donatella Guarino, Ignazio Infantino, Roberto Pirrone

Dipartimento di Ingegneria Automatica e Informatica - University of Palermo

and CERE-CNR - Palermo

Viale delle Scienze, I-90128 Palermo, Italy

{chella,pirrone}@unipa.it, dony@cere.pa.cnr.it, infantino@csai.unipa.it

Abstract

We describe an artificial high-level vision system for the symbolic interpretation of data coming from a video camera that acquires the image sequences of moving scenes. The system is based on ARSOM Neural Networks that learn to generate the perception grounded predicates obtained by image sequences. The ARSOM Neural Networks also provide a 3D estimation of the movements of the relevant objects in the scene. The vision systems has been employed in two scenarios: the monitoring of a robot arm suitable for space operations, and the surveillance of a EDP center.

1 INTRODUCTION

We describe an artificial high-level vision agent for the symbolic interpretation of data coming from a video camera that acquires image sequences of moving objects and persons. The agent generates the perception grounded predicates that suitably describe the dynamic scenes.

The agent integrates a *perception component* which is based on robust techniques of computer vision, with a *scene description component* based on ARSOM Neural Networks[Kohonen, 1997] [Lampinen and Oja, 1989] that generate the symbols describing the dynamic scene. The results of these components are the input of the *visualization component* that presents the data results by an advanced user interface.

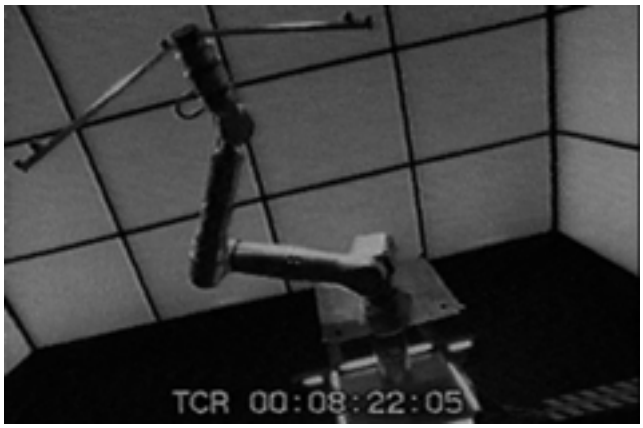

Figure 1: The SPIDER scenario.

Figure 2: The CUC scenario.

The vision agent is aimed at advancing the state of art in the field of robotics by introducing and integrating different AI techniques that offer a unique opportunity for providing effective greater degrees of autonomy for robotic systems [Chella et al., 1997] [Chella et al., 1998].

The vision agent has been employed in two scenarios of interest:

SPIDER In this scenario, images come from a video camera that acquires the movements of the SPIDER robot arm [Di Pippo et al., 1998] [Didot et al., 1998] [Mugnuolo et al., 1998] (built by the Italian Space Agency for space applications) during its operations. The agent generates the perception grounded predicates obtained by image sequences thus allowing the scientist user of SPIDER to receive meaningful feedback of his operations on the arm during a scientific experiment (Fig. 1).

CUC In this scenario, images come from a video camera posed at the entrance of the EDP Center (CUC) of the University of Palermo. The agent generates the description of the postures of a person at the entrance of the center, for generating attention degrees of the surveillance persons (Fig. 2).

2 THE PERCEPTION COMPONENT

The perception component of the proposed system processes the image data coming from a video camera that acquires the images of the moving scene. The main task of this component in both scenarios is to find the interesting points in the dynamic scene along with their motion.

In particular, in the SPIDER scenario the interesting points are the joint positions of the arm. It should be noted that this estimation, which is solely generated by the visual data, may be useful for fault identifications of the position sensors placed on the joints of the arm.

Similarly, in the CUC scenario, the interesting points are the characteristic points describing the posture of the persons at the entrance of the EDP center.

The images acquired by the camera are processed to extract the contours of the object of interest by a suitable algorithm based on *snakes* [Blake and Isard, 1998]. A snake is a deformable curve that moves in the image under the influence of forces related to the local distribution of the gray levels. When the snake reaches an object contour, it is adapted to its shape.

Formally, a snake as an open or closed contour is described in a parametric form by:

$$v(s) = (x(s), y(s)) \tag{1}$$

where $x(s)$ and $y(s)$ are the coordinates along the shape contour and s is the normalized arc length:

$$s \in [0, 1] \tag{2}$$

In the SPIDER scenario, the adopted snake model is based on circles and squares to better extract the arm components; in the CUC scenario the adopted snake model is based on a closed contour adapting to the person shape.

The snake model defines the snake energy of a contour E_{snake}, to be:

$$E_{snake}(v(s)) = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s))ds \tag{3}$$

The energy integral is a functional since its variable s is a function (the shape contour). The internal energy E_{int} is formed from a Tikhonov stabilizer and is defined by:

$$E_{int}(vvs)) = a(s) \left| \frac{dv(s)^2}{ds^2} \right| + b(s) \left| \frac{dv(s)^2}{ds^2} \right|^2 \tag{4}$$

where $|.|$ is the Euclidean norm.

The first order continuity term, weighted by $a(s)$, let the contours behave elastically, whilst the second order curvature term, weighted by $b(s)$, let it be resistant to bending. For example, setting $b(s) = 0$ at point s, allows the snake to become second-order discontinuous at point and to generate a corner.

The image functional determines the features which will have a low image energy and hence the features that attract the contours. In general, this functional is made up by three terms:

$$E_{image} = w_{line}T_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{5}$$

where w denotes a weighting constant. The w and E corresponds to lines, edges and termination, respectively.

The snake model adopted in our scenarios presents only the edge functional which attracts the snake to points with an high edge gradient:

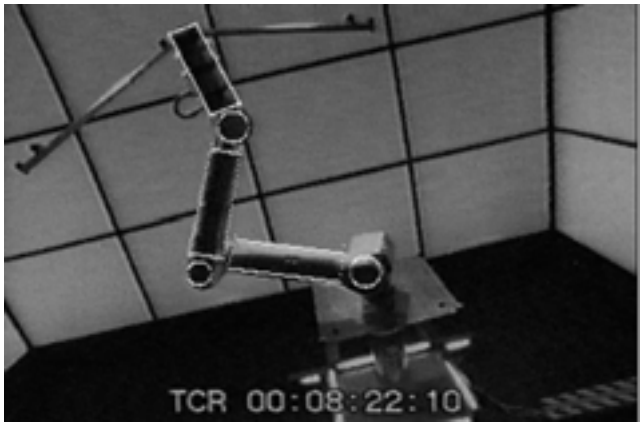

Figure 3: Contours extracted in the SPIDER scenario.

Figure 4: Contours extracted in the CUC scenario.

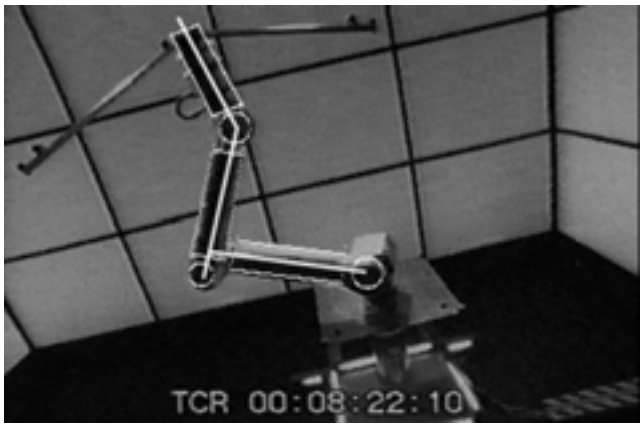

Figure 5: The skeleton extracted in the SPIDER scenario.

Figure 6: The skeleton extracted in the CUC scenario.

$$E_{image} = E_{edge} = -(G_\sigma * \nabla^2 I(x,y))^2 \tag{6}$$

This is the image functional proposed by Kass, Witkin and Terzoupolos [Kass et al., 1987]. It is a scale based edge operator that increases the locus of attraction of energy minimum. G_σ is a Gaussian of standard deviation sigma which controls the smoothing process prior to edge operator. Minima of E_{edge} lies on zero-crossing of $G_\sigma * \nabla^2 I(x,y)$ which defines the edges according to the theory of Marr [Marr, 1982].

The implemented snake allows to extract the interesting parts of the scenarios in a simple way and in short time. Fig. 3 shows the results in the SPIDER scenario and Fig. 4 shows the results in the CUC scenario.

After this step, we employ the well known skeletonizing algorithm of Zhang and Suen [Zhang and Suen, 1984] to extract the skeletons of the areas so found. Fig. 5 shows the skeleton in the SPIDER scenario and Fig. 6 shows the skeleton in the CUC scenario.

3 THE SCENE DESCRIPTION COMPONENT

From the extracted skeletons it is immediate to estimate the position of the interesting points previously described, characterizing the posture of the SPIDER arm or the posture of a person in the CUC scenario.

Let us consider a generic interesting point i of the scene in a scenario at time t; the point is characterized by its 3D coordinates:

$$x_i(t), y_i(t), z_i(t) \tag{7}$$

A generic posture at time t of the robot arm or of a person is characterized by the vector $\mathbf{x}(t)$ which individuates the m interesting points describing the posture itself:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t), y_1(t), z_1(t) \\ x_2(t), y_2(t), z_2(t) \\ \vdots \\ x_n(t), y_n(t), z_m(t) \end{bmatrix} \tag{8}$$

The snake information allows us to estimate only the first and the second coordinates of each point, i.e., their projection in the image plane:

$$\mathbf{x}'(t) = \begin{bmatrix} x_1(t), y_1(t), . \\ x_2(t), y_2(t), . \\ \vdots \\ x_n(t), y_m(t), . \end{bmatrix} \tag{9}$$

The scene description component receives as input the vector $\mathbf{x}'$ from the perception component and it generates a symbolic description of the posture. This component is based on the ARSOM neural network, a self-organizing neural network with a suitable explicit representation of time sequences [Kohonen, 1997] [Lampinen and Oja, 1989].

Each unit of the ARSOM is an autoregressive (AR) filter, able to classify and recognize variable inputs. Therefore, each unit characterizes a sequence of movements of the posture points. The map auto-organizes itself during an unsupervised learning phase, as a standard SOM map.

Let us consider a generic movement of the robot arm or of a person. The movement is characterized by a sequence of n posture points:

$$\mathbf{x}(t), \mathbf{x}(t-1), \ldots, \mathbf{x}(t-(n-1)) \tag{10}$$

The AR model associated with this movement is:

$$\mathbf{x}(t+1) = \mathbf{A}_0\mathbf{x}(t) \quad + \quad \mathbf{A}_1\mathbf{x}(t-1) + \ldots \tag{11}$$
$$\ldots \quad + \quad \mathbf{A}_{n-1}\mathbf{x}(t-(n-1)) + \mathbf{e}(t)$$

The order of this AR model is n, the $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{n-1}$ matrices are the weights of the model, and $\mathbf{e}(t)$ is the error matrix.

Let us denote with $\mathbf{B}$ the global matrix related to the weight matrices:

$$\mathbf{B} = [\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{n-1}]^T \tag{12}$$

and with $\mathbf{X}(t)$ the global matrix related to the time evolution of the posture points:

$$\mathbf{X}(t) = [\mathbf{x}(t), \mathbf{x}(t-1), \ldots, \mathbf{x}(t-(n-1))]^T \tag{13}$$

We may write Eq. (11) in a more compact form:

$$\mathbf{x}(t+1) = \mathbf{X}^T(t)\mathbf{B} + \mathbf{e}(t) \tag{14}$$

The optimal weights matrices are found by minimizing the error matrix $\mathbf{e}(t)$. We have adopted the *LMS* iterative method, that is:

$$\mathbf{B}_{new} = \mathbf{B}_{old} + h_{ci}\mathbf{e}(t)\mathbf{X}(t) \tag{15}$$

where h_{ci} is the neighborhood kernel:

$$h_{ci} = \begin{cases} 1/2r^2 & \text{if} \quad i \in N_c \\ 0 & \text{if} \quad i \notin N_c \end{cases} \tag{16}$$

In this equation, r is a suitable parameter and N_c is the width of the learning window.

The neural network, after a careful training phase, is able to classify the temporal sequences of movements of the interesting points into meaningful prototypical predicates.

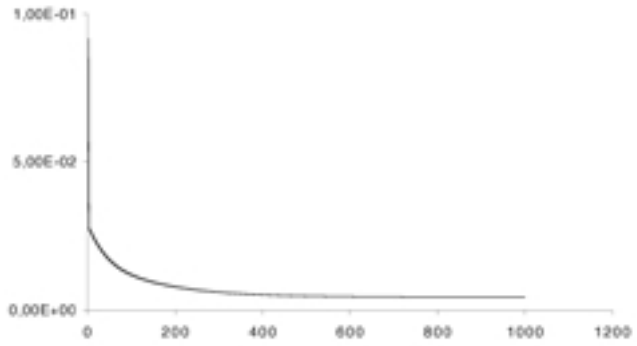

Figure 7: Error vs. learning epochs of the ARSOM network.

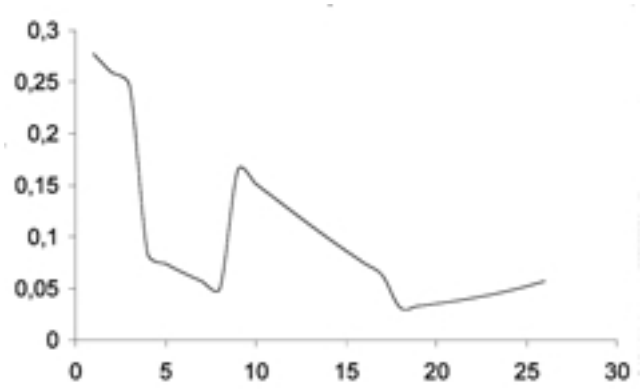

Figure 8: Prediction error of the ARSOM network.

Currently, we have two similar ARSOM neural networks: one for the SPIDER scenario and the other for the CUC scenario. We are experimenting the possibility to have one networks for both scenarios.

Fig. 7 shows the diagram of the error of the neural network during the training phase. It should be noted that, after a few hundred learning steps, the error of the network is near zero value. The figure is related with the SPIDER scenario; a similar behavior occurs in the CUC scenario.

When the estimation of the coordinates of the interesting point in the image plane are presented to the network:

$$\mathbf{x}'(t), \mathbf{x}'(t-1), \ldots, \mathbf{x}'(t-(n-1)), \tag{17}$$

the network is able to predict the full vector $\mathbf{x}(t+1)$, i.e., the vector with all the three coordinates of the posture.

Fig. 8 shows the prediction error of the network during its operations in the SPIDER scenario. It should be noted that the error, while is variable, it maintains in a reasonable limit. Similar behaviors have been observed in the CUC scenario.

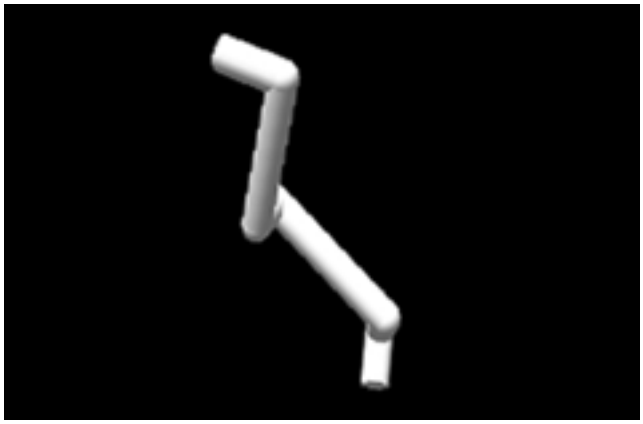

Figure 9: 3D recovery of the SPIDER scenario.

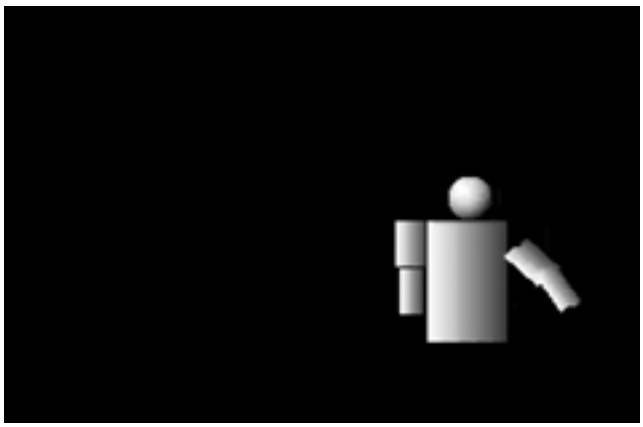

Figure 10: 3D recovery of the CUC scenario.

Figs. 9 and 10 show the recovered 3D situations, respectively in the SPIDER and CUC scenarios, as predicted by the ARSOM neural networks.

The network is also able to perform a classification of the global arm movement and to present as output a symbolic predicate describing the movement itself.

Examples of the learned predicates describing the operations of the arm in the SPIDER scenario are: *Stretching_up*, *Stretching_down*, *Seizing*, *Grasping*. Examples of the learned predicates in the CUC scenario are: *Entering*, *Exiting*, *Opening*, *Closing*, *Looking_inside*, *Staying*.

The neural network approach presents the main advantage that it avoids an explicit description of the discrimination functions for the arm operations, as this function is learned during the training phase.

Furthermore, the neural network is robust with respect to the noise, as it is able to correctly classify the arm operations also when the movements estimations of some links are missing or corrupted.

In the operation tests performed in the SPIDER scenario, the network has been able to perform the 100% success on the classification task. To analyze the operation of the network, tests are performed on the recognition task when some links information is missed. Tab. 1 reports the obtained results. It should be noted that in the worst case, when the two links 1 and 3 are missing, the network is able to perform 51% of success recognition.

Miss. link	% Rec.
0	100
1	75
2	74
3	62
1,3	51

Table 1: Recognition % vs missing link in the SPIDER scenario.

Also the performances of the system in the CUC scenario are good. Up to now, we have obtained about 94% of recognition success on the classification task. It should be noted that in this case, we have chosen to take into account only simple actions, as previously described. Currently, we are generalizing the system on a larger set of more rich and realistic situations.

4 THE VISUALIZATION COMPONENT

The scene description component of the system receives as input the data coming from the perception component and of the scene description component, and it generates a graphic 3D representation of the scene. The visualization component provides also the graphic interface for the whole agent. In the following we will describe the interface for the SPIDER scenario; similar consideration hold for the CUC scenario.

Fig. 11 shows the results of the visualization component of the system for the SPIDER scenario. The scientist user of the system may view the arm operations from different point of views and he may navigate in the reconstructed environment.

He may also supervise and intervene in all the processing steps occurring in the agent itself: e.g., he may change the parameters of the perception component modules or he may tune the learning phase of the neural network in the scene description component.

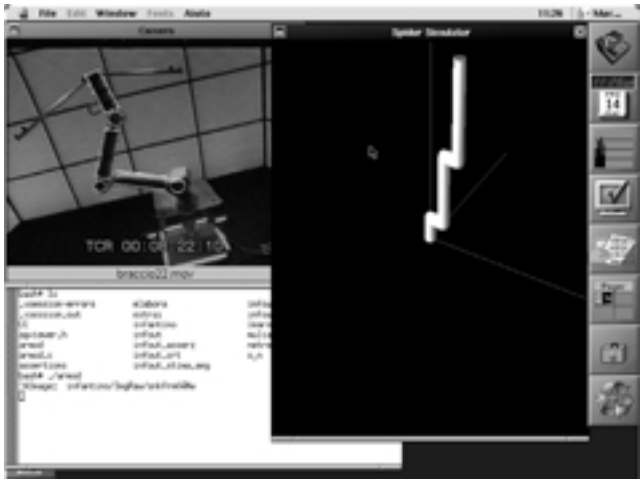

Figure 11: The user interface of the system.

The interface of the system presents several windows in order to provide the user scientist with a full control of the system.

The *camera* window shows the output image sequences of the video camera acquiring the real robot arm operations along with superimposition of the snake representing the output of the contour extraction module.

The 3D window shows the images representing the 3D reconstruction of the arm during its operations, and the *description* window shows the symbolic descriptions generated by the scene description component in terms of symbolic predicates.

A simple user interface based on buttons allows the scientist to modify the inner parameters of the agent in order to tailor the agent processing steps.

The graphical interface has been realized by using the OpenGL and the GLUT library [Kilgard, 1996] [Neider et al., 1996].

5 CONCLUSION

The described first results of the research demonstrated how the implemented artificial high-level vision agent may be an effective tool for monitoring operations. In the case of the SPIDER scenario, the user scientist of the arm can monitor his own operations by providing high-level feedback descriptions of the arm movements during the scientific experiments. In the case of the CUC scenario, the surveillance persons can be alerted of possible dangerous situations near the EDP center that require special attention.

The system is fully general and it may be employed in all the fields in which the interactive autonomy of the

space robotic systems is a mandatory requirement. The system will also give a valuable contribution to the use of the expensive and state of the art equipment related to space robotics and to surveillance robotics. Of great importance are the possible industrial application of the described system. It could be employed in all the applications that require high automatic tasks in interactive autonomy, as the submarine robots and autonomous systems acting in nuclear plants.

6 ACKNOWLEDGEMENTS

This work has been partially supported by ASI project “Un Sistema Intelligente per la Supervisione di Robot Autonomi nello Spazio” and by MURST project “Progetto Cofinanziato CERTAMEN”.

References

[Blake and Isard, 1998] Blake, A. and Isard, M. (1998). *Active Contours*. Springer-Verlag, Berlin.

[Chella et al., 1997] Chella, A., Frixione, M., and Gaglio, S. (1997). A cognitive architecture for artificial vision. *Artif. Intell.*, 89:73–111.

[Chella et al., 1998] Chella, A., Frixione, M., and Gaglio, S. (1998). An architecture for autonomous agents exploiting conceptual representations. *Robotics and Autonomous Systems*, 25(3-4):231–240.

[Di Pippo et al., 1998] Di Pippo, S., Colombina, G., Boumans, R., and Putz, P. (1998). Future potential applications of robotics for the international space station. *Robotics and Autonom. Systems*, 23(1-2):37–43.

[Didot et al., 1998] Didot, F., Dettmann, J., Losito, S., Torfs, D., and Colombina, G. (1998). JERICO. *Robotics and Autonom. Systems*, 23(1-2):29–36.

[Kass et al., 1987] Kass, M., Witkin, A., and Terzoupolos, D. (1987). Snakes: active contour models. In *Proc. of First Intern. Conf. on Computer Vision*, pages 259–268. Springer-Verlag.

[Kilgard, 1996] Kilgard, M. (1996). *OpenGL programming for the X Window system*. Addison-Wesley, Reading, MA.

[Kohonen, 1997] Kohonen, T. (1997). *Self-Organizing Maps, II ed.* Springer-Verlag, Berlin.

[Lampinen and Oja, 1989] Lampinen, J. and Oja, E. (1989). Self-organizing maps for spatial and temporal AR models. In *Proc. of the 6th Scandinavian Conference on Image Analysis*, pages 120–127, Oulu, Finland.

[Marr, 1982] Marr, D. (1982). *Vision.* W.H. Freeman and Co., New York.

[Mugnuolo et al., 1998] Mugnuolo, R., Di Pippo, S., Magnani, P., and Re, E. (1998). The SPIDER manipulation system (SMS). The italian approach to space automation. *Robotics and Autonom. Systems*, 23(1-2):79–88.

[Neider et al., 1996] Neider, J., Davis, T., and Woo, M. (1996). *OpenGL programming guide*. Addison-Wesley, Reading, MA.

[Zhang and Suen, 1984] Zhang, T. and Suen, C. (1984). A fast parallel algorithm for thinning digital patterns. *Comm. ACM*, 27(3):236–239.