

## AN ARTIFICIAL HIGH-LEVEL VISION AGENT FOR THE INTERPRETATION OF THE OPERATIONS OF A ROBOTIC ARM

Antonio Chella, Salvatore Gaglio, M. Donatella Guarino, Ignazio Infantino\*

Dipartimento di Ingegneria Elettrica, Università di Palermo and  
Centro di Studi sulle Reti di Elaboratori, CNR  
Viale delle Scienze, I-90128 Palermo, Italy,  
phone: +39.091.238111, fax: +39.091.6529124, e-mail: chella@unipa.it

\* Dipartimento di Ingegneria Elettrica, Università di Palermo  
Viale delle Scienze, I-90128 Palermo, Italy,  
phone: +39.091.6566273, fax: +39.091.488452

### ABSTRACT

We describe an artificial high-level vision agent for the symbolic and graphic interpretation of data coming from a video camera that acquires the image sequences of the SPIDER robot arm of the EUROPA system during its operations. The agent generates the perception grounded predicates obtained by image sequences and it provides a 3D estimation of the arm movements, thus allowing the scientist user of SPIDER to receive meaningful feedback of his operations on the arm during a scientific experiment.

### 1. INTRODUCTION

We describe an artificial high-level vision agent for the interpretation of data coming from a video camera that acquires the image sequences of the SPIDER robot arm of the EUROPA system [7,11,15] during its operations (see Fig. 1).

The described software module is related to the interpretation of sensory data in the framework of an ASI project aiming at the application of AI techniques to the design and realization of an effective and flexible system for the supervision of the SPIDER arm. The arm will work on board of the International Space Station (ISS) [8].

The framework project is an Italian three years research project [1,6,17] sponsored by the Italian Space Agency (ASI) involving AI researchers from the Universities of Rome, Turin, Genoa, Palermo, Parma, from the IP-CNR of Rome and from the IRST of Trento.

The main aim of the vision agent is the advancement of the state of art in the field of artificial vision for spatial robotics by introducing and integrating artificial vision

techniques that offer a unique opportunity for providing the SPIDER arm operations with effective greater degrees of autonomy [2,3].

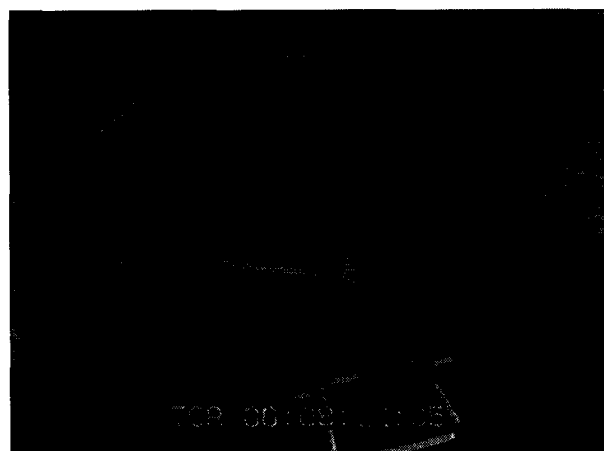


Fig.1. The SPIDER arm of the EUROPA system.

The valuable capabilities of the vision agent are:

- to individuate and segment the SPIDER arm also in contrasted and irregular backgrounds;
- to perform a 3D estimation of the position of the arm by camera images;
- to interpret complex movements of the arm acquired by a camera in terms of symbolic descriptions.

The implemented computer vision agent is based on three main components:

- (i) the perception component;
- (ii) the scene description component;
- (iii) the visualization component.

In the following, Sect. 2 describes the perception component of the system, i.e., how the system perform the low-level image processing in order to individuate and segment the SPIDER arm. Sect. 3 describes the scene description component, in which the acquired image is interpreted both in terms of 3D parameters and in terms of generated symbolic assertions. Sect. 4 describes the visualization component, in which the user may interact with the agent components, and Sect. 5 describes the implementation details of the system. Finally, Sect. 6 outlines some conclusions and future developments.

## 2. THE PERCEPTION COMPONENT

The perception component of the agent processes the image data coming from a video camera that acquires the operations of the SPIDER arm.

The main task of this component is to estimate the positions of the arm in the acquired image. It should be noted that the estimation, which is generated solely by the visual data, may be useful also for fault identifications of the position sensors placed on the joints of the arm.

The images acquired by the camera are processed by the contour module that extracts the arm contours by a suitable algorithm based on *snakes* [5,9,12].

The 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. In this way it is possible to extract the object shape of the image view.

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

$$v(s) = (x(s), y(s))$$

where  $x(s)$ ,  $y(s)$  are x,y co-ordinates along the contour and  $s$  is the normalized arc length:

$$s \in [0,1]$$

The snake model adopted is based on circles and squares, in order to better extract the arm components (see Fig. 2). The snake model defines the energy of a contour, named the snake energy,  $E_{snake}$  to be:

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

The energy integral is a functional since its independent variable is a function.

The internal energy,  $E_{int}$  is formed from a Tikhonov stabilizer and is defined:

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

where  $||$  is the Euclidean norm.

The first order continuity term, weighted by  $a(s)$ , makes the contours behave elastically, whilst the second order curvature term, weighted by  $b(s)$ , makes it resistant to bending. For example, setting  $b(s) = 0$  at point  $s$ , allows the snake to become second-order discontinuous at point and develop 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 made up of three terms:

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

where  $w$  denotes a weighting constant. Each of  $w$  and  $E$  correspond to lines, edges and termination respectively.

The snake used in this framework has only edge functional which attracts the snake to point at high gradient:

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

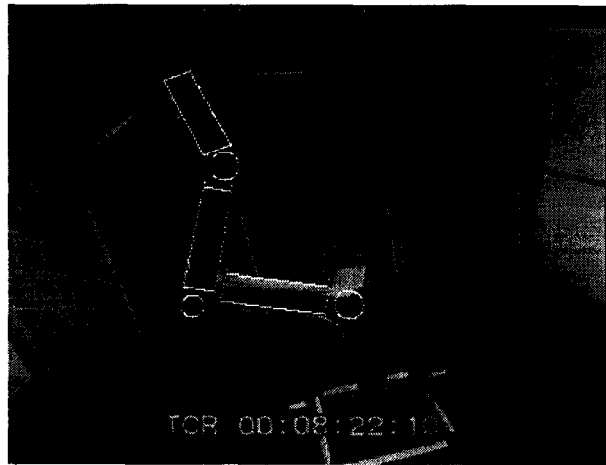


Fig. 2. Contour module extraction by the snake technique.

This is the image functional proposed by Kass [12]. It is a scale based edge operator that increases the locus of

attraction of energy minimum.  $G_\sigma$  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 edges in Marr-Hildreth [9,10] theory.

Scale space filtering is employed, which allows the snake to come into equilibrium on a heavily filtered image, and then the level of filtering is reduced, increasing the locus of attraction of a minimum.

The implemented snake allows to extract the arm shape in a simple way and in short time. Fig. 2 shows the results of the contour module.

From the extracted arm snake it is possible to estimate the position of the links of the arm in the image plane, i.e., without the depth information, which is recovered by the *scene description component*.

Let us consider a generic link  $i$  of the arm at time  $t$ ; the link is characterized by its 3D coordinates:

$$(x_i(t), y_i(t), z_i(t))$$

A generic posture of the SPIDER arm at time  $t$  is characterized by the vector  $\mathbf{x}(t)$  which individuates the seven links of the arm:

$$\mathbf{x}(t) = \begin{bmatrix} (x_1(t), y_1(t), z_1(t)) \\ (x_2(t), y_2(t), z_2(t)) \\ \vdots \\ (x_7(t), y_7(t), z_7(t)) \end{bmatrix}$$

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

$$\mathbf{x}'(t) = \begin{bmatrix} (x_1(t), y_1(t), \dots) \\ (x_2(t), y_2(t), \dots) \\ \vdots \\ (x_7(t), y_7(t), \dots) \end{bmatrix}$$

### 3. THE SCENE DESCRIPTION COMPONENT

The scene description component receives as input the data coming from the perception component and it generates a symbolic description of the arm operations. This component is based on a self-organizing neural network with a suitable explicit representation of time sequences [4,14].

Each unit of the ARSOM is an autoregressive (AR) filter, able to classify and recognize variable inputs. The map auto-organizes during an unsupervised learning phase. Each unit of the map characterizes a sequence of movements of the SPIDER arm.

Let us consider a generic movement associated with the SPIDER arm. The movement is characterized by a sequence of  $n$  postures:

$$\mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-(n-1))$$

The AR model associated with this movement is:

$$\mathbf{x}(t+1) = \mathbf{A}_0 \mathbf{x}(t) + \mathbf{A}_1 \mathbf{x}(t-1) + \dots + \mathbf{A}_{n-1} \mathbf{x}(t-(n-1)) + \mathbf{e}(t)$$

The order of the model is  $n$ , the  $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{n-1}$  matrices are the weights of the model, and  $\mathbf{e}(t)$  is the error matrix. Let us denote  $\mathbf{B}$  the global matrix related to the weight matrices:

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

and with  $\mathbf{X}(t)$  the global matrix related to the postures. We may write the previous equation in a more compact form:

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

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

$$\mathbf{B}' = \mathbf{B} + h_{ci} \mathbf{e}(t) \mathbf{X}(t)$$

where  $h_{ci}$  is the neighborhood kernel:

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

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

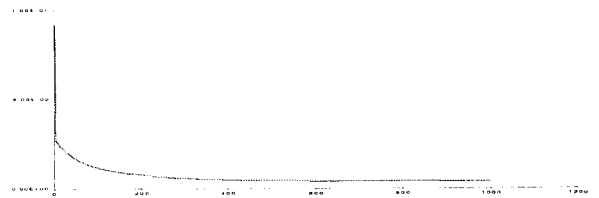


Fig.3. Error diagram vs training epochs.

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

Fig. 3 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.

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

$$\mathbf{x}'(t), \mathbf{x}'(t-1), \dots, \mathbf{x}'(t-(n-1))$$

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 of the arm links.

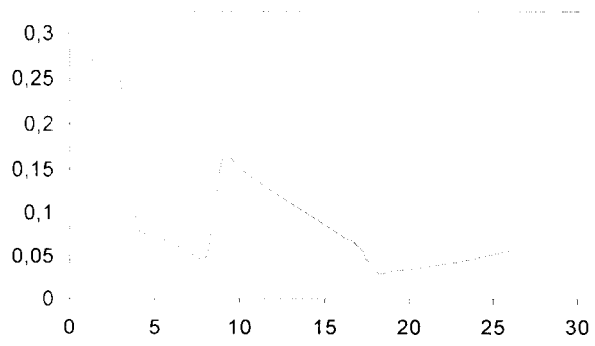


Fig. 4. Prediction error of the network.

Fig. 4 shows the prediction error of the network during its operations. It should be noted that the error, while is variable, it maintains in a reasonable limit.

Furthermore, 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 are: *Stretching\_up*, *Stretching\_down*, *Seizing*, *Grasping*.

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, 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. Table 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.

Missing links	Recognition %
0	100
1	75
2	74
3	62
1,3	51

Table 1. Recognition % with respect to the missing links.

#### 4. THE VISUALIZATION COMPONENT

The scene description component receives as input the data coming from the perception component, in the same way of the scene description component, and it generates a graphic 3D representation of the arm movements.

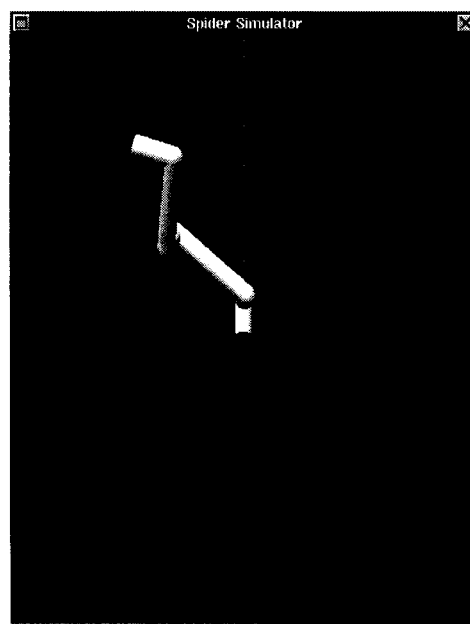


Fig. 5. The visualization of the robot arm.

This component provides an immediate, visual feedback of the arm operations that complements the symbolic description coming from the previous component. The visualization component provides also the graphic interface for the whole agent.

Fig. 5 shows the results of the visualization component of the agent. The scientist user of the agent 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.

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.

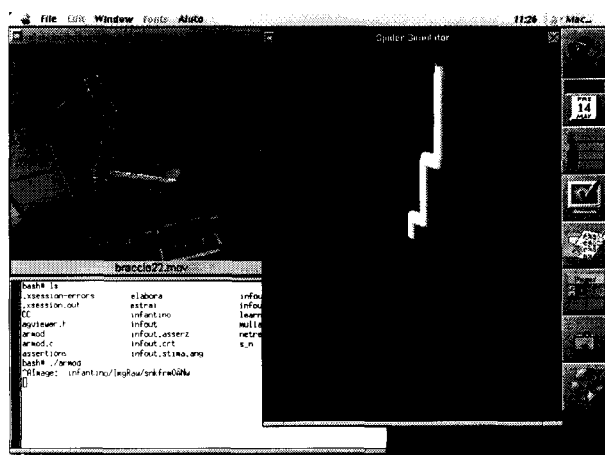


Fig. 6. The visualization environment of the vision agent.

## 5. IMPLEMENTATION DETAILS

The described artificial vision agent has been implemented in C under the Linux Operating System. The whole system currently runs on a Pentium II 400MHz and on a Apple iMac 266MHz.

The graphical interface has been realized by using the OpenGL [16] and the GLUT library [13].

## 6. CONCLUSIONS

The research demonstrated how the implemented artificial high-level vision agent may be an effective tool that helps the user scientist of the SPIDER arm to monitor his own operations by providing high-level feedback descriptions of the arm movements during the scientific experiments.

The described activity is aimed to the realization of a research product, which is innovative and complementary to the research activities of the ASI. It provides an effective scientific support with important effects of the development of new technologies within the ASI programs related to the utilization of the International Space Station.

The product of the described activity may be employed in all the fields in which the interactive autonomy of the space robotic systems is a mandatory requirement, as the exploration of the Moon and Mars.

The product of the present project will also give a valuable contribution to the use of the expensive and state of the art equipment related to space robotics owned by the ASI.

Of great importance are the possible industrial application of the product of the project. The system software 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.

## ACKNOWLEDGEMENTS

Authors would like to thank Dr. V. Venturini of Tecnospazio SpA and Dr. R. Mugnuolo of ASI for their kind collaboration on the described research.

This work has been supported by the ASI project "An Intelligent System for Supervision of Autonomous Robots in Space".

## REFERENCES

- [1] A. Cesta, M. Daniele, E. Giunchiglia, M. Piaggio, P. Riccucci, M. Schaerf, P. Traverso: JERRY: a system for the automatic generation and execution of plans for robotic devices -- the case study of the SPIDER arm, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).
- [2] A. Chella, M. Frixione, S. Gaglio: A Cognitive Architecture for Artificial Vision, Artificial Intelligence, 89, No. 1-2, pp. 73-111, 1997.
- [3] A. Chella, M. Frixione, S. Gaglio: An Architecture for Autonomous Agents Exploiting Conceptual Representations, Robotics and Autonomous Systems, 25, No. 3-4, pp. 231-240, 1998.
- [4] A. Chella, S. Gaglio, V. Mulia, G. Sajeve: An ASSOM Neural Network for Description of Autonomous Agent Actions, in: W. Gerstner, A. Germond, M. Hasler, J.D. Nicoud (eds.): Artificial Neural Networks - ICANN '97, pp. 799-804, Springer-Verlag, Berlin, 1997.
- [5] A. Chella, V. Di Gesù, I. Infantino, D. Intravaia, C. Valenti: A Cooperating Strategy for Object Recognition, in: R. Cipolla, D. Forsyth (eds.): Proc. of the Int. Workshop on Shape, Contour and Grouping in Computer Vision (invited paper), Lecture Notes in Computer Science, Springer-Verlag, Berlin 1998.
- [6] A. Dell'Arciprete, G. Rossi, A. Finzi, F. Pirri, M. Schaerf: A system integrating high and low level planning of complex tasks with a 3 dimensional visualizer, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).
- [7] F. Didot, J. Dettmann, S. Losito, D. Torfs, G. Colombina: JERICO, Robotics and Autonomous Systems, 23, No. 1-2, pp. 29-36, 1998.
- [8] S. Di Pippo, G. Colombina, R. Boumans, P. Putz: Future Potential Applications of Robotics for the International Space Station, Robotics and Autonomous Systems, 23, No. 1-2, pp. 37-43, 1998.
- [9] O. Faugeras: Three-Dimensional Computer Vision, MIT Press, Cambridge, MA, 1993.
- [10] B.P.K. Horn: Robot Vision. MIT Press, Cambridge, MA. 1986.
- [11] JERICO System Description Tech. Rep. JER-005-10/97 ESA, November 1997, Noordwijk, The Netherlands.
- [12] M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active Contour Models, in: Proc. of First Intern. Conf. On Computer Vision, pp. 259-268, Springer-Verlag, Berlin, 1987.
- [13] M.J. Kilgard: OpenGL Programming for the X Window System, Addison-Wesley, Reading, MA, 1996.
- [14] J. Lampinen, E. Oja: Self-Organizing maps for Spatial and Temporal AR Models, Proc. of 6<sup>th</sup> Scandinavian Conf. On Image Analysis, Oulu, Finland, pp. 120-127, 1989.
- [15] R. Mugnolo, S. Di Pippo, P.G. Magnani, E. Re: The SPIDER Manipulation System (SMS) The Italian Approach to Space Automation, Robotics and Autonomous Systems, 23, No. 1-2, pp. 79-88, 1998.
- [16] J. Neider, T. Davis, M. Woo: OpenGL Programming Guide, Addison-Wesley, Reading, MA, 1996.
- [17] L. Portinale, P. Torasso, G. Correndo: Knowledge representation and reasoning for fault identification in a space robot arm, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).