Proceedings of the 4th WSEAS/IASME Int. Conf. on System Science and Simulation in Engineering, Tenerife, Spain, December 16-18, 2005 (pp12-16)

Virtual Experiments Environment For Mobile Robots Design And Testing

EDUARD FRANTI[1], DAN TUFIS[3], MONICA DASCALU[2], ANGELA IONITA[3], SERGIU GOSCHIN[1], GHEORGHE STEFAN[4], TRAIAN BALAN[1], LUCIAN MILEA[4], TEOFIL TEODORESCU[4]

[1] National Institute for Research and Development in Microtechnologies, Erou Iancu Nicolae St. 32B, Bucharest, Romania, edif@atlas.cpe.pub.ro

[2] Centre for New Electronic Architecture, Romanian Academy, Bucharest, ROMANIA

[3] Research Institute for Artificial Intelligence, 13 Septembrie Street 13, sector 5, Bucharest, ROMANIA

[4] "Politehnica" University of Bucharest, Iuliu Maniu street 1-3, sector 6, Bucharest, ROMANIA

Abstract: - This paper refers to a virtual environment which represents the main support for experiments with mobile robots in the design and testing stage. This software environment is very useful because, compairing to the experiments with real robots, it allow the testing and evaluation of different types of interfaces and different working environments with diverse configurations. A very important facility of this interactive software environment is the fact that the designers of the robots sensors and interfaces are able to work in parallel to design test, optimize and realize different control devices for the robot.

Key-Words: - artificial intelligence, mobile robots, virtual experiments, virtual environment, simulator, synchronization

1 Introduction

The target of the robotics was and will probably remain for a long time, to produce an autonomous mobile robot.

Although remarkable progresses were made in the last 10 years, such an advanced autonomous system which can realize actions and have abilities similar to a human being, is not yet completely realized. Even so, it's realization is insistently required insistent in domains like: exploring unknown fields, intervention in dangerous environments and the control of the phenomena occurring there, medicine, military actions, entertainment etc.

In the results communicated by different researchers, we can find developments in which the autonomy is obtained for limited periods of time but the mission set is limited. Nevertheless even if they are not completely autonomous, the producing of mobile intelligent robots is insistently required by the necessity of deploying complex activities in fields as: exploring the unknown spaces (cosmic space, planetary ocean); intervention in dangerous environments (nuclear accidents, fires, natural disasters, industrial accidents, etc.) medicine (operations that involve high accuracy movements), military actions (enemy inspection, mine clearing, activities in irradiated environments, etc), domestic activities (museum visitors guiding, supermarket cleaning, entertainment, etc.).

The recent researches stage shows a tendency of merger of technics coming from distinctive fields of research as: machines teaching, automatic control theory, artificial intelligence, neurology, psychology, ethology etc. All the works of these domains are integrated in the so-called "embedded system approach to artificial intelligence" (ES) and here, the teaching is considered a very important potential component inside a general method of designing the robots control.

2 Simulators for robots testing

Most of the simulators produced until now, are making unrealistic assumptions such as: static operating environments, ideal sensorial systems, but the missions of the robots are passive (do not lead to significant modification of the geometry of the operating environment). The most usual movements were consisting on moving without collisions towards a target point (mostly immovable). In practice, the missions that has to be accomplished by the real robots are much more complex and need the projection of adequate controlling programs.

The difficulties aroused by the conception and elaboration of a software environment for simulating controlling programs for the mobile robots can be reduced by a modular conception. The simulation of the interactions between the robot and the operating environment is quite difficult because this is a dynamic process with a significant interdependence, on one hand, the robots actions can radically modify the configuration of the working environment, and on the other hand, the modifications in the working environment need modifications in the programs controlling the functioning of the robot.

An element of novelty of this simulator is the fact that, in this stage, the focusing is realized on the interaction between the agent and its operation environment and not the separate consideration of these components. The interactions between the agent and the environment are looked upon as to a dynamic coupled system in which the exit of the agent is transformed into the entering of the environment, and the exit of the last one is transformed into the entering of the agent. This simulator was made in a modular conception, each module having a very clearly set task: module generating the working environments, module for describing the robots (cinematic and dynamic modelling part), module for describing the controlling program and the module for coordinating and synchronizing the operation of the other components of the simulation environment. The simulation environment is composed from several modules each having very precise tasks:

"Operating environment" *module* – allows the user to generate, configure and modify diverse working environments which will be used for testing the robots and the controlling programs conceived for it.

"Robot_Model" *module* – allows the user to generate in the simulation environment one or more mobile robots (mobile platform with differential tractions are in view), the sensorial system is composed by a series of transducers for measuring the distance to the obstacles.

"Supervisor " *module* – is responsible with the coordination and synchronization of the activity of the other components of the simulator.

"Controlling program" *module* – each mobile robot defined in the simulated working environment, has to have a controlling program.

This controlling program has to have access to the information given by the sensorial system of the robot, to operate these signals following a preset algorithm and then elaborate a series of commands. The "Robot_Model" module takes over these commands and calculates the new position of the mobile robot. At the end of each step of the simulation, the new coordinates of the mobile robot will be taken over by the "Supervisor" module in order to update the scene.

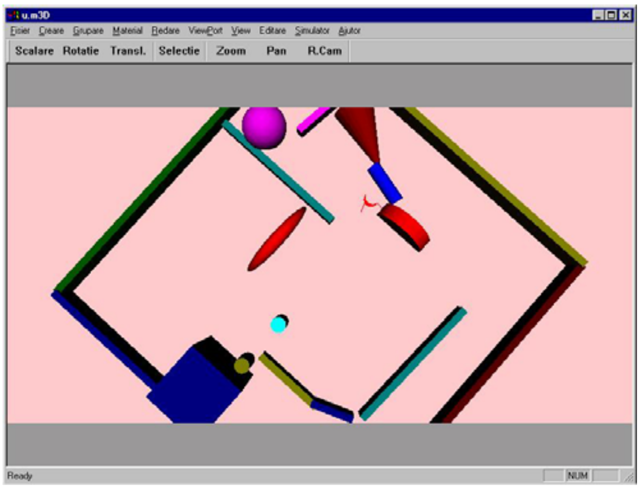

Fig.1 Global view over the environment

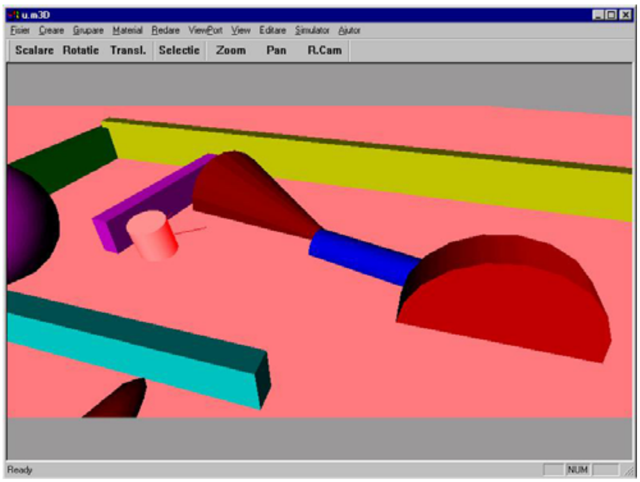

Fig.2 View from another angle

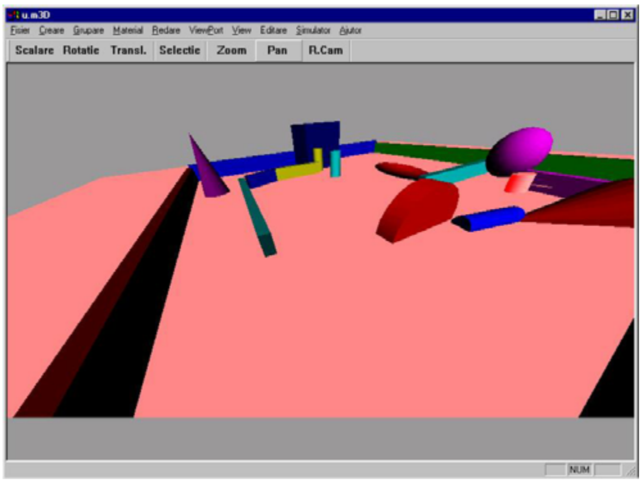

Fig.3 View from another angle

Proceedings of the 4th WSEAS/IASME Int. Conf. on System Science and Simulation in Engineering, Tenerife, Spain, December 16-18, 2005 (pp12-16)

3 The simulator Implementation

The simulator was developed under Visual C++ 6 environment and Open Gl library of graphical functions. A classification of the classes which may allow an easy as possible subsequent development was made. All the graphical primitives have as a parent class an abstract class called Prim Geom, which practically assures a mutual interface for all the objects and a possibility to work with generic lists of graphical primitives of any kind. The working at the environment module continued, and for facilitating the implementation of the robot module, some characteristics were added. The application was modularly constructed, to allow subsequent developments.

3.1 The "Operating environment" module

This module generates realistic and diverse working environments. This software module was integrally realized and was conceived to allow an easy interface with the other software modules which will finally form the complex simulating environment for the autonomous mobile robots.

This software package is remarkable by its flexibility and the very friendly graphical realization manner. It responds to a very large application area: from the simplest ones (in which the environment is manually described by the user and is characterized by a small number of static obstacles), to the most complex ones (in which the operating environment is automatically generated by an artificial evolution algorithm).

3.2 The "Robot Model" module

The major role of this software module is to allow the user to generate in the simulation environment one or more mobile robots.

In designing this module, the following requirements were in view:

-to generate mobile robots with differential traction (specifying for each robot the dimension of the motor wheels, the distance between the motor wheels, the transmission ratio between the electrical engine and the motor wheel, the maximum speed, the maximum acceleration, etc.);

-allows to equip the robot with a colour camera with the following specifications: the size of the horizontal view angle, the number of pixels on horizontal, number of pixels on vertical;

-allows to equip the robot with sensors for distance measurement, these can be with infrareds, ultrasounds or laser. For each distance measuring sensor, regardless of its nature, the size of the "visual" angle, the placement of the sensor towards the direction of the robots forwarding and the conversion table are specified;

-allows to endow the robot with an encoder for each motor wheel;

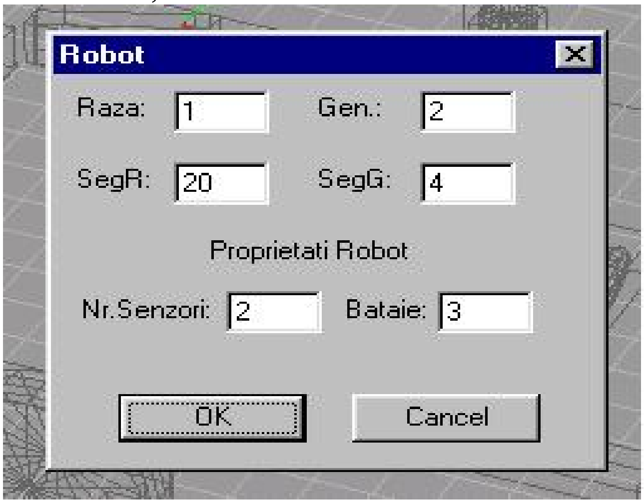

Fig.4 The dialogs box that apears when a new robot is made. Contains all the specific properties (geometrics and senzors).

-for the sensorial system to be closer to the reality, there is the possibility that every sensor, regardless of its nature, is to be affected by white noise;

-allows to specify the consume of the electrical engines and of the central command and control unit as well as the electrical energy stored in the battery;

-allows the development of the controlling program of the mobile robot in a software environment, the placement of this executable has to be specified;

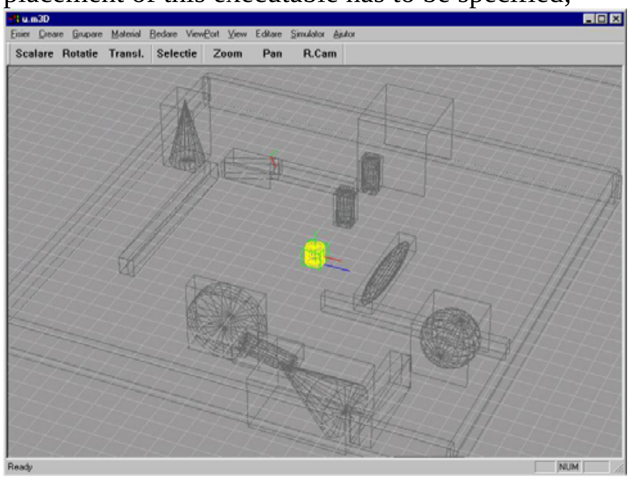

Fig.5 Global view over the edited environment: the robot with the sensors and the attached video camera can be seen.

Proceedings of the 4th WSEAS/IASME Int. Conf. on System Science and Simulation in Engineering, Tenerife, Spain, December 16-18, 2005 (pp12-16)

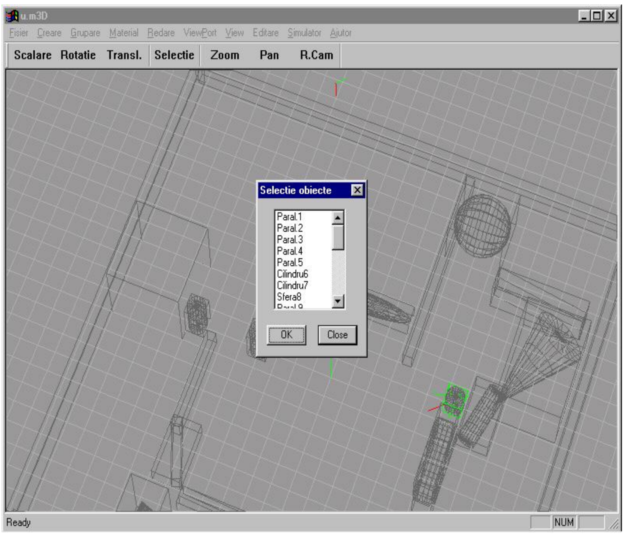

Fig.6 The dialogue window from which the objects can be selected from the environment and which can be edited eventually

3.3 The "Controlling program" module

The principle after which the control algorithm was developed was the one to assure an interface through which the information can be taken over from the robot and through which the robot is commanded, an interface easy to implement, and which does not need information regarding the way the application functions in the remnant. An abstract class ***Alg Control*** was implemented. Which has a virtual pure *control* function, which has to be implemented by all the subclasses.

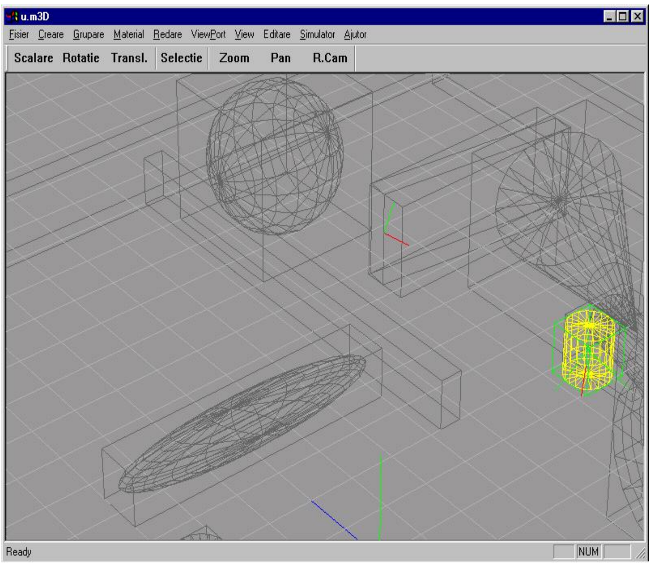

Fig.7 View from the edited environment: the robot and a central recording camera.

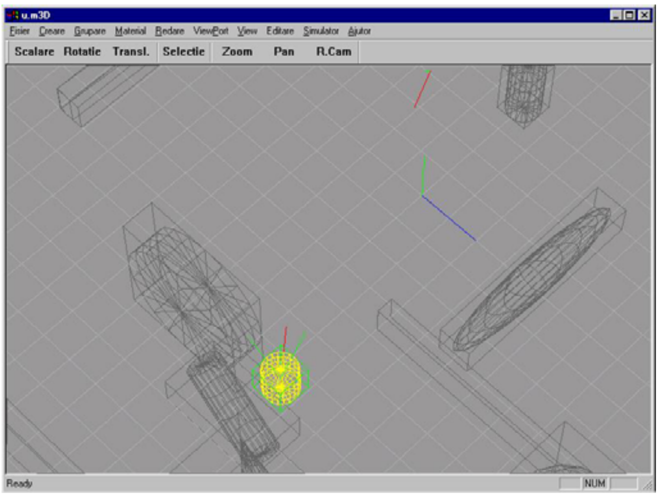

Fig.8 The robot with the sensors and the attached video camera.

3.4 The "Supervisor" module

This software module is responsible with the coordination and synchronization of the activities of the other components of the simulator: Environment Module, Robot Module and Controlling Program Module.

Generically speaking it allows the opening of an environment (previously edited), the possibility to introduce inside this environment one or more robots, the loading for each robot a controlling program on free choice and rolling the simulation using all these items. The exit is made partly of the data provided by the sensors (in a form of log files or data base – at free choice) and on the other hand, (if wanted) by a controlling program improved by evolution and/or learning.

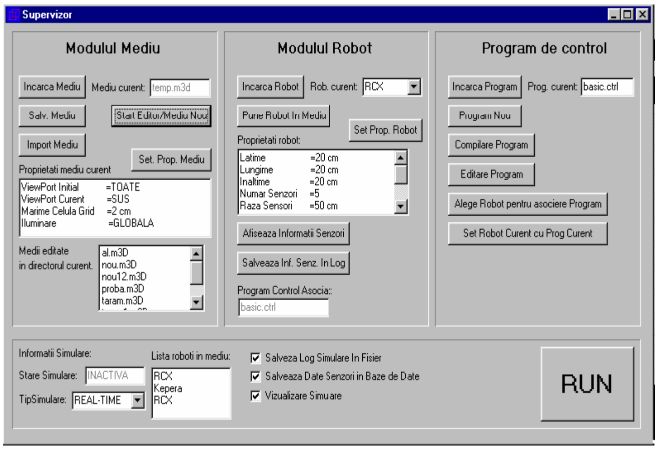

Fig.9 The window for the "Supervisor" module application

The *"Supervisor"* module is responsible with the coordination and synchronization of the activities of the other components of the simulator. At the end of every step of the simulation, the new coordinates of the mobile robot are taken over by the "Supervisor" module in order to update the scene. It allows the opening of an environment (previously edited), the possibility to introduce inside this environment one or more robots, the loading for each robot a freely chosen controlling program and rolling the simulation using all these items. The exit consists of the data provided by the sensors, and by a controlling program improved by evolution and/or learning.

4 Results

One important result was obtained in testing the performance of different types of video cameras and visual processing systems in order to implement the stereo vision system. In figure 4 is presented the 3D robot vision perception.

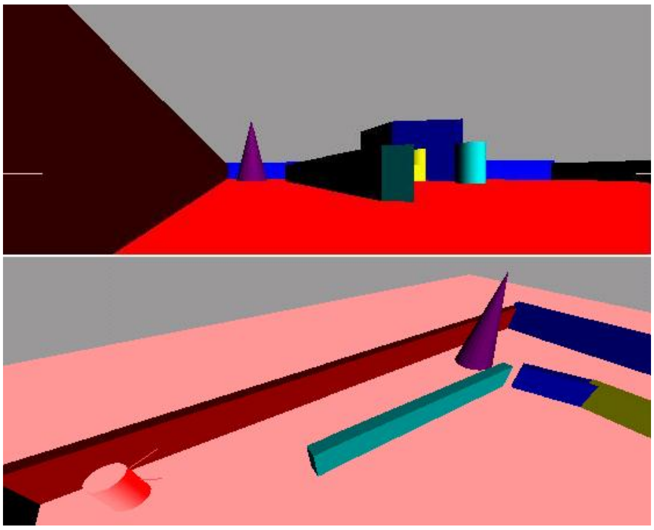

Fig.10 The top image is from the robot's camera, after the robot started the exploration of the environment. The lower image is from a camera mounted over the environment. It's the same thing, from another perspective.

5 Conclusion

The virtual environment presented by this paper is used for developing, testing and optimizing the control devices and interfaces for an autonomus robot.

As final data, this environment present and also evaluate the nature of the interactions and the behavior of the autonomous robot in diverse testing scenarios, according to the degree of teaching, adaptability and accomplishing the requested missions through the training program.

References:

[1] Borenstein, J., Everett, H. R. and Feng, L., "Where am I? – Sensors and Methods for Mobile Robot Positioning", *Prepared by the University of Michigan*, pp. 408-412, April 1996,

[2] Borenstein, J. and Feng. L**.**, "Correction of Systematic Dead-reckoning Errors in Mobile Robots". *Proceedings of the 1995 International Conference on Intelligent Robots and Systems (IROS '95*), pp. 569-574, Pittsburgh, PA, August 5-9, 1995

[3] Bredenfeld A, "Experience with Testing Robots", *in Proceedings of 3rd ICSTEST International Conference on Software Testing, ISBN 3-980 5073-9-4,* Tagung Düsseldorf, Germany, May 2002

[4] John D. Musa, *"More Reliable Software Faster and Cheaper"*, Software Reliability Engineering December 2004, Vol. 8, Number 1

[5] Ricardo Gutierrez-Osuna, Denise Fancher, Kyle Hoelscher, Michael Layton *"Omni-Directional Vision System for Mobile Robots"*, March 26, 2003

[6] E. Franti, S. Goschin, M. Dascalu, P.L. Milea, G. Stefan, T. Balan, C. Slav, V*irtual environment for robots interfaces design and testing*, in CAS-2005 Proceedings, pp. 275-278, Sinaia, Romania, October 2005