Educational autonomous robotics setup using mixed reality

Abraham Prieto García, Gervasio Varela Fernández, Blanca María Priego Torres, Fernando López-Peña
Grupo Integrado de Ingeniería
Universidade da Coruña
Spain
abraham.prieto@udc.es, gvarela@udc.es, blanca.priego@udc.es, flop@udc.es

Abstract—**This paper is concerned with the presentation of a mixed-reality operational environment for carrying out experiments in robotics. This set up was designed as an intermediate system between a real and a simulated scenario in order to work with real robots and to ease and simplify the configuration and modification of the rest of the elements of the experiments. The environment uses video projection over the arena of the experiment, image capture to obtain the current position of the robots and a computer which monitors and controls the scenario. The graphic representation of the virtual elements of the scenario is projected onto the arena and their state is updated based on world rules of the experiment and on the state and actions of the robots. The scheme followed to control and structure this environment is based on the one proposed in a previous work for the creation of a simulation environment named Waspbed, which was designed for the study of simulated coevolution processes in multiagent systems. This setup is used for getting students from engineering degrees used to deal with robotics and all its associated fields, such as artificial vision, evolutionary robotics, communication protocols, etc. in a very simple, quick and cheap way, which otherwise wouldn't be feasible taking into account the academic constraints of time and resources.**

Learning environments; multirobots system; virtual environments, educational robotics, autonomous robotics.

I. Introduction

Any autonomous robotics application and, in general any adaptive system application, usually involves two separate stages, this is, the training or adjustment of the behavior of the robots and the operation of the robots in the real scenario for which they were prepared. Most behavior training techniques involve algorithms based on experimentation to extract the adequate configuration parameters. This is because the appropriate behavior to accomplish any desired task is not known beforehand and several different configurations must be tested and evaluated so that changes can be applied to correct inefficiencies and obtain the most adequate one.

In practice it will be necessary to set the behavior for each robot and let the system act during a predefined testing time. The length of this test will be related to the complexity of the global behavior. Additionally, if there are changing initial conditions and a relevant amount of noise or randomness in the scenario, more than one test will be necessary to counter the variation between tests and to provide an average measurement. Therefore, a consequence of the need for experiment based training algorithms in this type of setups is that the evaluation of every configuration is expensive either in time or resources or both.

Additionally, training behaviors in autonomous robotics has been traditionally associated to artificial intelligence techniques such as reinforcement learning or evolutionary computation [1] [2] which use very little previous information about the problem but require a large number of evaluations of the proposed solutions. This is because relations between the configuration parameters of the behavior and the performance in achieving the global objective of the experiment are typically unknown as a consequence of the complexity of real scenarios and the amount of noise and non-determinism in their real behavior. As a result, those techniques employ more evaluations to do the training thus complicating the procedure for dealing with these problems.

Solutions have appeared to deal with this kind of problems within the field of evolutionary robotics and one of the most relevant is the embodied evolution methodology developed by Watson, Ficici, and Pollack [4] where the optimization algorithm is embodied in real robots while interacting on the scenario in an attempt to significantly reduce the number of evaluations. However, as pointed by some researchers, embodied evolution is inapplicable to most tasks because of the large number of robots required for an appropriate population size [3].

On the other hand, a straightforward and very widespread solution has been the creation of models to simulate the resulting behavior in a computer achieving this way a reduction in the time and resources required. In fact, nowadays every field in engineering takes advantage of the use of computer simulations, and robotics is not an exception. Therefore, it is easy to find a significant number of models for the simulation of robots; however, the models used for the simulation should represent not only the

978-1-4577-1127-5/11/$26.00 ©2011 IEEE

behavior of the robot but also the robot-environment interactions through its sensors and actuators [5], which is as important as the model of the robot to satisfactorily characterize the final behavior.

Currently we can see some examples of simulators that provide these capabilities such as the Stage two dimensional simulator [6], and the Gazebo three dimensional simulator [7] both using the Player server. Another existing commercial solution is the one provided by Webots [8]. We can also find fully open source simulation engines such as Delta3D [9] and USARSim [5].

It is clear that many advances in this field have been achieved and we can find several simulators with libraries for the most typical robots which are very helpful when using algorithms which require large amounts of processing time for the learning stage. However, one important issue still remains having to do with the transference between the behavior experimented for the robots in the simulation and the operation of the robots in a real environment. Even with very accurate models, when the simulated time grows or when the number of elements interacting is high, the divergence between the results obtained using real and simulated procedures, is inevitable. This implies that a subsequent stage is required for readjusting the solution. This stage, however, may distort the results obtained by the optimization process.

Consequently, regarding the field of autonomous robots we can appreciate that both strategies presented, optimization based on real operation of the robots or, optimization based on computational models, present important handicaps. These, when translated to educational constraints makes the teaching of autonomous robotics very cumbersome. Aiming to reduce these difficulties, we have decided to implement a hybrid solution and thus, in this work, we present a mixed reality environment for carrying out experiments in autonomous robotics.

II. SCENARIO SET UP

The setup is made up of the following elements (a diagram of the setup is shown in figure 1):

- An arena. A white board is placed on the floor of the room to serve as an arena on which robots and other elements of the scenario will interact.
- A video projector situated on top of the arena. All the virtual components of the scenario are generated by projecting them onto the white board from a video projector.
- Zenithal camera. Also above the arena, one Ethernet video cam is continuously capturing what is happening on the scenario and sending it to the main manager computer.
- Main manager computer. During the execution of the experiment a computer is running a simulation in a simulation platform developed in a previous work called Waspbed. It receives the visual information of the experiment from the zenithal camera to extract the state of all the real elements present in the scenario and, according to these and to some behavioral rules, modifies the state of the virtual elements.
- Robots. The set of robots are placed on the arena and they can interact with other robots and real elements of the scenario using their actuators and sensors. They can also interact with virtual elements of the scenario using virtual actuators managed by the main manager and using the camera of each robot to sense what is being projected. Currently we are using e-pucks for the first experiments which were designed with a series of features which make them very suitable for education in engineering [15].
- Real elements. Apart from robots there are some other real elements in the scenario to increase the possibilities and combine virtual and real elements and interactions such as, for example, walls, obstacles, objects to be transported, etc.
- Virtual elements. Finally, there are some virtual elements which are controlled by the main manager computer. These virtual elements have a graphical representation associated to each one of them which is projected onto the arena by means of the video projector. The behavior is defined using the Waspbed procedure and since it is represented by some preprogramed events it can be as complex as required.

As a result, we have a set of real robots which interact with a mixed (virtual and real) environment in which some elements are projected onto the ground of the arena from a video projector installed on top of the working space. On the other hand, the behavior of every virtual element is controlled by a main computer with some predefined rules. The use of this mixed environment opens up possibilities for creating any type of interaction by creating any non-available real element that we would like to use, virtually.

Regarding the flow of information, the robots are able to obtain information from the virtual elements only by means of their cameras and analyzing the projected image. On the other hand, the virtual elements capture information from the robots through the main computer which, at the same time, estimates the state of the robots by extracting information from the image provided by the zenithal camera.

Different communications strategies are allowed in the scenario. Both robot-robot communication and robot-virtual element communication through the main computer are achieved using bluetooth transmissions. Evidently, stigmergy communication, quite often used in some bio-inspired algorithms for autonomous robotics is also allowed using modifications of real or virtual elements present in the scenario.

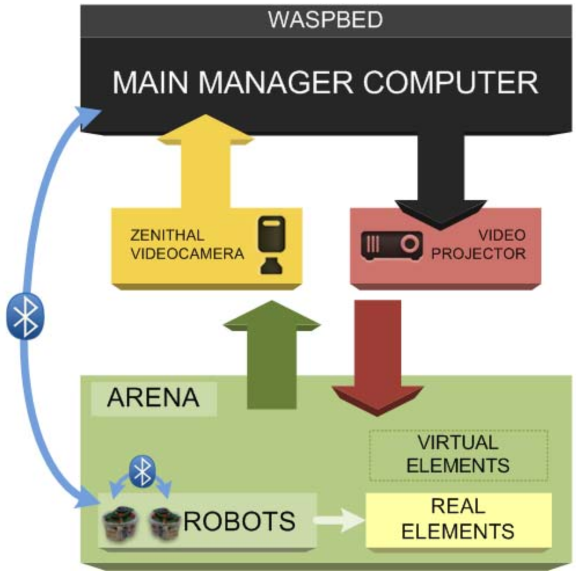

Figure 1. Schematic representation of the mixed reality environment.

Thus we have chosen to use real robots to avoid the aforementioned behavior transference and we have solved some of the problems associated with the set up and adaptation of the real environments. We are able to create any environment with any behavior of the elements contained within it easily and quickly. We are aware of the fact of that we have reduced the set of sensors of a robot to interact with the virtual scenario to a single camera. However, as can be gleaned from the most recent state of the art on the subject, as the robots and its behaviors have increased in complexity, the use of artificial vision to extract information from the environment has become more and more important because of the amount of information they provide. In fact, nowadays, most traditional sensors can be substituted by a camera (light intensity sensor, sonar, collisions detectors, etc).

Since we are using the set up presented for educational matters, the presented handicaps for training robot behaviors, teaching practical courses of robotics is affected by the same problems but they are more important because resources in time and apparatus are frequently more limited. The equipment required is expensive, not always adequate for simultaneous use (group working), and it is easy to damage if the user is not experienced. Also, the configuration of the scenario is, in some cases, a long process which complicates the coordination with strict academic schedules [10]. This mixed-reality environment solves some of these difficulties, makes possible the repeatability of the experiments helping to share environments and algorithms developed between universities to promote transnational education as presented in some other works [11] and provides an improvement in usability.

Within the state of the art, we can find several approaches in which a virtual or mixed environment is used for teaching activities but to our best knowledge, none has been used for teaching autonomous robotics. For example, in [12] a virtual environment was created to allow students to run virtual simulations remotely, simultaneously and without risk of damage. Nevertheless, though it can help students to get familiarized with the control of the robots, the whole process is virtual and it skips the part in which the pupil learns how to deal with the problems of a real environment. We have also found some works seeking the improvement of man-machine interaction: in [13] they explore the use of a virtual and augmented reality device to enhance, motivate and stimulate the learner's understanding of some issues which are complicated to achieve with traditional learning. In [14] a similar setup as that proposed in our work is presented for general purpose educational games.

III. SIMULATION SCHEME: WASPBED

In previous sections we have described how the hardware implementation of the setup has been assembled but, additionally, and in order to simplify the operation of the system, the definition of the environment for the teaching staff and the analysis of the experiments for the students or teachers, we have created a software framework that establishes a structured procedure for the construction of different experiments with the system. This is, every experiment which needs to be prepared will be defined filling certain configuration templates for those virtual and real elements present in the scenario and, defining some interaction events which will determine the rules for the variation of the state of the each virtual element.

The methodology created for the software framework uses an underlying structure presented in a previous work for the definition of a computer simulation environment called Waspbed (World-Agent Simulation Platform for BEhavior Design) [16][17]. This structure that was used for the creation of virtual scenarios in Waspbed is now transferred to the mixed scenario in which some of the constituent elements are real and others are virtual.

The scenario definition of this simulation platform provides the capability of changing the environments, the definition of the participating elements or the constraints with the minimum effort. This is, it permits having full control and freedom to introduce any components, rules, interactions, control systems and so on. Waspbed uses a creation template useful for all the different configurations of the different environments. Apart from that, the use of this template guides and simplifies the simulation definition process. In addition, Waspbed is provided with different and very configurable analysis tools.

In terms of educational advantages it is easy for the students to get used to the procedure for creating scenarios, it is straightforward to monitor the experiments analyzing some relevant parameters and it is simple for the teachers to modify existing scenarios to test the capabilities of the students for training new robots or preparing algorithms for extracting information from the data received through the sensors. The WaspBed core is composed of four main blocks:

- Main manager: responsible for initialization and linkage of all the modules, control of external inputs, file access, the graphic interface, etc. That part of Waspbed is in charge of high level management which is transparent to users and doesn't require any setting from students or teachers.In order to increase the generalization and simplify the definition of the environments and their interactions, each "scenario" is composed of just two types of structures: Elements and Events.
- Elements: they represent anything inside the world that contains a set of characterization parameters. These parameters can be associated to the element or to a group of element and can change or not throughout the simulation run (element state or descriptive parameters). In the mixed reality setup, every component of the scenario has its associated 'element' object inside Waspbed, real and virtual ones. The definition of the elements is very important to manage the simulations and work with them. Thus teachers should have some examples of elements definitions using the configuration templates to show to the students. In addition students should be able to define their own elements to create scenarios.
- Events: they are the representation of the different interactions that guide the simulation evolution, that is, any action that implies variations in the value of the state parameters is an event. Virtual elements have their set of events defined in the simulation and they change their state according to that but, on the other hand, real elements have no events associated to them. They execute their own actions which can affect real and virtual elements. Events are written in JAVA extending the *event.java* class and therefore pupils are required to have skills using this programming language.
- Configuration templates: they are the set of xml files that define the environment. There are three types: creators (initialization parameters of the elements), type definitions (parameter definitions and their default initialization values) and the compatibility board (definition of the associations between elements and events). Once the structure of the definitions is understood the creation of scenarios is really fast so, students should be familiar with them and is very recommendable for the first hours of teaching to focus on the creation of different scenarios and different events.

The Waspbed tool has been programmed in JAVA mainly due to its platform independence capabilities. The computation of the events algorithms is distributed amongst the elements that run as independent process threads. The simulation interface also allows associating a control system to each agent, managed by one or more decision threads. This control system is provided with a sensory apparatus and an internal memory. In addition it contains a learning block that provides means to adapt the control system of virtual agents.

Finally, and very important, in terms of analysis, the environment is basically endowed with what are called "monitoring agents" and several representation panels within the graphic interface. These agents collect information of single element parameters or complex combinations of them changing in real time, and represent them in the different panels while the simulation is running. These combinations of parameters, which may include statistics, trends, local trends, etc are defined by the user in the same way as the events, and represented through bars, graphs, or numerical values. Thus, the analysis tools allow for easy reconfiguration and adaptation to each specific case. As mentioned in the creation of elements and events, being able to define the appropriate monitoring agents is a very important part of the learning process in autonomous robotics. In fact, dealing with complex experimental setups is very complicated if we are not analysing properly what is happening on the arena.

Once students are familiar with the creation and monitoring of scenarios they can start working with real robots for defining their behaviours or the procedure for training them. Another important part of the learning process is to be able to extract information from the sensors of the robot, the camera to obtain the state of virtual and real elements, and those sensors used to detect real elements (sonar, infrared sensors…). Using the environment they can develop their own data analysis algorithms for any kind of sensor and test them in a controlled scenario.

The real experimental setup which corresponds to the schematic that was previously described is shown in figure 2 where we can see the simulated environment projected onto a white board which is the arena and the surface where the robots are wandering around. Figure 2 shows the videocamera and overhead projector connected to the environmental control computer. In the first experiments the camera has been used to detect the position of the robots using a segmentation algorithm and a marker on top of every robot to make them easy to distinguish.

With this assembly, the experimental operation follows this sequence:

- First, the main manager is executed in the computer, which initializes the states of all the virtual elements of the scenario.

- Second, the sensors of all elements get information from the rest of the virtual and real elements. Virtual elements have virtual sensors to detect parameters from other virtual elements. To detect parameters from real elements the main manager gives them the information coming from the sensors of the setup (in this case, the overhead camera).
- Third, bluetooth communications between robots and the main manager is carried out.
- After this the events associated to existing virtual elements are executed modifying some of the parameters.
- Finally, and in parallel with those four previous steps, the robots sense the environment, execute their controllers and use their actuators according to the predefined behavior.

As presented here, the operation of the whole system is conceptually equivalent to that of a Waspbed simulation, which is more accurately described in previous works, but with the addition of some new real elements.

IV. Some Scenarios

In order to show the operation of the mixed reality environment we will present some examples. The first experiment consists in a cleaning task. The arena represents a dirty area and the task of the robots is to clean it as thoroughly as possible. In this case, the environment control system reacts by changing the color of the arena when a robot covers one location: gradually increasing the darkness as time passes without any robot going over a particular area (dirt accumulation) and refreshing the color from dark to light whenever a robot goes over an area (dirt cleaning).

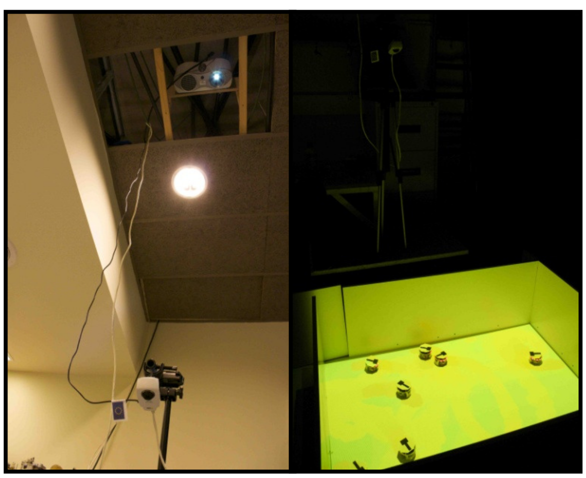

Figure 2. Scenario assembly. Projector and zenithal camera (left), projected arena and e-pucks (right).

A team of e-puck robots (see figure 3) was considered. In this case they make use of their infrared sensors and camera to sense other robots and the environment, their bluetooth connection to exchange genetic information and their wheels to move.

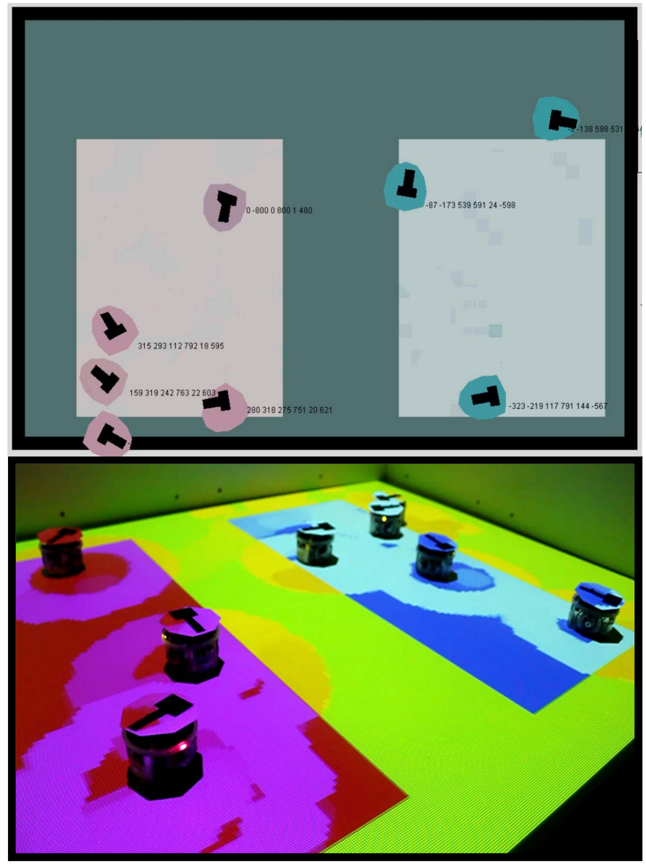

Figure 3. Cleaning task scenario. Virtual representation in Waspbed (top). Arena of the real experiment with epucks and two differentiated cleaning areas (bottom).

The virtual elements are: the grid to be cleaned and one monitor agent to compile information about the simulation, whereas the robots and walls are real elements. Using the mixed-reality environment allows the application of any desired change: updating rules of the grid, creating several areas to be cleaned through different actions… and the robots must use their sensors to get information about the environment (camera), other robots (camera and infrared) and walls (infrared) dealing with traditional problems of using real sensors and actuators as desired.

As indicated, for this specific experiment we have defined one scenario with real robots and walls to use the mixed reality environment. In addition, we have defined a completely virtual scenario in which we have created virtual robots defining their behavior with the required events. This way, the students were able to practice remotely with Waspbed and to create different configurations of the scenario in order to try them before implementing the final result in the mixed reality scenario during the classes. The educational purpose of this scenario, as the behavior of the robots is controlled with the r-Asico algorithm [17] and the students have no control over it, is to study the effect of the generation of changes in the scenario on the resulting behavior, and what is more important to become used to control the scenario events using the monitor agents.

Another experiment conducted in this environment was a competitive task for gathering balls on the scenario and placing them in some specific positions. In this case, it is assumed that the slope of different parts of the arena is not constant, and consequently, the projected scenario represents a hilly surface which affects the velocity of the robots when they modify their position. Figure 4 shows the environment during one of the classes using this scenario. Robots are able to detect the slope and balls and it affects their motion velocity. In this case the scenario was given to the students and they had to work in the implementation of behaviors for the robots to be able to solve the required task.

Regarding the educational aspects of this experiment, we have observed that using competitive tasks makes the classes more entertaining and helps to motivate the students to create better algorithms for training the controllers, analyzing information from the sensors and so on.

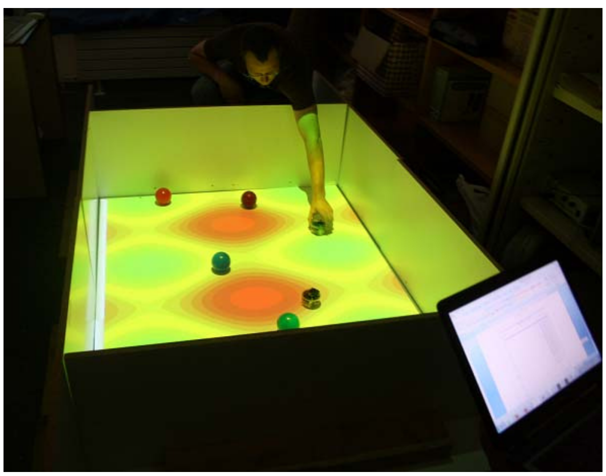

Figure 4. Students trying their controllers in the gathering task scenario during a practical class.

V. CONCLUSIONS

In this paper we have presented a mixed reality educational environment that provides a simple way for teachers of robotic courses to set-up experiments containing real robots without the hassle of constructing physical environments. This approach has been shown to be effective in getting students interested in performing experiments and playing with the different parameters of the physical robots. This has allowed for a better understanding of the complexities of real sensors and control using real hardware.

REFERENCES

[1] Lund, H. H., Miglino, O., and Nolfi, S. "Evolving mobile robots in simulated and real environments" Artificial Life , 1995, 4 (2), 417–434.

[2] D. Floreano and F. Mondada, "Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot". *From Animals to Animats 3: Proc. 3rd Int. Conf. on Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Cambridge, MA (1994).

[3] E. Stefan, U Eiji, D. Kenji and C. Henrik I. "Darwinian embodied evolution of the learning ability for survival". Adaptive Behaviour (2007).

[4] Watson, R., Ficici, S., & Pollack, J. (2000). "Embodied evolution: distributing an evolutionary algorithm in a population of robots." Technical Report CS-00-208, Department of Computer Science, Volen National Center for Complex Systems, Brandeis University, USA.

[5] Balakirsky, S., Scrapper, C., Carpin, S., and Lewis, M. (2006). "Usarsim: Providing a framework for multi-robot performance evaluation." In Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, pages 98–102.

[6] "Player/stage project," http://playerstage.sourceforge.net, 2005.

[7] Koenig N, Howard A. "Design and use paradigms for gazebo, and open-source multi-robot simulator." In IEEE/RSJ International Conference on Intelligent Robots and Systems Sendai, Japan; 2004:2149-2154.

[8] O. Michel, "WebotsTM: Professional mobile robot simulation," International Journal of Advanced Robotics Systems, vol. 1, no. 1, 2004

[9] Delta3D, http://www.delta3d.org/, 2006

[10] 1. Ch. Salzmann, P. Saucy, D. Gillet and F. Mondada, "Sharing of unique or expensive equipment for research and education," Informatik / Informatique, Magazine of the Swiss Informatics Societies, 4, 1999, pp. 32±33

[11] Bruno Baruque, Álvaro Herrero, Emilio Corchado and Javier Sedano. Implementation of the European Computer Science Course under the Spanish University Education System. International Conference on European Transnational Education (ICEUTE 2010). Burgos (Spain) September 2010. Universidad de Burgos. ISBN: 978-84-92681-18-1

[12] 3. F. A. Candelas, S. T. Puente, F. Torres, F. G. OrtõÂz, P. Gil and J. Pomares, "A virtual laboratory for teaching robotics," International Journal of Engineering Education, 19(3), (2003) pp. 363±370

[13] Pan, Z., Cheok, A.D., Yang, H., Zhu, J., Shi, J., 2006. "Virtual reality and mixed reality for virtual learning environments." Computers & Graphics 30 (1), 20–28.

[14] Chang, C. W., Lee, J. H., Wang, C. Y., & Chen, G. D. (2010). "Improving the authentic learning experience by integrating robots into the mixed reality environment." Computers and Education, 55(4), 1572-1578.

[15] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli, "The e-puck, a robot designed for education in engineering," in: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, vol. 1, 2009, pp. 59–65

[16] Abraham Prieto, Richard Duro, and Francisco Bellas, "Obtaining Optimization Algorithms through an Evolutionary Complex System", Abstracts booklet ECCS07, pp.132-1132, European Complex Systems Society, 2007

[17] Abraham Prieto, J.A. Becerra, F. Bellas, R.J. Duro, "Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time," Robotics and Autonomous Systems, Volume 58, Issue 12, Intelligent Robotics and Neuroscience, 31 December 2010, Pages 1282-1291, ISSN 0921-8890.