A Visualization System for Teaching Intelligent Mobile Robotics in SyRoTek

Miroslav Kulich, Jan Faigl, Jan Chudoba, Karel Košnar
Department of Cybernetics
Faculty Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Prahue 6, Czech Republic
{kulich,xfaigl,chudoba,kosnar}@labe.felk.cvut.cz

Abstract. This paper is concerned with a visualization of sensory data in the SyRoTek project that aims to create an e-learning platform for supporting teaching mobile robotics and artificial intelligence. The visualization is based on the Stage simulator that is extended to support multi-view configuration and combination of real scenes and simulated environments scenes in a single visualization window. The modified Stage becomes a standalone visualization application that can be used for on-line and off-line video creation. Moreover, the application can be used in the development process of robot control application. It helps students to debug their applications and to understand the robot perception of the real surrounding environment in real-time or from recorded/logged data.

Keywords: e-learning, robotics, visualization

1 Introduction

The first challenge in teaching intelligent mobile robotics is a demonstration of robot perception to students. Even though such demonstration can be based on a simulation, real mobile robots are more attractive as they provide an appealing way how computer programs can interact with the real world in meaning to make something with matter. The robot perception of the surrounding environment is crucial to achieve intelligent robot behavior, and therefore the robot has to be equipped with sensors and sensory data have to processed. A visualization of sensory data is therefore the key for students to realize how the robot see the world, and how sensory data can be used in robot control.

A combination of a real world view with data, albeit in a simple way, has been a part of the first virtual laboratories e.g. [1, 2]. Subsequently used Augmented Reality (AR) approaches provide a worthful service in better understanding and practical acquisition of robot control principles in education processes. Authors of [3] noted that even though several AR visualization systems have been proposed, very few tools are specifically designed for robotics debugging and evaluation. For this purpose, not

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

only sensory data, but visualization of robot states and the created world representation is useful. The currently most promising visualization system for robot developers is ARdev [4], which works with the Player framework [5].

In this paper, we describe our approach to combine a real world view with visualization of sensory data and robot's states. The proposed visualization system is a part of our project SyRoTek (system for robotic e-learning) [6]. Although the ARdev system provides valuable features, it doesn't directly support specific requirements of the SyRoTek project. The most important missing features are combination of several real world views captured by several cameras and support of multiple view combining real and simulated environments. These features are mainly required to support students' task evaluation by teachers that is based on automatic creation of video files consecutively available for download or live streaming on demand. The third reason for creation of the more powerful visualization system was an intention to create an easy to use system that can serve for creation of demonstration videos, on-line live visualization and also for a replay of recorded logs.

The rest of the paper is organized as follows. A brief overview of the SyRoTek project and description of the proposed visualization concept are presented in Section 2. Our proposed visualization system, based on the Stage simulator is described in Section 3 and its application in teaching robotics is described in Section 4. Concluding remarks are presented in 5.

2 Visualization Architecture in the SyRoTek Project

The SyRoTek project aims to create an e-learning system for education in mobile robotics and artificial intelligence fields. It is a representative of a robotic virtual laboratory in which a set of mobile robots is available for students to create a program that will control a mobile robot using real sensory data. The robots operate within an arena with dimensions 3.5× 3.8 m that is placed in a computer laboratory, see Fig. 1a. A description of the SyRoTek system architecture can be found [7, 6]. Here, the focus is on the visualization part of the whole architecture.

The visualization in SyRoTek comprises several real views to the arena and data. The data are values of robots' sensors like sonars, laser scanner etc. Besides, the robot position is provided by the global localization system based on image processing from a camera placed on top of the arena. Video files and data are provided by three types of presentations in SyRoTek, see Fig 1b. First, a single image of the current scene is provided through web pages. The second type are video files with mixed sensory data and real scene views that are created at the SyRoTek server called the control computer. The main advantage of this presentation is that a regular video player can be used to show the videos at user's workstation. The last type of the considered presentation options is visualization of sensory data in a dedicated application in which sensory data can be combined with videos files. The dedicated application can be used to visualize real robots and sensory data in a virtual environment that is advantageous for a low bandwidth connection to the SyRoTek server and for on-line

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

visualization. In this type of visualization, real data are presented in a simulated environment, and therefore we call this type of visualization the Mixed Reality [8].

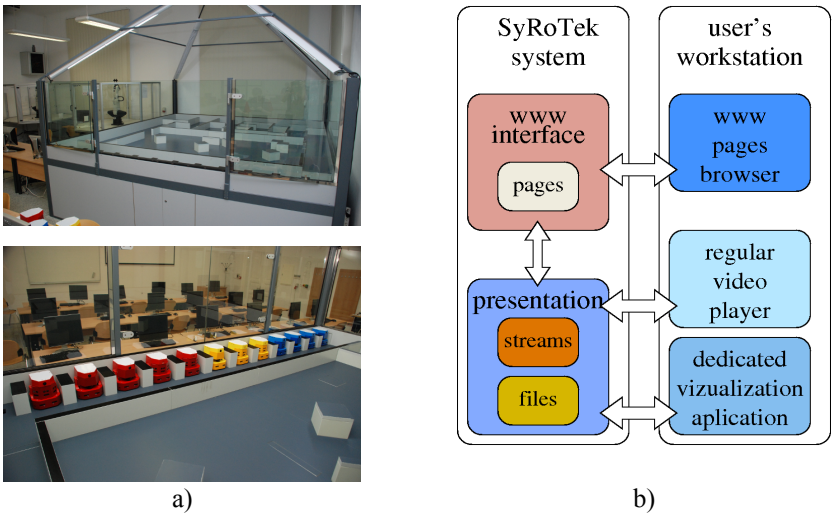

Fig 1. (a) SyRoTek arena and (b) users access to the visualization.

The live visualization is provided through streams, while video and sensor data files are provided as downloadable files for further visualization at the client machine or as streams for visualization on demand.

The main purpose of the visualization system in SyRoTek is to support development of users' applications that are a part of their assignments and also to support evaluation of the applications by teachers, which will work as follows. A final student's program that realizes autonomous behavior of a mobile robot, e.g. mobile robot exploration of unknown environment, is scheduled for commitment. Then, in the free time, where students do not actively use the robots in the arena, student's program is executed and a desired video of the robot behavior is captured. All required data are logged and according to the desired configuration resulting video files are created. In these video files, views from several cameras are combined with sensory data and particular results of the user's application to provide an overview of the robot performance. Such video files are then delivered to a teacher as a supporting material for user's assignment evaluation.

The required feature is a combination of several views in a single image, therefore the basic video processing pipe line consists of combination of several cameras' views into one image that is then used in the final image. The camera view can be either a real scene view or virtual view in a simulated environment. This processing pipe line is used in the aforementioned types of presentations, for on-line and off-line video presentation, and in the dedicated application for visualization of sensory data and views to real or virtual environments at the user's workstation. In order to combine sensory data and real videos in a meaningful way, images of the real scenes have to be timestampped to match the logged sensory data. Overview of the proposed visualization subsystem is depicted in Fig. 2.

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

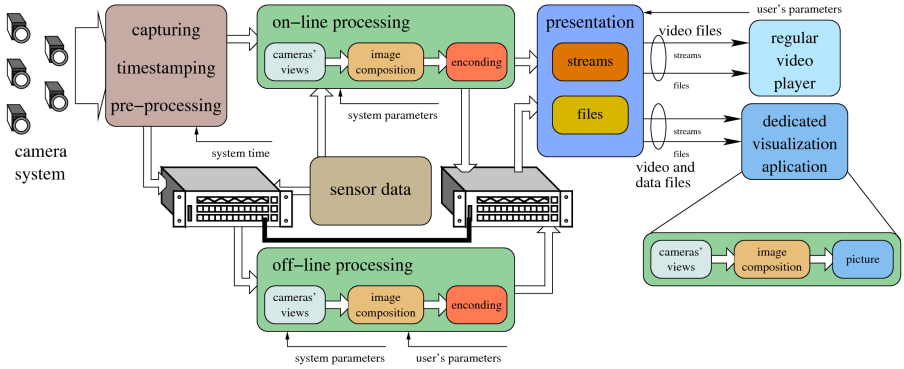

Fig. 2. Architecture of the visualization subsystem.

The core of the visualization component (the processing pipe line) being developed provides the Mixed Reality for creating video and visualization of live or logged robot behavior that is helpful for understanding the principles of real robot perception and debugging of the robot control application. The component is based on the Stage 3.x simulator augmented by visualization of video files and its main implementation aspects are described in the next section.

3 Implementation

Implementation of the visualization system in the SyRoTek project is based on Stage from the Player/Stage project [5]. Stage is primarily designed as a robotic simulator that simulates population of robots equipped with sensors in 2.5D environment. Moreover, it provides graphical environment for visualization of the simulated world and objects in it. Unfortunately, these two functionalities are tightly conjoined and so the current version of Stage cannot be used for visualization of a real world or real robots and their sensors. It is also not possible to visualize robots and sensors simulated in one instance in an other instance.

Both mentioned cases appear in a typical work with the SyRoTek system. In the first case, the user needs an overview over a situation on the field, behavior of the robots and sensory output. The second case arises in the moment, when a student works distantly on the control computer in a simulated environment and wants to display the actual state of the environment. It is possible to use some of remote desktop technologies but these are demanding on capacity of a communication channel. The better way is to run a visualization application directly on user's computer. The application gets data about the current state of the field, particular robots and their sensors from the control computer and display them.

To accommodate all requirements mentioned in previous paragraphs, Stage (version 3.2.2) was modified in the following way:

- The controller ***viewer*** has been created. This controller allows to run

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

Stage as a standalone visualization application. It can be configured so that it acts as a Player client and displays the data received from Player to the graphical window.

- A video player has been integrated into Stage, that is able to play both video files and video streams.
- Multi-view support has been added.

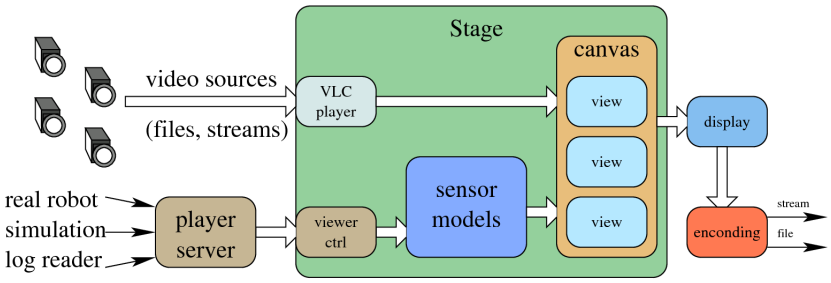

Fig. 3. An overview of the visualization system based on modified Stage.

The modified version of the Stage with these improvements stands for the core of the visualization system. An overview of the modified Stage is depicted in Fig. 3. Viewer controller is connected to a running Player server that controls real robots, simulated robots or/and playbacks previously gathered logs. Moreover, the video player in Stage reads a video source (either from a file or from a stream). Data from Player are processed by Stage's sensors models that are responsible for correct rendering of sensors and together with the particular images produced by the video player are displayed in views of the drawing window. The window can be finally presented to the user or encoded into another video. Particular added components to the modified Stage are described in the following subsections.

3.1 Viewer Controller

Stage in version 3.x is a simulator application that can be used with or without the Player server. A concept of controllers has been introduced in the version, which are plug-in modules to be attached at runtime to the models of Stage. A controller can be a sensor data filter or can substitute complete robot control without Player. This concept is used to make Stage a standalone visualization application.

The developed ***`viewer`*** controller reads data from a real robot (or the robot simulated in an other instance of Stage) and its sensors and displays them instead of data simulated in Stage itself. In other words, the controller is a Player client application that reads data from it and fills internal Stage structures.

3.2 Video player

The integration of video player into Stage is based on LibVLC [9] which is an open-

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

source library for handling and playing videos in a huge amount of formats. The key issue of combining real scenes with additional (artificial) data in Mixed Reality systems is a correct transformation to be applied to the data in order to display them at right positions in the video of the real scene. This problem can be solved by the camera calibration, i.e. finding intrinsic and extrinsic parameters of the camera, followed by the transformation of the artificial data according to the found parameters. There exist many algorithms for camera calibration that differ in the format of both input data and output parameters. Unfortunately, none of the algorithms or their implementations produces the calibration parameters in the form directly usable in OpenGL which is a library used in Stage for rendering the view. Therefore a transformation of the camera calibration result is needed. This transformation is described in the next paragraphs as we haven't found it in the literature, and it can be helpful for other developers.

For the camera calibration we use the method described in [10]. The method takes a set of 3D coordinates in the real world together with corresponding 2D coordinates in the video image. As a result, extrinsic and intrinsic parameters of the camera are returned. OpenGL uses projection and modelview transformations to define a desired transformation of drawn objects on the scene. The projection transformation defines how the 3D scene is projected in the 2D screen. The intrinsic parameters of the camera form the following matrix, which is used as a projection matrix

$$\begin{pmatrix} \frac{f_{prime}}{aspect} & 0 & 0 & 0 \\ 0 & f_{prime} & 0 & 0 \\ -2\frac{w_x}{w_{width}} & -2\frac{w_y}{w_{height}} & -1 & -1 \\ 0 & 0 & -2 & 0 \end{pmatrix}, \text{ where}$$

$$f_{prime}=\frac{2f}{dpy\, I_{height}},\quad aspect=\frac{I_{width}\, dpx}{I_{height}\, s_x\, dpy},\quad w_x=c_x\frac{w_{width}}{I_{width}}-\frac{w_{width}}{2},\quad \text{and}\quad w_y=-c_y\frac{w_{height}}{I_{height}}-\frac{w_{height}}{2},$$

z_{near} and z_{far} specify the distance from the viewer to the near and far clipping planes and they depend on a concrete scene. We set $z_{near}=1$ and $z_{far}=10^6$. f stands for effective focal length of the camera, I_{width} and I_{height} are width and height of the image in pixels, s_x is the scale factor desacribing uncertainty of camera horizontal scanline resampling and dpx and dpy are effective X and Y dimensions of a pixel in the camera (in mm/pixel). c_x and c_y are origin coordinates in the image plane and w_{width} and w_{height} are width and height of the presentation window.

The modelview matrix contains modeling and viewing transformations, which transform object space coordinates into eye space coordinates. As y and z matrices have a different orientation in the calibration tool and OpenGL, the modelview matrix should be rotated by the matrix R first, where

$$R=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The modelview is then translated (using `glTranslated` function) by the vector $[t_x, t_y, t_z]$, where t_x, t_y, and t_z are parameters provided by the calibration as translation parameters. Finally, a sequence of rotations is performed according to axes z, y, and x:

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

```
glRotated( φz, 0.0, 0.0, 1.0)
glRotated( φy, 0.0, 1.0, 0.0)
glRotated( φx, 1.0, 0.0, 0.0),
```

where φ_x, φ_y, and φ_z are the extrinsic camera parameters for rotation obtained from the calibration in degrees. The example of sensor data integrated into a video is shown on Fig. 4.

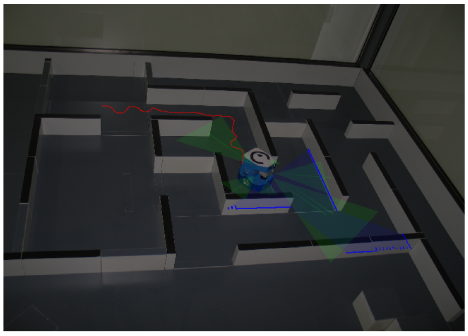

Fig. 4. Video integration with sensory information: laser data in blue, sonar data in green, and traversed trajectory in red.

An issue of the video processing is the problem of barrel distortion. For expensive cameras the distortion is negligible but it should be corrected for common cameras. One way is to perform the correction directly on the camera (if the camera allows programming), the other possibility is to do that in the visualization application just before presenting the image. For barrel distortion correction we use a standard tool described in [11].

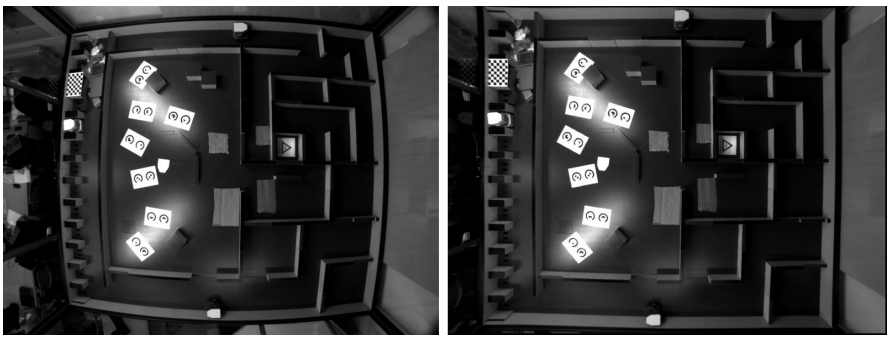

Fig. 5. Image before and after radial distortion correction.

3.3 Multi-view support

When controlling a robot at distance, it may be helpful to have different views to the scene - a general overview of the field, detailed view of the nearest robot

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

neighborhood, a view from the robot perspective, or views with data generated by the control application at the same time. The need of different views is stronger for multi-robot applications, where behavior of particular robots should be traced and controlled.

Multi-view support has been added into the Stage in order to satisfy these needs. All the views are fully configurable as a standard Stage window so the user can choose what kind of data and how will be displayed in each view. The implementation uses OpenGL viewports that allow to display a scene into a defined region of the window (canvas). For each view a viewport is defined where a whole scene is rendered in the same way the scene is rendered into a window in an original version of the Stage. Two different configurations of the Stage with multi-view support are shown in Fig. 6.

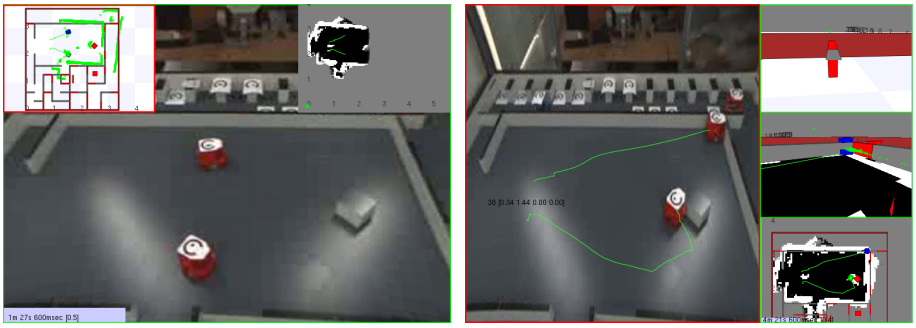

Fig. 6. Two different configurations of Stage with multi-view support. (a) Three views: the real scene, a map with sensor data, and an occupancy grid. (b) Four views: the real scene with robots trajectories drawn, two views from robots point of view in the simulated environment, and an occupancy grid.

4 Visualization System in Teaching Robotics

The implemented visualization application based on the modified Stage simulator is used in various contexts of the SyRoTek visualization architecture. The application is a flexible tool that supports visualization according to users' needs and their preferences, e.g. connectivity to the SyRoTek server. At least four types of visualization are supported by the application. These types are described in the following paragraphs together with particular methods of the presentation and expected video quality settings.

The first type of visualization are **informative** video files that provide an overview of the actual situation on the field. The aim of this visualization type is to mediate the user fast overview of the system in order to allow to monitor a running experiment. An informative video can be accessible from a web page of the e-learning system.

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

The visitors of the web page can be motivated to become acquainted with the system. To mediate better overview of the situation and to make the video more attractive, several views can be combined together with sensor data visualization. Informative videos are created on-line, because they capture the actual state of the system, and therefore they are made in the form of stream. With respect to a technical equipment of the user's computer it can be also useful to present this kind of video in the form of sequence of images, e.g. placed on the web page with refresh rate in seconds.

The second type of visualization is **for developing** a robot control application. Views from different positions as well as an overview of the whole scene are advantageous together with the current sensor data the application is processing. Moreover, application specific information embedded in the video can also improve the development process. The student which solves a real-time exercise like mobile robot control can observe events on the field and behavior of the debugged algorithm on-line but step-by-step debugging is not possible due to request to process sensor data as they are sensed and dynamics of the robot. Instead of this, a detailed record of the experiment is acquired that can be thoroughly analyzed.

The third type of visualization aims to support the **evaluation** process of student's assignment that is a creation of an application to control a real mobile robot. Basically this type of visualization is similar to the former case. The combination of several views provides overview of the application activity and also the requested results of the application can be part of the final video. The evaluation video can be streamed on-line during the final experiment or created off-line for further analysis.

The last type of the visualization are **demonstrative** videos that are significant parts of a teaching material or task description. Their main purpose is to demonstrate a principle how the introduced algorithms works. These videos should be attractive to hold student's interest, emphasis should be thus placed to its high quality - both technical and contentual. Concerning technical point of view, the video should have high resolution as well as high data flow. Such video can be accompanied by other files. For example, in case the video demonstrates a function of some algorithm, the raw sensor data processed by the algorithm can be made available to the student. Demonstration videos can be presented as files or streams on demand.

5 Conclusion

Visualization system for Mixed Reality in mobile robotics has been presented. The system is a part of the SyRoTek e-learning project aiming to provide support for teaching mobile robotics and artificial intelligence with real mobile robots. The developed visualization application is based on a set of commonly known open-source applications and libraries that are integrated into a complex but simply usable system providing visualization of all aforementioned types. It's uniqueness lies in a unified approach to visualization in the whole education process at all its levels – from preparation of teaching materials, through support during tasks solving, to support of task evaluation.

Proceedings of SIMPAR 2010 Workshops
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Darmstadt (Germany) November 15-16, 2010
ISBN 978-3-00-032863-3
pp. 494-503

The application is a part of the on-line, and off-line video processing and presentation and also it is used for combining real scene views with sensory date on the users' workstation. Moreover, we plan to integrate the visualization application as a component to an Integrated Development Environment, particularly NetBeans [12] is currently investigated. This activity is welcomed by students as it allows them to use their favorite environment for application development together with direct view of the real scene and it provides them a way how to visualize their applications results.

Acknowledgments. The work presented in this paper has been supported by the Ministry of Education of the Czech Republic under program "National research program II" by the project 2C06005. The support of the Ministry of Education of the Czech Republic, under the Project No. MSM 6840770038 "Decision Making and Control for Manufacturing III" to Miroslav Kulich is also gratefully acknowledged.

References

1. Simmons, R., Fernandez, J.L., Goodwin, R., Koenig, S., O'Sullivan, J.: Lessons learned from Xavier. Robotics and Automation Magazine (2000) 733 – 39
2. Masár, I., Bischoff, A., Gerke, M.: Remote experimentation in distance education for control engineers. In: Proc. of Virtual University 2004, Bratislava, Slovakia. (2004)
3. Chen, I.H., MacDonald, B., Wunsche, B.: Mixed reality simulation for mobile robots. (May 2009) 232 –237
4. Collett, T., MacDonald, B.: Augmented reality visualization for Player. (2006), 3954 – 3959
5. Gerkey, B.P., Vaughan, R.T., Howard, A.: The Player/Stage project: Tools for multi-robot and distributed sensor systems. In: In Proc. of the 11th International Conference on Advanced Robotics. (2003) 317–323
6. Faigl, J., Chudoba, J., Košnar, K., Kulich, M., Saska, M., Přeučil, L.: Syrotek - a robotic system for education. In: In Proceedings of Robotics in Education. (2010)
7. Kulich, M., Faigl, J., Košnar, K., Přeučil, L., Chudoba, J.: SyRoTek - On an e-Learning System for Mobile Robotics and Artificial Intelligence. In: ICAART 2009. Volume 1., Setúbal, INSTICC Press (2009) 275–280
8. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE Transactions on Information Systems E77-D(12) (December 1994)
9. http://wiki.videolan.org/LibVLC: (accessed 30th September 2010)
10. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. Radiometry (1992) 221–244
11. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: Proc. of the Seventh IEEE International Conference on Computer Vision, 1 (1999) 666–673
12. http://netbeans.org: (accessed 30th September 2010)