

Teleworkbench: An Analysis Tool for Multi-Robotic Experiments

Andry Tanoto¹, Jia Lei Du¹, Ulf Witkowski¹, and Ulrich Rückert¹

Heinz Nixdorf Institute
System and Circuit Technology
University of Paderborn, Germany
{tanoto, jialdu, witkowski, rueckert}@hni.upb.de

Abstract. This paper presents a tool, one component of the Teleworkbench system, for analyzing experiments in multi-robotics. The proposed tool combines the video taken by a web cam monitoring the field where the experiment runs and some computer generated visual objects representing important events and information as well as robots' behavior into one interactive video based on MPEG-4 standard. Visualization and data summarization enables the developer to quickly grasp a situation, whereas the possibility of scrolling through the video and selectively activating information helps him analyzing interesting events in depth. Because of the MPEG-4 standard used for the output video, the analysis process can be done in a wide range of platforms. This trait is beneficial for education and research cooperation purposes. ¹

1 Introduction

One way to design and develop multi-robot systems is the use of bio-inspired swarm principles. Swarm systems usually consist of many homogeneous agents that follow a small set of simple rules. Communication, either explicitly or implicitly via the environment, is strictly locally constrained. There is no central coordination, and cooperation among the agents and global effects result from the individual behavioral rules. When developing such robot swarm systems, the setup, observation and analysis of experiments can be tedious and challenging for the roboticist. Numerous robots need to be charged and the programs downloaded onto the robots. When executing the experiment, it is difficult to concurrently observe all robots, which run in real-time and possibly over a long period. It is our intention to provide an analysis tool for experiments in multi-robotics. This analysis tool is meant to help roboticists in assessing their robot programs or algorithms for any application when being tested in real experiments. Through this tool the observable as well as the unseen behavior of robots

¹ This work was developed in the course of the Collaborative Research Center 614 Self-optimizing Concepts and Structures in Mechanical Engineering University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

during experiments will become more transparent to the roboticists, which is an invaluable factor in debugging the robot programs.

For analysis purposes, robots are usually programmed to send or to record a lot of information. The question now is how to effectively summarize all information produced in a way that the actual situation of robots at any particular time can be conveyed to and easily comprehended by robot programmers. One way to do this is through visualization. Visualization can be used to show the behavior and especially to make complex states of behavior comprehensible to human.

The tool we propose works by processing information acquired from the Teleworkbench system [1] during an experiment and outputs one multimedia file visualizing acquired information. The visualization is built based on MPEG-4 standard. There are some reasons for using MPEG-4: its interoperability, flexibility, reusability, and interactivity. With MPEG-4, we can easily combine several multimedia sources, whether it is video or audio, into one file playable in a wide range of hardware and operating systems. This trait is advantageous for exchanging and storing the result of the experiment. Moreover, we can have a *“run once, analyze many”* analysis tool that requires us to run the analysis process only once but enables us to have many perspectives of the problems we are trying to solve at later time. Furthermore, with MPEG-4 we can also embed computer-generated objects into the video data, which is essential for providing a sense of reality for users. In the case of our analysis tool, we use one video taken by an overhead web cam monitoring a field where the experiment is executed and embed computer-generated objects on top of it. These video objects represent information important for analysis, e.g. robots' path, communication message, internal state, sensor values, or even images. To provide interactivity, there is a menu area with which users can select the most relevant information for a certain situation.

To date, there are some tools for analyzing robot experiments, ranging from general-purpose software such as MATLAB [2] and SysQuake [3] as well as application- and robot-specific software such as KITE [4], beTee [5], and PyKhep [6]. However, none of them offers all the features we mentioned above.

In this paper, we present the proposed tool for analyzing experiments using minirobot Khepera. However, it does not necessarily mean that this analysis tool can be used only for this type of robot. The idea of this tool is extensible to any robotic platform.

The paper is structured as follow. After shortly stating the problem and our proposed solution in Chapter 1, we will give a short overview of our Teleworkbench system in Chapter 2. Next, the description of the analysis tool will be presented in Chapter 3. Afterward, Chapter 4 will describe how our proposed tool can help us in analyzing an experiment in bio-inspired robotics. This paper will be concluded in a short summary in Chapter 5.

2 Teleworkbench - A Short Overview

We have presented the Teleworkbench System in [1]. Teleworkbench is a tele-operated platform or testbed for managing experiments involving one or many mini-robots Khepera. The idea behind the Teleworkbench is to provide remote setup and execution of experiments in multi-robotics and also to facilitate an easy analysis on the resulted data. To enable long-time experiments, the Teleworkbench is equipped with recharging stations to enable robots recharging their batteries during runtime. The Teleworkbench is connected to the Internet to allow easy access for remote users located in any part of the world.

2.1 Teleworkbench Features

- **Internet connectivity.** The Teleworkbench is connected to the Internet, which allows easy access for remote users located on every part of the world.
- **Remote experiment setup and execution.** Remote users can setup and execute experiments involving many mini-robots Khepera with various extension modules. The Teleworkbench has one field, measuring 2m x 2m, which is partitionable into four fields of size 1m x 1m.²
- **Robot positioning system.** The robot positioning system will track every robot captured by a web cam and extract robots' position relative to the field. At the current version, 36 robots can be identified and localized.
- **Wireless robot communication.** The Teleworkbench system uses Bluetooth technology as the medium for robot communication. With our Bluetooth module [7], the communication among robots is more reliable and faster (up to 57600 bps) than the normal Khepera Radio Turret.
- **Remote program-download to robot.** Remote users are allowed to directly control the behavior of the robots by downloading their own programs to robots.
- **Live-video of the experiment.** During experiments, users can in real-time watch how the experiments proceed through the live-streamed-video of the experiment taken by web cams monitoring the fields.
- **Events and messages logger.** Every occurred events and exchanged messages are recorded and retrievable.
- **Post-experiment analysis tool.** Immediately after the experiment completion, an interactive MPEG4-video is generated showing some important information recorded during the experiment. With this, we can have a “*run once, analyze many*” analysis tool. Thus, analysis of the experiment is easy and convenient. Moreover, since all information is stored in one file, the MPEG4-based video file, it is very comfortable for exchanging and presenting research result.

² The main field is more or less equal to a field of size 11.6m x 11.6m for robot Pioneer3-DX (44cm x 38cm x 22cm).

- **Interoperability.** The Teleworkbench system was designed to allow communication with other programs, which can be implemented simply by using socket communication. Result in [8] shows how one program acting as a gateway passes an XML-based messages from a remote user to the robot through the Teleworkbench system.

2.2 Teleworkbench Components

The Teleworkbench system (see Figure 1) comprises one field with several cameras monitoring it, a wireless communication system, and some computers connected to a local area network with tasks such as image processing, databasing, message redirecting, and web-hosting.

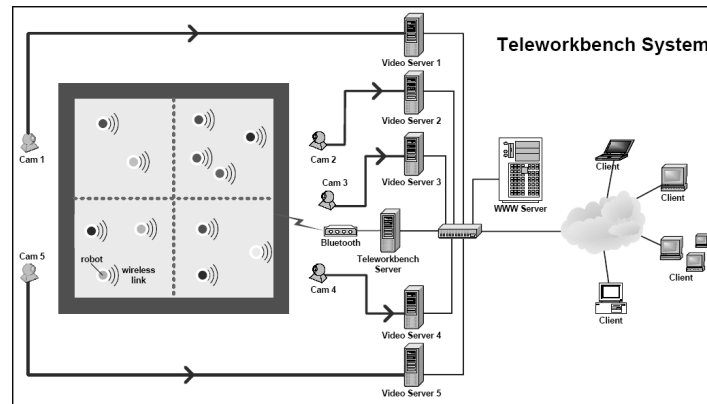


Fig. 1. The diagram of the Teleworkbench system showing the system architecture.

The experiments will be executed on the field which has the resources required as defined by the users during the experiment setup. During the experiment, the web cam above the field will become active and send the video information to the Video Server. This server will process the captured video data to extract any robot on the field and calculate the position and orientation of the robots. To allow the extraction of robot position, every robot has a color mark on top. The detected robots along with their position will be recorded in a log file called "*Position Log File*". Concurrently, this server will also encode the video and stream it to the Video Streaming Server, to provide live video of the experiment.

During experiment, robots can communicate wirelessly with each other or ask the Teleworkbench Server through the Bluetooth module. In some experiments, robots might need to send some internal information for analysis process. Every messages sent by the robot will be recorded in a log file called "*Communication Log File*".

If required by the users, it is also possible to involve some intelligent agents running on separate computers (For security reason, we allow only computers inside our LAN) connected to the Teleworkbench Server. These agents then can communicate or even control the robots during the experiment.

After the experiment is over, the Teleworkbench Server will call the Post-Experiment Tool module to generate the visualization of the experiment. This tool will generate an MPEG-4-based video with some computer generated objects superposed on it. These objects represent some important information needed by the roboticists for analysis purpose. Further detail on this analysis tool will be presented in the following chapter.

3 The Teleworkbench Post-Experiment Analysis Tool

The proposed tool consists of two main parts, which are MPEG-4-based video serving as a user interface and visualization generator. The former is actually the output of the latter.

3.1 The Analysis Tool

Video as User Interface A snapshot of the video is shown in Fig. 3(a). The part in the middle is the video taken by a web cam located above the field where experiments run. In the same area, some computer-generated objects will be superimposed onto the video near the corresponding robot. These objects representing information such as robots' body, communication messages, robots' path, battery level, internal states, and linear camera data.

In the area on the right side, there are two sub-areas, the first sub-area above is the menu area at which users can select the information they want to see at any particular of time. The second one below is the detail-information area in which users can have a more detailed visualization of selected information.

Visualization Generator The block diagram of the visualization generator is shown in Fig. 2. The input for the visualization is the position and communication log file as well as the video monitoring the platform where the experiment is running. The video and position log file are generated by the Robot Positioning system mentioned at the previous section, while the communication log file is provided by the Data Logger. At the other end, the process will produce a video serving as the user interface and the visualizer.

The visualization generator is basically composed of data extractor, scene generator and MPEG-4 scene encoder. In the following paragraphs, a more detailed description of each component will be presented.

Data Extractor Data extractor is responsible to extract information needed by the scene generator. The input data are position and communication log file. The position log file is generated by the robot positioning system, which

calculates the position of all robots taken by the overhead web-cam. Hence, it contains only robot external information. Meanwhile, communication log file captures all exchanged messages among the robots or between the server and the robots. Thus, it preserves robots' internal information.

Position log file provides two important information for the data extractor. The first one is the frame information in the form of frame number and time stamp. This information is required to synchronize the computer generated objects with the input video data. The second information is the position and orientation information in pixel unit. This information is required for drawing some objects at the right position and orientation.

In the case of communication log file, we can basically program the robot to send as various information as possible to provide robot programmers with their desiring information. But for the time being, we support only several information to be visualized, which are infra-red sensors, linear camera, robots' states, and other communicated messages. The last one means all messages which do not belong to any of the first three types of information. To differentiate the messages, we use a specific character as a header for each type of information.

Scene Generator Scene generator is responsible to generate scene description which is required by the MPEG-4 scene encoder for creating a computer-generated animation overlaying the input video data. The scene description is based on XMT-A, which is one of the Extensible Markup Language (XML)-based formats supported by the MPEG-4 standard.

The scene description contains the information describing how the robots move, how the visualization of the sensors' values, the communication messages, and the robot's internal state are varying over time, as well as how users interact with the visualization content.

MPEG-4 Scene Encoder MPEG-4 scene encoder has a function to generate an MPEG-4 file based on the scenery information written in XMT-A format. The heart of this component is the open source software called MP4BOX, a command-line MPEG-4 scene encoder. The output from MP4Box is a video file in MP4 format. There are two reasons that we are using this software. First, it supports many types of multimedia file, e.g. MPEG-4 video and audio, JPEG

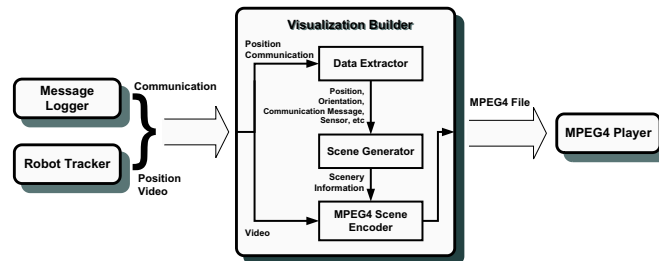


Fig. 2. The process flow of the visualization generator.

and PNG images, AVI, etc, which means we can easily combine many kinds of multimedia file into one file. Second, it is an open source software, which means it is free of charge. For more detail on the GPAC project in general and the MP4BOX in particular, interested readers are referred to the aforementioned reference.

3.2 The Implementation

At present, the data extractor and scene generator are built using C language and run under Linux. However, it will be easy to port these modules to other operating systems. In the case of MPEG-4 scene encoder, we downloaded and installed the GPAC project. From this package we get two important programs, which are MP4BOX and OSMO4. The former is an MPEG-4 scene encoder and the latter is an MPEG-4 player which we use for playing back the output produced by the former.

4 Experiment

In research, models have been developed to simulate the behavior of human crowds in panic and escape situations [9]. The findings gained from such simulations help engineers design escape routes of areas where mass panic situations may occur. When performing such simulations, researchers usually have a global view and are interested in total system behavior. In these simulations people are often modeled as particles. The resulting speed of a particle is calculated in dependence of the desired direction as well as attractive and repulsive forces of other particles and static obstacles like walls. The particles often have global knowledge to make their decisions and are able to sense and move omnidirectionally.

Inspired by the idea of escape simulations for large crowds, we aimed to develop evacuation strategies for multi-robot-systems. Evacuation strategies for robot systems may become necessary when an area must be vacated quickly through a limited number of exits or when a large number of robots have to board a transport robot. However, in difference to the particle-based simulation models above, the individual robots generally do not have global knowledge and cannot sense and move omnidirectionally. Our objective was to develop an evacuation strategy for multi-robot-systems based on simple, robust policies relying only on local sensing and information.

For our implementation and experiments we used the Khepera II robot equipped with our Bluetooth communication module, and the Teleworkbench. Using the Teleworkbench, the setup and execution of an experiment is significantly simplified. Several experiments can be pre-planned and are then automatically executed consecutively by the Teleworkbench. For example, if the effects of different program parameters need to be tested, it is possible to pre-compile

the robot software with the different parameters, transfer those programs to the Teleworkbench and let the Teleworkbench automatically run the tests.

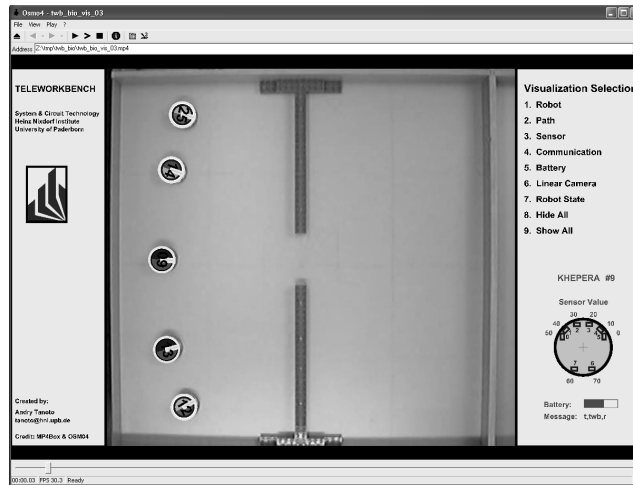
The setup of our experiment is shown in Fig. 3(a). The objective of the robots is to leave the left side of the operational area through the exit in the middle. One major challenge is to avoid mutual blockage and deadlock situations. For perception, the robots only use their integrated infra-red proximity sensors with a range of vision of about 4 cm. In our experiments the approximate direction of the exit is known to the robots. They use the integrated odometry system to keep track of their direction. We implemented a simple, distributed strategy in which robots first try to rush towards the exit individually. When the robots detect an obstacle or another robot, they try to circumnavigate it.

The analysis tool presented in this paper helps the developer in debugging multi-robot systems as it automatically matches and merges internal information from the robots with external information extracted from the recorded film. The result is presented in a single, interactive interface where the developer can fast forward or backward to situations and selectively activate information. Visualization and data summarization enables the developer to quickly grasp a situation, whereas the possibility of selectively activating information helps him analyzing interesting events in depth. The integration in a single user-friendly interface helps the developer concentrating on the analysis of relevant incidents and software debugging.

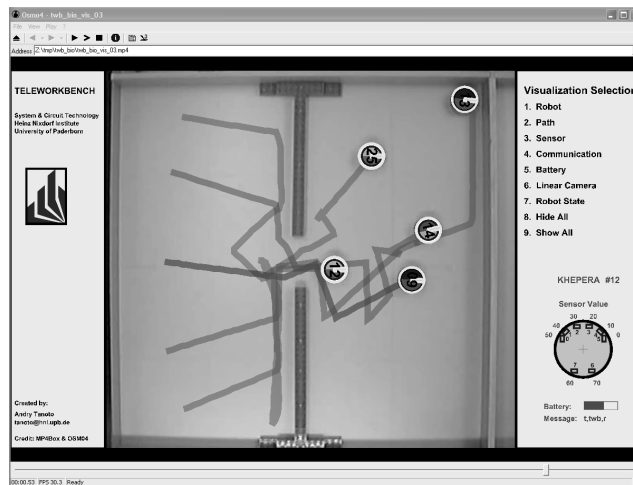
One concrete example for a useful feature in this particular case is the display of the superposed robot trajectories which allows us to analyze the global effects of our evacuation algorithm (see Fig. 3(b)). Furthermore, the presentation of the individual robot paths together with the respective sensor values and internal states in a single analysis window, and the possibility to fast forward and backward to important situations helps us to verify and debug the code as the analysis and explanation of the robot behaviors is significantly simplified.

5 Conclusion

We have described an analysis tool which is part of our Teleworkbench system. This tool can be very beneficial for robot programmers due to its ability to provide a video of the experiment embedded with computer-generated visual objects representing important events and information as well as the robots' behavior during an experiment. The use of MPEG-4 standard is very helpful because it eases the visualization process and gives us flexibility in providing interactivity between users and the content. We have demonstrated the functionality of the analysis tool in a bio-inspired robotic experiment. From the demonstration, we can see how the proposed tool can give us insight on the actual internal and external behavior of robot(s), which is invaluable for debugging.



(a) Setup of the experiment. The robots try to evacuate the left side of the area through the middle gate as quickly as possible.



(b) Analysis of the experiment. Visualization of the superposed robot paths. All robots passed through the gate efficiently except robot number 12.

Fig. 3. Snapshots of the output video of the proposed analysis tool in a bio-inspired experiment involving five mini-robots Khepera II.

References

1. A. Tanoto, U. Witkowski, and U. Rückert, “Teleworkbench: A teleoperated platform for multi-robot experiments,” in *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, Awara-Spa, Fukui, JAPAN, 20 - 22 Sept. 2005.
2. I. The MathWorks, “Matlab user’s guide,” 24 Prime Park Way, Natick, MA 01760, Jan. 1990.
3. SysQuake, “SysQuake.” [Online]. Available: <http://www.k-team.com/software/sysquake.html>
4. E. Sahin and P. Gaudiano, “KITE: The Khepera integrated testing environment,” in *Proceedings of the First International Khepera Workshop, Paderborn, Germany, 1999*, pp. 199–208.
5. A. Breidenfeld, “Behavior engineering for robot teams,” in *Proceedings of Autonomous Minirobots for Reseach and Edutainment (AMiRE 2001)*, Paderborn, 2001.
6. P. Stöhr, “PYKHEP: A python based visualisation toolkit for the khepera robot,” in *Proceedings of the First International Khepera Workshop, Paderborn, Germany, 1999*, pp. 209–218.
7. M. Grosseschallau, U. Witkowski, and U. Rückert, “Low-cost bluetooth communication for the autonomous mobile minirobot khepera,” in *IEEE International Conference on Robotics and Automation - ICRA05*, Barcelona, Spain, 18 - 22 Apr. 2005, pp. 4205–4210.
8. J. L. Du, U. Witkowski, and U. Rückert, “Teleoperation of a mobile autonomous robot using web services,” in *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, Fukui, Japan, 20 - 22 Sept. 2005.
9. D. Helbing, I. Farkas, and T. Vicsek, “Simulating dynamical features of escape panic,” *Nature*, vol. 407, no. 6803, pp. 487–490, Sept. 2000.