Rescue3D: Making rescue simulation attractive to the public

Alexander Kleiner and Moritz Göbelbecker

Institut für Informatik
Universität Freiburg
79110 Freiburg, Germany
{kleiner,goebelb}@informatik.uni-freiburg.de

Abstract. RoboCupRescue Simulation is a large-scale multi-agent simulation of urban disasters where, in order to save lives and minimize damage, rescue teams must effectively cooperate despite sensing and communication limitations. The annually increasing number of teams participating in this league shows clearly that there is a high demand on research in this field.
However, from our experience of participating at RoboCup as a team, but also from organizing RoboCupRescue as a public event [1,2], we learned about two strong limitations that arise practically during the competition: First, the current system offers only limited methods for comparing specific abilities of rescue teams. Second, the current presentation of the competition is only limited understandable for spectators. Within our effort in developing a new visualization of the rescue domain, we want to focus on these two limitations. We introduce a system for visualization that covers the demands of both developers and spectators.

1 Introduction

RoboCupRescue simulation has been part of the RoboCup competition [4] for four years now. The annually increasing number of teams participating in this league shows clearly that there is a high demand on research in this field.

However, from our experience of participating at RoboCup as a team, but also from organizing RoboCupRescue as a public event [1,2], we learned about two strong limitations that arise practically during the competition[1]:

Limited Team Comparison Due to the fact that the total score of a team of agents is based on the outcome of multiple tasks, such as path planning, exploration, target selection and team coordination, it seems, especially in the case of equal teams, dissatisfactory to compare them by the total score only. Particularly at run time it turns out to be difficult to get an idea about the performance of a team. Virtually, the team performance is observed by the group-extinguish action, whereas the percentage of

[1] These limitations mainly exist with the current official competition viewer. Former viewers, such as the viewer from Kuwata [6], were basically developed into the same direction as the viewer proposed in this paper.

discovered civilians or the percentage of cleared roads is seldom eye-catching to the spectator.

Hence we argue that it is very important to visualize statistics, such as the number of civilians found or the percentage of cleared roads, in order to shape a team's specialty but also in order of make watching a run even more exciting.

Accessibility to the public The current competition viewer provides an excellent visualization of the overall behavior of multiple agents. Though optimal for the agent programmer, it seems to be hardly possible for spectators to understand the complex agent-relations in real-time. As we noticed, most spectators even have problems to recognize the 2D visualization as a map of a city.

Within our effort in developing a new visualization of the rescue domain, we want to focus on these two limitations. We introduce a system for visualization that covers the demands of both developers and spectators.

The remainder of this paper is structured as follows. Section 2 provides an structural overview of the proposed viewer system, whereas section 3 discusses its implementation. In sections 4, 5 and 6 new features for scene selection and visualization of statistics and graphics are discussed. Finally we summarize the paper in section 7.

2 System overview

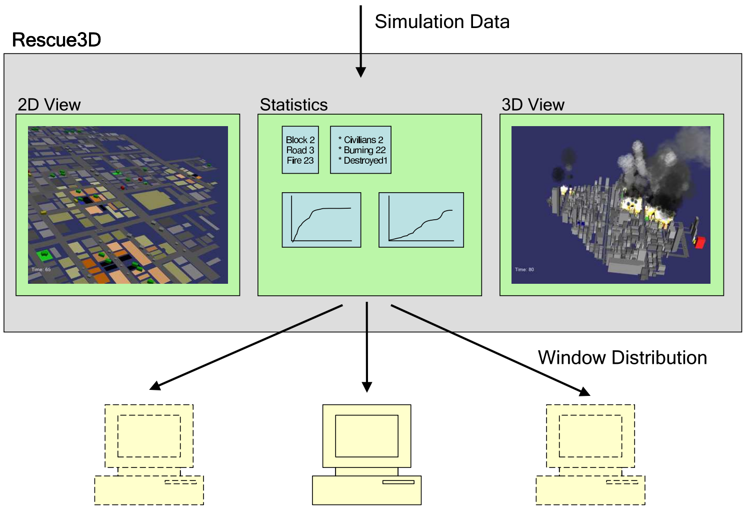


Fig. 1. An overview on the proposed viewer's capabilities: simultaneous scene rendering in 2D and 3D, and visualization of statistics on one or multiple screens, respectively

We believe that the current limitations mentioned in the introduction can only be relaxed by offering a 2D and 3D visualization and also statistical information of the team performance at the same time. Therefore the introduced viewer system is designed to create multiple views of the simulation (see figure 1), which might either be located on one screen or being distributed among many of them[2]. Additionally it will be supported to merge multiple views into one window by the technology of Head Up Displays (HUDs). Then, for example, one might have a large 3D visualization of the scene on the full screen and a small 2D visualization in the lower left corner, similar as known from computer games, such as Microsoft's Age of Empires. The number of views and whether they are merged into one window or distributed, can be defined within a configuration file. At any time and mode of the visualization, the scene currently rendered on the 3D screen will be indicated within the 2D view.

3 Implementation

The implementation of the viewer is based on the Open Scene Graph (OSG) library which is a C++ open source project for scene graph rendering, founded 1998 by Don Burns [3], and is based on *OpenGL*. The viewer project benefits from the library's fast rendering engine and portability. Particularly the latter allows it to successfully run the viewer under Linux as well as under Windows XP.

The current release of the viewer, available under the projects home page [5], supports the visualization of a simulation run from log files of any map, such as Kobe, Foligno or VC, and will soon also support online visualization via a connection to the kernel.

4 Scene selection

In order to make rescue simulation attractive to the public, it is important but also difficult, to select interesting scenes, such as a group of fire fighters extinguishing a fire or a team of ambulances rescuing civilians, for visualizing an appropriate camera perspective within the 3D window. Kuwata's viewer already implemented the possibility to select different perspectives, such as an agent-centered perspective [6]. We are currently working on mechanisms that automatically detect interesting actions, such as group extinguishing. However, we believe that with the information currently provided by the kernel to the viewers, the visualization of the agent's actions by automatically recognizing their plans, will always be one step behind. Therefore it will be once necessary that the agents inform the viewer about their intentions, i.e. which are their targets, at least on a group level.

At the current stage we intend to simplify the problem by providing the possibility to manually select agents and positions on the 2D map which will be successively rendered on the 3D window. Furthermore the selected perspective, precisely the camera's field of view, will be marked on the 2D view. By this it will be possible for the session

[2] Note that the distribution of various windows from a single application over more than one screen can be handled by modern graphics cards.

chair, or any member of the team, to emphasize the problems solved by the agents to the spectator. From a selected 2D pose (x, y, α) the right 3D pose $(x, y, z, \alpha, \beta, \gamma)$ will automatically be calculated, i.e. a perspective that shows a complete scene without buildings or agents in the way[3].

5 Visualization of statistics

The visualization of statistics is currently under development. Basically, the idea is to show the progress of key values that indicate the performance of the agents over time. In order to compare these values, all visualizations of statistics will allow to draw the data of the current run in comparison to all previous runs or in comparison to the best run of this round only. This feature, which one can enable or disable in the viewer's configuration file, will make watching runs much more interesting, since one gets a deeper comparison between competing teams. We intend to visualize the following information:

- Performance of the police forces
  - The percentage of blocked agents over time
  - The percentage of completely blocked roads over time
  - The percentage of partially blocked roads over time
- Performance of the ambulance teams
  - Percentage of dead agents over time
  - Distribution of civilian damage (chart)
- Performance of the fire brigades
  - Percentage of destroyed buildings over time
  - Percentage of saved (extinguished) buildings over time
  - Distribution of building fieryness (pie chart)
- Exploration
  - Percentage of found civilians over time
  - Percentage of explored buildings over time (i.e. how many building centers have been once in the sight of an agent)
  - Exploration coordination (i.e. how many buildings have been visited more often than necessary?)
- Path planning
  - How many times did agents run into blockades?
  - How optimal were the executed plans, i.e. by which percentage did they differ from optimal plans, given perfect knowledge of the world?

6 Graphics rendering

At the current stage, the viewer is able to render the animation of agents in three dimensions (see figure 2(a)) or two dimensions by projecting the 3D scene into the plane ((see figure 3(a)). Due to the high frame rate that is possible with the OSG engine and

[3] Note that α, β and γ denote the 3D rotations, also known as the roll-pitch-yaw angles.

a 3D graphics board, movements of the agents are easily traceable by the observer. The system enables the visualization of fires and smoke by *particle systems* (see figure 2(b)) and the usage of textures (see figure 3(b)) is supported. The visualization of other maps, such as that one of Foligno City or Virtual City, is possible without any additional changes (see figure 4). Furthermore it is possible to display the team name and the current time and score at each step of the simulation.

We are currently working on improving the 3D visualization by the following features:

- Animation of actions, such as extinguishing fires or clearing roads
- Displaying text that comments the actions currently taken by agents in the rendered scene
- Markings above buildings that indicate what and how many agents are inside
- The possibility of making building walls invisible for a selected position and radius in order to allow the observation of actions taking place around or within buildings

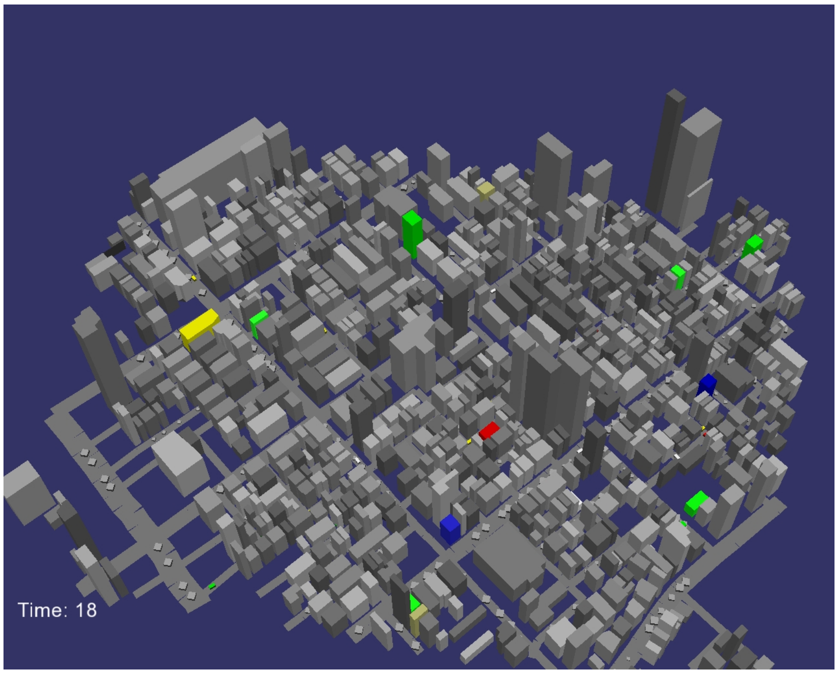
7 Summary

Everyone ever participating in the RoboCupRescue simulation league is aware of the fun but also the significance and challenges of this work one has to solve while implementing multi-agent behaviors under uncertainty and real-time. However, as we already stated in the introduction, current limitations make it hardly possible for spectators, and sometimes even for participants, to appreciate the performance of a rescue team. This is contrary to any other league at RoboCup and can only be changed by continously improving the visualization.

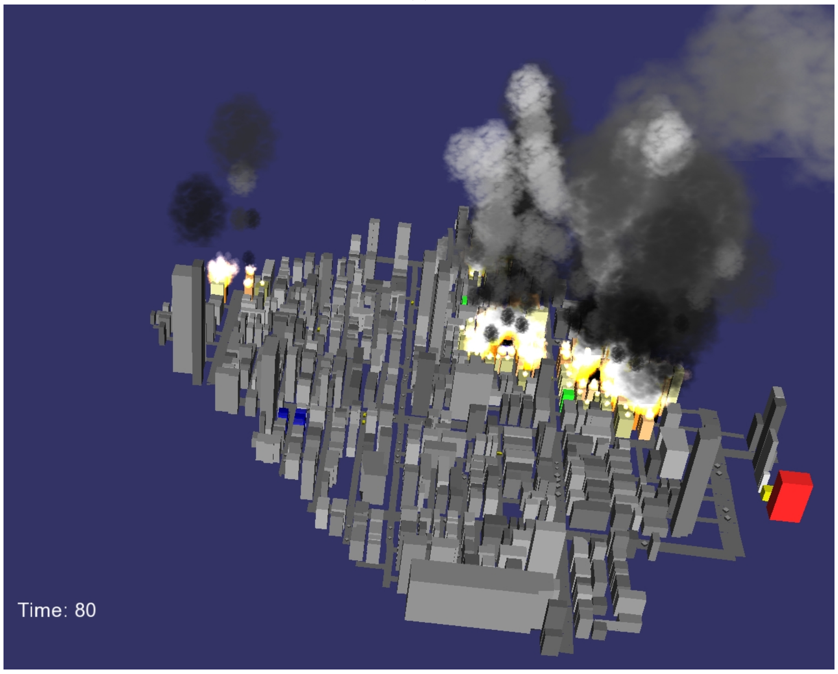
We hope that our proposal for an open source viewer system is one step towards making RoboCupRescue simulation accessible to the public and also that in the future more people will contribute to the viewer project.

References

1. Robocup german open 2003. `http://ais.gmd.de/GO/2003/`.
2. Robocup german open 2004. `http://ais.gmd.de/GO/2004`.
3. Don Burns. Open scene graph. `http://openscenegraph.sourceforge.net/main.html`.
4. H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. RoboCup Rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE Conf. on Man, Systems, and Cybernetics(SMC-99)*, 1999.
5. A. Kleiner and M. Goebelbecker. Rescue3d homepage. `http://kaspar.informatik.uni-freiburg.de/~rescue3D`.
6. Yoshitaka Kuwata. How to use robocup-rescue logviewer. 2001.

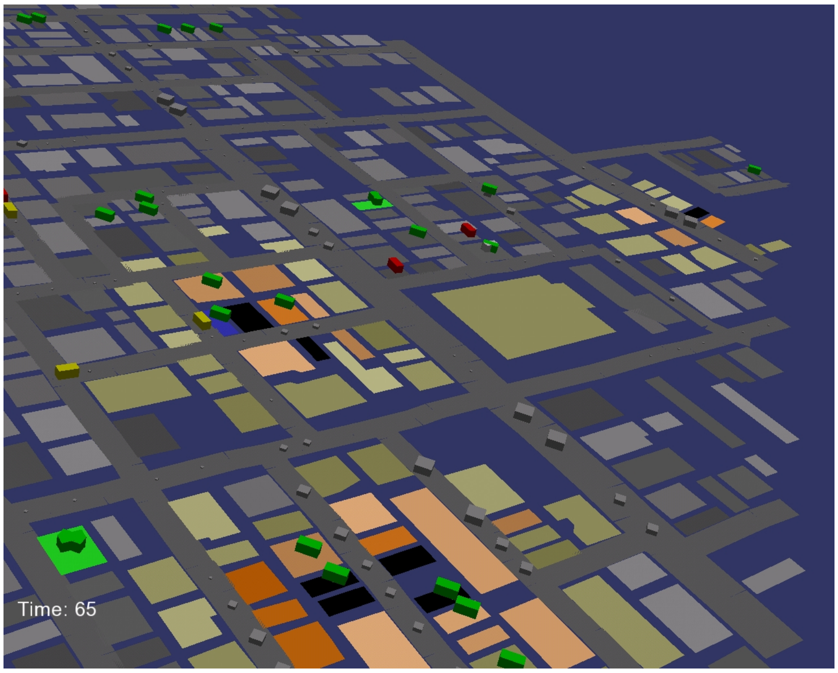


(a)

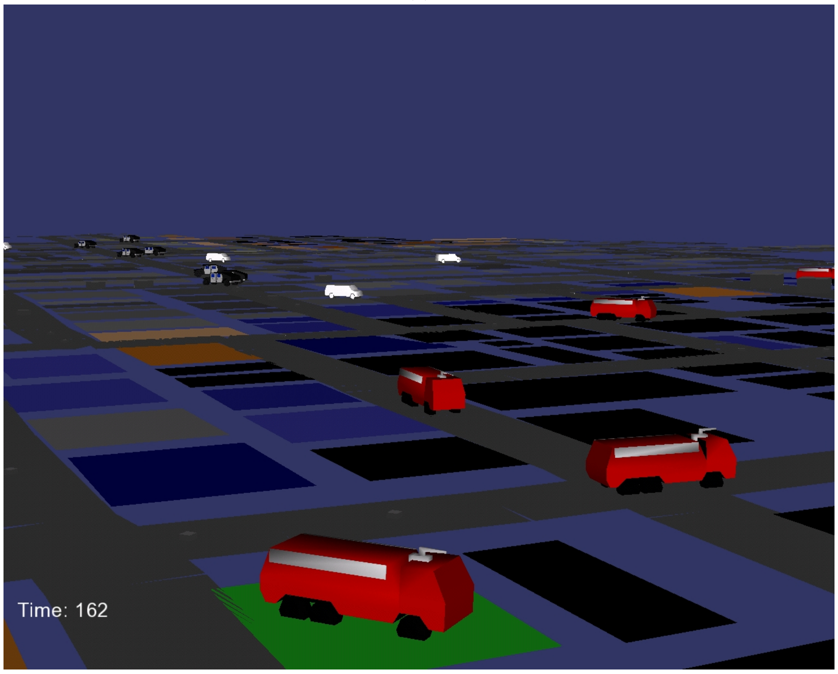


(b)

Fig. 2. Sub figure (a): 3D visualization of the Kobe City, Japan. Green buildings indicate refugees, red buildings fire stations, white buildings ambulance stations and yellow buildings police offices. Sub figure (b): Simulation and visualization of fire in the same city. Blue buildings indicate extinguished fires, whereas the colors from yellow via orange to black indicate the fieryness of a building.



(a)



(b)

Fig. 3. A two dimensional visualization of Kobe (a) and a visualization of objects with textures (b).

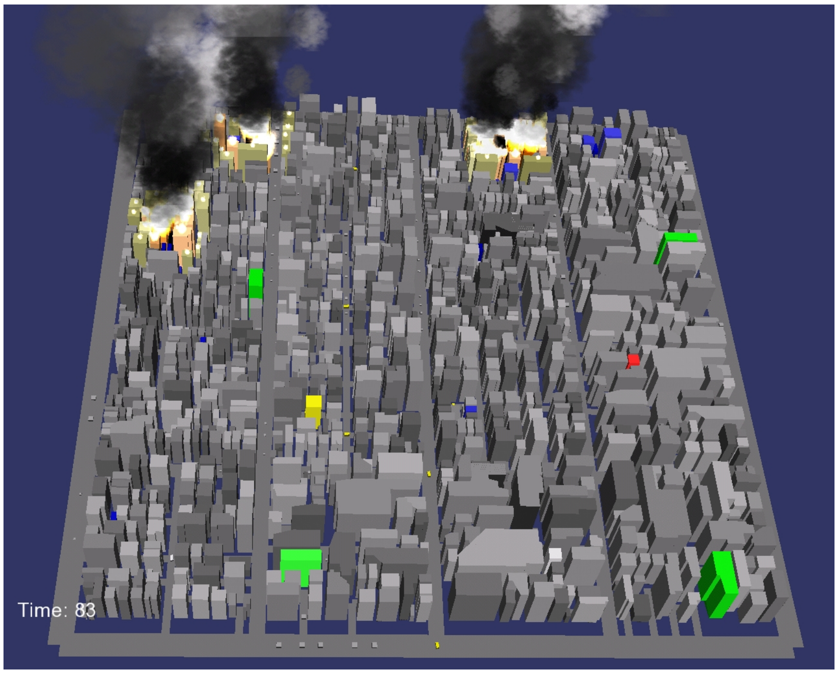


Fig. 4. Visualization of *Virtual City* (VC)