RoboCupRescue 2013 – Rescue Simulation League

Infrastructure Competition
Team Description
<AUT S.O.S. (Iran)>

SalimMalakouti, Seyed, Yoosef Golshahi, Hesam Hosseini, Mohammad Reza Modaresi, Sina Sheikholeslami, Hesam Akbari

Rescue Simulation Lab, Department of
Computer Engineering and Information Technology,
Amirkabir University of Technology,
No. 424, Hafez Ave.,
Tehran, Iran

{salim.malakouti yoosef.golshahi}@gmail.com, rht1369@yahoo.com, modaresi@aut.ac.ir, ssheikholeslami@live.com, hitax13@gmail.com

http://ceit.aut.ac.ir/~sos

Abstract. Robocup Rescue Agent Simulation league has been a part of Robocup competitions since 2001. The aim of the league is to provide a research environment for search and rescue problems during disasters. Rescue Agent Simulation uses various simulators to simulate a wide variety of world and human behaviors. Despite all efforts that have been put into the league, we believe there is still a big gap between decision-making in Rescue Agent Simulation environment and in real world problems. The output of the teams' works will not be applicable unless the simulated environment changes in a way that mimics the reality more precisely.

In this paper we propose the idea of a 3D Server in the Rescue Agent Simulation league as the first step to design a more realistic simulator. In our proposed 3D Server, we have modeled disaster environment in 3D and introduced new 3D entities such as tunnels and bridges.

1 Introduction

Robocup Rescue Agent Simulation league has been a part of Robocup competitions since 2001. The aim of the league is to provide a research environment for search and rescue problem during disasters (Tadokoro, et al., 2000). The system has been designed to be distributed and it utilizes heterogeneous agents. Some of its features make it ideal as a test bench for multi-agent research, such as dynamic environment (fires spread; victims get injured as buildings collapse), limited information (agents can only see a short distance), uncertain information (agents do not see the exact state of the world) limited communication (messages can be dropped or have noise), and limited processing time (agents may have a time limit to issue commands) (Skinner & Ramchurn, 2010).

The first implementation of the platform modeled a city as a graph in which roads were edges of the graph and intersections were vertices. In March 2009, the league committee decided to add some more complexity to the league and make the server more modular and extendable (Skinner & Ramchurn, 2010). Major new features of the new platform designed in 2009 can be listed as follows:

- Redesign of the core system to make it easy to extend.
- Standardization of configuration so that kernel, simulators and agents can share single configuration.
- Giving the users the ability to interact with the simulation by creating Implementations of human-controlled agents.
- Adding Geometrical complexities such as 2D areas as roads, which required new redesign of some simulators.

As it can be seen the objectives of the redesign process in 2009 were to make the server more efficient and modular. However, we believe the current server is still far behind the real world. We believe it is excessively significant to take a step further and design a new 3D world model among 3D simulators. This step is vital for the league for two main reasons:

1- As one of the most influential teams in the league, S.O.S. has been active in this league since 2003. During past years we have felt that the league lacks dynamism, and as a result there is not much difference in the strategies used by teams from one year to the other. The main reason for this issue is that this 2D server has been utilized since 2001 and there has been sufficient researches going on development of strategies in 2D environment. The 3D server arises new problems in the league that participants have not faced yet.

2- The Robocup Federation has described the first aim of Robocup Rescue Agent Simulation league as “*to develop simulators that form the infrastructure of the simulation system and emulate realistic phenomena predominant in disasters*” (RoboCup Rescue, 2006). Our goal is to introduce some new features to the league that will make us one step closer to the aim of the league. A realistic look into the current features of the league makes it clear that the results are not completely applicable to real world problem. Current version seems to be too far away from reality that even in countries such as Iran, which have numerous cities threaten by earthquake, governments and companies are not convinced to invest in the field.

This paper introduces the 3D Server project. We define a new 3D world model among changing current 2 dimensional simulators in order to satisfy needs of a 3D environment. We also add new 3D features to the world model entities and some brand new entities will be added to the server. Bridges and Tunnels will be the most important examples. Furthermore, the details of the new 3D simulators are explained in proceeding section.

2 3D World Model

In addition to 3D shapes such as building we added a 3D grid to all 3D object such as buildings. These 3D grids are added for two main reasons:

- Simplifying referencing algorithm in order to reduce computation time.
- Adding more details to the server (for example new fiery building model represented in section 3.5).

We also decided to add following entities and features to the world model since we believe they play a key role in determining city's condition after disaster. These features include:

- Bridges
- Tunnels
- Underground floors

2.1 CityGML

CityGML is used as the main file model of our proposed 3D design. CityGML is an information model for representation of 3D urban objects; and provides five different levels of details (LOD) that can be used simultaneously:

- **LOD0 -** regional, landscape
- **LOD1 -** city, region
- **LOD2 -** city districts, projects
- **LOD3 -** architectural models (outside), landmarks
- **LOD4 -** architectural models (interior)

At LOD0 it provides Simple 2D coordinates of city objects similar to simple GML. At LOD1, it provides prismatic 3D buildings similar to cubes and cuboids. At LOD2, which we decided to use for current version of our 3D Server, CityGML provides geometric information of outdoor shape of objects. Detailed structure information in three dimensions of buildings, roads, bridges and etc. are supported in this LOD. In contrast to previous LODs, a building in LOD2 has differentiated roof structures and thematically differentiated boundary surfaces. CityGML supports textures and more sophisticated semantic information of objects. It also provides outdoor city furniture in this LOD. At last, in LOD 4 it support indoor geometric information and furniture objects of buildings. We believe LOD 4 is a key feature of CityGML, which facilitates the process of merging two Rescue Simulation leagues in future.

CityGML also had features that made it superior to its competitors like KML, which is support by Google Maps. CityGML fully supports geometric and semantic information of urban objects. However, KML seemed to lack semantic information and appeared to support less sophisticated geometric information. CityGML and KML are compared in Table 1.

	KML	**CityGML**
Geometry	Basic	Sophisticated
Geo-referencing	**Sophisticated**	Very Sophisticated
Appearance	Basic	**Sophisticated**
Topology	No supported	**Sophisticated**
Semantics	Not supported	**Very Sophisticated**
Embedding	Very Sophisticated	**Very Sophisticated**

Table 1 Comparison of CityGML and KML

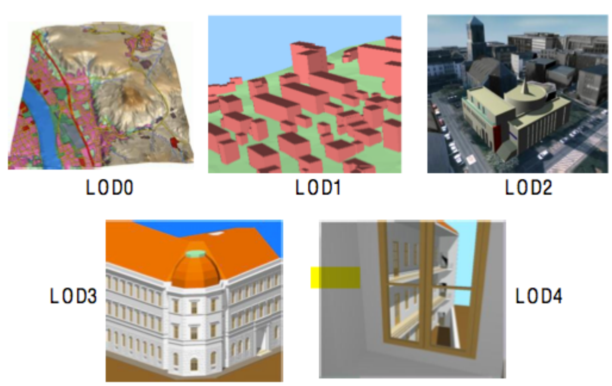


Figure 1 CityGML Levels Of Details

3 Simulators and Agents

3.1 3D Line of Sight

We proposed a new 3D model for line of sigh perception. Similar to current line of sight model agents are able to sense everything in line of sight. However in order to make it more realistic we decided to add noise to line of sight. In our model, agents' sense ability may be noisy due to the distance of objects to agents. In reality both humans and robots sense of the environment may become inadequate when objects are too far away. In our model, the amount noise applied to agents' sense is relevant to the object's distance to the agent. Additionally, in the current server, agents' line of sight may be interfered by the surround buildings. But, in our 3D model this has become more realistic since buildings' height is taken into consideration.

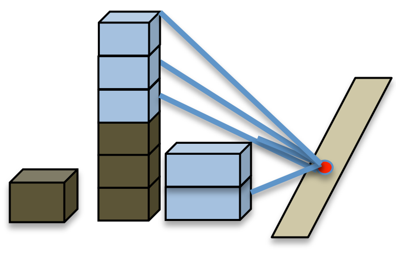


Figure 2 3D noise

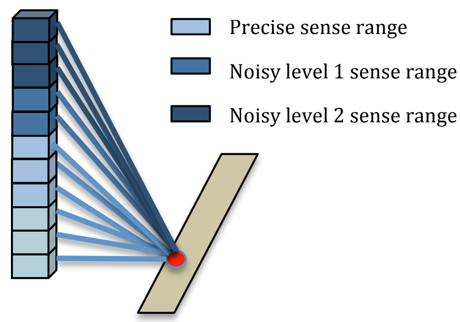


Figure 3 Different noise levels' sense

3.2 Collapse Simulator

In our proposed collapse simulator we considered many realistic parameters of buildings' collapse. We also designed a new 3D blockade entity, which is more relevance to the materials and properties of the collapsed building. The new collapse model is based on the models introduced in K.Meguro and M.Hakuno's paper [5], M.Kimiro and S.Tadayuki's work [6] and X.Lu's paper [7]. We also proposed a new collapse model for fiery buildings based on Li Yi's [8] and Dat Duthinh's papers [9], which consider damages of fire to buildings structure. In our models collapse is not only happened due to random variables and in random shapes, but, based on more realistic structural parameters of the building and according to buildings material and shape. Additionally, in order to make the simulation more realistic and with regard to buildings' 3D shapes in our model, collapsing affects their shape and structure which also affects agents' sense.

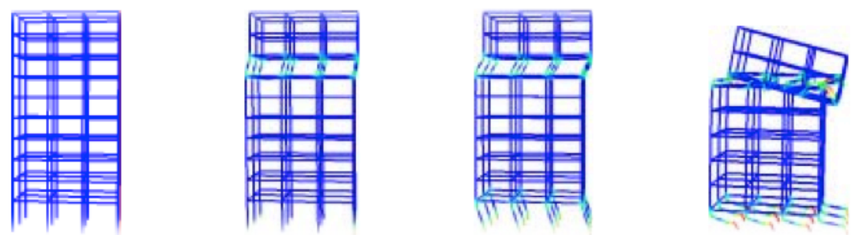


Figure 3 The pressure on beams is the main reason of buildings' collapse

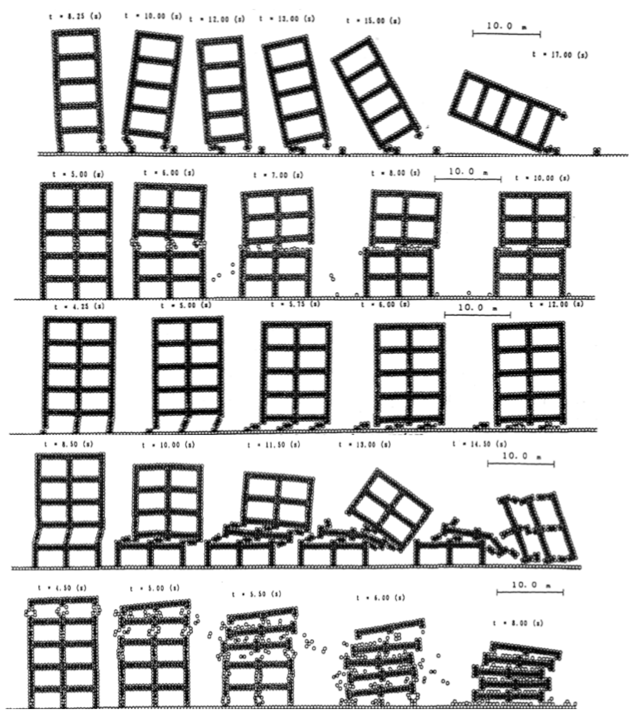


Figure 4 Simulation results for collapse models in [5]

```
ufp = 0                        // upper floor's pressure
ufd = new Vector(0,0,0)  // upper floor's pressure direction
collapse( Building b)
     for f in b.floors
       totalPressure = 0
       totalVector = new Vector(0,0,0);
       for beam in f.beams
               beamPressure = getRandomPressure()      // pressure on the floors
               pressureDirection = getRandomDirection()    // direction which this
               totalPressure += beamPressure                    // floors is collapsing
               totalVector += pressureDirection

       f.setCollapseParams(totalPressure+ufp,totalVector+uld)
       ufp+=totalPressure*alpha           // alpha is a coefficient
       ufd += totalVector
     collapseFloors(b)                      // checks if the pressure is above a threshold
                                            // collapse the floor in the direction. Collapse starts
                                             // from the bove floors going to the lower ones
```

pseudocode 1: pseudocode for generating blockades according supporting models represented in figure 3

We also propose the following method for collapse of bridges and tunnels. In this method bridges and tunnels are assumed to section (called decks in case of a bridge), which may collapse separately.

```
collapse( Bridge b)
   if( collapeMakesMapUnreachable(b))
       return;
     For d in b.decks
       d.force = randomForce()
       if(d.force>THRESHOLD)
               collapseDeck(d);
collapse(Deck d)
     makeAreaNotPassable(d)                    // clearing collapsed parts of
                                               // buildings does not make
                                               // The m passable
     createBlockades(d)                        // similar to current
                                               //method but 3D
```

Source-code 2 Source-code for collapsing bridges and tunnels

We also simulated collapse of buildings due to their fieriness and considering fire's damage to their structure. According to researches done by Dat Duthinh and Li Yi [9] fire makes beams' shape change causing buildings to collapse.

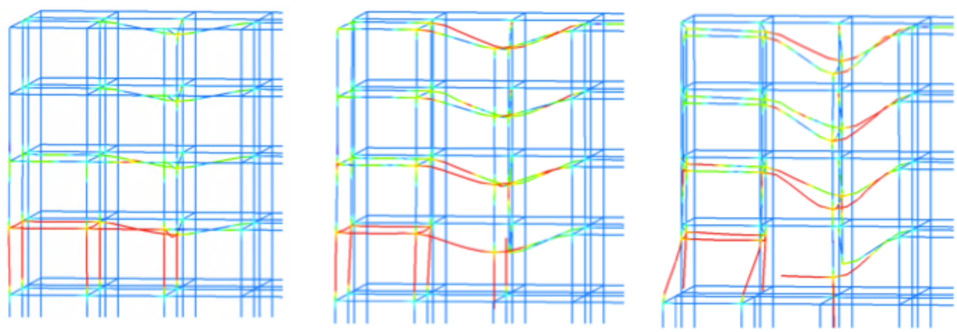


Figure 5 Damage of fire to building beams

3.3 Simplifying Blockade Models

Our suggested collapse model generates various sorts of blockades consisted of numerous parts since it is created by destruction of building structure. However, this model results in a very sophisticated blockade model requiring high computing powers for both agents and simulators. Therefore, in this section we propose a simplified blockade model, which is created due to collapsing direction, presented in pseudocode 1 and the volume of collapsed material.

Simplified blockades are represented by the following information:

- Apexes of the bottom surface
- Slope of the upper surface
- Grids to simplify implementation of clear method
- Volume that is computed according to the volume of collapsed floors' materials.

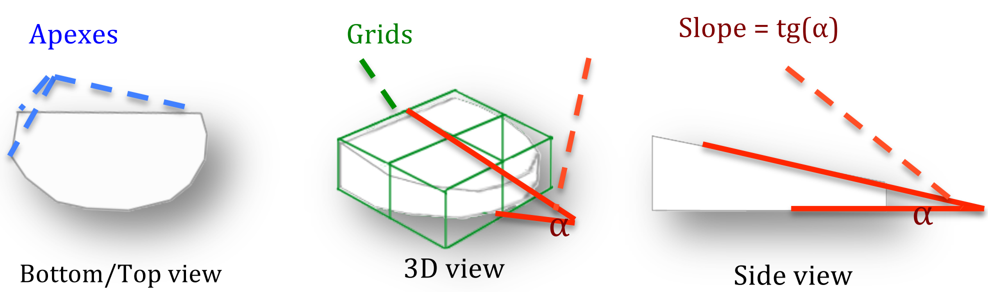


Figure 6 Three different views of simplified blockade model

3.4 Clear Method

The current clear method seems far away from the reality. Currently, clear method shrinks blockades from all sides. We propose a new clear method that is more realistic and suits our proposed blockade model more logically.

In our proposed clear method agents choose a path to clear instead of a single blockade. Clear simulator clears the path in form of a thick line until it reaches agent's clearing capability for each cycle. Clearing blockades consists of changing its bottom surface apexes and volume and is demonstrated in figure 7. This new clear method has the following advantages:

- Its more realistic
- Police force agents' positioning will become important
- Teams may have to optimize their clearing strategies which is significant in real world

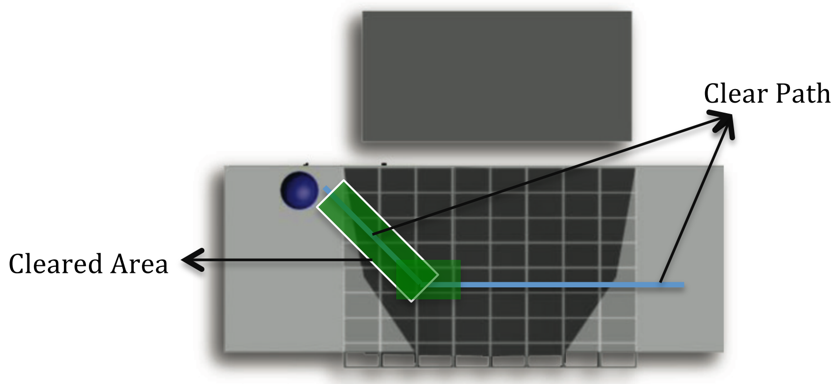


Figure 7 Clear method

3.5 Fire Simulator

The 3D Fire Simulator is going to be similar to the 2D simulator except the fact that world's grids would now become a 3D matrix as shown in figure 8.

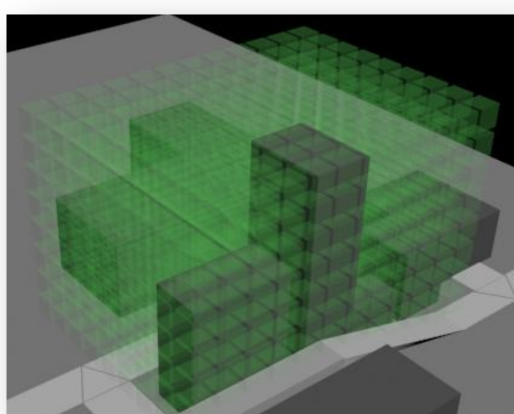
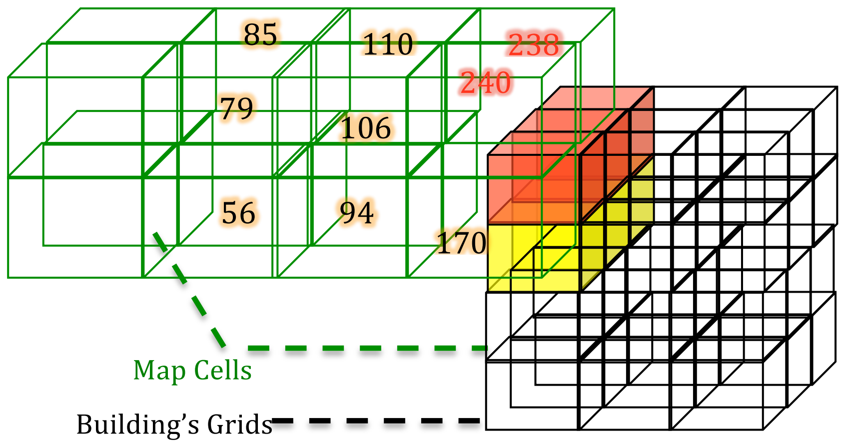


Figure 8 A part of world's 3D grids

The important difference of the 2D and 3D fire simulator is that in 3D fire simulator buildings are not threated as a single object. Building's each grid may become fiery separately. Grids' temperature also affects its neighbors. We believe our 3D model is much more realistic since in real world building floor go on fire separately and this has an enormous effect on the strategies of controlling fire zones. Our model is demonstrated in figure 9.



4. References

[1] S. Tadokoro, H. Kitano, T. Takahashi, I. Noda, H. Matsubara, A. Shin Joh, T. Koto, I. Takeuchi, H. Takahashi, F. Matsuno, M. Hatayamm, J. Nobe and S. Shimada, "The RoboCup-Rescue Project: A Robotic Approach to the Disaster Mitigation Problem," in *Proceedings of the 2000 IEEE lntemational Conference on Robotics & Automation*, San Francisco, CA, 2000.

[2] C. Skinner and S. Ramchurn, "The RoboCup Rescue Simulation Platform," in 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, 2010.
Building clustering for select fire zone

[3] "RoboCup Rescue," Robocup Foundation, 2006. [Online]. Available: http://www.robocuprescue.org/agentsim.html. [Accessed Feb 2013].

[4] "CityGML OGC 12-09 ," CityGML Official website. [Online]. Available: https://portal.opengeospatial.org/files/?artifact_id=47842. [Accessed Feb 2013].

[5] K.Meguro and M.Hakuno.1992.Simulation of collapse of structures due to earthquakes using the extended distinct element model. Earthquake Engineering. Tenth World conference. 1992 Balkema,Rotterdom

[6] M.Kimiro and S.Tadayuki. 1998.Simulation of Collapse of Structures Due to the 1995 Great Hanshin Earhquake. Eleventh world conference on Earthquake Engineering

[7] X.Lu and others. 2008. Numerical Simulation for the Progressive Collapse of Concrete Building due to Earthquake. World Conference on Earthquake Engineering.Beijing

[8] Li Yi and others. 2012.NUMERCIAL MODELS OF FIRE INDUCED PROGRESSIVE COLLAPSE ANALYSIS FOR REINFORCED CONCRETE FRAME STRUCTURES. ENGINEERING MECHANICS, vol 29(4):06-103.

[9] Dat Duthinh and others. 2008. Recent advances in fire–structure analysis. Science Direct, Fire Safety Journal:161-167