

Towards more realistic indoor environments for the Virtual Robot Competition

Francesco Amigoni, Matteo Luperto, and Alberto Quattrini Li*

Artificial Intelligence and Robotics Laboratory, Politecnico di Milano, Italy
{francesco.amigoni,matteo.luperto,alberto.quattrini}@polimi.it

Abstract. Simulations could make testing of mobile robot systems less strenuous in terms of time spent for setting scenarios and performing experiments. The USARSim simulator used in the Virtual Robot Competition has shown a good degree of realism in modeling the laws of physics that influence the behavior of robot sensors and actuators. However, the environments in which the robots of the Virtual Robot Competition move are not always realistic. These environments are randomly generated by a tool called World Generator and then manually adjusted. In this paper, we propose a system that exploits models of different real buildings for informing the automatic generation of more realistic indoor environments. The proposed system allows to improve the significance and the generality of simulated experiments.

Keywords: simulation, USARSim, building type, Virtual Robot Competition

1 Introduction

In the last years, autonomous mobile robotics has been spreading in several applications, especially in tasks that are difficult or dangerous for humans, like exploration and search and rescue. Good experimental methodologies are necessary to foster research and to ease the transfer of existing research results to market products, so that robots can be safely deployed in real scenarios. Currently, there is a lively debate in the autonomous robotics community on finding standard protocols to test robotic systems and, while some globally accepted benchmarks and well-established metrics for autonomous mobile robot systems are emerging, a general lack of sound methodologies has been recognized [1, 2].

In applications like search and rescue, setting real scenarios for robot testing could be very expensive in terms of effort, cost, and time. Also using existing facilities can be expensive in terms of traveling costs. In such cases, simulations could be a viable solution [3]. One of the important aspects considered for the validation of a simulator is its *physical realism*, namely how accurately the

* Authors are members of PoAReT team (<http://home.deib.polimi.it/amigoni/research/PoAReT.html>). This is the team description paper of the PoAReT entry for the 2014 Infrastructure Competition of the RoboCup Rescue Simulation League.

laws of physics that influence the behavior of robot sensors and actuators are represented. In this sense, USARSim¹ [4] is a realistic and reliable 3D robot simulator, employed, in addition to other uses, in the Virtual Robot Competition² of the Robocup Rescue Simulation League. This simulator has been quantitatively validated in terms of fidelity of behavior of robot sensors and actuators, by comparing the results obtained in real scenarios and in the corresponding simulated ones [5].

However, there is another important aspect that is usually neglected for validating a simulation: the plausibility of the simulated environment, namely the resemblance between the simulated environment and an analogous real one. We call this aspect *structural realism*. Some data sets acquired in real environments could be exploited for testing, like those of [6]; however their limited number and the relative sparsity of the collected data could represent an obstacle to their extensive use. Another option is the use of tools to automatically generate simulated environments. For example, within USARSim, there exists a tool for automatically building indoor environments, called World Generator and developed by NIST [7].

In this paper, we address the problem of generating realistic indoor environments. Specifically, we exploit the models found by analyzing different building types in [8] in order to extend the NIST World Generator tool. Our goal is to automatically generate more realistic indoor environments to be used both for testing mobile robot systems in USARSim and for making the Virtual Robot Competition more stimulating. In this way, the USARSim experiments performed to test autonomous mobile robot systems can acquire more significance, as there is a realism not only on aspects connected to the laws of physics (physical realism), but also on the environments in which the robots are deployed (structural realism or *structural plausibility*). Indeed, the results obtained with realistic simulated experiments could be more easily generalized to real settings, and the porting of a method tested in simulation to real robots could be facilitated. Last but not the least, with an increased realism, the Virtual Robot Competition (based on USARSim) could constitute a better benchmark for mobile robot systems.

This paper is structured as follows. The following section describes how simulated environments are currently generated in the context of the Virtual Robot Competition. Section 3 illustrates the models that we use in the proposed system for generating indoor environments. Section 4 motivates the development of the proposed system by comparing some features of an automatically generated environment, used during the last Virtual Robot Competition, with those of some real counterparts. Section 5 describes our proposed solution, while Section 6 presents its experimental analysis. Section 7 concludes the paper and discusses some possible future works.

¹ <http://sourceforge.net/projects/usarsim/>

² <http://www.robocuprescue.org/wiki/index.php?title=VRCompetitions>

2 Environment generation for the Virtual Robot Competition

The Virtual Robot Competition within the RoboCup Rescue Simulation League consists of a simulation of multiple mobile robots that are positioned in an initially unknown environment and have to explore it in order to find victims in a given time interval.

The simulation is run in USARSim, which is a realistic and reliable 3D robot simulator based on Unreal Tournament engine that provides different robot and sensor models. In the competition, in addition to some outdoor environments, several indoor environments are considered (see Fig. 1 for an example), mimicking search and rescue in real scenarios.



Fig. 1: The indoor environment used in the first final round of 2013 Virtual Robot Competition.

The creation of such indoor maps is performed within UDK (Unreal Development Kit)³ and involves several steps: firstly, the map designer draws the floor plan by defining individual rooms; secondly, (s)he places doorways; thirdly (s)he applies textures to the surfaces and lightning to the rooms; and, finally, movable and static objects, like chairs and desks, are placed in the environment.

As far as we know, the above steps to generate maps were manually performed, but, since 2010, the maps used in the competition have been automatically generated by a tool called *World Generator* developed by NIST [7] (see Fig. 2 for a screenshot of the user interface). With such tool, indoor environments are easily generated saving a lot of time and resources.

³ <http://www.unrealengine.com/udk/>

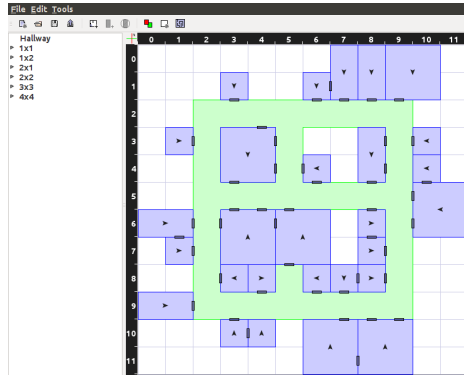


Fig. 2: Screenshot of graphical user interface of NIST World Generator.

The tool is able to generate random environments of a size specified by the user. Moreover, the user can specify different features of the map, like hallway widths or ramp slopes, by appropriately filling a configuration file. According to these settings, the algorithm creates a Manhattan-style grid and randomly fills it with rooms. Finally, it places doors between adjacent rooms/hallways, by applying some rules for constraining the connections, like placing at least one door for each room or removing doors between a room and a hallway when they are over-connected. Moreover, the tool can generate the building interior by using a pre-furnished rooms library.

Once the map has been generated, the technical committee of the Virtual Robot Competition makes some manual adjustments, so that some desired properties are satisfied (e.g., adding obstacles, so that a robot should follow a specific trajectory to reach a certain point).

This tool for generating maps has had a great impact on the Virtual Robot Competition, because the process of building maps has been simplified. However, as we show in Section 4, the generated environments do not resemble realistic indoor environments, as they are randomly created.

A recent work [9] extended the NIST World Generator by embedding a difficulty measure that can be set by the user to influence the map generation. The idea is that, depending on some of its characteristics, the environment is given a difficulty value related to how easily a robot can navigate in it and build its map. So, rooms and hallways are placed according to the selected difficulty value. For example, the authors consider features as the distance between hallways (the larger the distance, the more difficult the navigation) and the doorway probability (the lower the doorway probability, the lower the difficulty of map building). This extension to the NIST World Generator is of interest for performing different sets of experiments on maps of different difficulty levels. Nevertheless, the structure of the generated environments could be not realistic and the obtained results not easily generalizable to real-world settings.

3 Models for real-world indoor environments

Despite the fact that every building is unique (it is hard to find two different independent buildings that have the same floor plan), each building is immediately classifiable by a human being in a specific category, its *building type*. A building type (e.g., School, Office, House) represents the function of a building, namely the purpose for which it was designed and it is used.

Accordingly, each building belonging to a type has characteristics in common with other buildings of the same category, even if very different from each other: this is because both have the same function. The similarity between two buildings belonging to the same type can be explicit (as, for instance, in a school, where there are always hallways to connect classrooms) or implicit (as in an office building, where a big corporate open space can be very different from a small company office in a residential palace). A model of a building type can describe these features, shared by the buildings belonging to that type.

In this paper, we propose to create more realistic simulations of indoor environments for the Virtual Robot Competition by exploiting models of building types, like those presented in [8], namely *House* (residential buildings and houses), *School* (school buildings), and *Office* (office buildings and open spaces). Note that these three building types are particularly significant in the real world and are usually considered for experimental activities of autonomous mobile robots.

Here, we show the relevant aspects of the data sets for extending NIST World Generator (for more details, refer to [8]). Floor plans of 150 actual buildings (approx. 6,000 rooms) belonging to the above three categories are analyzed to extract the features of the models of the building types. Floor plans are divided in three separate data sets, according to their original building types: D_H , D_S , and D_O refer to House, School, and Office, respectively. Input floor plans were selected from eleven monographic books used for the design and analysis of buildings in architecture. In these books, rooms are labeled according to their function by the architects who designed the buildings. We mapped the labels provided by the architects to a smaller and more general set of labels, called *semantic labels*. We consider the following set of semantic labels: *small room*, *medium room*, *big room*, *corridor*, *hall*. They have been chosen to be simple enough to be immediately understandable by humans, but sufficiently descriptive of types of environments operated by robots. Each room in the data sets is also associated to a secondary semantic label, dependent on its building type, which explains the intended function of the room (e.g., class in Schools, office or conference room in Offices). It is interesting to point out how there emerges a strict mapping between the first and the second semantic label in every building type: for example in an office building almost all of the rooms functionally labeled as offices are semantically labeled as *medium room*.

Data sets are composed of one entry for each room found in a floor plan, represented as a vector of features. The features are chosen to capture (some of) the characteristics of the model underlying a specific building type. The features can be divided in two groups. The first group of features captures the shape of a

room and consists of the area a of the room and the axes ratio $rt = M/m$ of the major axis M and minor axis m of the room bounding box. The second group of features represents the structure of the building and the connections of the room with the rest of the environment (in particular, with adjacent rooms), as this is one of the distinguishing characteristics of a building type. The features of this second group include the number d of doorways present in a room r and the labels of the rooms directly connected to r . More precisely, for each semantic label s (s could be S, M, B, C, H for small room, medium room, big room, corridor, and hall, respectively), we consider the number of doorways l_s that connect r to a room with semantic label s . A room r is hence described by the feature vector $F_r = \langle a, rt, d, l_S, l_M, l_B, l_C, l_H \rangle$.

A quantitative statistical analysis of the above data sets is presented in Table 1. The mean μ and standard deviation σ of some of the room’s vector of

| | | Area a | | | | Doorways d $rt=M/m$ | | | |
|----------|---|----------|-------|-------|----------|-------------------------|----------|-------|----------|
| | | Label | % | μ | σ | μ | σ | μ | σ |
| H House | C | 23.9 | 14.0 | 11.4 | 4.0 | 1.8 | 3.4 | 3.6 | |
| | H | 8.0 | 61.5 | 25.1 | 3.3 | 1.5 | 1.8 | 0.8 | |
| | S | 28.0 | 6.0 | 2.6 | 1.0 | 0.2 | 1.6 | 0.6 | |
| | M | 26.3 | 17.4 | 4.8 | 1.3 | 0.4 | 2.0 | 1.0 | |
| | B | 13.7 | 33.7 | 7.8 | 1.8 | 1.0 | 1.5 | 0.4 | |
| S School | C | 15.4 | 36.1 | 42.2 | 5.7 | 3.8 | 7.4 | 7.6 | |
| | H | 5.6 | 205.1 | 172.9 | 6.9 | 4.5 | 2.0 | 1.3 | |
| | S | 55.6 | 10.0 | 9.7 | 1.1 | 0.4 | 1.6 | 0.9 | |
| | M | 21.3 | 54.8 | 14.8 | 1.6 | 0.8 | 1.3 | 0.4 | |
| | B | 2.0 | 102.9 | 22.3 | 2.1 | 0.7 | 1.7 | 0.7 | |
| O Office | C | 20.6 | 30.5 | 25.7 | 6.7 | 4.8 | 8.1 | 7.3 | |
| | H | 6.6 | 106.9 | 84.5 | 6.5 | 3.6 | 2.4 | 1.6 | |
| | S | 23.2 | 4.5 | 2.3 | 1.0 | 0.1 | 1.3 | 0.7 | |
| | M | 34.5 | 18.5 | 14.4 | 1.2 | 0.3 | 1.9 | 0.8 | |
| | B | 14.8 | 31.3 | 12.2 | 1.4 | 0.9 | 1.9 | 1.0 | |
| H+S+O | C | 20.3 | 26.4 | 27.9 | 5.7 | 4.1 | 6.5 | 6.8 | |
| | H | 6.8 | 110.9 | 109.2 | 3.7 | 3.7 | 2.2 | 1.3 | |
| | S | 32.3 | 7.1 | 6.9 | 1.0 | 0.4 | 1.5 | 0.8 | |
| | M | 29.3 | 24.5 | 18.9 | 1.3 | 0.6 | 1.8 | 0.8 | |
| | B | 11.5 | 35.0 | 18.0 | 1.6 | 0.9 | 1.8 | 0.9 | |

Table 1: Characteristics of the rooms in the data sets, where % is the percentage of semantic labels (C ‘corridor’; H ‘hall’; s ‘small room’; M ‘medium room’; B ‘big room’) present in the data set and μ and σ are the mean and the standard deviation of the corresponding feature, respectively.

features have been calculated. Some of the characteristics of the rooms of the same type are rather constant within a building type. For example, for House building type, all the rooms (except corridors and halls) have a standard deviation of the number of doorways not larger than one. The same happens for the other building types. Rooms with the same label belonging to buildings of different types are consistent with the typical size of the buildings. For example, a medium room in a house has an area a , on average, of 17.4 square meters, while a medium room in a school building has an average area a of 54.8 square meters. Moreover, a corridor in a house has an average axes ratio ($rt = M/m$) of 3.4, while a corridor in a school or in an office has a ratio of 7.4 and 8.1, respectively. Some distinctive characteristics of each building type are effectively captured by our analysis and represent the models for building types we exploit to generate realistic environments.

4 Are maps currently used in Virtual Robot Competition realistic?

| | C | H | S | M | B |
|-------------------------|----|---|---|---|---|
| CORRIDOR | 12 | 0 | 0 | 0 | 0 |
| HALL | 0 | 3 | 0 | 0 | 0 |
| OFFICE ROOM | 0 | 0 | 9 | 8 | 6 |
| EMPTY ROOM | 0 | 0 | 9 | 2 | 1 |
| MAZE ROOM | 0 | 0 | 0 | 0 | 2 |
| SERVICE ROOM | 0 | 0 | 0 | 0 | 4 |
| CONFERENCE/MEETING ROOM | 0 | 0 | 0 | 0 | 1 |
| OTHER | 0 | 2 | 0 | 0 | 1 |

Table 2: Correspondences between the set of semantic labels according to room size (C ‘corridor’; H ‘hall’; s ‘small room’; M ‘medium room’; B ‘big room’) and the set of functional semantic labels reported in the first column for the map of Fig. 1.

In this section we evaluate a map used in the first round of the final stage of the 2013 Virtual Robot Competition against the model of the previous section, in order to discuss its structural realism. In particular, we want to analyze whether or not the map has the structural characteristics typical of a real office. The map has 60 rooms and is shown in Fig. 1. We have semantically labeled the map using two taxonomies: the first one labels a room according to its size (Small, Medium, or Big), the second one according to its function (Office, Service room, Meeting room, Empty). Corridors and halls are considered special cases, and are semantically labeled in the same way in both categories. The results of our analysis are shown in Table 2 and in Table 3.

| | Label | % | Area a | | Doorways d | | rt=M/m | |
|---------|-------|------|----------|----------|--------------|----------|--------|----------|
| | | | μ | σ | μ | σ | μ | σ |
| RoboCup | C | 20.0 | 51.5 | 47.2 | 2.5 | 1 | 5.0 | 4.2 |
| | H | 8.3 | 73.8 | 42.4 | 2.6 | 2.0 | 1.8 | 1.3 |
| | S | 30.0 | 9.0 | 0.0 | 1.5 | 0.5 | 1.0 | 0.0 |
| | M | 16.7 | 18.0 | 0.0 | 2 | 0.9 | 2.0 | 0.0 |
| | B | 25.0 | 50.4 | 30.4 | 1.7 | 0.8 | 1.0 | 0.1 |
| | Label | % | Area a | | Doorways d | | rt=M/m | |
| | | | μ | σ | μ | σ | μ | σ |
| Office | C | 20.6 | 30.5 | 25.7 | 6.7 | 4.8 | 8.1 | 7.3 |
| | H | 6.6 | 106.9 | 84.5 | 6.5 | 3.6 | 2.4 | 1.6 |
| | S | 23.2 | 4.5 | 2.3 | 1.0 | 0.1 | 1.3 | 0.7 |
| | M | 34.5 | 18.5 | 14.4 | 1.2 | 0.3 | 1.9 | 0.8 |
| | B | 14.8 | 31.3 | 12.2 | 1.4 | 0.9 | 1.9 | 1.0 |

Table 3: Characteristics of the rooms in the map shown in Fig. 1 compared to those found in the Office data set. The format of the table is the same as Table 1.

The main difference between the map used in the competition and a real building is in the relevance of corridor rooms. In a real office almost all of the rooms are connected to a corridor and, since most of the rooms have only one door, the 70% of the rooms are connected only to a corridor and almost no rooms are connected with each other. This means that, in order to explore a building, a robot needs to travel through the corridors and then selectively enter

into the rooms; the path between two distant rooms is usually a list of connected corridors. In the simulated map, corridors do not have the same relevance: they have few doors and are typically connected only with each other. Corridors often do not lead to anything, ending up in a dead-end or in an empty room. The other rooms (e.g., offices) have more doors than their real counterparts, and are often connected between them. Examples of this unnatural structure can be seen in the vertical row of rooms at the left of the map in Fig. 1: these rooms are all connected to each other and are not connected to the corridor that runs parallel to them. Another recurring structure in the simulated map is a series of offices directly connected only to each other, like a maze (in the map there are also two actual mazes, but they are not relevant to this analysis). To reach the last room of these office-mazes, a robot must enter the first office, and then go through all other offices through a fixed path. This type of behavior (i.e., to neglect corridor and to explore paths between offices) is encouraged in the simulation but is seldom performed in reality. It is also interesting to point out that, as can be seen in Table 2, an office in the simulated map can be a small, medium, or big room. Instead, in a real building, usually every office is a medium room.

Just to add another figure, in a real building, approximately 98% of the offices are directly connected to at least one corridor while, in the simulated map, only 27% of offices present this feature. Even if these simulated structures can be challenging for teams participating to the competition, they are never encountered in real world and, hence, results obtained by simulated robots could be hardly transferable to real robots. Our proposal aims at contributing to overcome this situation.

5 Increasing structural realism of indoor environments

Our proposed system is an extension and partial redevelopment of the NIST World Generator tool, described in Section 2. The main focus of our work is the algorithm used for generating new UDK maps. The previous randomized policy is replaced by a new one, which uses a generative model that exploits the building type features extracted from the data sets. In particular, we consider the data set D_O (presented in Section 3) to derive our model, as NIST World Generator is currently oriented only to office buildings. The use of this model guarantees the structural realism of the newly-generated simulated UDK maps.

Fig. 3 (top) is a schema of the current process performed by the NIST World Generator to obtain a new simulated environment (see also Section 2). The user is asked to decide the size of the grid of the map. This seems to bias the user to choose square-shaped buildings, instead of other common shapes (e.g., rectangular-shaped ones). The structure of the building is strongly determined by a path of corridors that run parallel to the map’s perimeter, in a circuit. The space contained in this circuit is then filled with a random sampling from a selection of possible rooms. The main limitations introduced by this approach are due more to the existence of the external hallways circuit and to a limited

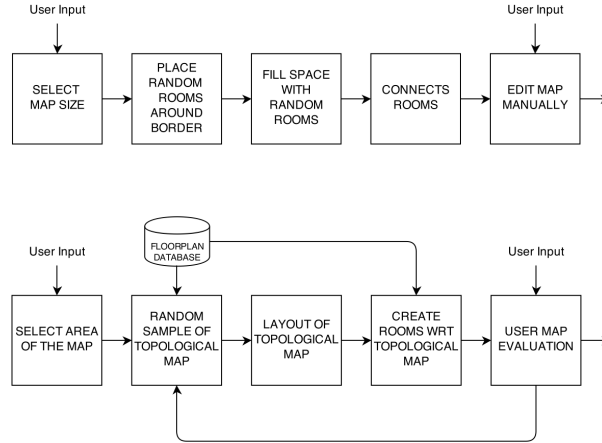


Fig. 3: Basic process flow in the current NIST World Generator (top) and in our extension (bottom) in order to obtain a randomly generated map.

interaction with an experienced user, than to the random policy used for the room selection.

Our approach aims at reducing these constraints, as presented in Fig. 3 (bottom). We let the user select the area of the map, i.e., the number of cells of the grid map. The two dimensions (width and height) of the grid rectangle are imposed later according to the layout of the generated environment, thus facilitating different building layouts as output (e.g., a narrow and wide building, as it leans on a side of a road). While the current random generative policy samples each room independently from another, we provide a sampling policy based on the type of the room, on its connection to the rest of the environment, and on the whole structure of the simulated environment. To perform this step, we use an abstract topological graph-based representation of the environment, where every room is represented with a node and an edge between nodes exists if and only if the two rooms are directly connected through a door. The sampling method is derived from [10] and exploits the concept of subgraphs and edit operation probability between sets of subgraphs. We use gSpan algorithm [11] for deriving the most frequent subgraph from the data set of topological maps. The result of the sampling phase is a topological map of the new environment. The graph is then displayed with a 2D layout using a heuristic method and, subsequently, each node is associated with a 2D room, exploiting again the data set D_O as a knowledge base.

Using this basic method, artifacts or unrealistic situations can be created in the randomly generated environment. Hence, we let the user (possibly an expert of the domain) decide whether to accept the generated environment, rebuild it from scratch, or partially regenerate some local portions deemed unrealistic or problematic.

The result of the generative process is an XML file, which can be modified, updated, or improved using the manual editing operation provided by the current NIST World Generator. Fig. 4 shows a toy example of how the topological map

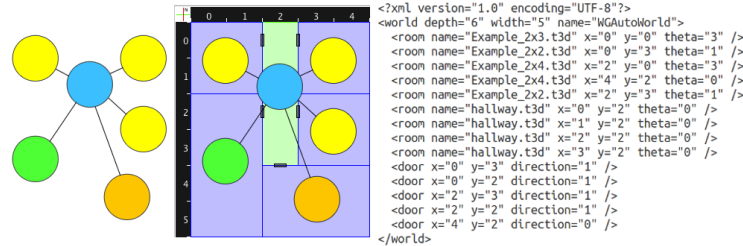


Fig. 4: A small example of the steps presented in Fig. 3 (bottom).

of the environment is used to generate a floor plan in the XML format used by the NIST World Generator. In the figure, a topological map (left) is used to determine the floor layout (center) and the resulting XML file (right).

The data sets and the tool described in this paper will be made publicly available at <http://home.deib.polimi.it/luperto/datasets/floorplans>.

6 Test case analysis

In this section we provide an analysis of a realistic environment created according to the principles described in the previous sections. Fig. 5 shows an aerial view of a realistic office building. The layout is highly symmetrical and the offices are all connected to the main corridors. Specifically, the building is divided in two main corridors, connected between them by two other smaller corridors (which are connected to the toilets and to two empty rooms marked with A and B). The rooms A and B could represent the stairs that connect this floor to other floors (as in real buildings). Note that NIST World Generator is currently designed to create only single-floor buildings. Nevertheless, the locations of rooms A and B can be used as initial spawning points for robots in an exploration task, since they could be considered the main entrance to the environment, thus providing realistic starting points for testing a robotic system. The right side of the map is a big open-space office. The main features of the rooms and of the whole building are consistent with those of a real building, as presented in Section 4.

7 Conclusions and future work

In this paper we have motivated the need for an improved realism of environments used in the Virtual Robot Competition. Realism of the simulations in robotics has often been dealt with as an entirely physical concept: a simulated



Fig. 5: A simulated office building generated according to our method.

environment is realistic if the laws of physics are applied in a consistent manner. While physical realism is of paramount importance, we have shown how some simulated environments are far from their real-world counterparts and how, as a result, the significance of simulated tests could be reduced. We propose to consider a broader concept of realism for simulations, including the structural plausibility, referring to the likelihood of an environment, namely to the fact that it adheres to a common cultural model. We have discussed and analyzed a possible representation of this cultural model, called building type. Accordingly, we presented our extension of the NIST World Generator tool, which is able to generate realistic offices environments, and we also showed an example of generated environment.

The proposed extension can improve the potential of USARSim as a simulation tool, both as part of RoboCup (also in other competitions, like @Home) and as a testing platform for research in robotics. Future work will address the possibility to generate UDK maps of other building types besides offices, like schools, houses, and malls. An example can be seen in Fig. 6, which shows the floor plan of a shopping mall created using the NIST World Generator. This map was manually created starting from the floor plan of a real building and was used for experiments in [12]. All the rooms are empty, since the models of the features of a shopping mall are still missing. We are seeking the possibility of automatically generating realistic buildings of this kind, which can greatly enhance the capabilities of USARSim as experimental tool.

Acknowledgements

The authors thank Nicola Sturari for contribution in developing the system.

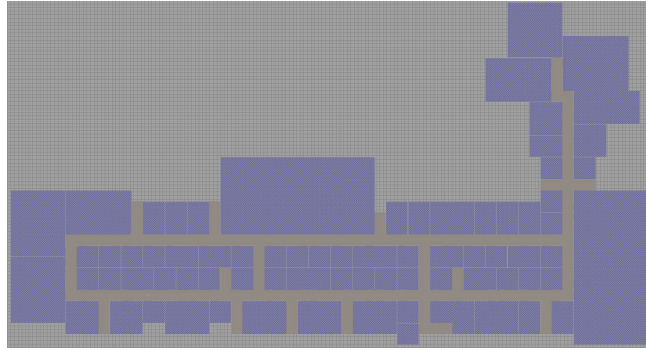


Fig. 6: Floor plan of a big shopping mall (Oriocenter, Orio al Serio (BG), Italy), generated with NIST World Generator using template and custom rooms.

References

1. Amigoni, F., Schiaffonati, V.: Autonomous mobile robots as technical artifacts: A discussion of experimental issues. In Magnani, L., ed.: *Model-Based Reasoning in Science and Technology*. Volume 8 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*. Springer (2014) 527–542
2. Amigoni, F., Reggiani, M., Schiaffonati, V.: An insightful comparison between experiments in mobile robotics and in science. *Auton Robot* **27**(4) (2009) 313–325
3. Amigoni, F., Schiaffonati, V.: Good experimental methodologies and simulation in autonomous mobile robotics. In Magnani, L., Carnielli, W., Pizzi, C., eds.: *Model-Based Reasoning in Science and Technology*. Volume 314 of *Studies in Computational Intelligence*. Springer (2010) 315–332
4. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: USARSim: A robot simulator for research and education. In: *Proc. ICRA*. (2007) 1400–1405
5. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: Bridging the gap between simulation and reality in urban search and rescue. In Lakemeyer, G., Sklar, E., Sorrenti, D., Takahashi, T., eds.: *RoboCup 2006: Robot Soccer World Cup X*. Volume 4434 of *Lecture Notes in Computer Science*. Springer (2007) 1–12
6. Howard, A., Roy, N.: *The robotics data set repository (Radish)* (2003)
7. Balakirsky, S., Brent, T., Dukto, J.: NIST Simmers Team Description Paper Infrastructure Competition RoboCup 2010. *RoboCup2010 CD* (2010)
8. Luperto, M., Quattrini Li, A., Amigoni, F.: A system for building semantic maps of indoor environments exploiting the concept of building typology. In: *Proc. RoboCup*. (2013)
9. Zwennes, O., Weiss, A., Visser, A.: Adapting the mapping difficulty for the automatic generation of rescue challenges. In: *Proc. IranOpen*. (2012) 482–487
10. Aydemir, A., Jensfelt, P., Folkesson, J.: What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments. In: *Proc. IROS*. (2012) 4675–4682
11. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: *Proc. ICDM*. (2002) 721–724
12. Cipolleschi, R., Giusto, M., Quattrini Li, A., Amigoni, F.: Semantically-informed coordinated multirobot exploration of relevant areas in search and rescue settings. In: *Proc. ECMR*. (2013) 216–221