Realistic Cities in Simulated Environments - An Open Street Map to Robocup Rescue Converter

Moritz Göbelbecker and Christian Dornhege

Institut für Informatik**
Universität Freiburg
79110 Freiburg
{goebelbe, dornhege}@informatik.uni-freiburg.de

Abstract. A general problem when developing large scale disaster simulation environments is to acquire GIS data. In this work, we tackle the problem of map generation from public sources. Usually the major problem is not only the data conversion itself, but to get access to the data at all. We solve this problem by using the website OpenStreetMap.org, that provides mapping data for the whole world in a wiki-style concept, as our source of data, thus being able to generate maps for almost any city. The data is converted to the format required by the Robocup Rescue Simulation System, enabling simulations on various real-world scenarios.

1 Introduction

Disaster simulation environments enable researchers to investigate techniques for mitigating large scale disaster scenarios. The Robocup Rescue League was founded with the idea to provide solutions for events as the Great Hanshi-Awaji earthquake in Kobe [1, 2]. The league is based on the Robocup Rescue Simulation System, that provides a collection of realistic simulators [3] for developing sophisticated multi-agent systems and therefore we chose this system as a target application to integrate real-world data.

(a) (b) (c)

Fig. 1. The figure shows the conversion of OpenStreetMap data to a rescue map for the city of Graz. (a) The raw data in the JOSM editor. (b) The map in JOSM adapted to Rescue specific needs. (c) The converted map in the Morimoto Viewer.

One major issue still is the creation of realistic maps as this usually has to be done manually in a time consuming process. This shows in the limited selection of maps that is available. For example the Kobe map, along with others, has served as one of the default maps for years. It is obvious, that agent development should not focus on a very limited subset of maps. A rich selection of different scenarios is generally very desirable for a simulation environment.

Another benefit of real-world scenarios is that it does not only allow realistic simulations, but also allows to integrate real-world data into the simulation environment. One such application is to generate an overview map of the rescue forces' positions acquired by the deployed units themselves [4] that enables incident commanders to make informed decisions. Additionally current observations as the spread of a fire, trapped victims, etc. can be easily integrated into the map.

There have been previous approaches to generate maps based on GIS data [5]. The recurring issue is the acquiration of such official GIS data, that is not generally available for any city. We provide a solution to this problem by using OpenStreetMap [6] as our source of GIS data. OpenStreetMap is a wiki-style website dedicated to provide an open mapping resource for everybody. Almost any big city

** This research was partially supported by DFG as part of the collaborative research center SFB/TR-8 Spatial Cognition R7

and even many small cities are accurately mapped, providing GIS data for the whole world. The data is licensed under the Creative Commons licence making it available for reuse in the Rescue League for free. OpenStreetMap provides several editors, one of which is JOSM, that incorporates a plug-in interface. Our contribution is such a plug-in for the Rescue Simulation league, that allows us to access OpenStreetMap data, process it, and finally output maps in the rescue format.

In the remainder of this paper we describe the structure of the GIS data in section 2, the processing steps are shown in section 3 and 4, and in section 5 we present examples of converted maps for different cities.

2 GIS Data in the OpenStreetMap World Model

Compared to Robocup Rescue, OpenStreetMap uses a very simple, but at the same time flexible data model. There are only three primitive types [7]:

- **Nodes** have an id and a location, given in longitude and latitude coordinates.
- **Ways** consist of an ordered list of nodes. Depending on its context, it may represent a line (e.g. a street) or an area.
- **Relations** can contain an arbitrary number of other primitives (including other relations) and can be used to model higher level structures.

The underlaying data model is XML, so all semantic information is encoded in *tags*, key-value pairs of which each primitive can have an unlimited amount. Although tags are in principle arbitrary, standardized tags have emerged to represent commonly used information. Thus, the presence of a "highway" tag states that a way represents a road, whose type is further specified by the value of the tag. Table 1 lists a number of common tags that are relevant for rescue maps [8].

tag	semantics
highway	a road of any kind
oneway	true if the road is a one-way street
lanes	number of driving lanes
area	treat the way as an outline of an area
building	for a way: the outline of a building
shop, amenity, building	for nodes: indicate buildings
landuse	general type of a larger area

Table 1. Commonly used tags in OpenStreetMap and their meanings.

There are several tags that are not of direct interest for the rescue simulation, but which might help us to interpolate missing data. For example, the "shop" tag on a node indicates the presence of a buiding at that point, even if there is no outline present.

In the same way, the "landuse" tag on a way can give the automatic building generator hints which kinds of buildings (if any at all) should be placed in that area (e.g. no buildings should be placed in areas with "landuse"="forest") .

3 Map conversion

The map converter was implemented as a plug-in to the OpenStreetMap map editor JOSM [9]. We perform the conversion in two seperate steps. First, we extract the relevant information into an *intermediate representation*. All processing is executed on this intermediate representation. In the second step, this data is saved as a rescue map.

3.1 Intermediate representation

The intermediate representation is a map layer in the OpenStreetMap data model, which contains all the information required by the kernel and simulators in an easy to extract fashion. We chose this approach, because the conversion of data between the different models requires some amount of interpretation of the source data. All the processing will now take part when converting to the intermediate representation.

As this representation is valid OpenStreetMap data, it can be edited, saved and loaded after the lossy parts of the conversion have finished. Conversion to Rescue maps is then a simple task, as is only requires writing of the already present data.

As all other semantics in OpenStreetMap, we model the Rescue specific data using a set of custom tags. The neccessary information is commonly stored in the form of properties, so we simply create a tag "rcr:*property-name*" = "*property value*" for each property. We introduce a variety of new tags to express different aspects of Rescue maps. These are listed in table 2.

These tags are only added to the OpenStreetMap data, all original data remains untouched. We can then use this additional data (e.g. street names) for other purposes, like visualization or improved communication with human teams. To facilitate this usage, the ids of exported objects are also written in the OpenStreetMap file, thus allowing easy retrieval of the element corresponding to a simulation object.

tag	data type	semantics
rcr:type	{node, road, building}	The basic type of the object
rcr:id	number	The id of this object in the Rescue map (written during export)
rcr:outline	id	The id of the building outline
rcr:entrances	ids	A list of entrance nodes
rcr:refuge	bool	This building is a refuge
rcr:fire	bool	This building is an ignition point
rcr:ambulances	number	Number of ambulances on this position
rcr:firebrigades	number	Number of fire brigades on this position
rcr:policeforces	number	Number of police forces on this position
rcr:civilians	number	Number of civilians on this position

Table 2. Tags used in the OpenStreetMap representation of Rescue maps

3.2 Roads

Roads in OpenStreetMap are generally identified by the "highway" tag. Unfortunately, simply mapping all ways, containing such a tag, to rescue roads is problematic due to the following reasons:

- Roads are mapped for the general public, not for rescue services. There is, e.g., no distinction, if a way tagged as "highway"="pedestrian" is merely reserved for pedestrians, or if it is physically impossible to reach by rescue vehicles.
- Sometimes, especially at larger junctions, each lane is mapped seperately, creating a very complex road graph.
- In OpenStreetMap, ways can also represent areas, that cannot be represented in the Robocup Rescue domain.

Therefore we employ postprocessing on the raw data. First, we add all roads that are certainly passable by rescue forces. Next, among possibly passable roads (like pedestrian zones, cycleways, footways) we select those, that are required to reach all buildings already on the map. The intuition being, that each building will have at least one road connection in the real world.

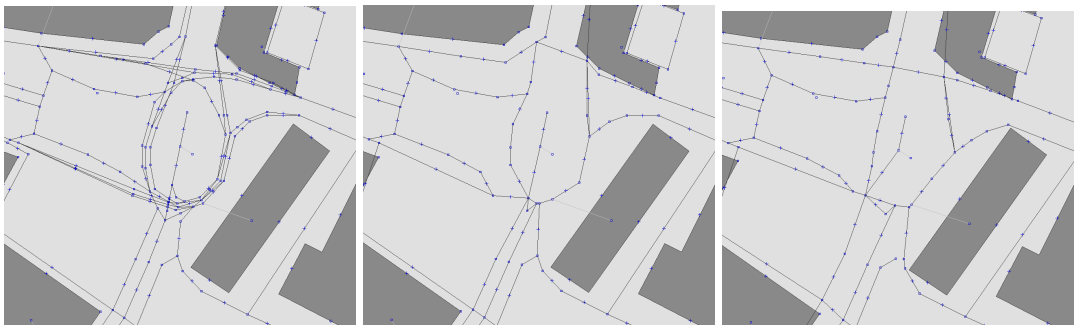
Finally, we perform a reachability check of all roads and remove those, that cannot be reached from the main part of the map, thus eliminating small isolated clusters, that might have been introduced as a map only contains a rectangular section of the world.

To remove redundant roads, we only regard those, that have the "oneway" tag set. We then iterate through each way, and at each crossing decide for each set of way segments with the same orientation (i.e. outgoing or incoming), if they can be merged into one segment. We merge the largest possible set of segments for which the average angular difference is below a certain threshold and for which the average direction of ways on the opposing ends of the segments is roughly similar, thus avoiding merging opposite lanes.

This simplification can introduce some irregularity into the roads, so afterwards we apply a simple smoothing algorithm to the modified roads. Figure 2 shows the different stages of our method.

3.3 Buildings

Buildings can be mapping in OpenStreetMap as ways, that have a "building" tag. The Rescue map format doesn't represent the building itself as a polygon from nodes and edges, but only stores the outline in the building itself. Therefore, when converting a building we construct a *building node*, that is located at the centroid of the building polygon and holds an "rcr:outline" tag, pointing to the corresponding building outline. This building node consequently receives all rescue specific tags as, e.g., "rcr:area". When saving a map, the outline is automatically written to the map.

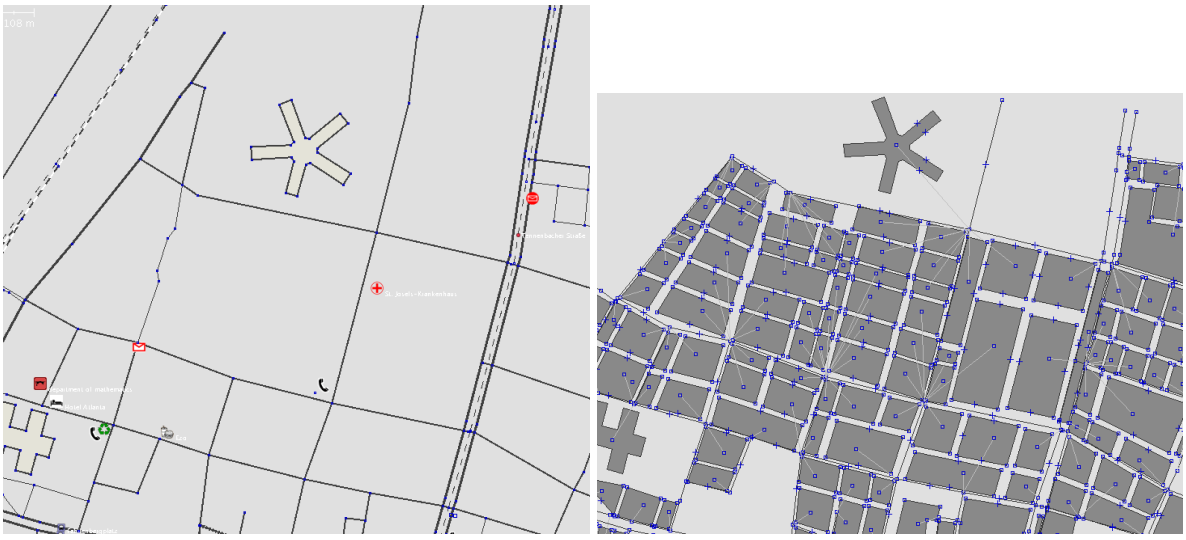


(a) (b) (c)

Fig. 2. This figure shows the steps of our road simplification algorithm. (a) a complex junction with multiple lanes (b) the same junction after lane merging (c) the final result after smoothing.

Building entrances are usually not modelled in GIS data, but necessary for rescue operations. Fortunately buildings in OpenStreetMap data can have an address tag, that we can use to select the correct street to connect a building to. If no adress is given we fall back to selecting the nearest road.

One general issue with GIS data, that also applies to OpenStreetMap data, is, that in contrast to roads, not all buildings are modelled. The Rescue League relies on the presence of buildings, so we implemented a building generation algorithm to fill unmapped areas. Basically, the algorithm follows the general idea, that can, for example, also be seen in the rescuecore's random map generator [3]. First we identify city blocks, that are surrounded by streets. Secondly, depending on the size of the block, we split it horizontally and vertically to gain the final buildings. An example of this algorithm can be seen in figure 3.



(a) (b)

Fig. 3. An example of our building generation algorithm. (a) shows the original data. (b) shows the same scene with added buildings. Note, that the already mapped building in the top is kept during the conversion.

During this autogeneration, we try to use as much information as the base GIS data allows us to. We obviously retain every already manually mapped building. Sometimes buildings are only mapped as nodes or addresses hinting the location of a building. In such a case, we also place a building at that node. Additionally important information can be gained from marked areas or landuse tags. Those might mark parking areas, parks, forests, and other similar things prohibiting us from placing buildings, thus keeping open places, that naturally appear in the structure of most cities. Finally, when choosing parameters for constructing buildings as their size or the spacing between buildings, we obey "landuse" tags, that might mark areas as "industrial", "residential", and others. Residential areas, for example, are usually formed from many small buildings, that stand farther apart, which

will especially influence the spreading of fires in those areas. Although we do not always have the original building data, by using those geographic annotations, we still think, that we can reproduce the non-uniform structure of a city in a representative manner for the Rescue Simulation League.

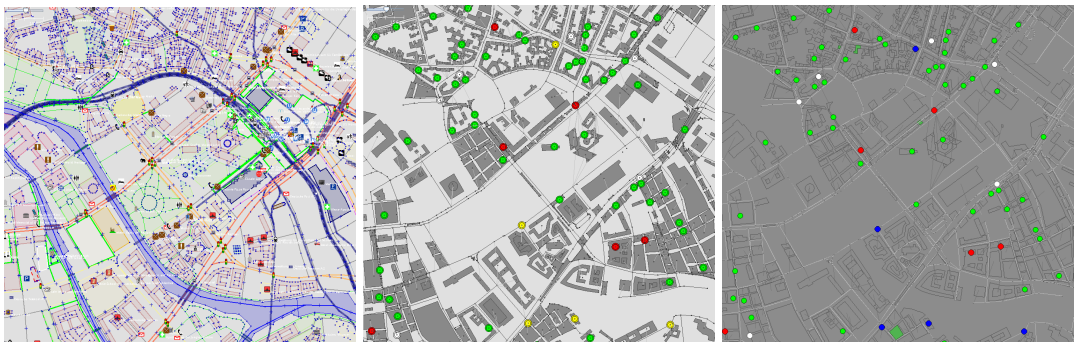
4 Scenario generation

Rescue maps not only consist of pure GIS data, but also describe a rescue scenario. This consists of marking specific buildings as the stations and refuges, placing agents and civilians as well as choosing fire points. Our plug-in can also edit those Rescue specific entities and save them to a map, thus providing a full rescue scenario.

We provide a simple algorithm, that places a given number of those entities on a converted map. As in previous sections we propose the use of semantic information, if applicable. In this case, we can place refuges at buildings, that have been tagged as hospitals. Additionally the probability for fire points can be significantly higher in industrial areas. A simple example is presented in figure 6.

5 Results

In this section we present exemplary conversions of different cities as well as an insight to our implemented tools and the conversion procedure.



(a) (b) (c)

Fig. 4. The figure shows the conversion of OpenStreetMap data to a rescue map for the city of Berlin. (a) The raw data in the JOSM editor. (b) The map in JOSM adapted to Rescue specific needs. (c) The converted map in the Morimoto Viewer.

The conversion procedure is presented in figure 4 and figure 5. The process of generating a rescue map of any city follows the following few simple steps:

- Start the JOSM-editor and enable the Rescue plug-in.
- Via the download feature in JOSM, choose an excerpt from the world and download the GIS data of this area.
- Click the *Make Rescue Map* button to create the intermediate representation.
- Optionally, the intermediate representation can be edited manually, or a scenario can be generated.
- Save the map to the Rescue format and run it.

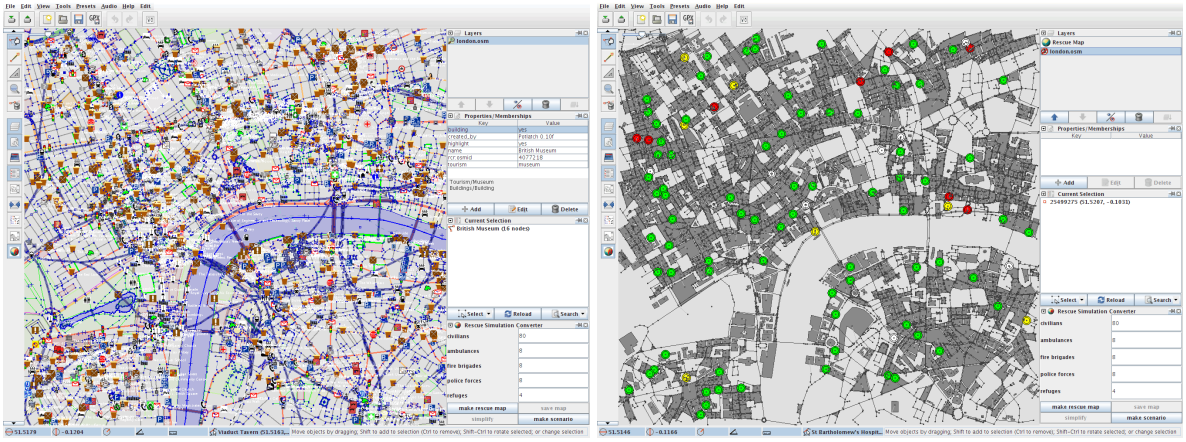
The scenario generation is shown in figure 6 during the conversion of London. Scenarios can be adapted by simple parameters shown in the plug-in's dialog. As this data is present in the JOSM map editor all relevant data can also still be edited, so that we not only have a simple conversion program, but a full editor.

6 Conclusion

We created a converter from OpenStreetMap GIS data to Robocup Rescue Simulation maps. By using OpenStreetMap as our source, we solve the common problem of getting the actual GIS data for almost any city in the world. As we have demonstrated the conversion process is as simple as possible and will enable agent developers to easily test their system on a variety of scenarios. Additionally it is now possible to use the Rescue Simulation System as a platform for integrating real-world data in a GIS representation. Such a system opens the opportunity for future contributions integrating position tracking of rescue forces, incident reports, and emergency calls and, looking further ahead, also the integration of agent frameworks as decision advisory.

(a) (b) (c)

Fig. 5. The process of map conversion for the city of Freiburg. (a) shows the downloaded data in the JOSM editor. (b) shows the intermediate representation in a Rescue layer, that can be saved to Rescue format. (c) is a screenshot of the same map running in the rescue kernel's viewer.



(a) (b)

Fig. 6. This figure shows screenshots from within the JOSM map editor for the city of London. (a) shows the raw GIS data of London. (b) presents the intermediate Rescue layer, that can still be edited before being saved. On the lower right side, the options dialog of our Rescue plug-in is visible.

References

1. Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shimada, S.: Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agent research. In: IEEE Conference on Man, Systems and Cybernetics. (1999)
2. RescueSystems: Robocup rescue. `http://www.rescuesystem.org/robocuprescue/` (2008)
3. Robocup: The rescue kernel. `http://sourceforge.net/projects/roborescue/` (2008)
4. Kleiner, A., Behrens, N., Kenn, H.: Wearable Computing meets Multiagent Systems: A realword interface for the RoboCupRescue simulation platform. In: SRMED2006 - Third International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster. (2006)
5. Takahashi, H., Tanigawa, M., Takahashi, T.: Tool kits for using open source gis data as robocup rescue gis maps. In: Robocup 2005. (2005)
6. Openstreetmap.org: Openstreetmap. `http://www.openstreetmap.org/` (2009) (date; 28. February 2009).
7. Openstreetmap.org: Osm protocol version 0.5. `http://wiki.openstreetmap.org/wiki/OSM_Protocol_Version_0.5` (2009) (date; 28. February 2009).
8. Openstreetmap.org: Map features. `http://wiki.openstreetmap.org/wiki/Map_Features` (2009) (date: 28. February 2009).
9. Scholz, I., Stoecker, D.: Josm. `http://josm.OpenStreetMap.de` (2009) (date: 28. February 2009).