Sampling-based floor plan analysis on BIMs

Christian Schlette and Jürgen Roßmann

Institute for Man-Machine Interaction (MMI), RWTH Aachen University, Germany
Email: schlette@mmi.rwth-aachen.de, rossmann@mmi.rwth-aachen.de

Abstract –

This contribution explores a novel combination of methods from robotics and BIM by working on Building Information Models (BIMs) with the Open Motion Planning Library (OMPL), an extensible programming interface and reference collection of various algorithms for sampling-based motion planning. We present a seamless workflow from industrial BIM generation to sampling-based planning of collision-free paths for mobile robots on 3D floor plans extracted from BIMs.

Beyond robot motion planning, we apply the workflow to generate huge amounts of paths for floor plan analysis. We assume the paths to resemble walking patterns where many paths crossing a given floor tile match with heavy usage of this space. The resulting paths are sampled in heatmaps, where pixels then correspond to floor tiles and the color value corresponds to the number of crossing paths, yielding a semi-automatic, easy-to-use method to analyze floor plan layouts in different scenarios.

Keywords –

BIM, Sampling-based Motion Planning, Floor plan analysis

1 Introduction

In robotics, we are used to see CAD data for mechanisms, machines and plants from industrial sectors such as automotive and aerospace, but based on the underlying approaches in computer aided engineering, the construction industry developed a similar concept called Building Information Modeling (BIM). At their core, Building Information Models (BIMs) are databases with well-defined layers for modeling and managing the relevant information for a wide range of share- and stakeholders of modern construction enterprises.

As the main application of BIMs is to hold the necessary data for construction, BIMs contain central parameters and semantic information about all structural elements, e.g. classes, dimensions and materials of walls and doors. Using Constructive Solid Geometry (CSG), the elements can be evaluated to generate full 3D geometric models (see Figure 1), yielding semantically labeled 3D models.

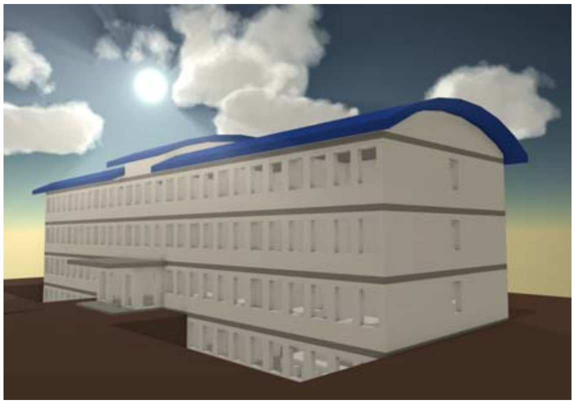

Figure 1. Semantically labeled 3D model of a multi-floor office building from BIM

Due to its significance in the construction industry, architects and construction engineers already explore new applications of BIMs, e.g. by using BIMs for building simulation and analysis [1][2]. As an example of how methods from robotics can contribute to such developments, we demonstrate the application of sampling-based algorithms for robot motion planning as well as floor plan analysis in this contribution.

Many well-established methods for motion planning are available for mobile platforms, with sampling-based motion planning being the most prominent approach for high-performance, model-based path generation in complex environments. Here, the Open Motion Planning Library (OMPL) provides a thoughtful Application Programming Interface (API) as well as an extensible set of reference implementations of historical and current algorithms (see Section 2). The models for collision testing in these algorithms are usually provided as CAD data for known environments, sensor data to enable planning in (partially) unknown environments [3][4] or plans sketched by human operators [5].

In section 3, we describe the workflow and prerequisites for making BIMs available in our software environment for robot simulation and control. Since motion planning for mobile platforms is a central task in indoor working environments, we demonstrate the usage of BIMs in section 4 by running OMPL on the model data. In section 5, we present first results of this combination of two strong concepts – BIM from the construction industry, OMPL from the robotics community – and discuss specific aspects of our solution for motion planning on BIMs and floor plan analysis.

2 State of the Art

Early approaches in motion planning for manipulators and mobile robots started planning in sandbox environments with fully known semantics and geometries, e.g. [6][7]. But since the a priori availability of detailed plans and models was not realistic and thus doubted at that time, research on mobile platforms turned towards "bottom-up", sensor-based localization and navigation in unknown environments. With BIM, this situation changes dramatically, as semantically labeled 3D models are becoming the standard data basis for modern construction enterprises. So far, only a few attempts have been made to link BIMs with navigation and motion planning. Isikda et al are extending BIM with a new data layer for (human) indoor navigation, e.g. to support emergency operations [8]. Borkowski et al are correlating semantic information from BIMs with sensor findings from mobile platforms in order to support "semantic navigation" [9], an approach which is later extended by Siemiatkowska et al towards "semantic localization" [10].

2.1 Open Motion Planning Library

Sampling-based methods are the most prominent approach for solving problems in fully defined, tight and cluttered workcells [11][12]. OMPL [13] is a C++ library developed by Kavraki et al. [14]. The development is visible since 2012 and offers many versions of "Rapidly-exploring Random Trees" (RRT) [15], "Probabilistic Roadmap Method" (PRM) [16], Sparse Roadmap Spanner algorithm" (SPARS) [17], "Expansive Space Trees" (EST) [18] as well as many other algorithms, in parts as contributions from OMPL users. The library is a Linux-driven development with the option to run on Windows and Mac. OMPL is the library behind "MoveIt!", the motion planning package for the "Robot Operating System" (ROS) [19].

2.2 Building Modeling Information

The BIM methodology encompasses processes and methods to create and manage virtual 3D building models (BIMs). As a central database for construction projects, BIMs contain semantic information, parameters and geometries of all structural elements, but also tracking information to share developments, changes and comments with share- and stakeholders. Although BIM started as a computer aided engineering approach for architects, construction engineers and civil engineers, BIMs are managed and applied today throughout the whole lifecycle of buildings – beyond design, planning and construction also for energy and facility management, automation and destruction [21][22].

BIM generation is currently limited to specialized 3D construction tools. Most of the tools are developed as parts of bigger software packages for 3D construction, which also offer solutions for managing and rendering BIM data. Today, the integration and usage of BIM data beyond the given capabilities of these dedicated software packages requires exporting the data to an intermediate format – e.g. exporting to VRML for rendering 3D views of the building [23].

A more general approach to exchange BIMs independently from specific software packages is provided by the "Industry Foundation Classes" (IFC) [24]. IFC is a neutral, ISO-certified (ISO/PAS 16739) data format standardized by buildingSMART, an international alliance of the developers of the leading BIM software packages. IFC data contains three major subsets, so-called "Views", *CoordinationView* for building design and construction management, *StructuralAnalysisView* for building simulation and analysis, and *FMHandOverView* to link buildings to computer aided facility management software. The basic structure of IFC data follows the "Standard for the Exchange of Product Model Data" (STEP) for the file-based exchange of geometric data (ISO 10303).

3 Concept

Our approach of running OMPL on BIMs is based on the software environment VEROSIM® for robot simulation and control, which we co-design and co-develop at the Institute for Man-Machine Interaction (MMI) for our research on eRobotics [25]. The eRobotics methodology provides a development platform for roboticist to exchange ideas and to collaborate with experts from other disciplines. The central method in eRobotics are Virtual Testbeds, where complex technical systems and their interaction with prospective working environments are first designed, programmed, controlled and optimized in 3D simulation, before commissioning the real system. Typical working environments are open landscapes, forests, cities, buildings, factories etc. In consequence, BIM on the one hand provides an important standard for modeling relevant environments and simulating their interdependencies with complex technical systems. On the other hand, modern buildings themselves are complex, highly automated, technical systems, which can profit from the simulation-driven development processes in Virtual Testbeds. We described this idea of “Virtual BIM Testbeds” in [23].

3.1 Prerequisites

The general idea of VEROSIM is to provide a micro-kernel architecture for simulation, where a slim core provides the central functionality to structure and manage a database which represents the simulated scenario. Individual simulation methods, datatypes and UIs are added via plugins. We briefly outline the system features which contribute most to our approach of running OMPL on BIMs.

Following the principles of Object-Oriented Modelling (OOM), the central database of VEROSIM is an active database, that contains all simulation entities and information of the environment as well as extensive meta data/ meta type information. The database is active, in the sense that it contains functionalities to connect and define the behavior of simulation entities directly in the database. This behavioral information differentiates the database from simple scene graph representations. The ability to interpret and manage meta data and meta type information enables the generic import, mapping and access of external database formats, such as IFC data.

For motion planning, the Kinematics plugin of VEROSIM provides the underlying functionalities to control robots in Cartesian or configuration space to sample the validity of given states. Collision detection in our system is carried out based on the simple approach to hierarchically compare the intersection of 1) axis-aligned bounding boxes, 2) user-defined simplified collision geometries, and 3) full polyhedrons. In case of BIMs, the geometries are simple enough to sustain performant collision detections without the need to introduce additional simplified geometries for level 2) tests.

3.2 Workflow

Figure 2 depicts the basic workflow of our approach. BIMs are typically modeled in 3D construction tools such as Autodesk Revit®, which have much in common with 3D CAD tools for other disciplines. The interaction with BIMs at this stage is mainly based on manufacturer-specific, proprietary database formats. As described in section 2, the major exchange format to transfer BIM data between the construction tools of different manufacturers is IFC. Using the IFC Engine DLL [26], our software is able to import and map IFC data directly into our own database (see Section 4 for details). As part of the import process, the IFC Engine DLL generates 3D geometries from CSG evaluations of the structural elements and their properties. In combination, the IFC data and the geometries yield semantically labeled 3D models, in which geometries are associated with their generator classes, the structural elements. The direct accessibility of the IFC data in our own database allows us to a) use the methods and tools in our software environment to work on and interact with the BIM, and b) to refer to and mix the IFC elements with our own Versatile Simulation Database (VSD) elements for robot simulation and control.

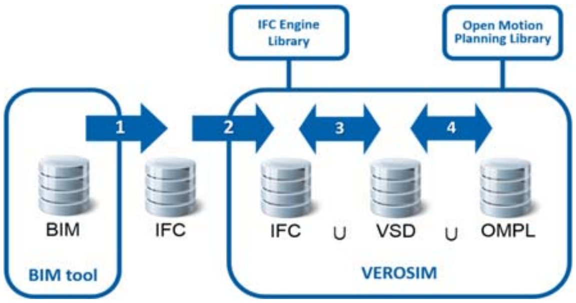

Figure 2. Workflow of running OMPL on BIM: 1) Export of IFC data from BIM tool, 2) Import of IFC data into VEROSIM database, 3) Preparation of IFC data with additional VSD elements, 4) OMPL queries and solution based on OMPL adapter classes

In particular, we use this access to prepare the building data for collision detection. Based on the semantics and a database search, we can identify doors directly from their class names and remove them in order to open rooms and corridors for passages. Also based on the semantics and a database search, we can collect all geometries which belong to a given entity (e.g. a selected floor or wing of a building) and connect them to the collision detection methods in our system.

In order to use OMPL on any 3D model in VEROSIM, we developed a set of adapter elements to make OMPL functionalities available in our database (see Section 4 for details). The adapter classes allow us to parameterize, trigger and store any number of OMPL queries and solutions directly by referring to existing elements in the database, e.g. by referring to robots, workcell geometries and 3D poses in the VSD or the IFC data. Also via the adapter classes, paths generated with planners in OMPL are directly available for robot simulation and control.

4 Running OMPL on BIMs

VEROSIM provides an importer based on the IFC Engine DLL by RDF Ltd. [26] for integrating and accessing IFC data [23]. The importer is set up by reading the meta data/ meta type schemas of the IFC language, thus being fully and generically in sync with a given version of the IFC data format. Then, the IFC schema is bidirectionally mapped to the VSD schema for associating incoming IFC data types with corresponding representations in the VSD (and vice versa). The import of concrete IFC data makes use of this so-called “schema mapping” by instantiating and initializing corresponding VSD types for the incoming IFC instances, thereby representing structure and details of the given IFC data in our own database. After this so-called “instance mapping”, we are able to refer to and mix the IFC elements with our own VSD elements in a third step called “functional mapping”. More details on this three-step mapping process are available in [27].

For interfacing OMPL, we developed a set of adapter classes to represent and map the main classes and parameters for formulating single query planning problems for the OMPL library (see Figure3 for details).

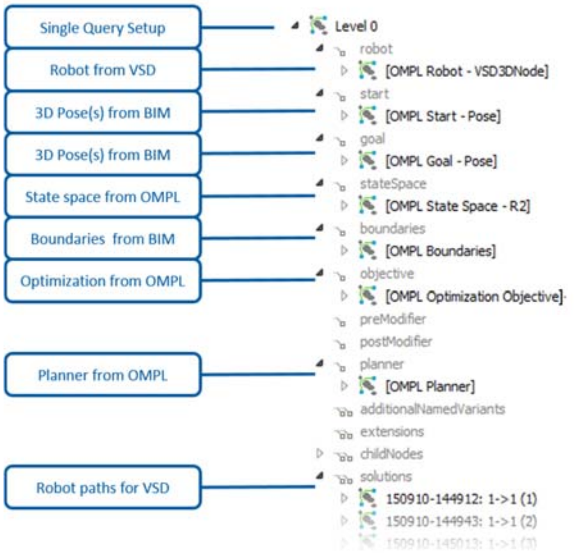

Figure 3. Setup with OMPL adapter classes

For a basic single query setup, we select a state-space model (e.g. *SE(3)*) and a planner (e.g. RRT*), and define start and goal states of the planning problem by selecting 3D poses from the database. In addition, we select a robot (including geometries, forward and inverse kinematics) and link it to an object derived from *ompl::base::StateValidityChecker*. During planning, this object signals OMPL requests to check the validity of proposed states in order to advance the sampling-based building of a collision-free path to reach the goal. For responding to the request, our derived implementation of the validity checker converts states proposed by the planner to our internal pose representation for Cartesian frames or joint configurations. The robot is set to this pose for a moment with the (invisible to the user) effect of rearranging the 3D workcell accordingly. In this rearranged situation, which represents the state proposed by the planner, we call the collision detection in VEROSIM and store the result. We then set the robot back to its original pose and respond to the request of the planner: Only if the rearranged situation did not exhibit any undesired collisions, the state is assessed to be valid (and invalid otherwise). The communication between OMPL and our system is depicted in Figure 4.

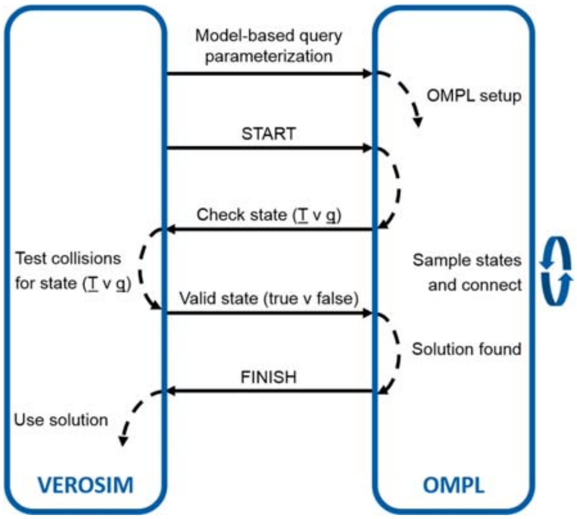
Model-based query parameterization
OMPL setup
START
Check state (T v g)
Test collisions for state (T v g)
Sample states and connect
Valid state (true v false)
Solution found
FINISH
Use solution
VEROSIM
OMPL

Figure 4. Interfacing OMPL with our system

5 Results

Following our workflow, running OMPL on BIMs is easily set up by importing a building model into our system, preparing it for collision detection and defining OMPL queries. As we are looking into motion planning for a mobile platform working on indoor tasks, we reduce the complexity of the full 3D problem in *SE(3)* by planning on 2D planes on individual floors in R^2 (see Section 5.1). In order to compensate for the missing transfer connections between floors, we then introduce “portals” on each floor, i.e. pre-defined sets of start resp. goal poses for entering and leaving a given floor (see Section 5.2). In section 5.3, we demonstrate how OMPL can be applied for BIM-based building simulation and analysis by evaluating floor plans by compiling numerous paths between arbitrary poses to learn about often used, thus critical passages.

In our experiments, we are using RRT*, an asymptotically optimal derivative of RRT by Karaman et al [20] which converges on the optimal path resp. shortest path in our experiments. This allows us to allocate a fixed runtime of 30 [s] for queries, in which RRT* will first find a collision-free path and then continue to minimize its path length to provide a high quality solution. Beyond a certain threshold of about 3 [s] for the construction of a first connection between start and goal pose, runtimes in our experiments can thus be assumed to scale linearly with the number of queries.

5.1 Single Floor Planning

For each floor $j \in \{-1, 0, 1, 2\}$, we select one representative start pose per room as well as one goal pose per exit resp. entry to the floor. In the given example, the floors are connected by central stairs, yielding one goal pose per floor, with only one additional exit on ground level. For each floor, we run RRT* three times to generate solutions from each start pose to the goal pose resp. both poses on Floor 0. The result is depicted in Figure 5.

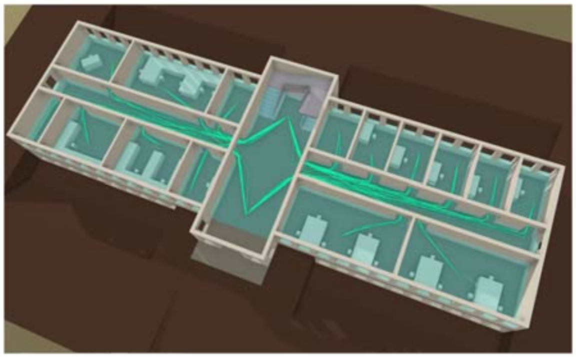

Figure 5. RTT* on Floor 0: 15 start poses (1 per room), 2 goal poses (portal "Stairs" and Exit), avg. solution length: 17.34 [m], runtime: 4.5/45.0 [min] (low/high quality)

5.2 Multi Floor Planning

We introduce "portals" for evaluating motions between individual floors and for describing the topology of multiple floors (see Figure 6). Portals are pre-defined poses at exits resp. entries to a floor and thus the essential start and goals poses for single floor motion planning. Using a portal for transfer between floors is assumed to add specific costs P to the solution, which represent average times for waiting and usage, e.g. for stairs and lifts. Equation (1) formulates a simple cost function for entering a portal i on floor j_{in} and leaving it on floor j_{out}:

$$P_i = c_{i_{j_{in}}} + \sum_{j_{in}}^{j_{out}} p_i w_{i_j} + c_{i_{j_{out}}}, \qquad (1)$$

where $c_{i_{j_{in}}}$ resp. $c_{i_{j_{out}}}$ represent the costs to enter resp. leave this specific portal on these specific floors. p_i is a portal-specific factor to ascend resp. descend one floor. The additional weight w_{i_j} allows for considering floor-specific variations from p_i as well as priorities and penalties.

The costs to use a portal can be expressed in terms of additional solution length, which allows for comparing multi floor planning solutions by assessing various combinations of average solutions lengths from single floor planning and portal costs.

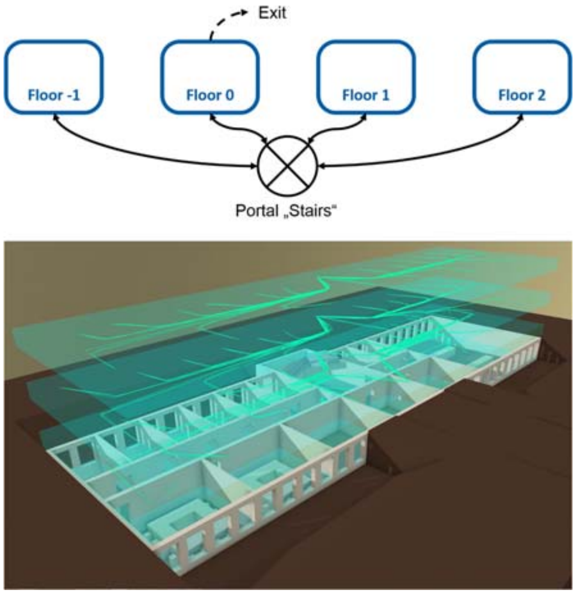

Figure 6. (**Top**) Floor topology with portals; (**Bottom**) RTT* on all floors: 64 start poses (1 per room), 1 goal pose (portal "Stairs") plus 1 goal pose on Floor 0 (Exit), avg. solution length: 16.29 [m], runtime: 9.6/96.0 [min] (low/high quality)

5.3 Floor Plan Analysis

So far, we demonstrated the combination of OMPL and BIM for robot motion planning in building maintenance tasks. On the other hand, the combination can also contribute to BIM-based building simulation and analysis. For floor plan analysis, we span a grid of 6 × 21 poses over a given floor (see Figure 7), which may well include invalid poses permanently colliding with walls and furniture. In a basic approach, we run RRT* to generate solutions from all grid poses to a given goal pose. The resulting solutions are then sampled in a heatmap, where a pixel position corresponds to a floor tile and the color value corresponds to the number of solutions crossing this floor tile. Figure 8 depicts the solutions "grid to exit" and the corresponding heatmap. We refine the basic approach by using the grid to define not only the start poses but also the goal poses, yielding a total of 15750 combinations in our example (however, many invalid due to collisions). Figure 9 depicts the solutions "grid to grid" and the corresponding heatmap. Clearly visible now are heavily used floor tiles in the right corridor in front of the larger offices, which may indicate critical passages, e.g. in case of emergency.

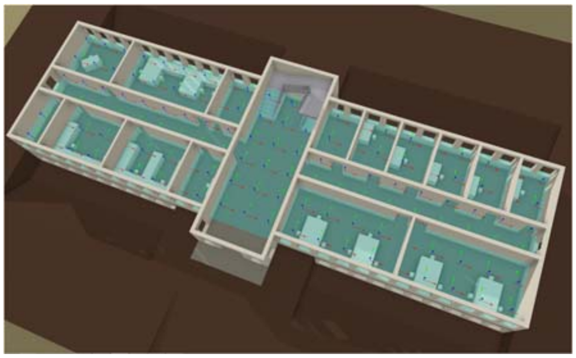

Figure 7. Grid of 126 poses on Floor 0

6 Conclusions

In this contribution, we are combining two strong concepts – BIM from the construction industry, OMPL from the robotics community – to allow for sampling-based motion planning directly on semantically labeled 3D building models.

In robotics, the availability of semantics and geometries of working environments has been assessed to be unrealistic a priori knowledge for many years. Now BIM has the potential to change future developments in robotic indoor applications significantly, as professionally generated, semantically labeled 3D models will become more and more available. For robotic applications such as building maintenance, BIM thus provides profound data for navigation and motion planning, which of course has to be additionally enriched and correlated with sensor data to cope with time-variant obstacles such as furniture and human co-workers.

Beyond motion planning for mobile platforms in indoor environments, we demonstrate the application of algorithms from motion planning for floor plan analysis as an example of how methods from robotics can contribute to BIM-based building simulation and analysis. In particular, the proposed floor plan analysis allows for identification of "shortest path" movement patterns between rooms and consequential heavily used floor segments, which may indicate critical passages that need closer attention.

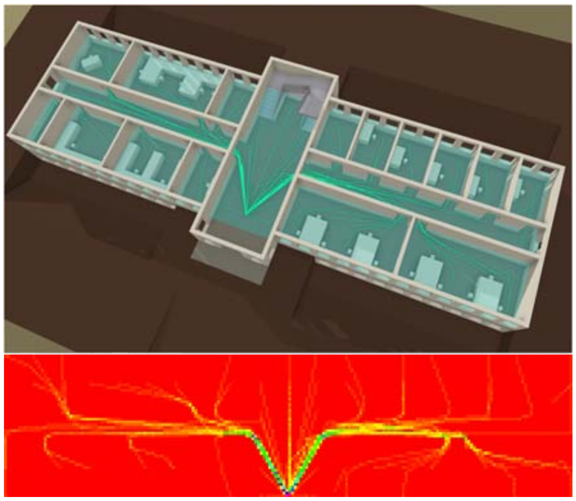

Figure 8. (**Top**) RTT* on Floor 0: 126 start poses from grid (69 invalid), 1 goal pose (Exit), avg. solution length: 14.50 [m], runtime: 8.5/85.0 [min] (low/high quality); (**Bottom**) Heatmap: 200 x 50 [px], 10 samples per [m]

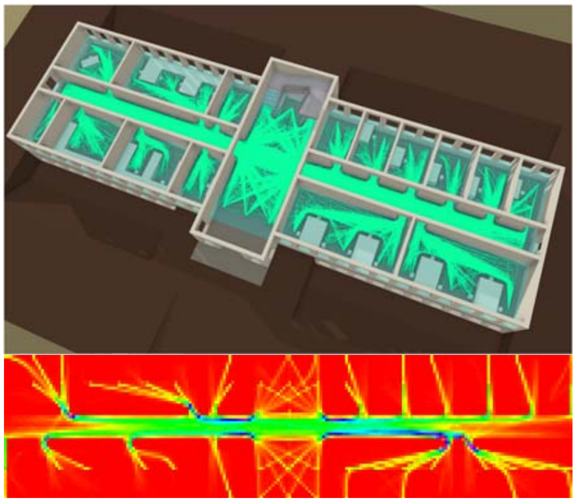

Figure 9. (**Top**) RTT* on Floor 0: 126 start poses from grid (69 invalid), 126 goal poses from grid (69 invalid), avg. solution length: 18.18 [m], runtime: 0.3/3.3 [d] (low/high quality); (**Bottom**) Heatmap: 200 x 50 [px], 10 samples per [m]

Acknowledgment

We would like to thank Peter Bonsma, main developer of the IFC Engine DLL at RDF Ltd., for sharing experiences and giving advice on using the engine on Windows 8.

References

[1] Bazjanac, V. IFC BIM-based methodology for semi-automated building energy performance simulation. In *Proc. CIB W78: Improving the management of construction projects through IT adoption*, pp. 292–299, 2008.

[2] Li, J., Hou, L., Wang, X., Wang, J., Guo, J., Zhang, S. and Jiao, Y. A project-based quantification of BIM benefits. *Int. Journal of Advanced Robotic Systems*, 11(123), 2014.

[3] Um, D., Gutierrez, M., Bustos, P. and Kang, S. Simultaneous planning and mapping (SPAM) for a manipulator by Best Next Move in unknown environments. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 5273–5278. 2013.

[4] Masehian E. and Katebi, Y. Sensor-based motion planning of wheeled mobile robots in unknown dynamic environments. *Journal of Intelligent & Robotic Systems*, 74(3-4):893–914, 2014.

[5] Li, J., Xu, A. and Dudek, G. Graphical State Space Programming: A visual programming paradigm for robot task specification. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 4846–4853, 2011.

[6] Nilsson, N. *Shakey the robot.* SRI International, Menlo Park, CA, US, Tech. Rep. Technical Note no. 323, 1984.

[7] Lozano-Perez, T.A simple motion-planning algorithm for general robot manipulators. *IEEE Journal of Robotics and Automation*, 3(3):224–238, 1987.

[8] Isikdag, U., Zlatanova, S. and Underwood, J.A. BIM-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41:112–123, 2013.

[9] Borkowski, A., Siemiatkowska, B. and Szklarski, J. Towards semantic navigation in mobile robotics. In *Graph transformations and model-driven engineering*. Springer, Heidelberg, Germany, pp. 719–748, 2010.

[10] Siemiatkowska, B., Harasymowicz-Boggio, B., Przybylski, M., Rozanska-Walczuk, M., Wisniowski, M. and Kowalski, M. BIM based indoor navigation system of Hermes mobile robot. In *Romansy 19 - Robot Design, Dynamics and Control*. Springer, Vienna, Austria, pp. 375–382, 2013.

[11] LaValle, S.M. *Planning algorithms*. Cambridge University Press, New York, NY, USA, 2006.

[12] Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G.A., Burgard, W., Kavraki, L.E. and Thrun, S. *Principles of robot motion*. MIT Press, Cambridge, MA, USA, 2005.

[13] Kavraki Lab, Rice University. Houston, TX, USA. Open Motion Planning Library. Online: http://ompl.kavrakilab.org, Accessed: 03/2016.

[14] Sucan, I.A., Moll, M. and Kavraki, L.E. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[15] Kuffner J. and LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 995–1001, 2000.

[16] Kavraki, L.E., Svestka, P., Latombe, J.-C. and Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[17] Dobson, A., Krontiris, A. and Bekris, K.E. Sparse roadmap spanners. In *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, pp. 279–296, 2013.

[18] Hsu, D., Latombe, J.-C. and Motwani, R. Path planning in expansive configuration spaces. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2719–2726, 1997.

[19] Sucan, I.A. and Chitta, S. MoveIt! Online: http://moveit.ros.org, Accessed: 03/2016.

[20] Karaman S. and Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30(7):846–894, 2011.

[21] Eastman, C., Teicholz, P. and Sacks, R. *BIM Handbook: A guide to Building Information Modeling for owners, managers, designers, engineers and contractors.* Wiley, Hoboken, NJ, USA, 2011.

[22] Hardin B. and McCool, D. *BIM and construction management: Proven tools, methods, and workflows*, Wiley, Hoboken, NJ, USA, 2015.

[23] Roßmann, J., Schluse, M., Hoppen, M., Losch, D., Hempe, N. and Schlette, C. Virtual BIM Testbeds. In *Proc. IEEE Int. Conf. Developments in eSystems Engineering (DeSE)*, 2015.

[24] buildingSMART International Ltd. Industry Foundation Classes (IFC). Online: http://www.buildingsmart-tech.org, Accessed: 03/2016.

[25] Schluse, M., Schlette, C., Waspe, R. and Roßmann, J. Advanced 3D simulation technology for eRobotics. In *Proc. IEEE Int. Conf. Developments in eSystems Engineering (DeSE)*, pp. 151–156, 2013,

[26] RDF Ltd. Bankya, Bulgaria. IFC Engine DLL. Online: http://rdf.bg, Access: 03/2016.

[27] Hoppen, M., Schluse, M., Roßmann, J. and Weitzig, B. Database-driven distributed 3D simulation. In *Proc. Winter Simulation Conference (WinTeSys)*, pp. 1–12, 2012.