3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera

Peter Biber *, Henrik Andreasson †, Tom Duckett †, Andreas Schilling *

* University of Tübingen
WSI/GRIS
Tübingen, Germany
email:{biber,andreas}@gris.uni-tuebingen.de

† Örebro University
Dept. of Technology
Örebro, Sweden
email: {henrik.andreasson,tom.duckett}@tech.oru.se

***Abstract*— We present a method to acquire a realistic, visually convincing 3D model of indoor office environments based on a mobile robot that is equipped with a laser range scanner and a panoramic camera. The data of the 2D laser scans are used to solve the SLAM problem and to extract walls. Textures for walls and floor are built from the images of a calibrated panoramic camera. Multiresolution blending is used to hide seams in the generated textures.**

I. Introduction

A 3D model can convey much more useful information than the typical 2D maps (e.g. occupancy grids) used in most current mobile robotic applications. By combining vision and laser range-finder data in a single representation, a textured 3D model can provide remote human observers with a rapid overview of the scene, enabling visualization of structures such as windows and stairs that cannot be seen in a 2D model. For these reasons, 3D models are well suited to a variety of tasks such as surveillance for security and rescue applications, self-localization, or as a background model for detection and tracking of people.

In this paper, we present an easy to use method to acquire such a model. A mobile robot equipped with a laser range scanner and a panoramic camera collects the data needed to generate a realistic, visually convincing 3D model of large indoor office environment. Our geometric 3D model consists of planes that model the floor and walls (there is no ceiling, as the model is constructed from a set of bird's eye views). The geometry of the planes is extracted from the 2D laser range scanner data. Textures for the floor and the walls are generated from the images captured by the panoramic camera. Multi-resolution blending is used to hide seams in the generated textures stemming, e.g., from intensity differences in the input images. After a brief summary of relevant techniques for generation of 3D models of real scenes, especially 3D indoor models, our method is outlined in the next section.

A. Geometric approaches

Geometric representations of scenes include triangle meshes, curve representations or simply point clouds to model surfaces. Material properties, light sources and physical models provide the basis for rendering them. While it is possible to build mobile platforms that are able to acquire surface models of real world scenes by range scan techniques [17], [10], [16] even in real-time, estimation of material properties or light sources is a hard problem in general. So to render visual information convincingly without reconstructing or simulating physical properties it has been proposed to represent real scenes directly by images.

B. Image-based approaches

Image-based rendering is a now well established alternative to rendering methods based on geometric representations. The main promise is that it is able to generate photorealistic graphics and animations of scenes in real-time [14]. Nowadays, panoramic views are the most well known variant of image-based rendering and can be discovered everywhere in the web. A user can rotate his/her view freely and can zoom in real-time (but only with a constant position). To allow all degrees of freedom, the so-called plenoptic function has to be sampled. For a static scene, this is a six-dimensional function, and is thus hard to sample and to keep in memory. Aliaga et al. [2] presented a system that allows photo-realistic walk-throughs in indoor environments. A panoramic camera mounted on a mobile platform captures a dense "sea of images", that is, the distance between two camera positions is only around 5 cm. Advanced compression and caching techniques allow walk-throughs at interactive speed. For the calculation of the camera positions, battery powered light bulbs were placed at approximately known positions. The largest area covered was $81m^2$, requiring around 10.000 images. The disadvantage of such a model is that despite its high memory requirements, only walk-throughs are possible: the user is not permitted to move too far away from a position where an image has been recorded.

It is now common to attempt to combine the best of both worlds in so-called *hybrid* approaches.

C. Hybrid approaches

Debevec et al. combined still photographs and geometric models in a hybrid approach [6]. In their work, the user had to interactively fit parametrized primitives such as boxes to the photographs to build a basic model. This model in turn was the basis of a model-based stereo algorithm, which enriched the basic model with depth maps. Finally, *view-dependent texture mapping* was used to simulate geometric details not recovered by the model. This system allows generation of photo-realistic renderings from new viewpoints, as long as there exists a still photograph taken from a position close to that new viewpoint. *Texture mapping* per se, that is, mapping the color information of an image onto a plane, belongs to the oldest class of hybrid techniques, and is still the most commonly used method in computer graphics, so acquisition of textures from real world scenes is an important topic. A representative study was done by Früh and Zakhor [8]. They presented a system that is able to generate 3D models of a city by combining textured facades with airborne views. Their model of downtown Berkeley, which is really worth a glance at, allows walk-throughs as well as bird's eye views.

Our method can be seen as a similar approach for indoor office environments, since we use a basic geometric model together with advanced texture creation and mapping methods. We emphasize especially blending methods to hide the seams when textures are generated from multiple images. In contrast to the "sea of images" approach, we recover also camera positions automatically by applying a simultaneous localization and mapping (SLAM) algorithm to the laser range-finder data.

However, our goal is not to produce photo-realistic results. Using a mobile robot driving on the floor as an image acquisition system, current techniques would allow only for walk-throughs (or drive-throughs) at a constant view height using view-dependent texture mapping. As we want our model to be viewable also from distant bird's eye views, our goal is to create visually convincing models. The acquired indoor model presented here is much larger than other indoor models reported, yet it is possible to view it as VRML model in a standard web-browser. In essence, our approach is much more similar to that of Früh and Zakhor than Aliaga et al.

II. Overview of the paper

This section gives an overview of our method to build a 3D model of an office environment by remotely steering a mobile robot through it.

At regular intervals, the robot records a laser scan, an odometry reading and an image from the panoramic camera. The robot platform is described in section III. From this data, the 3D model is constructed. Fig. 1 gives an overview of the method and shows the data flow between the different modules. Four major steps can be identified as follows (the second step, data collection, is omitted from Fig. 1 for clarity).

1) Calibration of the robot's sensors, described in section IV.
2) Data collection. The robot was controlled remotely by a human operator with a teleoperation interface via a client-server architecture. The server program running on the robot sends the sensor data and a visual image stream to the client PC (P4, 1200 Mhz), while the client sends the motor commands selected by the user back to the robot. In our implementation, new scans were recorded whenever the current robot pose changed by a distance of at least 50 cm or an angle of at least $15°$.
3) Building a 2D map. For that, the simultaneous localization and mapping (SLAM) problem first has to be solved. We do that by scan matching (section V). Lines are extracted from the generated map: these are used to provide the walls later. This is the step of geometry creation, described in section VI.
4) Creation of textures. The floor and the walls are textured to improve the visual appearance of the 3D map. The textures are generated from the images from the panoramic camera. If necessary, generated textures from different input images are fused by multi-resolution blending. This process is described in section VII.

Our method consist of manual, semi-automatic and automatic parts. Recording the data and calibration is done manually by teleoperation, and extraction of the walls is done semi-automatically with an user interface. The rest of the processing is fully automatic.

We demonstrate our method on a medium size data set covering parts of a region of about 60×50 meters. Finally we report results on this data set in section VIII and conclude with an outlook and possible future extensions.

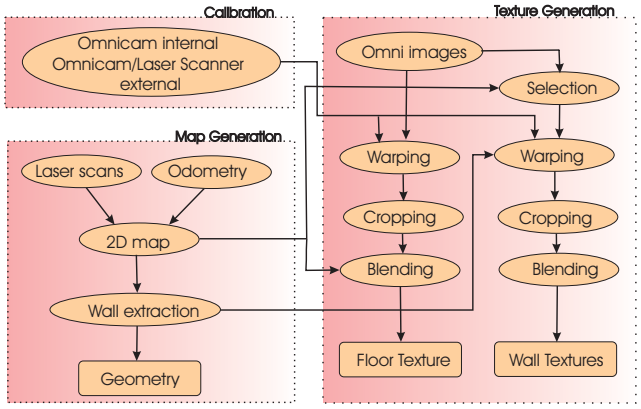

Fig. 1. An overview of our method to build a 3D model of an office environment. Shown is the data flow between the different modules.

III. Hardware platform

The robot platform used in these experiments is an ActivMedia Peoplebot (see Fig. 2). It is equipped with a SICK LMS 200 laser scanner and a panoramic camera consisting of an ordinary CCD camera with an omni-directional lens attachment (NetVision360 from Remote Reality). The panoramic camera has a viewing angle of almost 360 degrees (a small part of the image is occluded

by the camera support) and is mounted on top of the robot looking downwards, at a height of approximately 1.6 meters above the ground plane.

Fig. 2. ActivMedia Peoplebot. It is equipped with a SICK LMS 200 laser scanner and panoramic camera (NetVision360 from Remote Reality).

IV. Calibration

A. Calibration of the panoramic camera

Since the geometrical shape of the mirror inside the omni-directional lens attachment is not known, a calibration procedure was applied to map metric coordinates p onto pixel coordinates p_p (see Fig. 3). We assume that the shape of the mirror is symmetrical in all directions θ, hence it is only necessary to perform calibration in one direction, i.e., to map 2D world coordinates onto 1D pixel coordinates. Several images with known positions (r, z) and measured corresponding pixels r_p were collected. From this data the parameter h, the camera height, was estimated using $\tan\varphi = \frac{r}{h-z}$. To handle conversions from φ to r_p a polynomial of degree 3 was fitted to the data. The polynomial function was then used to interpolate on the calibration measurements for 3D modeling.

B. Joint calibration of laser, panoramic camera and ground plane

All methods in the rest of the paper assume that the laser scanner and the panoramic camera are mounted parallel to the ground plane. It is difficult to achieve this in practice with sufficient precision. While a small slant of the laser scanner has less effect on the measured range values in indoor environments, a slant of the panoramic camera has

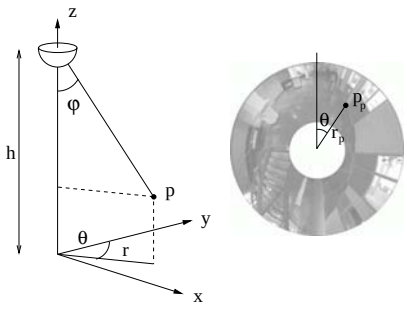

Fig. 3. Left: Geometry of the panoramic camera calibration (the half-sphere represents the surface of the mirror inside the lens attachment). Right: Geometry of the panoramic images.

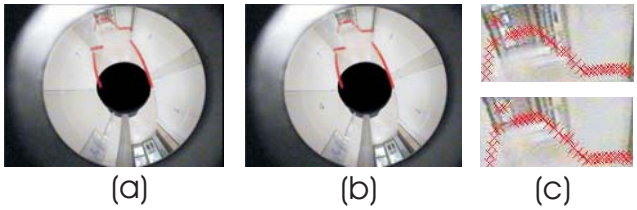

Fig. 4. Joint external calibration of laser, panoramic camera and ground plane tries to accurately map a laser scan to the edge between floor and wall on the panoramic image. (a) without calibration (b) with calibration (c) zoom

considerably more effect. Fig. 4(left) shows one panoramic image along with the corresponding laser scan mapped onto the ground plane under the above assumption. Especially for distant walls, the alignment error is considerable. As a mapping like this is used to extract textures for walls, we have to correct this error.

A model for the joint relation between panoramic camera, laser scanner and ground plane using four parameters turned out to be accurate enough: three parameters for the rotation of the panoramic camera and one for the rotation of the laser scanner around the y- ("up") axis. The parameters can be recovered automatically using full search (as the parameters' value range is small). To get a measure for the calibration, an edge image is calculated from the panoramic image. We assume that the edge between floor and wall produces also an edge on the edge image and therefore count the number of laser scan samples that are mapped to edges according to the calibration parameter. Fig 4(right) shows the result of the calibration: the laser scan is mapped correctly onto the edges of the floor.

V. Building the 2D map by scan matching

An accurate 2D map is the basis of our algorithm. This map is not only used to extract walls later, it is also important to get the pose of the robot at each time step. This pose is used to generate textures of the walls and floor. As longer walls require the fusion of textures from multiple input images, the poses (especially the orientation) need to be as accurate as possible.

Our approach belongs to a family of techniques where the environment is represented by a graph of spatial relations obtained by scan matching [13], [9], [7]. The nodes of the graph represent the poses where the laser scans were recorded. The edges represent pairwise registrations of two scans. Such a registration is calculated by a scan matching algorithm, using the odometry as initial estimate. The scan matcher calculates a relative pose estimate where the scan match score is maximal, along with a quadratic function approximating this score around the optimal pose. The quadratic approximations are used to build an error function over the graph, which is optimized over all poses simultaneously (i.e., we have $3 \times$ `nrScans` free parameters). Details of our method can be found in [4]. Fig. 5 shows a part of the map's graph and the final map used in this paper.

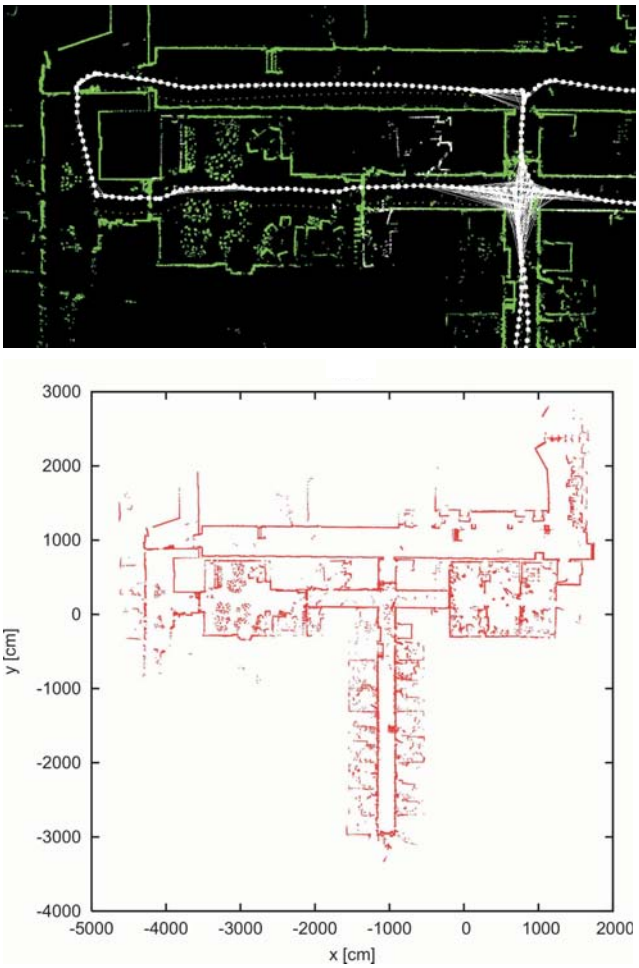

Fig. 5. Part of the graph that the map consists of (top) and final map (bottom)

VI. Generation of geometry

The geometry of our 3D model consists of two parts: the floor and the walls. The floor is modeled by a single plane. Together with the texture generated in the next section, this is sufficient: the floor's texture is only generated where the laser scans indicate free space.

The walls form the central part of the model. Their generation is the only semi-automatic step of the process, for reasons described here. The automatic part of this process assumes that walls can be identified by finding lines formed by the samples of the laser scans. So in a first step, lines are detected in each single laser scan using standard techniques. The detected lines are projected into the global coordinate frame. There, lines seeming to correspond are fused to form longer lines. Also, the endpoints of two lines that seem to form a corner are adjusted to have the same position. In this way, we try to prevent holes in the generated walls.

This automatic process gives a good initial set of possible walls. However, the results of the automatic process are not satisfying in some situations. These include temporarily changing objects and linear features, which do not correspond to walls. Doors might open and close while recording data, and especially for doors separating corridors, it is more desirable not to classify them as walls. Otherwise, the way would be blocked for walk-throughs. Also, several detected lines were caused by sofas or tables. Such objects not only caused the generation of false walls, they also occluded the real walls, which were then not detected. So we added a manual post-processing step, which allows the user to delete, edit and add new lines. Nearby endpoints of walls are again adjusted to have the same position. In a final step, the orientation of each wall is determined. This is done by checking the laser scan points that correspond to a wall. The wall is determined to be facing in the direction of the robot poses where the majority of the points were measured.

VII. Generation of textures

The generation of textures for walls and for the floor are similar. First, the input images are *warped* onto the planes assigned to walls and floor. A floor image is then cropped according to the laser scan data. Finally, corresponding generated textures from single images are fused using multi-resolution blending.

The calibration of the panoramic camera, the joint calibration of robot sensors and ground plane, and the pose at each time step allows a simple basic acquisition of textures for floor and for walls from a single image. Both floor and walls are given by known planes in 3D: the floor is simply the ground plane, and a wall's plane is given by assigning the respective wall of the 2D map a height, following the assumption that walls rise orthogonally from the ground plane. Then textures can be generated from a single image by backward mapping (*warping*) with bilinear interpolation, as is included in many image processing packages.

A. Walls

The construction of the final texture for a single wall requires the following steps. First, the input images used to extract the textures are selected. Candidate images must be taken from a position such that the wall is facing towards this position. Otherwise, the image would be taken from the other side of the wall and would supply an incorrect texture. A score is calculated for each remaining image that measures the maximum resolution of the wall in this image. The resolution is given by the size in pixels that corresponds to a real world distance on the wall, measured at the closest point on the wall. This closest point additionally must not be occluded according to the laser scan taken at that position. A maximum of ten images is selected for each wall; these are selected in a greedy manner, such that the minimum score along the wall is at a maximum. If some position along the wall is occluded on all images, the non-occlusion constraint is ignored. This constraint entails also that image information is only extracted from the half of the image where laser scan data are available (the SICK laser scanner covers only 180°), if this is possible. Finally, a wall texture is created from each

selected image, then these are fused using the blending method described in section VII-C.

B. Floor

The generation of a floor texture from a single image is demonstrated in Fig. 6. The image is warped onto the ground plane. Then it is cropped according to the laser scanner range readings at this position, yielding a single floor image. This entails again that one half of the image is not used. Such a floor image is generated from each input image. Then, these images are mapped onto the global 2D coordinate frame.

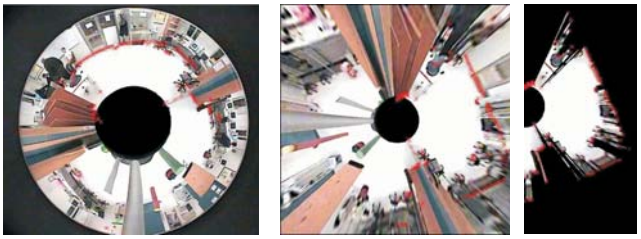

Fig. 6. Generation of floor texture from a single image.

C. Blending

Both floor and wall textures are fused from multiple input images (Fig. 7 shows an example). The fusion is faced with several challenges, among them

- image brightness is not constant,
- calibration and registration may be not accurate enough,
- parts of the input image may be occluded by the robot or support of the panoramic camera, and
- walls may be occluded by objects in front of them and thus effects of parallax play a role.

Additionally, the quality along a wall texture degrades with the distance from the closest point to the robot position (this effect is due to scaling and can be seen clearly in Fig. 7). Similar effects can be observed for floor textures. These problems also exist in other contexts, e.g. [3], [15].

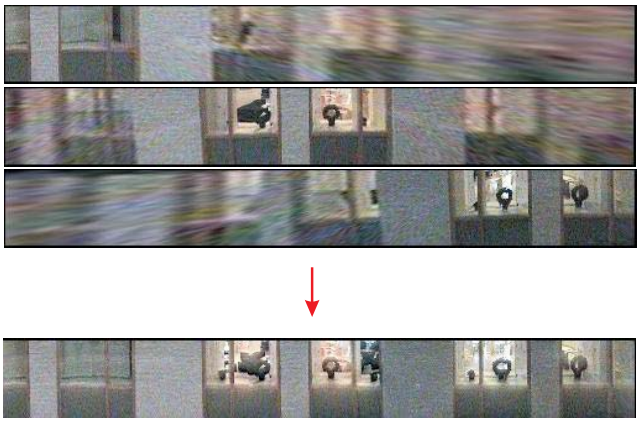

Fig. 7. Final textures of walls are generated by blending multiple textures generated from single panoramic images. Shown here are three of ten textures which are fused into a single texture.

We use an adaption of Burt and Adelson multiresolution blending [5]. The goal of the algorithm is that visible seams between the images should be avoided by blending different frequency bands using different transition zones.

The outline is as follows: a Laplacian pyramid is calculated for each image to be blended. Each layer of this pyramid is blended separately with a constant transition zone. The result is obtained by reversing the actions that are needed to build the pyramid on the single blended layers. Typically, the distance from an image center is used to determine where the transition zones between different images should be placed. The motivation for this is that the image quality should be best in the center (consider e.g., radial distortion) and that the transition zones can get large (needed to blend low frequencies). To adapt to the situation here, we calculate a distance field for each texture to be blended, which simulates this "distance to the image center". For the walls, this image center is placed at an x-position that corresponds to the closest point to the robot's position (where the scaling factor is minimal). Using such a distance field, we can also mask out image parts (needed on the floor textures as in Fig.6 to mask both the region occluded by the robot and regions not classified as floor according to the laser scanner).

VIII. Results and Conclusion

A data set of 602 images and laser scans was recorded at Örebro by teleoperation. The built map and a part of the graph containing the spatial relations was shown in Fig. 5. The graph finally contained 3299 spatial relations. Two screenshots of the resulting 3D model can be seen in Fig. 8. The model can be visualized by an own application based on Java3D. Additionally, the model can be exported as a VRML model, so that it can be viewed in a webbrowser with a VRML plugin (the screenshots are taken from the VRML viewer). The model is available for download on the project's website [1].

The resulting model allows walk-throughs as well as bird's eye views and is well suited for visualization of additional information like displaying the result of self-localization on a remote machine. Using a 3D model for tasks like this is a natural part of future work. A large part of our work was concerned with setting up the pipeline for the generation of 3D models from a mobile robot. Improving the quality of the final 3D model can be done using a second laser scanner or using stereo algorithms, with the goal of including additional objects like tables and chairs into the 3D map.

Acknowledgments

Peter Biber gratefully acknowledges his funding by a DFG grant (STR 465/11-1).The authors would also like to thank Per Larsson for developing the client-server architecture used to teleoperate the robot.

References

[1] Project's webpage. `www.gris.uni-tuebingen.de/~biber/indoor3D`.

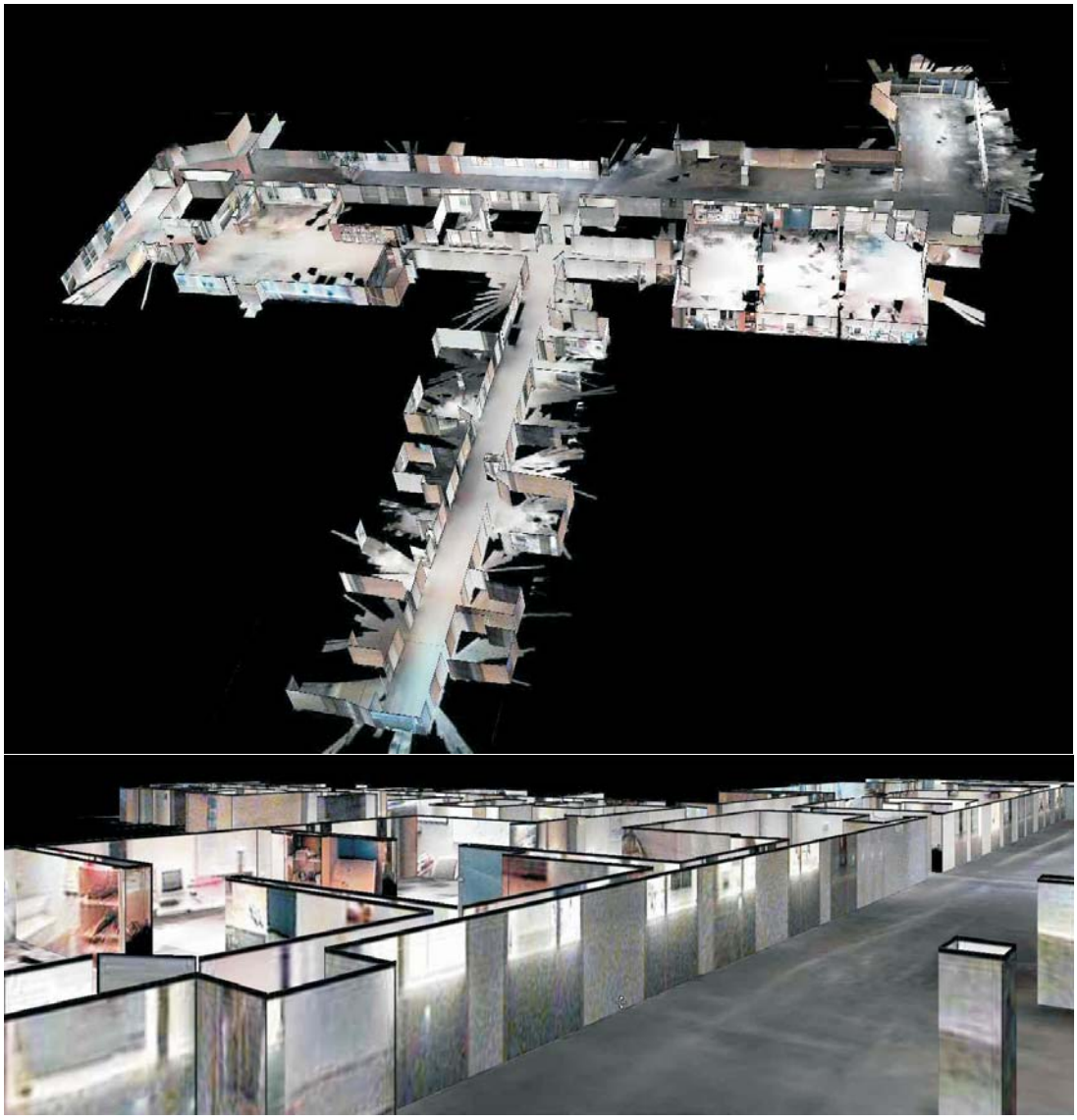

Fig. 8. Two views of the resulting VRML model.

[2] D. Aliaga, D. Yanovsky, and I. Carlbom. Sea of images: A dense sampling approach for rendering large indoor environments. *Computer Graphics & Applications, Special Issue on 3D Reconstruction and Visualization*, pages 22–30, Nov/Dec 2003.

[3] A. Baumberg. Blending images for texturing 3d models. In *Proceedings of the British Machine Vision Conference*, 2002.

[4] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[5] P. J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[6] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *SIGGRAPH 96*, 1996.

[7] Udo Frese and Tom Duckett. A multigrid approach for accelerating relaxation-based slam. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003)*, 2003.

[8] C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. *Computer Graphics and Applications*, November/December 2003.

[9] Jens-Steffen Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*.

[10] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1), 2003.

[11] L. Iocchi and K. Konolige. Visually realistic mapping of a planar environment with stereo. In *International Symposium on Experimental Robotics (ISER)*, 2000.

[12] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models with mobile robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[13] F. Lu and E.E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[14] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *SIGGRAPH*, 1995.

[15] Claudio Rocchini, Paolo Cignomi, Claudio Montani, and Roberto Scopigno. Multiple textures stitching and blending on 3D objects. In *Eurographics Rendering Workshop 1999*, pages 119–130.

[16] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4), 2003.

[17] S. Thrun, W. Burgard, and D.Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2000.