Planning Robot Motion for 3D Digitalization of Indoor Environments

Andreas Nüchter **Hartmut Surmann** **Joachim Hertzberg**

Fraunhofer Institute for Autonomous Intelligent Systems
Schloss Birlinghoven
D-53754 Sankt Augustin, Germany
{nuechter,surmann,hertzberg}@ais.fraunhofer.de

Abstract

3D digitalization of environments without occlusions requires multiple 3D scans. Autonomous mobile robots equipped with a 3D laser scanner are well suited for the gaging task. They need an efficient exploration scheme for the digitalization. We present a new approach for planning the next scan pose as well as the robot motion. Our approach calculates a collision free trajectory regarding complicated objects, e.g., with jutting out edges. A closed loop and globally stable motor controller navigates the mobile robot. The results of a 3D digitalization experiment in the main hall of castle Birlinghoven is presented.

1 Introduction

Digital 3D models are demanded in rescue and inspection robotics, facility management and architecture. Especially mobile systems with 3D laser scanners that automatically perform multiple steps such as scanning, gaging and autonomous driving have the potential to greatly advance the field of environmental sensing. Furthermore 3D information available in real-time enable autonomous robots to navigate in unknown environments.

This paper presents a planning module for an automatic system for gaging and digitalization of 3D indoor environments. The complete system consists of an autonomous mobile robot, a reliable 3D laser range finder and different software modules. The software modules are composed of a scan matching algorithm based on the iterative closest points algorithm (ICP), which registers the 3D scans in a common coordinate system and relocalizes the robot. The second component, the next best view planner, computes the next nominal pose based on the 3D data acquired so far, while avoiding 3D obstacles, by calculating a suitable trajectory.

Motion Planning has to be extended beyond basic path planning under the presence of obstacles. Visibility requirements are present and the question we are interested in is: How can a robot keep the number of sensing operations at a minimum, while allowing the data to be registered and merged into a single representation? This question combines motion planning with sensing and model construction.

The paper is organized as follows. After discussing the state of the art in the following part we present the mobile robot and the 3D laser range finder. Section 2 describes the autonomous mobile robot and AIS 3D laser range finder. The next two sections describe the next best view planner followed by a brief description of the motor controller. Section 5 concludes the paper.

Some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [1], Früh et al. [2] and Zhao et al. [3] use two 2D laser range finder for acquiring 3D data. One laser scanner is mounted horizontally and one is mounted vertically. The latter one grabs a vertical scan line which is transformed into 3D points using the current robot pose. Since the vertical scanner is not able to scan sides of objects, Zhao et al. use two additional vertical mounted 2D scanner shifted by 45° to reduce occlusion [3]. The horizontal scanner is used to compute the robot pose. The precision of 3D data points depends on that pose and on the precision of the scanner. All of these approaches have difficulties to navigate around 3D obstacles with jutting out edges. They are only detected while passing them. Exploration schemes for environment digitalization are missing.

A few other groups use 3D laser scanners [4, 5]. A 3D laser scanner generates consistent 3D data points within a single 3D scan. The RESOLV project aimed

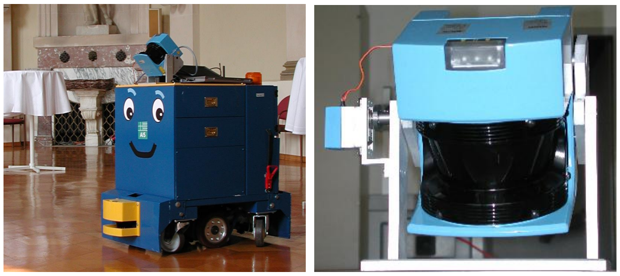

Figure 1: Left: The Ariadne robot platform equipped with the 3D scanner. Right: The AIS 3D laser scanner.

to model interiors for virtual reality and telepresence [4]. They use ICP for scan matching and a perception planning module for minimizing occlusions. The AVENUE project develops a robot for modeling urban environments [5]. They give a planning module [6], which calculates set intersections of volumes to calculate occluded volumes and to create an solid model.

2 The Autonomous Mobile Robot and the AIS 3D Laser Scanner

The ARIADNE robot (fig. 1) is an industrial robot and about 80 cm × 60 cm large and 90 cm high. The mobile platform can carry a payload of 200 kg at speeds of up to 0.8 m/s. The core of the robot is a Pentium-III-800 MHz with 384 MB RAM for controlling the AIS 3D laser range finder. An embedded PC-104 system is used to control the motor. The 3D laser scanner [7] is built on the basis of a 2D laser range finder by extension with a mount and a servomotor. The 2D laser range finder is attached to the mount for being rotated. The rotation axis is horizontal (pitch). A standard servo is connected on the left side (fig. 1) and is controlled by a laptop running RT-Linux [7].

The area of $180°(\mathrm{h}) \times 120°(\mathrm{v})$ is scanned with different horizontal (181, 361, 721) and vertical (128, 256) resolutions. A plane with 181 data points is scanned in 13ms by the 3D laser range finder, that is a rotating mirror device. Planes with more data points, e.g. 361, 721 duplicate or quadruplicate this time. Thus a scan with 181 × 256 data points needs 3.4 sec. In addition to the distance measurement the 3D laser range finder is capable of quantifying the amount of light returning to the scanner. Fig. 2 (left) shows the hall of castle Birlinghoven wheras each voxel has an intensity value. This scene is used throughout the paper.

The basis of the map building and planing module are algorithms for reducing points, line detection, surface extraction and object segmentation. Descriptions of these algorithms can be found in [7]. While scanning a scene, lines are detected in every scanned slice. These lines are merged into surfaces and are the basis of object segemntation, which marks occupied space. Fig. 2 shows the result of these algorithms.

Several 3D scans are necessary to digitalize environments without occlusions. To create a correct and consistent model, the scans have to merged in one coordinate system. This process is called registration. Variants of the iterative closest points algorithm [8] are used to calculate a rotation $\mathbf{R}$ and translation $\mathbf{t}$, which aligns the 3D scans. This transformation also corrects the estimated robot pose [9]. The 3D digitalization and map building is a stop, scan, plan and go setting. The next section describes the next best view planning module.

3 The Next Best View Planner

The autonomous robot has to plan and to drive to multiple positions to efficiently construct a complete 3D model. The next best view is the pose (x, y, θ) which has a high gain of information, is accessible by the robot and the overall robot path is short. Traditional next best view algorithms assume that the sensor head can freely move around some object [10]. In mobile robotics, the sensor head has lower degrees of freedom, in our case even a fixed height. The sensor is inside the scene and the accuracy of the pose is limited. Thus Banos et al. concluded that traditional next best view algorithms are not suitable [11].

The calculation of viewpoints, i.e., where shall we place the 3D sensor to scan the whole building without occlusions, is similar to the *art gallery problem*, i.e., given a map of a building, where shall someone place watchmen to see the whole building [12]. Map building with mobile robots requires a competitive online strategy for finding the locations from which the whole environment is seen. The next part describes an approximation of the art gallery problem and derives an online greedy version based on the algorithm of Banos et al [11]. The art gallery is modeled by a horizontal plane (2D map) through the 3D scene. The approach is extended by considering several horizontal planes at different heights to model the whole 3D environment.

3.1 Approximating Art Galleries

Suppose a 2D map of an art gallery is given by a polygon P, having n corners (vertices) and n walls

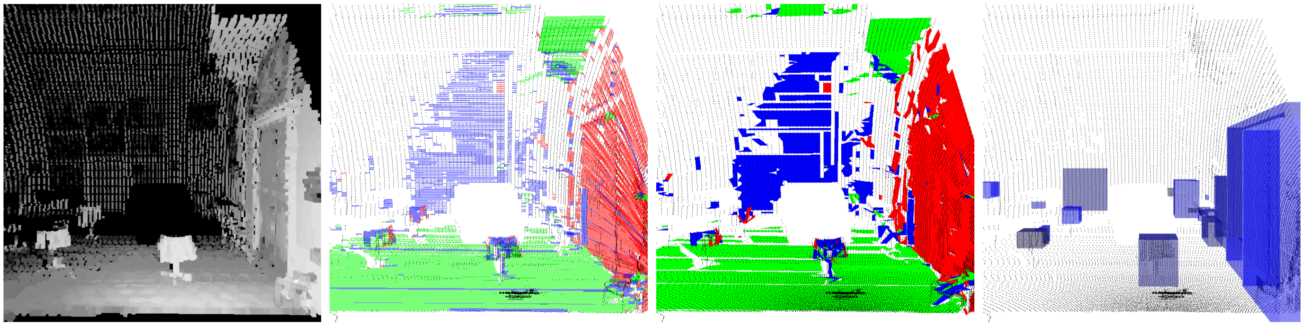

Figure 2: 3D range scan in the main hall of castle Birlinghoven. Left: The scene is shown (see fig. 5 (top row, third image) for the reflectance image). Middle: Line extraction and surface approximation. Right: Computed bounding boxes around objects.

(edges). If a watchman sees everything that lies on a straight line from his position, then the maximal number of guards needed is $\lfloor \frac{n}{3} \rfloor$ [12]. Finding the minimum number of watchmen needed for this task is NP hard, since it can be reduced to the 3-SAT problem [12]. Banos et al. reduce the art gallery problem to set cover and approximate the latter one. Set cover is defined as follows: Given a pair (X, F), where X is some finite set. F is a subset of the power set of X, i.e., $(F \subset P(X))$ and $X = \bigcup_{s \in F} s$. Find the $C \subset F$, such that $X = \bigcup_{s \in C} s$ and the set C has a minimum number of elements. The reduction of the art gallery to the set cover problem is randomized and can be stated as: 1. Generate randomly n_K candidate positions in the interior of the polygon (map). 2. Calculate for all candidate positions the part of the polygon that is visible. 3. Approximate the set cover problem, i.e., find a minimal set of candidate positions from which the entire polygon is visible. A greedy algorithm approximates the set cover problem fast and with a good approximation ratio.

3.2 Planing the Next Best View

Exploration starts with a blank map i.e., the environment is unknown. The robot's first action is to acquire a 3D scan. Based on the first scan, an initial ground plane is generated. The Hough transformation is used to detect horizontal lines in the ground plane, including lines of legth 0 (fig. 3 (left)). The found lines are transformed into polygons, whereas the edges are either labeled as *seen* or as *unseen* (fig. 3). The *seen* edges are the detected lines which are connected by *unseen* edges (third picture). The seen lines have to be sorted for connecting them. The sorting criterion is the smallest angle between the end points of a line and the scanner position. Fig. 3 (second picture) shows how to connect two lines. $\alpha_2 < \alpha_2$ and no further α exists between them.

The second step of the next best view algorithm generates randomly candidate position (x, y) in the interior of the polygon. Every position has the same probability (fig. 4 (left)).

The candidate with the most information gain is selected according to the position to the unseen lines, i.e., the direction of the next measurement has to be towards the unseen lines, since behind these lines lies unexplored area. For estimating the information gain, a vertical laser scanner with an apex angle of 180° and a distance range, e.g., [0.5m, 15m], dependent on the slice height, is simulated. The number of intersections with the polygon determines the information gain $V(\vec{p})$ and the direction θ_p of the candidate location $\vec{p} = (x, y)^T$

$$\begin{aligned} V(\vec{p}) &= \max \left\{ \sum_{i=\varphi}^{\varphi+\pi} f(\vec{p}, i) \middle| 0 \leq \varphi < 2\pi \right\} \\ \theta_p &= \operatorname{argmax}_{\varphi} \left\{ \sum_{i=\varphi}^{\varphi+\pi} f(\vec{p}, i) \middle| 0 \leq \varphi < 2\pi \right\} + \frac{\pi}{2} \\ &= \varphi_{max} + \frac{\pi}{2}, \end{aligned}$$

with $f(\vec{p}, \varphi) = 1$, if the laser beam scans an unseen edge, 0 otherwise. Fig. 4 shows 200 candidate positions inside a polygon (left) and the result of the evaluation (right).

The next best view pose (x, y, θ) has three properties: 1. The evaluated information gain value $V(\vec{p})$ of the candidate position $\vec{p}$. 2. The distance from the current position. 3. The angle to the current position. The next best view pose is an optimum of:
$\hat{V}(\vec{p}) = V(\vec{p}) \exp(-c_1 ||\vec{r} - \vec{p}||) \exp(-c_2 ||\theta_r - \theta||)$,
with $\vec{r} = (x_r, y_r)^T$ is the current robot position and θ_r the orientation. The second item prevents the robot

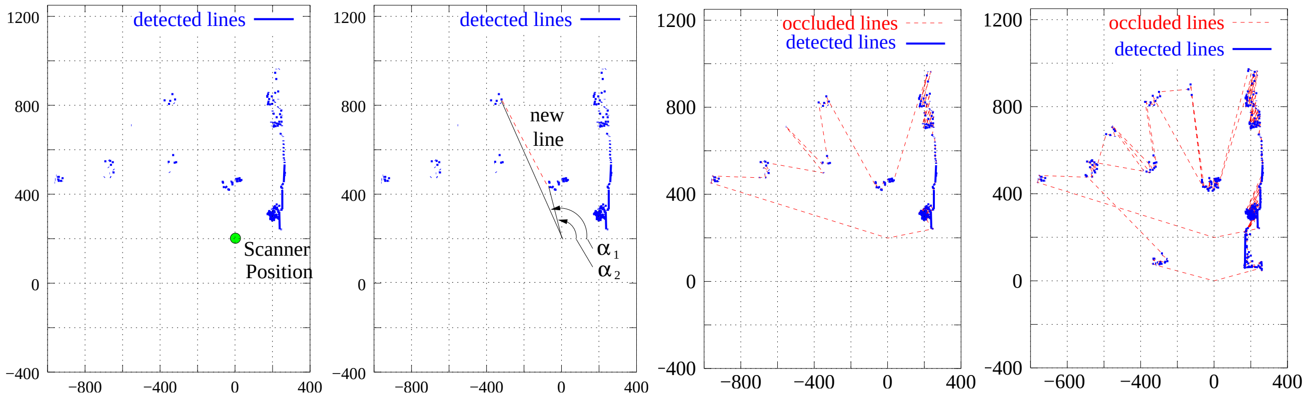

Figure 3: A ground plane of the scene inside castle Birlinghoven (fig. 2) taken at height of 50cm ± 5cm. Left: The detected lines. Second: Lines are sorted by using the smallest angle between the lines and the scanner position. Third: The polygon. Right: Two polygons with edge labels prior merging. The resulting map is given in fig. 5 (top row, rightmost illustration).

from oscillating between distant areas of high information gain. The third part penalties rotation and also prevents oscillation. The constants c_1 and c_2 determine the optimum, e.g., $c_1 = 0.05\text{cm}^{-1}$, $c_2 = 0.01$. To plan the next best view in 3D, several positions in different slices are computed and compared. In our experiments, we used one to five slices and it turned out, that one slice is sufficient in the majority of cases.

Extention to multiple 3D scans – Map building. The calculation of the next best view from multiple 3D scans is an extension to the method described above. Scan matching is used to align the 3D scans to calculate the precise scan position. Polygons are calculated from each single 3D scan and aligned with the precise position. A modified version of Vatti's polygon clipping algorithm [13] is used to merge the polygons. The modification is necessary to ensure the labeling of the edges (seen, unseen), while creating the union of the polygons (fig. 3). A new 3D scan is clipped against the union of the previous scans.

The performance of the proposed planning module can be estimated as follows: The first step converts data points into lines. The Hough transform runs in $O(d^2)$ (d is the maximal distance of a data point in a single 3D scan). Generating candidate points is done in $O(n_c n_l)$ and their evaluation is also in $O(n_c n_l)$ (n_c is the number of candidates and n_l the number of lines). Vatti's polygon clipping algorithm runs in $O(n_p)$ (n_p is the sum of edges of both polygons). The whole planning algorithm takes on scenes of $20m \times 30m$ up to 2 seconds, running on a Pentium-III-800.

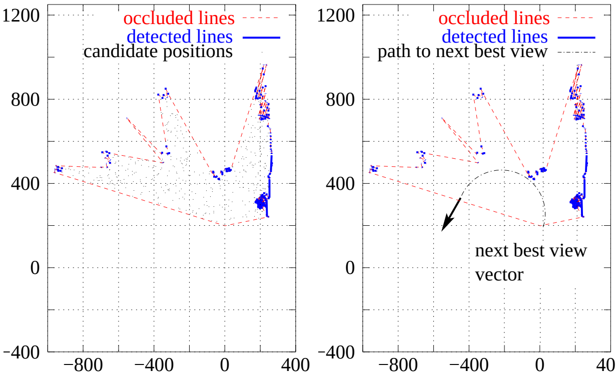

Figure 4: Left: Canidate positions in the map as generated in fig. 3. Right: A trajectory to a candidate position cannot be taken since the path intersects a bounding box of an object (fig. 2).

4 The Robot Controller

After determining the next best view, the robot has to drive to that pose. The current pose has to be connected to the target pose by a trajectory, i.e., a continuous path. The robots wheel encoders are used to calculate and estimate the robot position. Since the scan matching calculates precise positions on the base of 3D data, small odometry errors and slip can be tolerated. The computed transformation for registering the 3D scans are used to correct the self localization of the robot at every scan point.

The non holonomic robot vehicle is controlled by a closed loop, time invariant and globally stable motor controller, developed by G. Indiveri [14]. The target configuration is always approached on a straight line and the vehicle is requested to move in only one specified forward direction, thus avoiding cusps in the paths and satisfying a major requirement for the implementation of such strategy on many real systems. Let (x_G, y_G, ϕ) be the robot pose in the target centered coordinate system. The controller is based on a Cartesian kinematic model described by: $\dot{x}_G = u \cos\phi$, $\dot{y}_G = u \sin\phi$ and $\dot{\phi} = \omega = uc$. Thereby u is the robot's linear velocity, ω the angular velocity and c the (bounded) curvature. $(0, 0, 0)$ is the final position and corresponds to the next best view. The transformation of the Cartesian coordinates with $e = \sqrt{{x_G}^2 + {y_G}^2}$, $\theta = \mathrm{ATAN2}(-y_G, -x_G)$ and $\alpha = \theta - \phi$ into polar like coordinates results in: $\dot{e} = -u \cos\alpha$, $\dot{\alpha} = u(c - \frac{\sin\alpha}{e})$ and $\dot{\phi} = u\frac{\sin\alpha}{e}$.

G. Indiveri uses for the robot speed the equation $u = \gamma e$ with $\gamma > 0$ and a Lyapunov-like based control law synthesis to derive the following fomula for the curvature:

$$c = \frac{\sin\alpha}{e} + h\frac{\theta}{e}\frac{\sin\alpha}{\alpha} + \beta\frac{\alpha}{e},$$

with $h > 1$, $2 < \beta < h+1$ [14]. These two formulas for the velocity and curvature (or angluar velocity) form the closed loop, time invariant and globally stable motor controller.

One major problem in mobile robotics is obstacle avoidance. Objects with jutting out edges, e.g. standing tables are often not detected by standard sensors and the mobile robots hit these objects. The 3D laser scanner software as described above (fig. 2) computes bounding boxes in a 3D scan. These bounding boxes are joined with a planned trajectory. The trajectory is calculated with the closed loop motor controller and the physical motion model of the mobile robot. The robot drives to a candidate position if and only if a collision free trajectory exists (fig. 4).

5 Conclusions

We have presented a next best view planner for gaging and digitalization of 3D indoor environments without any intervention. The next best view planner computes next nominal poses under the conditions: Maximizing the information gain, reducing the overall robot path length, minimizing the rotation angles, and avoiding obstacles, including tricky ones like jutting out edges. Furthermore we presented a collision free trajectory planner with a closed loop stable motor controller.

3D laser range finders on mobile robots enable the automatic acquisition, surveying and mapping of entire interiors. Future work will concentrate on building semantic maps from the digitalized environment, i.e., a high level description of the scanned scene.

References

[1] S. Thrun, D. Fox, and W. Burgard, "A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000.

[2] C. Früh and A. Zakhor, "3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans," in *Proc. of the Computer Vision and Pattern Recognition Conf.*, 2001.

[3] H. Zhao and R. Shibasaki, "Reconstructing Textured CAD Model of Urban Environment Using Vehicle-Borne Laser Range Scanners and Line Cameras," in *Second Int. Workshop on Computer Vision System*, 2001.

[4] V. Sequeira, E. W. K. Ng, J. Goncalves, and D. Hogg, "Automated 3D reconstruction of interiors with multiple scan–views," in *Proc. of SPIE, Electronic Imaging '99, The Society for Imaging Science and Technology*, 1999.

[5] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer, "AVENUE: Automated Site Modelling in Urban Environments," in *Proc. of the Int. Conf. on 3DIM*, 2001.

[6] P. Allen, M. Reed, and I. Stamos, "View Planning for Site Modeling," in *Proc. of the DARPA Image Understanding Workshop*, (Monterey), 1998.

[7] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg, "A 3D laser range finder for autonomous mobile robots," in *Proc. of the of the 32nd Int. Symp. on Robotics*, 2001.

[8] P. Besl and N. McKay, "A method for Registration of 3–D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239 – 256, 1992.

[9] A. Nüchter, *Autonome Exporation und Modellierung von 3D-Umgebungen, GMD Report 157 (in German)*. 2002.

[10] J. Banta, Y. Zhieng, X. Wang, G. Zhang, M. Smith, and M. Abidi, "A "Best–Next–View" Algorithm for Three–Dimensional Scene Reconstruction Using Range Images," in *Proc. SPIE (Intelligent Robots and Computer Vision XIV: Algorithms), vol 2588*, 1995.

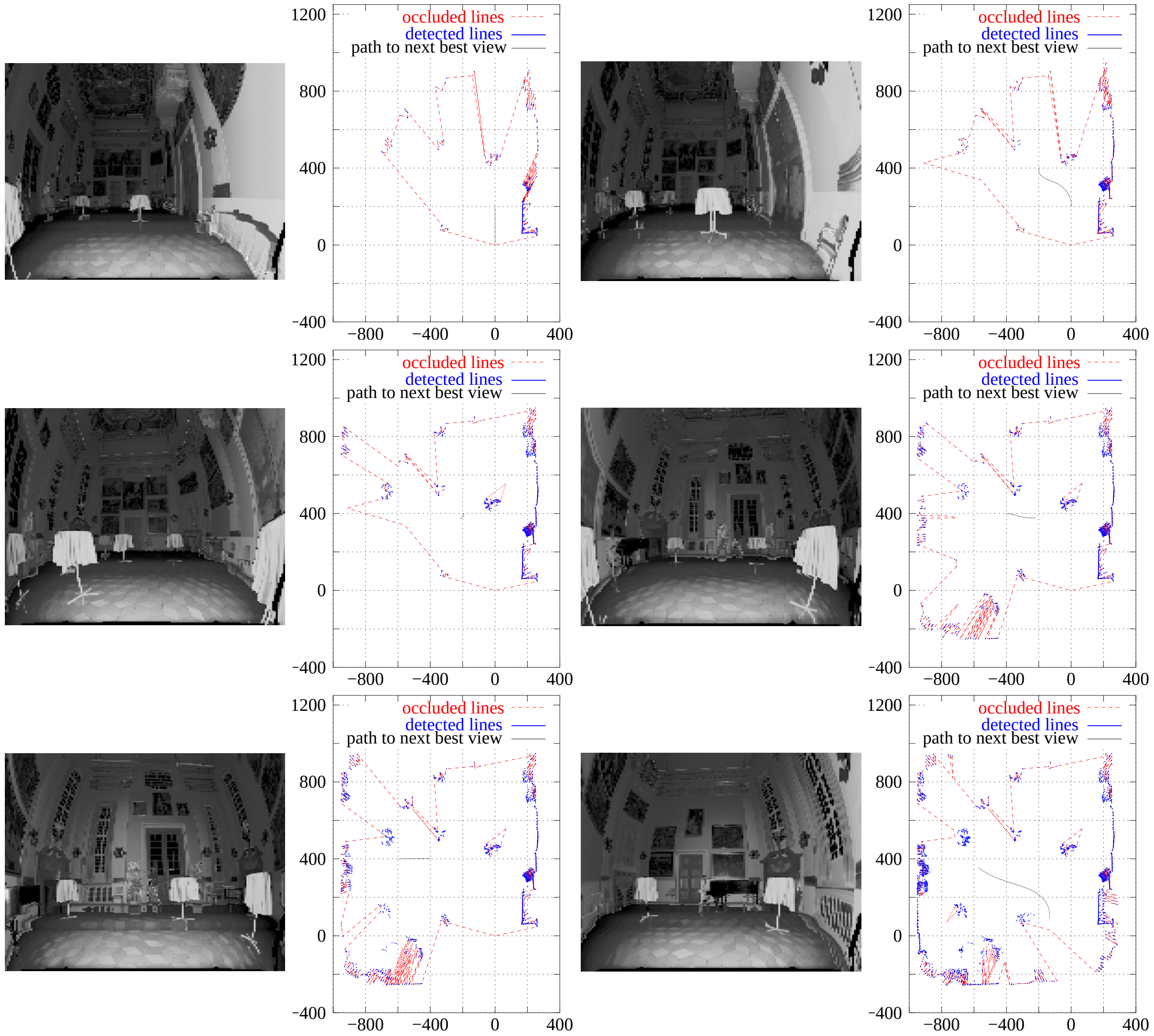

Figure 5: The first 6 scans and explorations path in the main hall of castle Birlinghoven. Reflectance images with distorted views of the scene and a horizontal map are shown (reading form left to right). The reflectance image is systematically distorted, because one scan line of the figure corresponds to a slice of the 2D scanner, thus the rotation of the scanner is not considered.

[11] H. Gonzalez-Banos, E. Mao, J. Latombe, T. Murali, and A. Efrat, "Planning Robot Motion Strategies for Efficient Model Construction," in *Robotics Research – The 9th Int. Symp.*, 2000.

[12] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[13] B. Vatti, "A Generic Solution to Polygon Clipping," *Communications of the ACM*, vol. 35, no. 7, pp. 56 – 63, 1992.

[14] G. Indiveri, "Kinematic Time-invariant Control of a 2D Nonholonomic Vehicle," in *Proc. of the 38th Conf. on Decision and Control*, 1999.

Acknowledgments

Special thanks to Kai Lingemann for creating the surface and object detection algorithms and for proofreading the paper.