Proceedings of the 32nd ISR(International Symposium on Robotics), pp. 153 - 158, 19-21 April 2001

A 3D laser range finder for autonomous mobile robots

Hartmut Surmann, Kai Lingemann, Andreas Nüchter and Joachim Hertzberg

GMD - German National Research Center for Information Technology,
AiS - Institute for Autonomous intelligent Systems
53754 Sankt Augustin, Germany
E-mail: {surmann,lingemann,nuechter,hertzberg}@gmd.de.

Abstract

This paper presents a high quality, low cost 3D laser range finder designed for autonomous mobile systems. The 3D laser is built on the base of a 2D range finder by the extension with a standard servo. The servo is controlled by a computer running RT-Linux. The scan resolution ($\sim$ 5 cm) for a complete 3D scan of an area of 150 (h) $\times$ 90 (v) degree is up to 115000 points and can be grabbed in 12 seconds. Standard resolutions e.g. 150 (h) $\times$ 90 (v) degree with 22500 points are grabbed in 4 seconds. While scanning, different online algorithms for line and surface detection are applied to the data. Object segmentation and detection are done offline after the scan. The implemented software modules detect overhanging objects blocking the path of the robot. With the proposed approach a cheap, precise, reliable and real-time capable 3D sensor for autonomous mobile robots is available and the robot navigation and recognition in real-time is improved.

1. Introduction

Prognoses at the beginning of the nineties claimed for the new millennium a number of about 50.000 independently operating autonomous service robots in different areas of production and service sectors [1]. The reality is different. In industrial environments guided vehicles, i.e. vehicles guided by a magnetic or optical track are standard [2]. Autonomous mobile service systems, i.e. systems not restricted by a track, are used extremely rarely although many research groups are working on this since several years – particularly with mobile systems for transportation tasks, e.g. [3, 4, 5, 6]. One of several reasons for the gap between prognoses and reality is the lack of good, cheap and fast sensors that allow the robots to sense the environment in real-time and to act on the basis of the acquired data.

This paper presents a 3D laser range finder designed for autonomous mobile systems (fig. 1). A large number of today's autonomous robots use 2D laser range finders as a proximity sensor. They are very fast (processing time $\sim$ 30 ms), precise ($\sim$ 1 cm, 180°) and becoming cheaper ($\sim$ \$3000) since there are at least two competing products [7, 8].

Figure 1: The mobile robot equipped with 3D laser range finder.

A few number of groups build 3D laser range finders and/or special algorithms on the basis of 2D laser range finders to overcome the drawback of only 2D proximity information that may cause problems with overhanging objects (e.g. chairs, tables or stairways) [9, 10, 11]. Furthermore, the map building and navigation process [12] is improved since objects blocking the path are detected and classified.

Thrun et al. use two laser range finders, one shifted by 90 degrees [9]. The 3D information is generated while the robot is moving. The accuracy of the acquired data depends on the accuracy of the robot pose. Object detection without moving the robot is not possible. Waltheim et al. [11] and Kristensen et. al. [10] use an amtec rotation module [13] with high accuracy at high costs ($\sim$ \$3500).

Our approach extends a standard 2D laser range finder by a low cost rotation module based on a servo motor. Combining such an extension with a set of fast algorithms has resulted in good object detection and obstacle avoiding system.

2. Constructing a 3D laser range finder

The presented 3D laser range finder is built on the basis of a 2D range finder by extension with a mount and a servomotor. The 2D laser range finder is attached to the mount so that it can be rotated. The rotation axis is horizontal. A standard servo ($\sim$ \$75) is connected on the left side (fig. 2). Annotation: An alternative approach is to rotate the range finder

around the vertical axis. Throughout this paper we will only discuss the approach based on a horizontal rotation, but all presented algorithms can be used in the same way. The differences between both approaches are the orientation of the apex angle and the effects of dynamic objects moving through the scene, e.g. persons. Using vertical scanning, a perturbation either appears with low probability within a few scans making them useless for further data processing, or does not appear at all. The first approach on the other hand shows perturbations throughout the whole scene, but these data can still be used for robot navigation and object detection (see also fig. 9).

Figure 2: The 3D laser range finder. The servo is mounted at the left side. A camera on top is used to get texture images for the realistic appearances of a 3D scene.

The given setup determines an intrinsic order of the acquired data. The data coming from the 2D laser range finder is ordered anticlockwise. In addition the 2D scans (scanned planes) are ordered due to the rotation. A digital camera for texture mapping is mounted on top of the 3D laser range finder. While rotating, serveral photos are taken so that the texture mapping can be applied to a larger area.

The 3D laser range finder uses only standard interfaces of a computer. The servomotor is directly connected to the parallel port, the 2D laser range finder to the serial port and the camera is connected to an USB port. Nowadays, every computer (esp. laptops) does have these interfaces and the built 3D laser range finder can therefore easily be used on mobile platforms.

The mount and the servo are inexpensive and no special electronic devices are used, so the price of the whole system mainly depends on the used 2D laser range finder.

The maximal scan resolution for a complete 3D scan of an area of $150°$(h) $\times$ $90°$(v) is up to 115000 points and can be grabbed in 12 seconds. Standard resolutions e.g. $150°$(h) $\times$ $90°$(v) with 22500 points are grabbed in 4 seconds. The grabbing speed can be increased by a factor of ~ 4 using a RS-422 instead of a RS-232 which is the standard interface of the laser range finder. The scan resolution is basically determined by the resolution of the laser range finder and is $\sim$ 5 cm [8]. It can be increased to $\sim$ 1 cm with other laser range finders [7].

3. Real time control and online algorithms

The servo of the new 3D laser range finder is controlled by a computer running RT-Linux, a real-time operating system which runs LINUX as a task with lowest priority. The servomotor expects a signal every 20 ms, the length of the signal determines the position of the motor (1 ms = leftmost position, 1.5 ms = middle position, 2 ms = rightmost position). This very time critical job has to be done by a real time operating system since a latency of about 10 μs corresponds to a deviation of $\sim 1°$. Real time Linux has an average latency of about 5 μs (PII-333) and thus the rotation can be realized with an average deviation of $0.5°$.

While scanning, different online algorithms for line and surface detection are applied to the data, thus no extra time is needed. Two different kinds of line detection algorithms have been tested.

The first algorithm is a simple straightforward matching algorithm running in $\mathcal{O}(n)$, (n is the number of points) with small constants. The algorithm implements a simple length comparison. The data of the laser range finder (points $a_0, a_1, \ldots, a_n$) is ordered anticlockwise so that one comparison per point is sufficient. We assume that the points $a_i, \ldots, a_j$ are already on a line. For a_{j+1} we have to check if

$$\frac{\|a_i, a_{j+1}\|}{\sum_{t=i}^{j} \|a_t, a_{t+1}\|} < \varepsilon(j). \tag{1}$$

To obtain better results this algorithm runs on preprocessed data. Data points located close together are joined so the distance from one point to the next point is almost the same. This process minimizes the fluctuations within the data and reduces the points to be processed. Hence this algorithm runs very fast, but the quality of the lines is limited.

The second line detection algorithm implemented is the well known Hough-Transform [14]. A (ρ, θ) parametrisation of the space of lines is used, where θ is the angle of the normal to the line and ρ the distance along this normal from the line to the origin of the image. Using this parametrisation, all points of a line fulfill the following equation:

$$\rho = x\cos(\theta) + y\sin(\theta). \tag{2}$$

Solving this equation for every point (x, y), that is calculating ρ for every angle θ, results in a histogram (see fig. 3). The maximum of the histogram corresponds to a line and the number of points belonging to this line is maximal.

The iteration of the following three steps returns all line segments.

1. find maximum = find straight line and calculate the line equation $y = a \times x + b$
2. tag all points belonging to this line
3. remove these points from the histogram and make line segments.

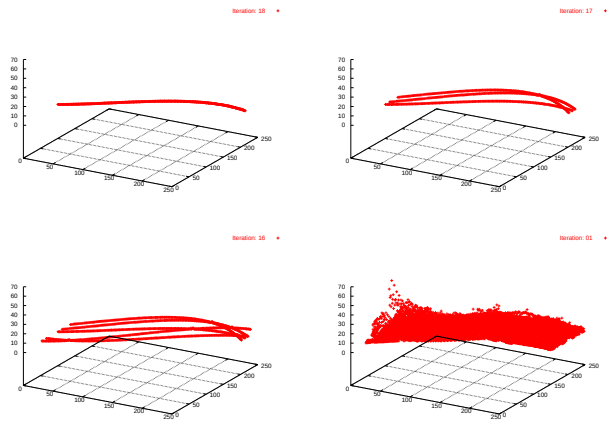

Figure 3: Histogram of the Hough transformation with 1, 3, 5 and n points

The final step uses the intrinsic order of the data as it comes from the laser range finder. Due to the intrinsic order of the data on a line only one check is necessary to determine whether a point belongs to a line segment. An advantage of the Hough transformation is that it computes the general line equation and all of the belonging line segments in one step which is helpfull in further processing steps (fig. 4).

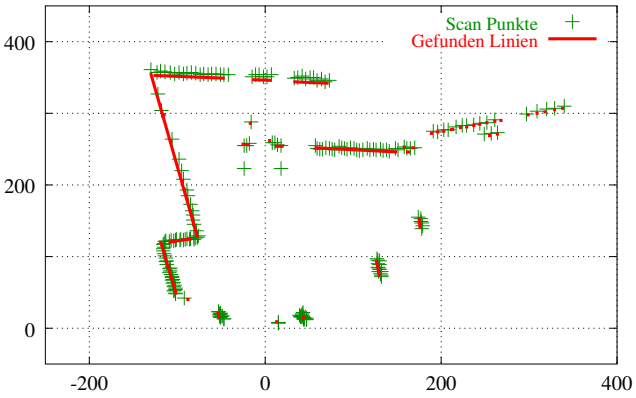

Figure 4: Lines found by the Hough transformation (one horizontal plane of 3D image of figure 9)

The Hough-Transformation runs in $\mathcal{O}(d \cdot n)$, where d is the distance to the furthest data point. This maximum distance is currently limited to 10 m, so that the transformation can be done in real-time. For indoor environments this limitation is sufficient.

After line detection the transformation of the 2D coordinates into 3D coordinates is done. All following data processing steps operate on the three dimensional data.

3.1. The third dimension

All algorithms described so far run on 2 dimensional data. After line detection is done the data is converted into 3D. Based on the detected *lines*, the following algorithm tries to detect *surfaces* in the 3-dimensional scene.

Scanning a plane surface, the line detection algorithm will return a sequence of lines in successive 2D scans approximate the shape of this surface. The task is to recognize such structures within the 3D data input and to concatenate these independent lines to one single surface. The surface detection algorithm proceeds the following steps:

0. The first set of lines – coming from the very first 2D scan – is stored.

1. Every other line is being checked with the set of stored lines. If a matching line is found, these two lines are transformed into a surface.

2. If no such matching line exists, the line may be an extension of an already found surface.
   In this case, the new line is matching with the top line of a surface. This top line is being replaced by the new line, resulting in an enlarged surface (fig. 5).

3. Otherwise the line is stored as a stand-alone line in the set mentioned above.

To achieve real time capabilities, the algorithm makes use of the characteristics of the data as it comes from the range finder, i.e. it is the order by the scanned planes. Therefore the lines are sorted throughout the whole scene (with regard to their location within the virtual scene) due to their inherited order. Thus an efficient local search can be realized.

Two criteria have to be fulfilled in order to match lines: On one hand the endpoints of the matching line must be within an ε-area around the corresponding points of the given line (fig. 5). On the other hand the angle between the two lines has to be smaller than a given value δ. The second constraint is necessary for correct classification of short lines, since they fulfill the distance criterion very easily.

To illustrate this aspect, the reader may imagine the 3D scan of a complex object, a chair, for example. The specific structure of this object will yield in a set of small, differently oriented lines, approximating the many small surfaces of the chair (fig. 11). Because of the closeness of these lines they would be transformed into some fewer, lager surfaces.

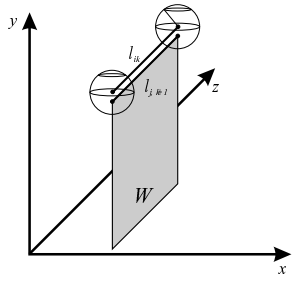

Figure 5: Expansion of a surface by a new line

This step, merely a joining of lines found in former scans, can be done online, too, since no data from future scans is necessary. This means that the robot or a user gets much information about objects in the scenery right during the scan.

4. Offline algorithms

Offline algorithms are used to create a 3D map of the room and to enable a safe navigation of the robot. Despite of the prior algorithms, this step requires information about the *whole* scene and has to be done after the scan process is finished.

Object segmentation is done by sequentially merging conglomerations of points, lines and surfaces into one object:

1. Start with one arbitrary element, e.g. a surface. This element is already treated as a "real" object in this state of the algorithm.
2. Iterate over the previously found elements and check whether there are any elements "near enough" to this object.
3. If such an element is found, the object is enlarged until it encloses this element.
4. Repeat until no more elements fit the given object.

During this process, all elements belonging to the object are marked. The object segmentation algorithm starts again, until no more conglomerations of elements exists, i.e. until no more objects can be found within the scanned scene.

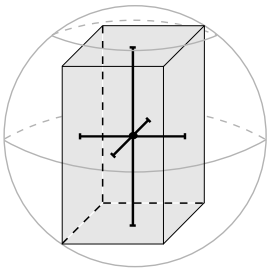

Figure 6: Representation of an object

The objects are characterized by bounding boxes (cubes) (fig. 6). A bounding box is given by 3 orthogonal lines, corresponding to the axes of the virtual world coordinate system. In addition, each object is surrounded by a sphere which encloses the whole bounding box. This sphere is used to check very fast whether a single point is near enough to this object to be examined as part of the bounding box.

The algorithm first checks if the point is inside the sphere. This can be done by a test of the euclidian distance. If the element passes this test, the algorithm has to determine whether the point is near enough to the actual bounding box. In this case, the lines defining the bounding box have to be re-adjusted to enclose this new point, the sphere likewise.

This procedure can be generalized to handle other kinds of elements as well, since lines are represented by 2 points and surfaces by 2 lines.

5. Results

To visualize the data, different viewer programs based on OPENGL and JAVA3D were implemented. The task of these programs is the projection of the 3D scene to the image plate, i.e. the monitor. A graphical user interface developed with JAVA (fig. 7) enables a simple and easy way to use the scanner application and the external viewers.

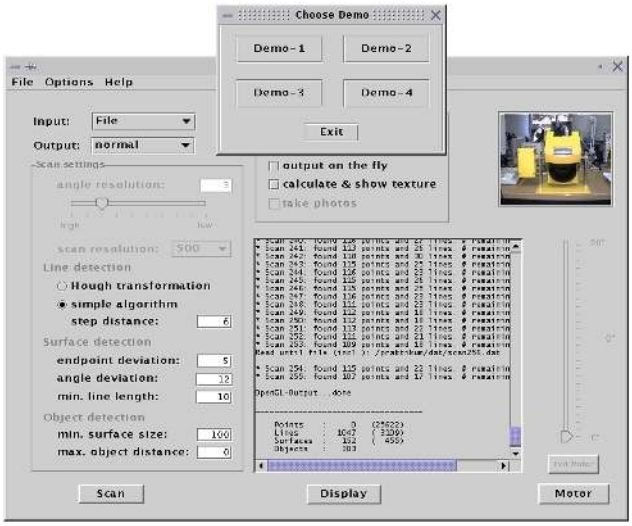

Figure 7: The JAVA user interface

The first example (fig. 8) shows a man sitting on a chair in a corridor. Durring the 12 seconds scan, a person is crossing the scene two times (at the right side). Figure 9 shows the 115000 scan points as they come from the scanner. The two pertubation can be seen as white stripes at the right side.

Figure 8: The third author sitting on a chair

Based on figure 9 the implemented algorithms compute a line view as shown in figure 10. This picture demonstrates clearly that the line detection is done in 2D planes coming from the scanner.

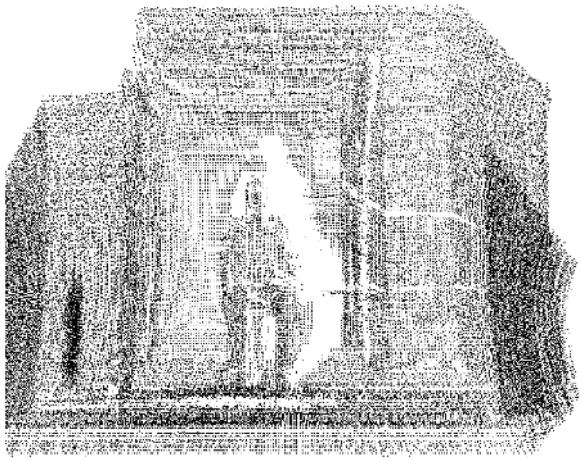

Figure 9: Scanned points

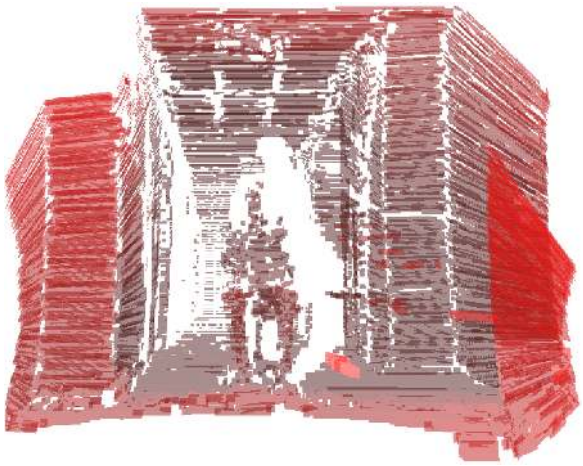

Figure 10: Detected lines in fig. 9

Figure 11: Detected surfaces in fig. 10

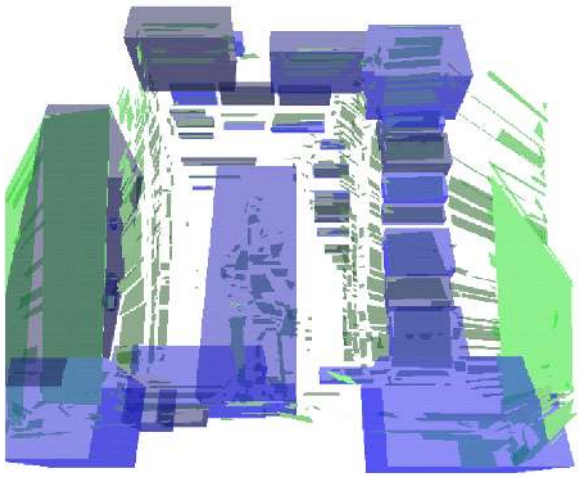

Figure 12: Detected surfaces and objects in fig. 11

Figure 11 shows the detected surfaces derived from the lines of figure 10. On the left side the lines are joined to a large surface representing the wall very well. In contrast the other side consists of several small surfaces, due to the person crossing the scene while scanning (for the dynamic view of the planes see the url [15]. This perturbation did not change the general look of the scene as shown in the previous figures.

Figure 12 illustrates the results of the object detection step. The person in the middle consisting of a bunch of surfaces is placed within one single bounding box and clearly marked as an obstacle. The pertubated surfaces on the right side are placed in different boxes. These boxes still can be used for safe navigation of a mobile robot, to derive a path around all obstacles, regardless of the scanning error.

The second example shows a stairway without any dynamic objects during the scanning process (fig 13). It shows that fine structures like the staircase railing can be detected by the system. Surfaces with a similar orientation (that is, with a sufficiently small angle between their normal vectors) get the same color. This labeling enables the algorithm to detect a wall even if it is separated into several smaller surfaces. Furthermore it is possible to detect walls as a whole even if they were interrupted by doors, for example.

6. Conclusion

This paper has presented a cheap, precise and reliable high quality 3D sensor for autonomous mobile robots. With the proposed approach a real-time capable 3D sensor is available and the robot navigation and recognition in real-time is improved. The 3D laser is built on base of an ordinary 2D range finder and uses only standard computer interfaces. The 3D scanner senses the environment contactless without the necessity of setting landmarks. The scan resolution for a complete 3D scan of an area of 150 (h) $\times$ 90 (v) degree is up to 115000 points and can be grabbed in 12 seconds. The precision is mainly determined by the laser scanner ($\sim$ 5 cm).

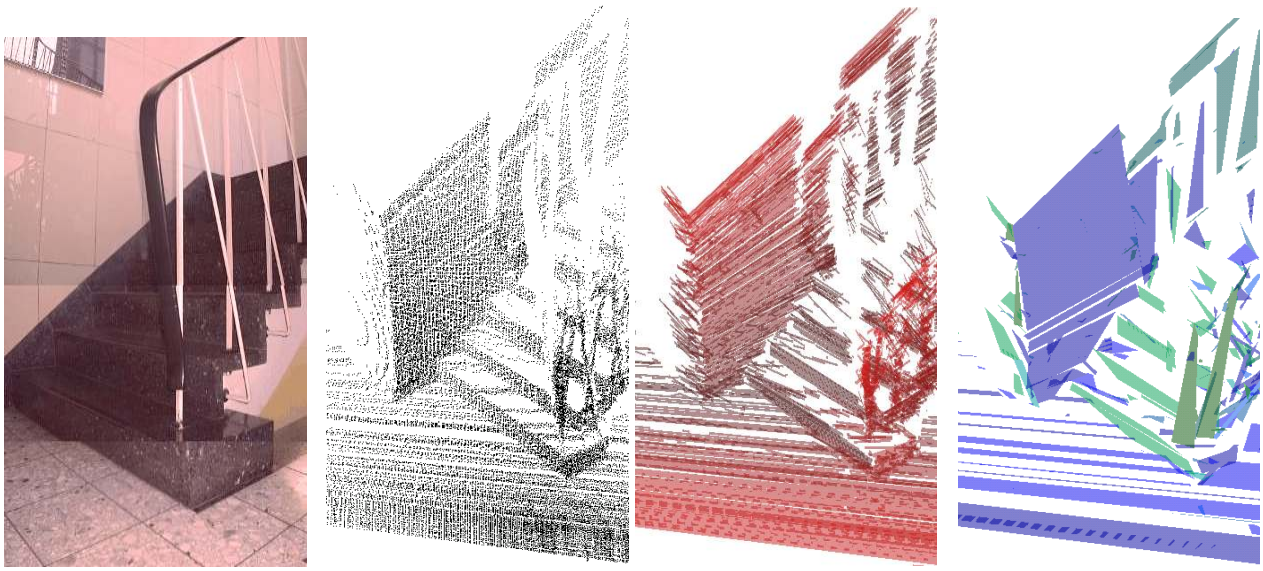

Figure 13: Picture, scan of the stairway in our building, detected lines and surfaces. The picture is assembled of 3 photos.

Standard resolutions e.g. 150 (h) × 90 (v) degree with 22500 points are grabbed in 4 seconds. While scanning, different online algorithms for line and surface detection are applied to the data. Object segmentation and detection are done offline after the scan. The implemented software modules detect overhanging objects blocking the path of the robot.

Needless to say much work remains to be done. Future work will concentrate on four further aspects.

- Improving object classification while fusing camera information and an object database.
- Improving facility management systems by constructing and updating digital building models including the inventory (scan matching).
- Increasing the rotation angle of the servo.
- Detecting non-flat or quadrangular surfaces.

References

[1] R.D. Schraft and H. Volz, *Serviceroboter, Innovative Technik in Dienstleistung und Versorgung*, Springer-Verlag, Berlin, Heidelberg, 1996.

[2] PSLT, *FTS-Anlagen-Analyse Europa: Technischer Bericht*, PSLT Hannover, 1998.

[3] Joseph F. Engelberger, "Health-care robotics goes commercial: the 'HelpMate' experience," *Robotica*, vol. 11, pp. 517–523, March 1993.

[4] P. Weckesser R. Graf, "Roomservice in a hotel," in *IFAC Symposium on Intelligent Autonomous Vehicle, Madrid*, 1998, pp. 641–647.

[5] S. J. Vestli, "Mops - a system for mail distribution in office-type buildings," *Service Robot: An International Journal*, vol. 2, no. 2, pp. 29–35, 1996.

[6] Hartmut Surmann and Antonio Morales, "A five layer sensor architecture for autonomous robots in indoor environments," in *International symposium on robotics and automation ISRA'2000, Monterrey, N.L., Mexico*, 10-12 Nov. 2000, pp. 533–538.

[7] URL: Sick optic electronic, "PLS: Definition der Telegramme zwischen Benutzerschnittstelle und Sensorsystem über RS-422 / RS-232," in *http://www.sickoptic.com/laser.htm*, 2000.

[8] URL: Schmersal EOT, "LSS 300: Telegramm zwischen Benutzerschnittstelle und Sensorsystem V.1.14," in *http://www.schmersal.de/d/produkte/n2000/laserlss.html*, 2000.

[9] Sebastian Thrun, Dieter Fox, and Wolfram Burgard, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *IEEE International Conference on Robotics and Automation, San Francisco*, 2000.

[10] Steen Kristensen and Patric Jensfelt, "Active global localisation for a mobile robot using multiple hypothesis tracking," in *Proceedings of the IJCAI'99 Workshop on Reasoning with Uncertainty in Robot Navigation, Stockholm, Sweden*, 1999, pp. 13–22.

[11] Axel Walthelm and Amir Madany Momlouk, "Multisensoric active spatial exploration and modeling," in *Dynamische Perzeption: Workshop der GI-Fachgruppe 1.0.4 Bildverstehen, Ulm*. 2000, pp. 141–146, AKA Akad. Verl.-Ges., Berlin.

[12] Hartmut Surmann and Liliane Peters, *MORIA - A Robot with Fuzzy Controlled Behaviour*, vol. 61 of *Studies in Fuzziness and Soft Computing*, chapter Layer Integration, pp. 343–365, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 2001.

[13] URL: amtec, "amtec product homepage," in *http://www.-powercube.de/index_products.html*, 1999.

[14] P. V. C. Hough, "Methods and means for recognising complex patterns," in *U.S. Patent 3 069 654*, Dec 1962.

[15] URL: GMD-AiS 3D Laserscanner, "Animated gif of a 3d scan (multiple 2d layers)," in *http://capehorn.gmd.de:8080/pictures/demo1.gif*, 2001.