Copyright © IFAC Intelligent Components and Instruments
for Control Applications, Annecy, France, 1997

AN OPTIMIZED SEGMENTATION METHOD FOR A 2D LASER-SCANNER APPLIED TO MOBILE ROBOT NAVIGATION

Ali SIADAT$^{(1)}$, Axel KASKE$^{(1)(2)}$, Siegfried KLAUSMANN$^{(1)(2)}$, Michel DUFAUT$^{(1)}$ and René HUSSON$^{(1)}$

(1) Centre de Recherche en Automatique de Nancy; CRAN-ENSEM, INPL; CNRS URA 821; 2,avenue de la Forêt de Haye; 54516 Vandœuvre-lès-Nancy Cedex (France)

(2) Universität Karlsruhe; Kaiserstraße 12; 76128 Karlsruhe (Germany)

e-mail: Ali.Siadat@ensem.u-nancy.fr

Abstract: This article presents a system for segmentation the data of a laser scanner on board a mobile robot. Three algorithms were tested in three different environments. They were compared in terms of their robust segmentation for indoor and outdoor environments, good representation of the environment and their handling for the map building. For environment modelling a polygonal representation was chosen. Furthermore, it should have good real-time capability to be able to integrate the information of the laser-scanner into the navigation algorithms of the mobile robot.

Keywords: segmentation, environment modelling, least-squares estimation, real time systems, mobile robots.

1. INTRODUCTION

For navigation a mobile robot needs a system to see its environment. Environment perception, sensor data processing and environment modelling are fundamental problems of autonomous navigation in an unknown environment [Hin88].

In recent years laser-scanners have often been used for environment perception and automatic map building in 2D or 3D especially in an outdoor environment. There are several advantages over the passive sensors or other active sensors. They are independent from illumination and object reflection. They are also more precise and cost effective. In addition, the distance scanned is longer (up to $50m$), which is an important factor for automatic map building in an outdoor environment. The disadvantage of this system is the length of time needed for a 3D-scan; a camera based system may be faster [Aya89,Gon95].

The robot also needs a computer system with special algorithms to build up its model of the world and to determine its actual position within this model. It must be able to do all the data processing and computing in real-time [Hin88].

In this paper, first, we will present a short overview of the system descriptions, the different segmentation methods, the tested algorithms and our segmentation results. Then, the representation of the environment and the navigation map building methods are described.

2. SYSTEM DESCRIPTION

This work was realised in the frame of the ROMANE project (RObot Mobile Autonome pour la Navigation en Exterieur). The aim of the project is to construct a robot that is capable of navigating autonomously in an outdoor environment. It has been equipped with a laser-scanner to get external information about the robot's environment.

The mobile robot is constructed at the base of a wheelchair. It is equipped with a VME bus carrying three MC68030 processors running at $25MHz$, a DRAM and a SRAM card. The mobile robot is connected by an Ethernet radio bridge to the workstation cluster. As an operating-system VxWorks5.2 is used, this allows an almost transparent integration of the robot-CPU's into the UNIX-environment of the workstations via Ethernet and TCP/IP (Fig. 1).

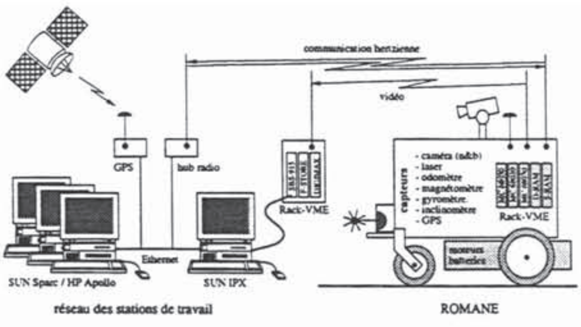


Fig. 1. Structure of ROMANE

The robot has several sensors. The odometer, gyrometer and magnetic compass are currently used to calculate the relative position of the robot while it is moving. This position determination process is done using the Kalman filter technique [Von96].

The laser scanner (PLS 101 from SICK) is mounted at the front of the vehicle and it is connected to a CPU MC68030 via RS232. To give a command to the laser, a specific telegram has to be constructed and sent to the laser via the serial device. The serial port is driven at 38400*baud* for fast data transmission. It is possible to measure the environment in a plane semicircle. This allows a minimal scan periodicity of 250*ms* (using maximal angular resolution) (Fig. 2).

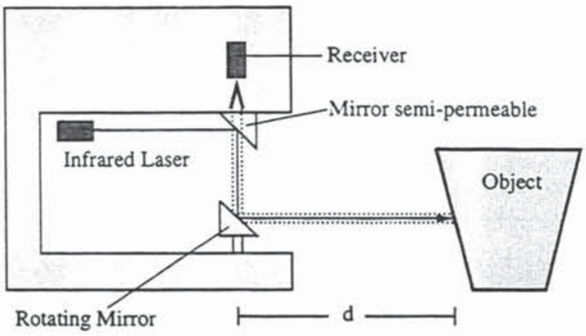


Fig. 2. Laser-Scanner

The laser-scanner has one rotating mirror (25*Hz*, counter-clockwise) and measures the distance by the principle of time of flight. The internal 3*GHz* counter (stop-watch) allows a resolution of ±5*cm* [Hey95, Sic95]. It has an angular resolution of 0.5° and a measurement range of between 7*cm* and 50*m*. The laser-scanner is able to supply its measurements of the semicircle every 40*ms* (continuously). During a first period of 20*ms* the distance measurement over the whole semicircle is done, while it takes another 20*ms* for the rotating mirror to finish a whole turn. The robot moves while measuring, introducing an additional error. The maximal values for translation speed and angular velocity of cap are listed below. With these values the maximal displacement of the robot during the scan can be estimated as:

$$V_{\max} \cong 0.4\frac{m}{s} \Rightarrow \Delta V_{\max} \cong V_{\max} \times 20_{ms} = 8_{mm} \quad (1)$$

$$\omega_{\max} \cong 10\frac{deg}{s} \Rightarrow \Delta\omega_{\max} \cong \omega_{\max} \times 20_{ms} = 0.2^\circ \quad (2)$$

This shows that the introduced error is negligible in comparison to the measurement error (±5cm) of the laser-scanner itself. The angular velocities of inclination in X and Y depend on the soil and they are also negligible. This laser-scanner is ideal for a slow moving vehicle.

3. SEGMENTATION ALGORITHMS

To get geometrical information of the data of the laser scanner, a segmentation process is required. Two main principles of treating the data are distinguished, the successive process and the total process. During the successive process the measured points are treated one by one, where as the total process works on a whole scan. The advantage of the successive algorithms is, that we do not need to wait for all data to be transmitted and so the overall time of measurement, transmission and segmentation is reduced compared to the total time of performing the three steps separately. But the total algorithms can better segment the data, because they "know" all measurements. The algorithms can also be distinguished by the segmenting technique. There are Edge Following algorithms and Line Fitting algorithms (Fig. 3).

All of the presented algorithms produce line segments. The implicit representation of a straight line is used:

$$ax + by + c = 0 \quad (3)$$

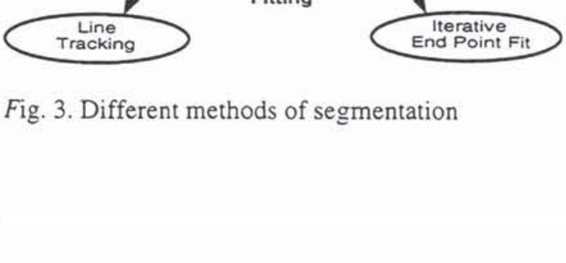
To improve the results of the segmentation a linear regression can be applied to the found segments. It fits the segments better to the original points by a least square technique (4)(5)(6)[Pre88].

$$a = \sum_i^n x_i \sum_i^n y_i^2 - \sum_i^n y_i \sum_i^n x_i y_i \quad (4)$$

$$b = \sum_i^n y_i \sum_i^n x_i^2 - \sum_i^n x_i \sum_i^n x_i y_i \quad (5)$$

$$c = \left(\sum_i^n x_i y_i\right)^2 - \sum_i^n x_i^2 \sum_i^n y_i^2 \quad (6)$$

It is not necessary to use the regression after a line tracking segmentation because the result of this algorithm is already a least square fit [Kla96, Kni91].



*F*ig. 3. Different methods of segmentation

4. TESTED ALGORITHMS

To chose the best algorithm for the data segmentation, three algorithms(Iterative End Point Fit "IEPF", Successive Edge Following "SEF" and Line Tracking "LT") were implemented and tested in different indoor and outdoor environments. The best algorithm should ensure a robust segmentation. This means that the number of segments should be as stable as possible against variations of the algorithm's segmentation parameters, and as a second point the number of segments should be as stable as possible against variations from the complexity of the environment. Furthermore the algorithm must not create artefacts.

The SEF works directly on the distances measured by the laser-scanner (no transformation to x/y Cartesian co-ordinates). A segment is completed when the difference between a distance and his following distance exceeds a given threshold (Fig. 4).

The LT does a linear regression over *n*-points. If the distance d_n (the distance of the point P_n to the already estimated straight line) is greater than a given threshold the end of a segment is found at the point (n-1) and a new segment starts (n is reset to 0). If not, we set n=n+1 and we restart with the linear regression (Fig. 5).

The IEPF is a recursive algorithm. A line defined by the first (P_0) and the last (P_n) point of the scan. Then the point with the maximum distance to this line is detected (P_j). This point divides the scan in two intervals and the algorithm starts recursively again until the maximum distance is smaller than a certain threshold (Fig. 6 & 7) [Cro89, Dud 73, Kni91].

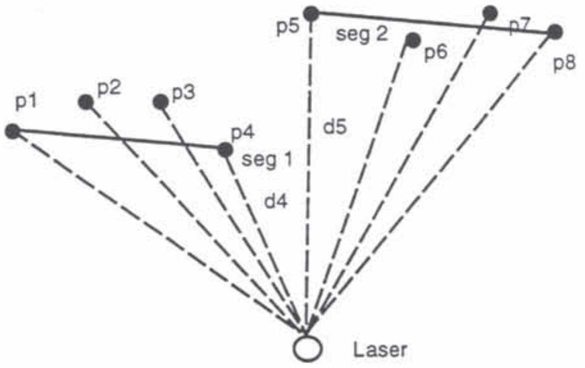


Fig. 4. SEF terminates a segment, if the difference of two successive distances is greater than the threshold T (|d4- d5| > T).

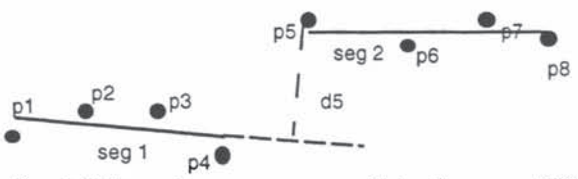


Fig. 5. LT terminates a segment if the distance of Pi to the already estimated line is grater than the threshold (d5>T).

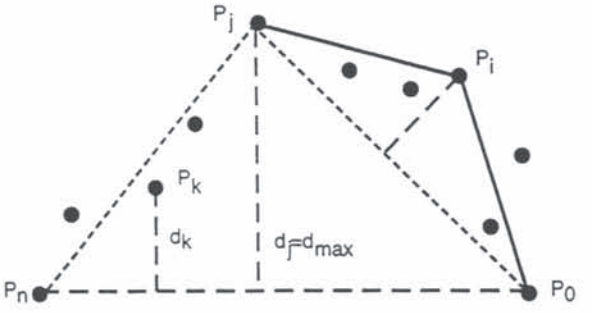


Fig. 6. IEPF algorithm searches the point P_j with the greatest distance to the straight line passing through P_0 and P_n. If this distance dj is greater than the threshold T, the line P_0,P_n is broken into two lines P_0,P_j and P_j, P_n and another segmentation step is applied for these two lines.

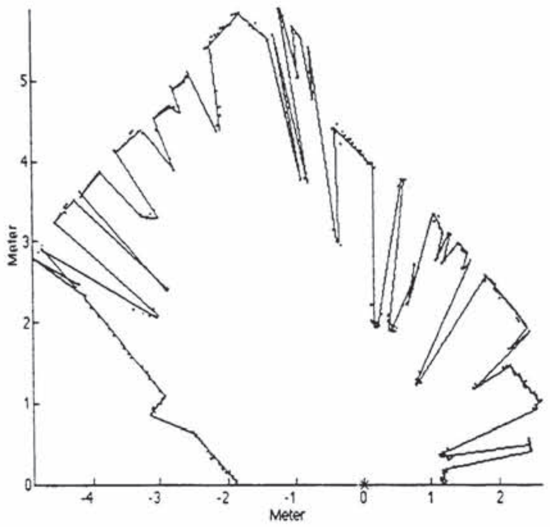


Fig. 7. IEPF segmentation of an indoor scan.

5. EXPERIMENTAL RESULTS

SEF and LT algorithms are fast, but an important improvement is the adaptation of the threshold T to the absolute distance of the point. The minimal distance between two successive points has a linear dependency with the absolute distance of the point and also the measurement error depends on the distance of the point to the laser scanner (Fig. 8).

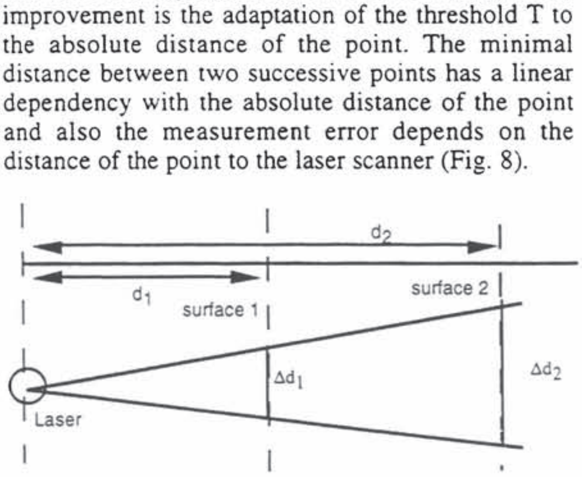


Fig. 8. The distance of two successive points (Δd1 & Δd2) depend on the distance of surface to the laser scanner (d1 & d2)

SEF normally doesn't need any pre-processing (Cartesian transformation). But linear regressing (post-processing) works on the Cartesian co-ordinate, so the pre-processing is necessary. For LT algorithm the post-processing is not necessary. It creates the least square fit segments.

IEPF is slower. It needs for pre- and post-processing and it must wait for all the data to be scanned (250 ms). But it is very robust in both indoor and outdoor environments.

The results of these algorithms for three different environments (complex indoor, less complex indoor and complex outdoor) are shown. All presented results are obtained on the mobile robot ROMANE which was following a pre-planed trajectory.

The measured computation times give a good idea of the real-time behaviour of the different algorithms. In the tables the row "T" shows the used threshold. The next four rows show the mean times of *pre*-processing, of *segmentation*, of *post*-processing and of the overall time *total*. The row "*number of segments*" shows the minimum, maximum and the mean number of segments in all scans. The mean value, s_{mean}, is the most meaningful value. It shows in combination with the criterion *visual*. Therefore a good environment representation does not correlate to the number of segments. d_{mean} is the geometric average of measure point-to-segment distances and d_{max} is the average of the maximal error of every single scan. The user symbols express very good (++) to unacceptable (- -).

Table 1: Complex indoor environment (162 scans)

complex indoor environment											
Algo-rithm	T in *cm*	computation time in *ms*				no of segments			error in *cm*		visual
		t_{pre}	t_{seg}	t_{post}	t_{total}	s_{min}	s_{mean}	s_{max}	d_{mean}	d_{max}	
SEF	6	9.8	3.3	91.5	105.3	69	128.0	198	0.36	0.70	+
	9	9.8	2.5	84.9	98.3	60	116.7	184	0.39	0.79	+
	12	**11.2**	**2.8**	**64.3**	**79.0**	**46**	**91.1**	**151**	**0.49**	**1.11**	++
	15	10.6	3.2	62.7	77.4	44	87.3	148	0.52	1.23	+
	20	10.6	2.6	54.8	68.8	34	73.3	129	0.69	1.72	+
IEPF	4	8.9	128.8	64.7	203.7	52	93.9	135	0.58	1.07	+
	6	**10.7**	**110.9**	**42.8**	**164.9**	**28**	**60.4**	**101**	**0.90**	**1.81**	++
	8	9.0	103.6	37.4	151.1	17	48.8	87	1.13	2.40	+
	10	9.4	98.0	32.5	141.7	12	42.1	83	1.34	3.00	+
	15	11.0	91.1	26.0	129.1	7	33.0	69	1.82	4.47	o
LT	4	10.4	95.7	0.0	107.0	39	86.5	144	0.96	1.92	+
	6	**10.1**	**73.7**	**0.0**	**84.7**	**21**	**50.7**	**96**	**2.16**	**4.90**	+
	8	11.0	66.7	0.0	78.3	13	38.2	75	3.42	8.52	o
	10	9.5	63.5	0.0	74.0	8	31.6	63	4.80	12.78	o
	15	11.0	58.4	0.0	69.8	6	23.3	43	8.08	24.06	o

Table 2: Structured indoor environment (110 scans)

less complex indoor environment											
Algo-rithm	T in *cm*	computation time in *ms*				no of segments			error in *cm*		visual
		t_{pre}	t_{seg}	t_{post}	t_{total}	s_{min}	s_{mean}	s_{max}	d_{mean}	d_{max}	
SEF	6	10.2	3.1	70.4	84.9	61	96.6	132	0.27	0.53	+
	9	10.5	4.0	65.5	80.2	60	90.0	119	0.27	0.54	+
	12	**8.5**	**3.1**	**47.1**	**59.8**	**34**	**61.4**	**84**	**0.28**	**0.72**	++
	15	12.4	2.7	43.8	59.5	34	69.2	82	0.29	0.75	o
	20	11.5	2.9	36.5	52.0	26	47.4	62	0.42	1.13	-
IEPF	4	11.1	121.5	43.8	177.3	17	58.8	76	0.66	1.30	+
	6	**11.1**	**80.7**	**20.5**	**113.5**	**8**	**23.2**	**44**	**1.11**	**2.48**	++
	8	10.4	61.1	15.1	88.0	4	12.6	25	1.40	3.50	+
	10	8.9	52.9	12.5	75.3	3	8.7	18	1.63	4.33	+
	15	10.0	46.0	10.4	67.3	3	6.6	13	1.99	5.55	o
LT	4	10.0	71.1	0.0	82.2	15	42.7	68	0.70	1.66	+
	6	10.2	55.1	0.0	66.2	6	16.7	34	1.14	3.04	+
	8	**10.0**	**51.6**	**0.0**	**62.5**	**3**	**9.5**	**18**	**1.74**	**4.85**	+
	10	8.9	50.5	0.0	61.1	3	7.3	14	2.35	6.67	o
	15	9.3	50.4	0.0	60.7	3	5.4	11	3.68	14.47	o

Table 3: Complex outdoor environment (110 scans)

complex outdoor environment											
Algo-rithm	T in *cm*	computation time in *ms*				no. of segments			error in *cm*		visual
		t_{pre}	t_{seg}	t_{post}	t_{total}	s_{min}	s_{mean}	s_{max}	d_{mean}	d_{max}	
SEF	**15**	**9.9**	**1.7**	**100.6**	**113.4**	**84**	**147.0**	**189**	**0.50**	**0.93**	++
	20	9.4	1.3	84.6	96.5	60	117.9	158	0.75	1.56	+
	25	8.6	2.0	73.4	85.1	50	98.5	134	1.06	2.37	o
	30	8.6	2.6	60.7	73.4	47	90.0	126	1.41	3.22	-
	35	8.3	2.4	54.1	66.2	42	75.2	118	1.78	4.04	--
IEPF	10	8.4	65.1	57.2	132.8	49	65.5	91	1.17	2.20	++
	15	8.8	60.9	50.3	120.7	44	56.2	76	1.64	3.19	++
	20	**8.8**	**61.7**	**34.9**	**107.9**	**38**	**49.0**	**63**	**2.15**	**4.28**	++
	30	8.8	56.1	31.7	98.0	30	39.2	53	3.40	7.20	+
	40	8.1	46.2	28.4	83.7	23	32.3	46	4.47	10.48	o
LT	**10**	**9.0**	**75.0**	**0.0**	**85.7**	**49**	**63.2**	**89**	**1.67**	**3.52**	++
	15	11.2	66.2	0.0	79.4	36	50.3	66	3.50	7.82	+
	20	9.5	63.9	0.0	74.7	31	42.8	58	5.80	14.20	o
	30	9.5	60.8	0.0	71.6	28	36.1	47	11.68	33.50	-
	40	9.7	53.6	0.0	65.0	17	26.2	34	17.08	47.06	-

6. ENVIRONMENT REPRESENTATION

A model or representation of the world can be created by two methods, the grid based representation and the geometric representation. The first one can be handled very easily, but is not usable in larger indoor or outdoor environments, because of the memory requirements. The second representation method is based on segments or more generally on geometric objects (primitives as lines, arcs etc. for 2D and as triangles, spherical segments, spheres etc. for 3D). The memory requirements are low and the algorithms of geometrical transformations are fast, but the automatic map building becomes more difficult than it is in the grid based methods [Cro89, Gon94, Kni91].

For the robot's representation of the environment a polygonal representation was chosen. The polygons in this representation consist of line segments. In this representation the environment is modelled by a list of line segments. Neighbouring segments are often connected to polygons. In most cases visible and invisible regions are distinguished. For all calculations, the implicit presentation of a straight line 'ax+by+c=0' is used. Thus we do not have any problems with lines parallel to the axis Y (or X) as we have with explicit presentations y=mx+b (or x=m'y+b') (Fig. 9).

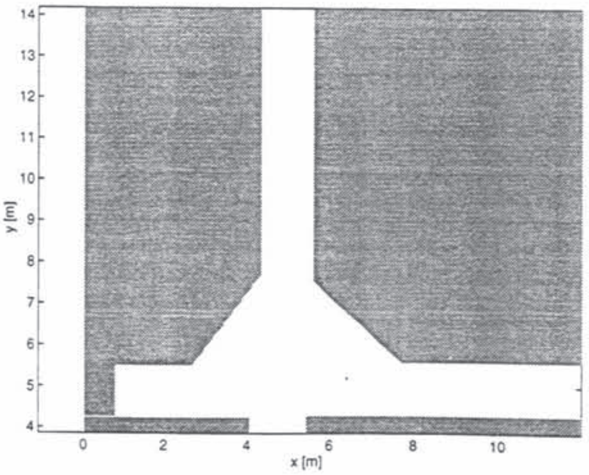


Fig. 9. Polygonal representation.

To avoid the further usage of explicit representations the vector form is used. Instead of the points P_1 and P_2 the vectors V_1 and V_2 are defined. To speed up position correction and map construction algorithms, some other informations about the line segment are held. These are the direction vector, the normal vector, the length of the segment, the number of points, and its perpendicular distance to the origin. In this representation a segment requires 48 bytes of memory (Fig. 10).

The implicit representation allows also a very fast calculation of the point-to-line distance. Firstly we need the normalised version of eq. (2):

$$\frac{(ax_i + by_i + c)}{\sqrt{a^2+b^2}} \quad (7)$$

and the normalised parameters:

$$\tilde{a} = \frac{a}{\sqrt{a^2+b^2}}$$

$$\tilde{b} = \frac{b}{\sqrt{a^2+b^2}} \quad (8)$$

$$\tilde{c} = \frac{c}{\sqrt{a^2+b^2}}$$

We obtained an oriented distance from the following equation:

$$d_i = \tilde{a}x_i + \tilde{b}y_i + \tilde{c} \quad (9)$$

Oriented distance implies that the distance is a signed value depending on whether the point P_i is located between the origin of the co-ordinate system and the line (d_i<0), or on the outer side (d_i>0) . This may be important information in a further step of automatic map building.

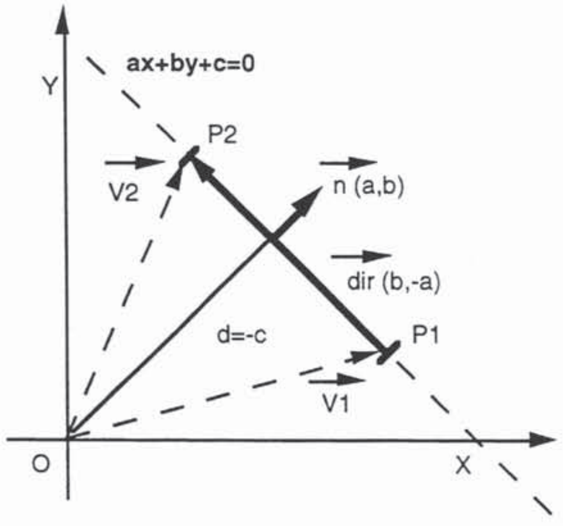


Fig .10. Vector representation.

7. NAVIGATION MAP

For the construction of a map, it is very important to have a correct position and orientation of the robot at all times. We already use an internal localisation algorithm for tracking of position and orientation [Von96]. This method accumulates the errors in movement and is very sensitive to slipping. Therefore the error between the real position and estimated position increases when the robot moves. To correct these errors, external information must be used. In our case, laser data are employed for guiding and for building a navigation map (Fig. 11).

The basic navigation map is the first segmented scan. Successively, we can terminate the position and orientation errors by correlating the present laser image and last navigation map. The new segments of laser image can be merged in the last navigation map.

We are interested in the segment based correlation techniques. Our current work is testing two correlation methods. The first is a correlation between the segmented navigation map and a segmented laser image and the second consists of the correlation between the segmented navigation map and a no-segmented laser data.

For the segment-segment correlation method IEPF algorithm is not fast enough. SEF and LT create many segments and they are not optimised for this method. Now, we are testing the segment- point correlation techniques. In this case, the laser data segmentation task can be executed both the correlation task and at the end, the segmented laser image can be used for update to our navigation map.

A Point-Point correlation method is also tested. It is not very fast and also it is not compatible for segmented navigation map building [Wei94].

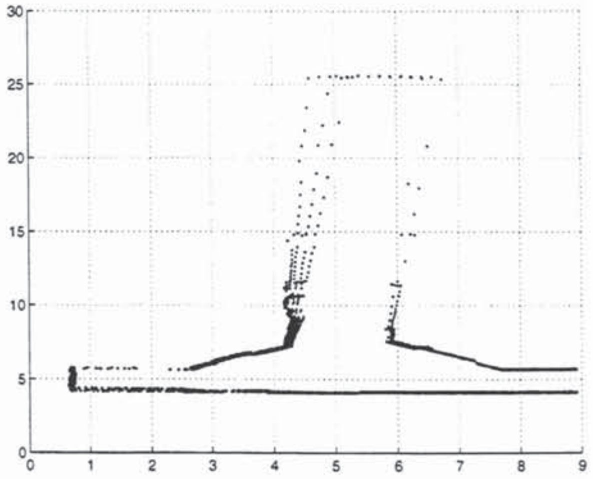


Fig 11. Navigation map with the internal localisation algorithm.

8. CONCLUSIONS

For the ROMANE perception of environment, we have chosen an industrial 2D laser-scanner. It is fast, robust, low in cost and precise enough for our application. The multi-task operating system (vxWorks) and three independent CPU allow the execution of several algorithms for processing the laser data; such as environment perception, data segmentation, environment modelling and laser based navigation in real time.

The segment based algorithms for the above tasks are usually faster and the segmented maps take up less memory. Therefore we have developed and tested several segmentation algorithms.

"Iterative End Point Fit" algorithm gives the best result and it is ideal for environment representation and building navigation maps. But it is too slow (overall time = 150ms calculation time + 250ms scan time = 400ms) for laser based navigation and position estimation. "Line Tracking" and "Successive Edge Following" give bad segmentation in certain environments. The threshold criteria must be changed for each environment, on the other hand they are fast enough for real time navigation.

Our current work is building the real time 2D navigation map, the fusion of the laser data with vision data for 3D perception and developing a 3D laser system relying on the same principles and hypothesis.

This system will be used in a co-operation project with APF (Association des Paralysés de France) for developing a semi-autonomous wheelchair for physically disable people.

9. REFERENCES

[Ayache N., Faugeras O. D. (1989)]. Maintaining Representations of the Environment of a Mobile Robot. IEEE Transactions on Robotics and Automation, vol. 5, no. 6, p. 804-819.

[Crowley J.L.. (1989)]. World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. Int. Conf. Robotics and Automation, p.674-680.

[Duda R.O. and Hart P.E.. (1973)]. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc.

[Gonzalez J., Ollero A., Reina A. (1994)]. Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder, IEEE, p. 1904-1909,1994.

[Gonzalez J., Stentz A., Reina A. (1995)]. A Mobile Robot Iconic Estimator Using a Radial Laser Scanner. Journal of Intelligent and Robotic Systems 13, p.161-179, 1995. Kluwer Academic Publishers, Netherlands.

[Heyrowsky F., Kerprich D. (1995)]. Erweiterung eines Laserscanners auf räumliche Abtastung und Integration in die Navigationssensorik eines "Autonomen Mobilen Systems". Diplomarbeit, Fachhochschule Bochum 1995, Fachbereich Elektrotechnik / Nachrichtentechnik.

[Hinkel R., Knieriemen T. (1988)]. Environment Perception With A Laser Radar in a Fast Moving Robot. IFAC Robot Control, Kalsruhe FRG, P. 271-277.

[Klausmann S. (1996)]. Segmentierung und auswertung der daten eines laserscanners für die navigation eines mobilen autonomen roboters. Diplomarbeit, Institut für prozebrechentechnik und robotik, Universität Karlsruhe.

[Knieriemen T. (1991)]. Autonome Mobile Roboter: Sensordateninterpretation und Weltmodellierung zur Navigation in unbekannter Umgebung. Bd. 80 aus der Reihe Informatik; BI-Wissenschaftsverlag.

[Press W. H., Flannery B. P., Teukolsky S. A. and Vertterling W. T. (1988)]. *Numerical Recipes in C*. Cambridge University Press.

[Sick (1995)]. *Sick PLS Laser Scanner Operating Instructions*. Erwin Sick GmbH; Optik-Elektronik; Waldkirch, Germany.

[Von der Hardt H. , Wolf D. , Husson R. (1996)]. The dead reckoning localization system of the wheeled mobile robot ROMANE. IEEE/SICE/RSJ, P. 603-610.

[Weib G., Wetzler C., Puttkamer E. (1994)] Keeping Track of Position and Orientation of moving Indoor Systems by Correlation of Range-Finder Scans. University of Kaiserslautern, Germany.