ELSEVIER

Robotics and Autonomous Systems 14 (1995) 199-212

Robotics and Autonomous Systems

Path planning and guidance techniques for an autonomous mobile cleaning robot

Christian Hofner, Günther Schmidt *

Department for Automatic Control Engineering (LSR), Technical University of Munich, 80290 München, Germany

Abstract

In the past mobile robot research was often focused to various kinds of point-to-point transportation tasks. Service tasks, such as floor cleaning, require specific approaches for path planning and vehicle guidance in real indoor environments. This article discusses automatic planning of a feasible cleaning path considering a 2D-map as well as kinematic and geometric robot models. Path construction makes use of two typical motion patterns. Each pattern is defined by a sequence of subgoals indicating robot position and orientation. Results of automatic path planning are illustrated by realistic examples of typical robots and cleaning environments. Vehicle guidance includes initialization of robot location, path execution, accurate path tracking and detection of unexpected environmental changes. Path tracking is achieved by subgoal modification *during* cleaning motion using data from the dead-reckoning and landmark localization systems. If obstacles permanently block the preplanned path, an automatic map update and path replanning is performed. Experimental results with the mobile robot MACROBE confirm the feasibility of the developed planning and guidance system.

Keywords: Automatic cleaning path planner; Long distance path tracking control; Automatic sensor application planner; Cleaning path adaptation

1. Introduction

Many challenging research problems arise in the application of free navigating mobile service robots to floor cleaning or related tasks in extended public areas such as corridors, halls or platforms [11]. Although most of today's cleaning and sweeping machines are still guided by human operators, there exist already several semi-autonomous cleaning robots for commercial usage [12]. These systems assist a human operator in cleaning of rectangular or polygonal shaped areas. Obstacles cause stops, until either the operator or the obstacle clears the blocked path. A few publications [4,2] discuss application of a mobile robot to the cleaning task without making use of explicit a-priori knowledge about geometry and complexity of the cleaning area. In spite of these developments there exist many open problems in intelligent path planning and vehicle guidance for cleaning robots due to advanced requirements as listed in Table 1.

* Corresponding author. E-mail: gs@lsr.e-technik.tu-muenchen.de; Fax: +49 89 2105-8340.

SSDI 0921-8890(94)00034-4

Table 1
Advanced requirements for an autonomous floor coverage robot

Task-related requirements	Safety/operational requirements
Identification of the cleaning area	Appropriate man-robot interface for operator assistance
Path planning considering technological and economical aspects	Collision and tumbling avoidance
Accurate long distance path tracking for satisfactory cleaning performance	Detection of obstacles with the result of velocity adaptation and path replanning
Detecting, memorizing and cleaning temporarily occupied parts of the cleaning area	

In [8,9] we have introduced basic ideas of our planning and guidance approach for autonomous floor-covering operations in extended indoor environments. In Section 2 of this paper we describe in more detail the rule-based planning system which automatically generates the motion command sequence for an appropriate cleaning path according to robot geometry and kinematic restrictions. Results from automatic planning are illustrated for robots with various degrees of agility in typical cleaning environments. In Section 3 we discuss a vehicle guidance scheme comprising regular robot localization, i.e. estimation of position and heading, as well as appropriate compensation of drift errors during path execution. Unlike obstacle avoidance approaches known from point-to-point operations we propose in Section 4 a new approach for detection and handling of obstacle situations during cleaning motion. In Section 5 we discuss the implementation of the total control structure as well as results from a floor coverage experiment evaluating the proposed path planning and tracking techniques.

2. Planning of a cleaning path

Reasonable guidance of a cleaning robot in real indoor environments requires an appropri-

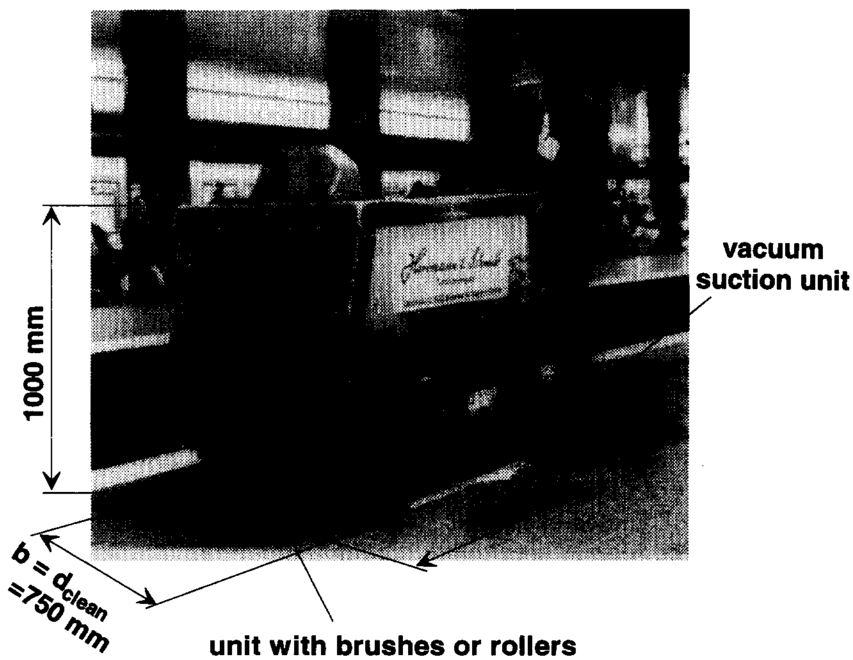


Fig. 1. Manual cleaning machine.

ately planned cleaning path. A few publications have already introduced grid-based [13] or rule-based [4] planners for complete floor coverage. However, a flexible path planner must pay attention to the technological feasibility of a cleaning path in real indoor environments and to problems arising during path execution. Thus robot geometry and agility as well as appropriate adaptation of the cleaning path to the specific environment must be considered. In this section we describe a semi-automatic approach for generation of a robot motion program which ensures most complete coverage of a specified area.

2.1. *Geometry and agility of the robot and its cleaning units*

To define relevant design parameters of mobile robots for floor coverage operations we refer to the features of typical cleaning machines as illustrated in Fig. 1. Most of these conventional systems are equipped with two or more cleaning units [6]. The front unit comprises brushes soaked in detergent. They scrub the dirt from the floor. Dirt is then removed by a vacuum suction unit at the rear end of the vehicle. For path planning we will always assume *rectangular shaped vehicles* with car-like non-holonomic kinematics but varying degrees of agility. Agility is characterized by the robot's minimum turning radius $r_{\min}$. Furthermore we will assume the existence of *two cleaning units* mounted at the front and rear end of the vehicle being as wide as the robot itself, i.e. $d_{\text{clean}} = b$.

2.2. *Representation of the cleaning area*

With respect to the environment we assume the existence of a 2D-map of a-priori known walls, pillars, staircases or other fixed objects. These items are represented by closed 2D-object polygons as illustrated in Figs. 4b and 5b.

Planning starts with operator-based identification of the borders of the cleaning area in a visualized 2D-map which is part of the planning system. Both, the borders and all object polygons inside the cleaning area are represented by contours. Each contour c is described by Cartesian end points SP_c and EP_c with respect to an inertial reference system Σ, see Fig. 3.

2.3. *Selection of a start location*

To determine the alignment of motion patterns to be planned, the operator selects in the visualized map an appropriate start position and a reference contour. Typically a point at a corner and a neighbouring wall are chosen for this purpose. Based on this information the robot's start location S as illustrated in Figs. 4b and 5b is automatically calculated. S is defined by an initial path frame $P_0 = (x_0, y_0, \psi_0)$ representing vehicle position (x_0, y_0) and heading ψ_0 with respect to Σ.

2.4. *Automatic planning of a cleaning path*

For satisfactory floor coverage a cleaning path consists typically of overlapping parallel tracks. Two basic changing maneuvers are used for connecting neighbouring tracks.

- *U-Turn* defined by desired turning radius r_{turn} and a turning angle of 180°,
- *Side-shift* defined by a concatenation of two circular arcs with different sign of curvature.

U-turn maneuvers are preferred for path planning, because no change in motion direction is required and operating time as well as floor cov-

Table 2
Basic path planning templates

Template	Purpose
TM	Move straight forward/backward
UT	Move straight forward and change track using a simple U-turn
SS	Move straight forward and change track using a Side-shift
UT-stretched	Move straight forward and change track using a stretched U-turn
UT-squashed	Move straight forward and change track using a squashed U-turn

erage redundancy can be reduced. Because of vehicle swing out during U-turns, safety distances between objects and the vehicle have to be considered [8]. On the other hand Side-shift maneuvers are preferably employed in narrow areas or in case of a lateral approach to a wall.

Once all parameters of the robot and its environment are defined, the planning procedure *automatically* constructs a *technologically feasible cleaning path* based on basic path planning templates.

Basic path planning templates

Five templates proposed for path planning are defined in Table 2 and Fig. 2. Application of a certain template depends on the robot's agility (r_{min}). For this reason, path planning distinguishes two types of robots.

$r_{min} \leq r_{turn}$: Type I robot,

$r_{turn} < r_{min} \leq 2 \cdot r_{turn}$: Type II robot,

$$\text{with } r_{turn} = \frac{d_{clean} - d_{overlap}}{2},$$

r_{turn} is the radius required for a U-turn to connect neighbouring tracks. Each track has an efficient cleaning width d_{clean} including an overlapping width $d_{overlap}$. Most complete floor coverage is achieved for both Type I and Type II robots, if the templates as illustrated in Fig. 2 are used for path construction. The determination of template parameters is discussed next.

With each template a geometric motion corridor (*rectangular, circular or combined*) is defined.

It is used for map-based collision analysis between the robot and known contours. The result is a list of discrete subgoals $P_i = (x_i, y_i, \psi_i)$ defining a collision-free section of the robot's motion path. The procedure of subgoal calculation is discussed in more detail for the *UT-template* as shown in Fig. 3. Assuming that the robot is currently located at P_{i-1}, the last subgoal from application of a preceding template, collision analysis is performed by the algorithm given in Table 3.

From the subgoals for the UT-template illustrated in Fig. 3 the planner generates the follow-

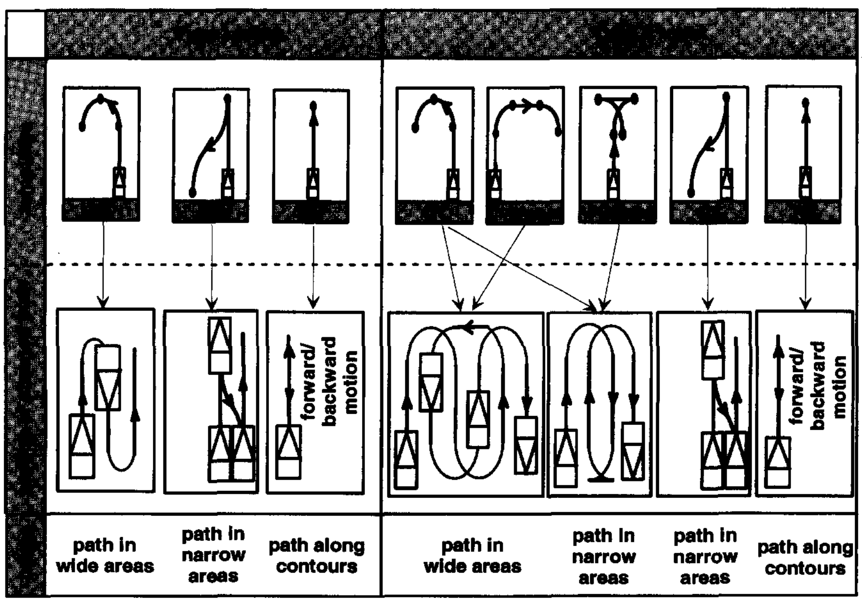


Fig. 2. Templates for path construction.

ing list of motion commands for a mobile robot with a Cartesian motion controller [10].

MOVE_RELATIVE(x_i, y_i, ψ_i, FORWARD, i),

MOVE_RELATIVE(x_{i+1}, y_{i+1}, ψ_{i+1}, FORWARD, $i+1$),

MOVE_RELATIVE(x_{i+2}, y_{i+2}, ψ_{i+2}, FORWARD, $i+2$).

Automatic selection of basic path planning templates

A complete cleaning path as illustrated in Figs. 4b and 5b is constructed by successive selection of basic path templates according to a logical sequence of rules. The *rule sequence for a Type I robot* is illustrated in Fig. 4a and can be easily verified by the planning example given in Fig. 4b. From the operator-defined start location $S = P_0$ inside the selected cleaning area the appropriate shift direction is determined. For the first section of the path in Fig. 4b the planner selects a *UT-template* with shift direction to the right. If the UT-template algorithm returns SUCCESS, the robot will be located at P_3. From P_3 the new shift direction is determined and again a *UT-template* is selected. The concatenation of feasible UT-templates leads automatically to a *Snake-trail pattern* with overlapping tracks, see Fig. 4b. If a U-turn cannot be performed due to environmental restrictions such as a side wall, an *SS-template* is called upon. The radius of the two arcs is automatically calculated for a collision free track change. Successful application of *SS-templates* leads to a *Side-shift pattern*, see Fig. 4b. Application of an *SS-template* fails if a lateral border of the cleaning area is detected inside the specified motion corridor. In this case floor coverage is

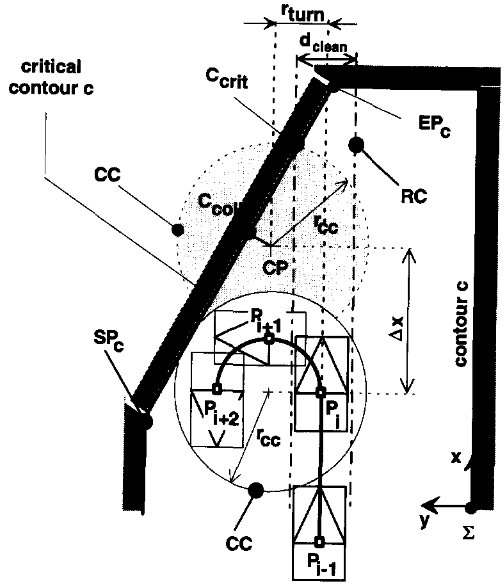


Fig. 3. Collision analysis for UT-template.

Table 3
Algorithm for subgoal calculation in case of a UT-template

1.	Define a rectangular motion corridor RC with width d_{clean} from current robot location P_{i-1}. Transform contours with respect to P_{i-1}.
2.	Calculate critical collision point $C_{crit} = (x_{crit}, y_{crit})$ between contours c and RC with respect to P_{i-1}.
3.	For a circular motion corridor CC with radius r_{cc} [a] calculate its center point CP for C_{crit} intersecting CC. The lateral displacement of CP with respect to P_{i-1} is given by r_{turn}.
4.	Consider contours to be part of CC. Determine critical contour by calculating point C_{coll} which is part of this contour while having the smallest radial distance to CP.
5.	Calculate the vertical displacement Δx of CP for CC touching the critical contour.
6.	If $\Delta x > r_{CC}$ then return FAILURE and stop else calculate P_i relative to P_{i-1} : $P_i = (x_i, y_i, \psi_i)$ and return SUCCESS.
7.	In case of SUCCESS calculate subgoals $P_{i+1} = (r_{turn}, \pm r_{turn}, \pm 90°)$ and $P_{i+2} = (r_{turn}, \pm r_{turn}, \pm 90°)$ to complete the collision-free U-turn description.

[a] r_{cc} depends on the vehicle's geometric parameters and r_{turn}.

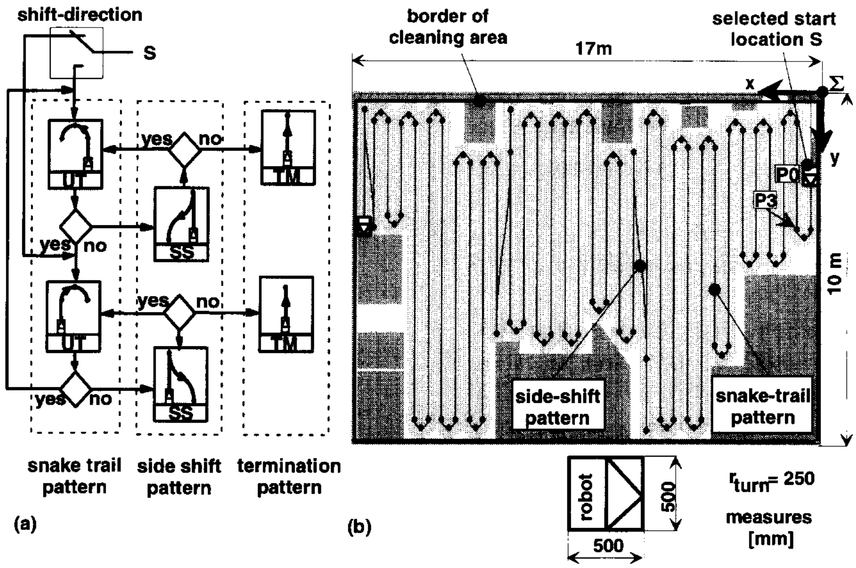


Fig. 4. Construction of a complete cleaning path for Type I robot. (a) Outline of template selection procedure. (b) Path planning example.

assumed to be completed and path planning is terminated.

Due to their size, weight and velocity Type II robots usually operate in extended cleaning areas. To achieve reasonable floor coverage with this type of robot the set of path templates needs to be extended by the so-called *stretched UT-template* and *squashed UT-template* as illustrated in

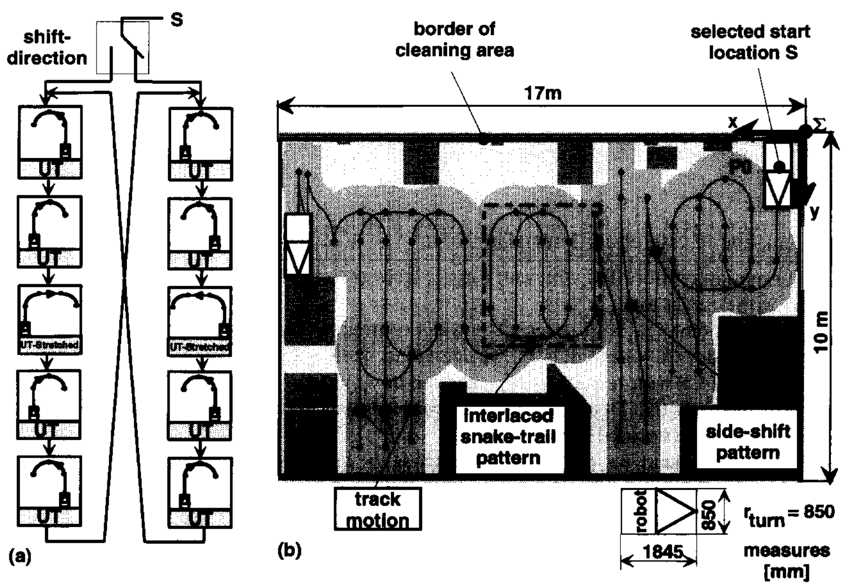


Fig. 5. Construction of a complete cleaning path for Type II robot. (a) Major cycle of template selection procedure. (b) Path planning example.

Fig. 2. Application of this set of templates requires a more sophisticated selection procedure, given by the *rule sequence for Type II robots*. In Fig. 5a only the major selection cycle is illustrated which concatenates feasible path templates to an *interlaced Snake-trail pattern*. This is equivalent to the UT-template cycle in Fig. 4a. It is obvious that additional template selections are required for completing the cleaning path, if application of a template returns FAILURE.

Application of *interlaced Snake-trail patterns* to cleaning areas with complex structured boundaries leads to unsatisfactory floor coverage. To cope with this drawback, the planning procedure tests before and after each feasible U-turn whether an additional forward/backward track motion can cover additional floor by using a *TM-template*. Examples are shown in Fig. 5b.

2.5. Evaluation of floor coverage

The ultimative objective of autonomous floor cleaning is complete coverage of a specified area at minimum operating time. However, technological restrictions and navigation in real indoor environments make it rather difficult to specify an "optimal" cleaning path. Experience has shown that reasonable floor coverage is achieved by a cleaning path composed of appropriate *Snake-trail* and *Side-shift patterns*. To evaluate floor coverage for a planned cleaning path the following performance measures are defined

$$t_{\mathrm{op}}\,[\mathrm{min}] = \sum_{i=1}^{m} \frac{s_i}{v_i} + \sum_{j=1}^{n} t_{\mathrm{acc}_j},$$

$$q_{\mathrm{total}}\,[\%] = 100 \cdot \frac{A_{\mathrm{covered}}}{A_{\mathrm{spec}}},$$

$$q_{\mathrm{red}}\,[\%] = 100 \cdot \left(\frac{A_{\mathrm{path}}}{A_{\mathrm{covered}}} - 1 \right),$$

$$p_{\mathrm{cov}} \left[\frac{m^2}{h} \right] = 60 \cdot \frac{A_{\mathrm{covered}}}{t_{\mathrm{op}}},$$

t_{op} represents the operating time for floor coverage and is calculated according to the velocity profile of the robot along its planned path. This profile considers path sections s_i with constant velocity v_i, as well as (de-)acceleration phases t_{acc_j} due to kinematic restrictions and changes in motion direction.

q_{total} represents the ratio of the actually covered area A_{covered} and the specified cleaning area A_{spec}. Due to complexity of motion patterns and the modelled environment, q_{total} is estimated by segmentation of the visualized cleaning area with methods known from image-processing.

q_{red} considers the ratio of A_{covered} and the area covered according to the planned cleaning path A_{path}. This measure indicates redundancy caused by overlapping tracks, interlaced Snake-trail and Side-shift patterns. A_{path} is determined as follows.

$$A_{\mathrm{path}} = d_{\mathrm{clean}}(h+f) + \overbrace{\sum_{i=1}^{m} A_i}^{\text{tracks}} + \overbrace{\sum_{j=1}^{n} A_j}^{\text{circles}},$$

with

$$\begin{aligned}
A_i &= d_{\mathrm{clean}} \cdot l_i, \\
A_j &= R(\varphi_{j_{\mathrm{clean}}} - h) \\
&\quad + (R^2 + h^2) \arctan(h/R) \\
&\quad + (\varphi_j/2) \cdot (f^2 - d_{\mathrm{clean}}^2), \\
R &= r_{\mathrm{turn}} + d_{\mathrm{clean}}/2,
\end{aligned}$$

h, f : distance from robot's turning axis to its rear end, resp. front end,
d_{clean} : efficient cleaning width,
l_i : length of a single track,
φ_j : turning angle,
r_{turn} : turning radius.

Cleaning performance p_{cov} is a common term in cleaning technology and represents the floor

Table 4
Results from floor coverage evaluation

Geometric data			Fig. 4b	Fig. 5b
Path length	l	[m]	236.6	144.6
Average velocity	$\bar{v}$	[mm/s]	430	460
Specified floor	A_{spec}	[m^2]	124.6	124.6
Covered floor	A_{covered}	[m^2]	113.1	101.1
Coverage along the path	A_{path}	[m^2]	122.8	158.3
Performance value			**Fig. 4b**	**Fig. 5b**
Operating time	t_{op}	[min]	9.5	6.4
Total coverage	q_{total}	[%]	90.8	81.0
Coverage redundancy	q_{red}	[%]	8.5	56.6
Cleaning performance	p_{cov}	[m^2/h]	715.7	953.0

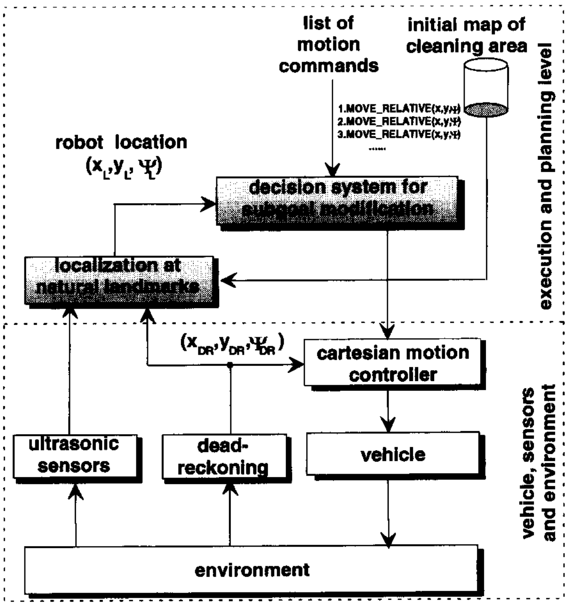


Fig. 6. Control structure for path tracking.

space in square meters which can be covered within one hour.

The described path planning system automatically generates performance values for a planned cleaning path. Table 4 summarizes evaluation results for the examples of Figs. 4b and 5b. Analysis of the quality values indicates, that achieved floor coverage and cleaning performance depend on the specific environment and the appropriately selected robot technology (geometry, agility etc.). The proposed quality measures strongly support further improvement of path planning techniques with respect to the ultimative objectives of cleaning operations.

3. Vehicle guidance

Two important issues must be considered for successful execution of a planned cleaning path.

- Finding the specified start location S of the cleaning path from an arbitrary robot location in the vicinity of S, an operation which we call initialization.
- Compensation of path errors in order to achieve satisfactory floor coverage over long distances.

Both operations are supported by a combined sensor and map based self-localization procedure.

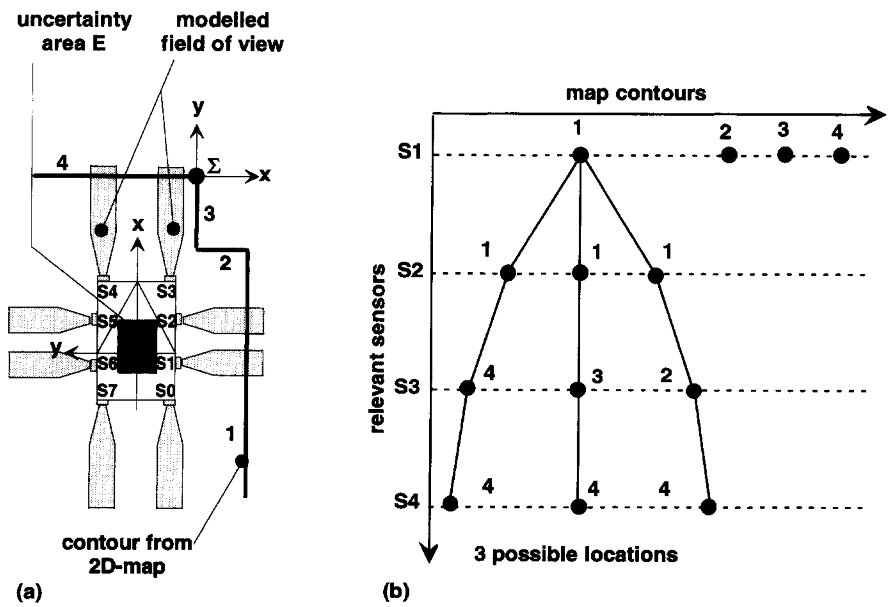


Fig. 7. Self-localization at natural landmarks. (a) Configuration of range sensors. (b) Correspondence tree.

The related robot control structure is illustrated in Fig. 6.

3.1. Initialization of robot motion

Cleaning motion starts at the selected start location S of the planned cleaning path, e.g. a room corner as illustrated in Figs. 4b and 5b. Since a human operator cannot be expected to position the robot precisely at S, but somewhere in its vicinity, self-localization is performed and the robot maneuvers to S. The maneuver is planned and executed by a local path planning and execution module which is part of the robot control system as described in [1]. Self-localization performed during initialization and path execution is briefly discussed in the following section.

3.2. Self-localization

For estimating robot position and heading in the environment, natural landmarks detected by sensors are matched with corresponding contours in the 2D-map. For *landmark detection ultrasonic sensors* with a maximum range of 1.5 m [3] are used. A minimum configuration of eight sensors mounted around a rectangular shaped robot is illustrated in Fig. 7a. Taking into account typical problems arising with ultrasonic range sensors, a reliable absolute localization technique as reported in [7] was adopted for vehicle guidance. The main processing steps for estimating robot location $L = (x_L, y_L, \psi_L)$ by means of ultrasonic range measurements and *dead-reckoning data* are summarized in Table 5.

Position and heading of the robot cannot be completely determined if either the robot moves along or towards a single contour or if faulty sensor measurements have to be eliminated. In this case missing components of L are replaced by the corresponding values of the dead-reckoning system. This avoids undefined situations when path tracking errors must be compensated.

3.3. Automatic planning of sensor application

Automatic planning of sensor application, i.e. selection of appropriate localization points, is supported by the subgoals already determined during path planning. Since subgoals are typically located in the vicinity of landmarks such as walls or pillars, they are easily detectable by ultrasonic sensors. Localization points are selected from subgoals according to the following criteria:

- Landmarks in the vicinity of a subgoal must be detectable by at least two sensors. This is a prerequisite for deriving a good estimate of current heading which is a most sensitive parameter for accurate path tracking during cleaning. For checking this condition the set of intersections between the modelled sensors' field of view and the respective contours of the 2D-map is tested for the robot positioned at the subgoal, see Fig. 7a.
- Path length between two localization points should not exceed a threshold value given by the errors of the dead-reckoning system and

Table 5
Main steps of location estimation

1. Self-localization starts with a rough estimate of the robot's current location characterized by an uncertainty area E, see Fig. 7a.
   - For robot initialization, E is parameterized by the path planner around the selected start location S.
   - During path execution, E is parameterized around the current robot location $O = (x_{DR}, y_{DR}, \psi_{DR})$, given by the dead-reckoning system.
2. When approaching a landmark a correspondence tree as shown in Fig. 7b is generated. Each branch represents the set of map contours detectable by relevant ultrasonic sensors. Different branches are generated by variation of robot location in E. By use of appropriate models of the sensors' field of view, the number of possible branches can be kept small.
3. Based on map and current sensor data a possible robot location and a quality measure is calculated for each branch using least-squares-fitting, see [7].
4. The robot's true location $L = (x_L, y_L, \psi_L)$ is selected from the set of possible locations by use of the quality measures.

the specified overlap d_{overlap} of neighbouring tracks.

If a subgoal meets the above requirements, the corresponding motion command is marked by a so-called localization flag. If such a command is passed to the Cartesian motion controller during path execution, vehicle guidance initiates self-localization when the robot approaches the subgoal. Localization points as selected by automatic sensor application planning are given in Fig. 10b.

3.4. Compensation of path tracking errors

For compensation of tracking errors during path execution the following data are available

- current subgoal $P_i = (x_i, y_i, \psi_i)$,
- current location $L = (x_L, y_L, \psi_L)$ estimated by self-localization, see Section 3.2,
- current location $O = (x_{DR}, y_{DR}, \psi_{DR})$ provided by the dead-reckoning system.

As all data refer to the same reference system Σ, transformation vectors $\boldsymbol{\delta}_L$ between L and P as well as $\boldsymbol{\delta}_{DR}$ between O and P are calculated. The robot's errors in heading and position are then given by $\boldsymbol{\epsilon} = |\boldsymbol{\delta}_L - \boldsymbol{\delta}_{DR}|$. Error compensation is activated according to the following criterion:

$$\begin{array}{c}\text{Compensation}\\ \text{activated}\end{array} \begin{cases} \text{YES} & \text{if } \bigvee_{i=1}^{3} |\delta_{l_i} - \delta_{DR_i}| > \epsilon_{\max_i}, \\ \text{NO otherwise.} & \end{cases}$$

The threshold $\boldsymbol{\epsilon}_{\max}$ is known from experiments. Errors are compensated *during* robot motion by appropriate modification of subgoal values generated by the path planner. If localization fails locally, compensation is not initiated and the robot continues its path to the current subgoal. A more detailed description of this procedure is given in [8].

4. Obstacle avoidance

Obstacle avoidance strategies are constrained by requirements of satisfactory floor coverage. This means among others that temporary obsta-

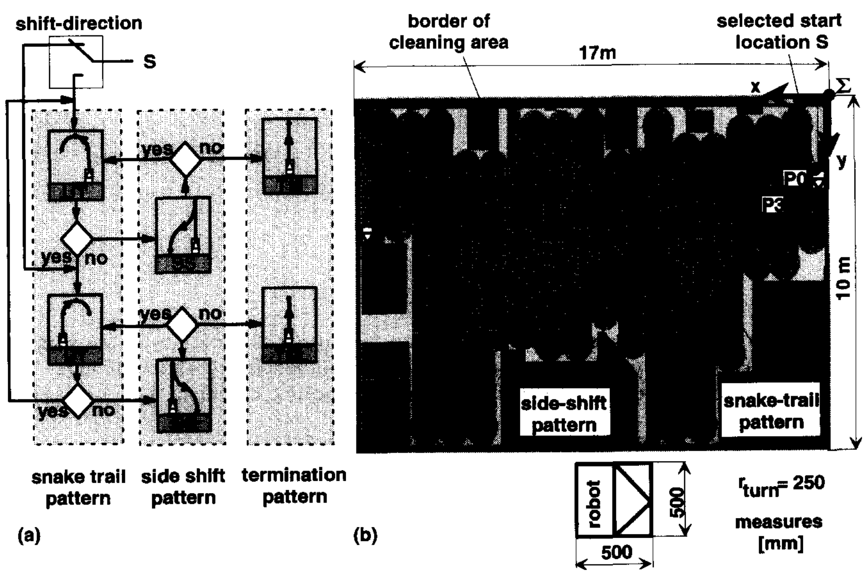


Fig. 8. Obstacle avoidance. (a) Obstacle detection. (b) Replanned cleaning path after map update.

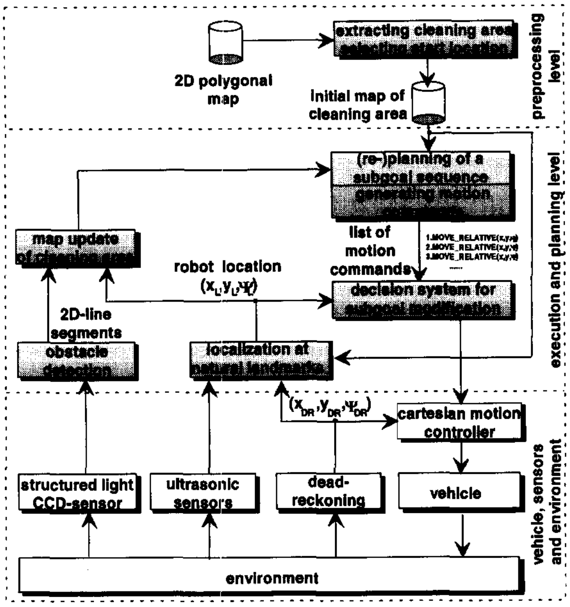


Fig. 9. Overall control structure for floor coverage robot.

cles should not cause immediate path modification, while permanent obstacles should be passed according to task-related requirements.

4.1. Obstacle detection

A structured light CCD-sensor as described in [5] is used for obstacle detection. It provides 2D line segments in a 3 m × 3 m field of view in front of the vehicle. According to Figs. 8a and 8b, a line segment represents an obstacle if it intersects the surveillance corridor (SC) of the robot moving to its current subgoal. Detected objects, which will never be reached by the robot along its planned path, are not considered as obstacles.

For a blocked path the robot estimates collision distance d. If $d < d_{crit}$, a softstop is performed. The robot slows down smoothly, but does not stop in case of temporary obstacles, while a safe stop occurs in front of permanent obstacles. If the path is not cleared within a given time, path replanning is initiated.

4.2. Path replanning

The path planner as described in Section 2.4 is used online to construct a new cleaning path. Passing an obstacle on this path complies with the requirements of good floor coverage. For

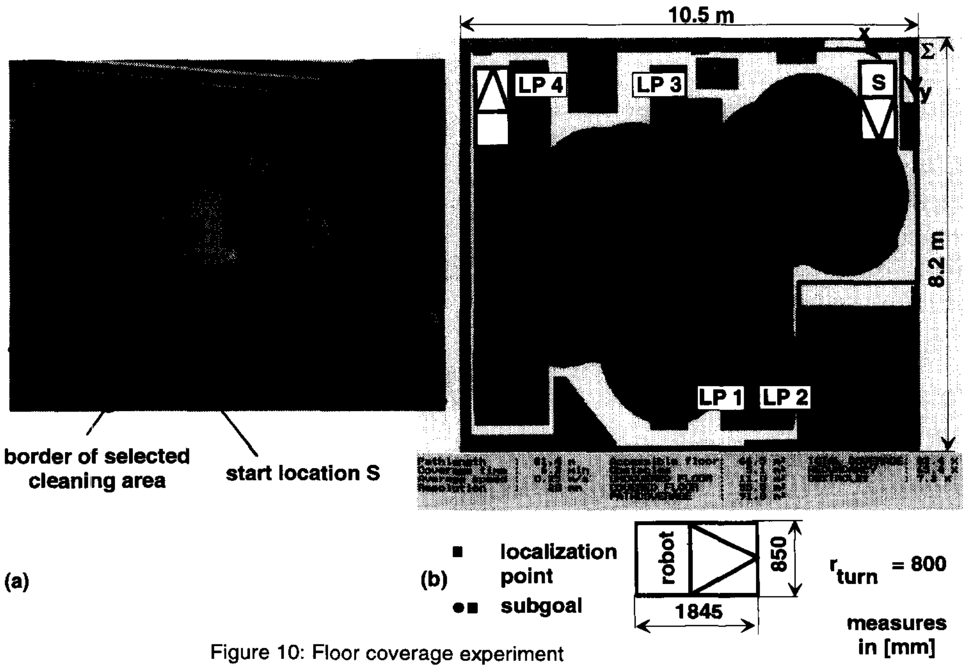


Fig. 10. Floor coverage experiment. (a) Cleaning environment. (b) Results of planning.

path replanning the map of the cleaning area is updated by the line segments representing an obstacle. Current robot location is considered as the new start location S*. A simulation result of path replanning and resulting floor coverage with a Type II robot is shown in Fig. 8b.

5. Results

5.1. Implementation issues

MACROBE, which must be classified as a Type II mobile robot, is used for experimental validation of the proposed planning and guidance techniques. The overall control structure is shown in Fig. 9. Two cascaded loops of cyclic perception, planning and execution ensure both accurate long distance path tracking and adaptive cleaning motion in case of obstacles. The gray shaded blocks have been implemented on a notebook computer.

5.2. Floor coverage experiment

The following experiment was performed in a FMS pilot factory. A view into the cleaning environment is shown in Fig. 10a, the corresponding map and results from planning are presented in Fig. 10b.

Using a joystick, the operator positions the robot in the vicinity of S. Next the robot initializes its location and maneuvers to S. The result of initial self-localization in Table 6 demonstrates that the operator had positioned the robot with

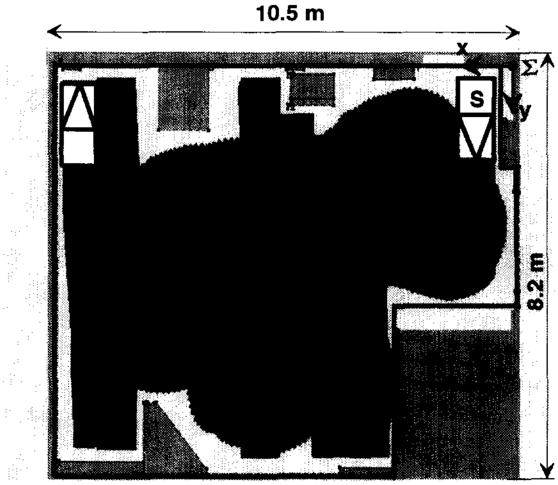


Fig. 11. Floor coverage achieved in experiment.

deviations up to 1 m in motion direction and up to 0.5 m in lateral direction with respect to S.

During path execution the tracking error is determined whenever the vehicle approaches one of the planned localization points LP 1 to LP 4, see Fig. 10b. If the error threshold $\epsilon_{max} = (\Delta x, \Delta y, \Delta \psi) = (50$ mm, 50 mm, 0.5°) is exceeded, compensation is initiated. Numbers given in Table 6 demonstrate, that path tracking errors are kept small and overlapping of neighbouring tracks is guaranteed. The '*' indicates, that at LP3 and LP4 the robot's lateral displacement and lateral error cannot be determined by self-localization.

Processing time for location estimation is in the order of 300 msec on a 486/33-MHz computer. This includes generating the correspon-

Table 6
Initialization and path tracking errors

Task	Lateral error [mm]	Longitudinal error [mm]	Orientation error [°]	Compensation
Initialization	−963	+510	+1.1	yes
Tracking (LP 1)	+2	*	−0.4	no
Tracking (LP 2)	+45	+2	−0.5	yes
Tracking (LP 3)	*	+10	+0.9	yes
Tracking (LP 4)	*	−10	−2.7	yes

dence tree as well as calculating the current location. Experiments demonstrate, that vehicle guidance satisfies real-time requirements for velocities up to 500 mm/sec. This is a velocity typical for conventional cleaning machines.

Fig. 11 illustrates floor coverage achieved in the experiment according to recorded data based on MACROBE's dead-reckoning system and the localization module. This and further experiments [9] have demonstrated that the presented planning and guidance approach meets major requirements as given in Table 1.

6. Conclusions

This paper proposed an automatic path planner which generates a technologically feasible cleaning path and the corresponding list of Cartesian motion commands. Execution of this path leads to most complete floor coverage both in narrow and wide cleaning areas.

Stable path execution is supported by regular self-localization at preplanned points in the vicinity of natural landmarks. Obstacle detection and avoidance is achieved by use of a new structured light CCD-sensor and online application of the proposed path planner.

Experiments in real-world environments have demonstrated that the implemented sensor and control approach can improve capabilities of current cleaning machines at reasonable additional cost.

Further work will focus on intelligent behaviour of the robot in case of obstacles. For example, memorizing of temporarily inaccessible subareas can help to cover the remaining uncleaned floor after completion of the nominal cleaning task.

Acknowledgement

The work reported in this paper was supported by the *Deutsche Forschungsgemeinschaft DFG* as part of an interdisciplinary research project on "Information Processing Techniques in Autonomous Mobile Robots (SFB 331)". The authors would like to thank colleagues and students of the laboratory for their valuable contributions to this work.

References

[1] W. Bott et al., Automatische Planung und Ausführung lokaler Fahrmanöver für Roboterfahrzeuge; *VDI-Bericht*, Nr. 1094 (VDI, Düsseldorf, v, 1993) 763–772.

[2] V.P. Burhanpurkar, *Design of Commercial Autonomous Service Robots* (Cyberworks Inc., Ontario, 1993).

[3] C. Fröhlich et al., Multisensor system for an autonomous robot vehicle, Information processing in autonomous mobile robots, *Proc. Int. Workshop*, (Springer-Verlag, Heidelberg, München, Germany, 1991) 61–76.

[4] H. Furuhashi et al., Development of an autonomous cleaning robot with an external power cable, *Proc. Int. Conf. on Motion and Vibration Control*, Yokohama, Japan (1992) 499–504.

[5] A. Gilg, Laser-Lichtschnittsensor mit homogener Auflösung durch asphärischen Spiegel, Internal report, Dept. for Automatic Control Engineering, München, Germany, 1993.

[6] HAKO-Company, Information brochures (Bad Oldesloe, Germany, 1994).

[7] U. Hanebeck, Kurzübersicht zur Lokalisierung mit Hilfe von Ultraschallsensorik, Internal report, Dept. for Automatic Control Engineering, München, Germany, 1993.

[8] C. Hofner and G. Schmidt, Flächendeckende Kursplanung und Führung autonomer Reinigungsfahrzeuge, *Proc. 9th Workshop on AMS*, München, Germany (1993) 27–38.

[9] C. Hofner and G. Schmidt, Path planning and guidance techniques for an autonomous mobile cleaning robot, *Proc. Int. Conf. on Intelligent Robots and Systems* (IROS), München, Germany (1994) Vol. 1, 610–617.

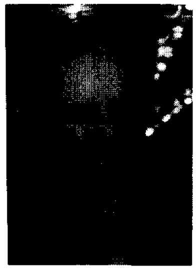
[10] J. Horn, Cartesian motion control of a mobile robot, *Proc. Intelligent Vehicles 94 Symposium*, Paris, France (1994) 443–448.

[11] R.D. Schraft et al., Service robots: The appropriate level of automation and the role of users/operators in the task execution; *Proc. Int. Conf. on Systems Man and Cybernetics*, France (1993) Vol. 4, 163–169.

[12] F.-J. Villmer, Einsatzfelder und Applikationen von Dienstleistungsautomaten in der Reinigung, *Proc. IPA-Technologie-Forum, Innovative Technologien für Dienstleistungen*, Stuttgart, Germany (1994) 81–120.

[13] A. Zelinsky et al., Planning paths of complete coverage of an unstructured environment by a mobile robot, *Proc.* of the *ICAR 93* (1993) 533–538.

Christian Hofner was born in Olching, Germany, in 1967. He received the Dipl.-Ing. degree in electrical engineering from the Technische Universität München, Germany, in 1992. Since 1992 he is a research assistant at the Department for Automatic Control Engineering (LSR) at the Technische Universität München, Germany. His research interests include path planning, sensing and guidance techniques for mobile robots in service applications.



Günther Schmidt was born in Wiesbaden, Germany, in 1935. He received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the Technische Universität Darmstadt, Germany, in 1960 and 1966, respectively. He was awarded a postdoctoral fellowship by the Max Kade Foundation, New York, in 1967. During that year he conducted control research in the Division of Engineering Mechanics at Stanford University, Stanford, CA. In 1968 he became Head of the Flight Control and Electronics Group of the Dornier Aerospace Company, Friedrichshafen, Germany. In 1972 he joined the Faculty of Electrical Engineering at the Technische Universität München, Germany, as Professor and Director of the Automatic Control Laboratory. His current research work and interests include methodologies for monitoring and control of complex industrial systems and advanced robotics. He has authored many publications and is on the Editorial Board of several international journals.