Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

A New Exploration Strategy for Mobile Robots based on a Cost Function Approach

Volker Sommer
Innovationen & Beratung
Schwabstedter Weg 6
D-13503 Berlin
Germany
drvsom@aol.com

Andreas Röcher
Robert Thomas GmbH & Co KG
Hellerstraße 6
D-57290 Neunkirchen
Germany
a-roecher@robert-thomas.de

Abstract - A new exploration method for mobile robots is presented applicable especially in home and office environment. For calculating new robot positions, the method uses an iterative algorithm based on a cost function. The closer area surrounding each position is scanned and simultaneously cleaned, using an extensible arm with an end-effector. The new approach enables the robot to clean any floor space fast and thoroughly, requiring only low processing power.

I. INTRODUCTION

Despite of certain improvements during the years, vacuum cleaning remains a time consuming and exhausting housework. Therefore, especially during the last decade much effort has been spent in order to develop service robots for autonomous working in home and office environment, since a growing demand can be anticipated for such devices [1].

Although robots for professional cleaning applications are already available, these bulky machines, equipped with sophisticated sensors and powerful control units, are much too expensive and inflexible for private deployment.

Especially in private households, various obstacles and narrow areas require highly adaptive exploration methods for service robots. Besides, since those devices must be sufficiently cheap, only simple sensors and standard microcontrollers may be applied.

Nowadays, first home vacuuming robots are commercially available [2]. The existing navigation methods can be divided into those based on deterministic path planning and others relying on pure stochastic movement, sometimes combined with contour scanning algorithms and other elementary deterministic movements. However, these methods of controlling autonomous vacuum cleaners show severe drawbacks:

Navigation methods which require a manual route planning are too complicated and inflexible, because especially in households, the floor area to be cleaned changes continuously because of new and moving objects.

Controlling methods, which before starting the actual cleaning process detect the border lines of the floor surface to be cleaned and determine their course with this information, will fail, if many obstacles, e.g. furniture, force them to frequent evading movements. Apart from that, in case of an initial detection of the border lines, it takes a long time until the actual cleaning process starts, and the method only works in closed rooms. Moreover, it is not possible to determine a position for the robot to start the cleaning process from.

Stochastic controlling methods without map-building, although winning the 'Cleaning Robot Contest' in 2002 [3], work unsatisfactorily as well, because dependent on the room topology certain areas are covered very often whereas other areas are covered rarely or not at all, so that a non-uniform cleaning effect is achieved. Since the efficiency of this simple control method decreases exponentially with increasing room size, it is clearly restricted to small- and single room applications without any potential beyond.

Another proposal uses an exploration method based on neural networks [4], with the robot seeking for unclean floor space adjacent to its path. However, due to the high complexity this approach does not seem to be suitable for home applications.

Thus, there is still a need to provide a flexible and fast cleaning method, which on the one hand is able to adapt to any kind of topology with any kind of obstacle, and on the other hand guarantees a uniform coverage of the floor to be cleaned. Besides, the cleaning process should be able to start at any point selected by the user without detecting the contours of the room at first.

To provide a serious alternative to hand held vacuum cleaners, an ideal vacuuming robot should be able to cover the complete floor surface as well as to clean edges of furniture as well as in narrow niches. Furthermore, a high cleaning effect must be achieved and damages must be avoided. The device has to be big enough to contain a sufficient capacity of batteries and to enable an efficient shielding of noise.

Moreover, the robot should be constructed as simple as possible, be robust and dispense with complicated sensors to render possible a low cost production.

0-7803-7860-1/03/$17.00 © 2003 IEEE

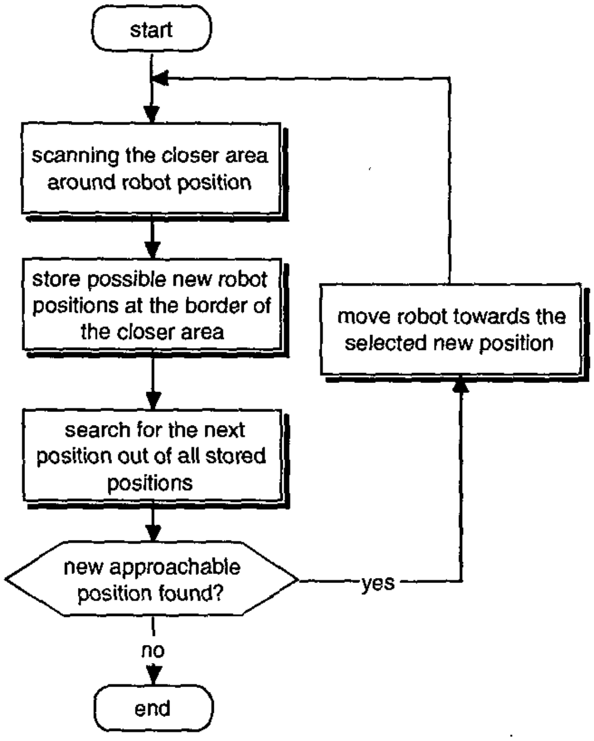

Fig.1 General flow chart of the control method

II. THE NEW EXPLORATION METHOD

In order to cover a given area as completely as possible while adapting to arbitrary obstacles, an iterative approach is proposed, which combines a deterministic scanning of the area around the robot with a statistic selection strategy for the next position, respectively.

According to figure 1, the procedure consists of the following steps:

1. First, an individually determined closer area around the current position of the robot is scanned and simultaneously cleaned.
2. Then, dependent on the result of the scanning, possible new positions for the device at the outer border of this closer area are stored.
3. In the next step, one of those possible new positions stored in the current or a previous step is selected by means of minimizing a cost function.
4. Finally, the device is moved to the selected new position applying a backtracking algorithm if needed.

This sequence is continued until a given area is completely covered or no new position can be selected.

A. Scanning of the closer area

Generally, the scanning of the closer area around the robot could be performed by means of any kind of deterministic algorithm and with a robot equipped with short range or contact sensors to detect obstacles. In this case, the extension of the closer area can be determined arbitrarily, but in order to ensure a simple shape of the closer area, its size should be adapted to the topology of the floor space to be cleaned.

Alternatively, distance sensors, based on infrared or ultra sonic could be applied, with the extension of the closer area depending on the range of those sensors.

In agreement with [5], the use of an extensible arm connected to the robot seems to be promising, since it allows a simple cleaning of contours without complex trajectories of the whole robot, and provides a high accessibility to narrow regions, which in households occur rather frequently.

The scanning principle is depicted in fig. 2 with the closer area chosen as a segment of a circle (in this example with a 180° sector): Starting with the retracted arm the robot first turns to the nearest border of the sector, then turning repeatedly to the right and the left with increased arm length at each turn, until a given maximum arm length is reached and the arm is then, after a last turn, retracted again.

Regarding the mechanical complexity of the robot, it should be noted that for implementing the arm, only one additional electric drive for ex-/retracting is needed, since the rotation can be performed by opposite movements of the two wheels of the robot. Besides, since the head at the front part of the arm acts as a third support for the robot, no additional supporting wheel is necessary and any cleaning equipment at the head is always in close contact to the floor surface, also in uneven areas or e. g. at small carpet steps.

By means of the scanning principle according to fig. 2 it is guaranteed, that unknown areas are always first covered by

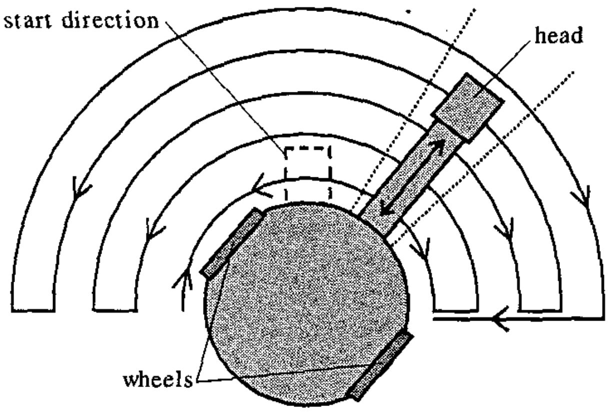

Fig.2 Scanning the closer area with an extensible arm

the front end of the arm, which apart from the end-effector also is equipped with a sensitive bumper including a contact sensor. In order to avoid mechanical contacts, additional IR distance sensors at both sides of the arm, whose beams are indicated by the dotted lines in fig. 2, are used.

When detecting an arbitrary obstacle, the arm is guided along its contour at the smallest possible range by means of a specific contour following algorithm. In order to avoid multiple scanning of the same obstacle, the arm is shortened when reaching those angles where an obstacle has already been detected at an earlier turn of the robot.

B. Building a grid based global map

To allow global orientation of the vacuum cleaner, the area covered by its head as well as the sensor readings are stored in a grid based map. Here, different states are stored, which can be assigned to each area element. In comparison to feature extraction methods [6], the creation of a bit map can be realized with low processing power, and especially for indoor applications only a very limited RAM space is needed: With four different states and a grid size of 2 × 2 cm^2, which is sufficient for the resolution of the sensors, an area of 20 × 20 m^2 can be stored in a RAM space of 250 KBytes without any data compression.

The information in the map is used to mark possible new positions, to calculate the route to these positions and to determine the optimum size of the closer area, respectively. The following four states are distinguished:

0: This state is the default-value in the map when starting the robot in an unknown environment. It will be overwritten as soon as the robot's head has covered the respective floor space for the first time, or if any obstacle is detected.

1: This state is assigned to those elements in the map, which have been covered by the head and which do not represent an obstacle for the movement of the robot.

2: This state is used to mark obstacles which have been detected by the sensors. A field marked with this state cannot be traversed by the robot when approaching a new cleaning position.

3: With this state, fields along the outer border of a closer area, which previously must exhibit the state 0, are marked as a possible new position for the robot. If those areas later on are covered by the head, the field will obtain the state 1 or 2. When checking a possible new cleaning position, state 3 shows that the surrounding floor area has not been scanned before.

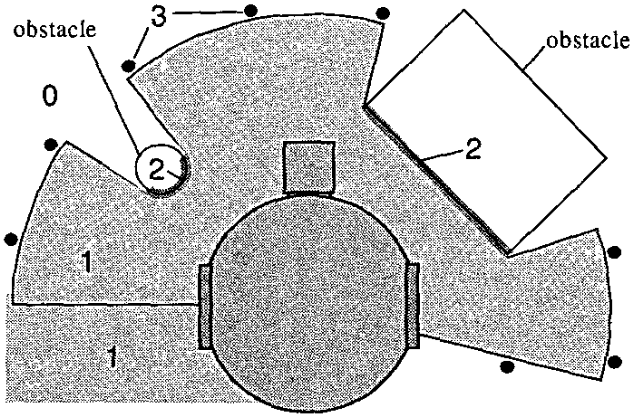

Fig.3 Storing positions at the border of the closer area

By evaluating the information stored in the map it is ensured that the scanning of the floor surface is only conducted in those areas which have not yet been covered by the head before from a neighboring position. This is achieved by means of adapting the opening angle of the circle segment between 20° and 240° in order to minimize the overlap between neighbored closer areas.

To limit the number of stored positions after the scanning of the closer area, new positions are only stored where no obstacles have been detected and where no areas border, which have already been scanned from a previous position of the robot. This is achieved most simply by storing only new positions at those borders of the closer area which are marked in the data field as not yet scanned (state 0).

Fig. 3 shows an example of a map after the robot has scanned a new closer area in the form of a circle segment. During the scanning, parts of the contours of two obstacles have been detected and marked in the map with the state 2. Since most of the background has not yet been scanned, it is still marked with state 0, only at the left side of the robot an already scanned area marked with the state 1 has been found. The black dots at the outer border of the close area in figure 3 indicate new possible positions for the robot. In addition to the marking in the map, these possible new positions with their co-ordinates in x- and y- direction are also stored in a list of all possible new positions found so far.

When scanning again an area already marked in the map, the states of this area are updated according to the new sensor readings. Thus it is realized, that because of the small overlap of the scanning areas the map always contains the most current information, especially if obstacles are located in an area already scanned.

C. Selecting and approaching new positions

After scanning the closer area, a new position for the next step out of all positions stored during the previous steps is selected. To determine the new scanning position, several criteria are evaluated as follows:

First, when selecting a new position for the device, only not yet scanned positions are taken into consideration, with the state 3 still assigned to the respective position in the map. This criterion provides a fast and effective method to consider only positions at the borders of the whole floor area already covered by the robot. If a stored position is marked as already scanned, it can be deduced that the surrounding area as well has been already covered from a neighboring position, due to the usually compact form of the closer areas. Therefore, those positions can be deleted from the list, since they do not indicate floor space still to be cleaned.

After this pre-selection, it is then ensured by evaluating the map, that the possible new position can be reached by the device from the current position on a direct route, allowing only areas to be traversed which have already been scanned and which are not marked as an obstacle. Although more sophisticated methods for calculating optimum trajectories have been proposed [7], this simple check can be performed very fast and run-time efficiently.

In order to limit the influence of odometry errors, the "age" of a possible new position is taken into consideration. While checking the possible reach of a position in the map, this is achieved by requiring an additional safety distance without obstacles at the borders of the path which is to be traversed. Its width depends on the covered distance since storing that position.

For all possible new positions which can be reached, then a cost function is calculated with the position exhibiting minimum costs being chosen for the next step. In this function, the following criteria are used:

- The distance between a possible new position and the current position of the robot. By means of this criterion, positions in the vicinity of the robot are privileged and unnecessary movements of the robot are avoided, thus reducing the time needed for cleaning.
- The maximum size of the closer area which can be covered first from the possible new position. The size is determined by counting all map entries with the state 0 within reach of the arm at the respective robot position. This criterion provides a handicap for inefficient positions, which enable access only to small floor surfaces.
- The total angle, by which the robot must be turned in order to reach and to scan the new closer area. This criterion minimizes the rotations of the robot with a positive effect on cleaning time and slipping.
- The distance between a possible new position and a reference position. Initially, the starting position of the robot is chosen as reference position, which will be changed if the distance between two consecutive robot positions exceeds a threshold. This criterion ensures that the device mainly covers coherent areas.

When the next position is selected, the new closer area is determined by evaluating the map, so that a small overlap with already scanned neighbored areas occurs.

If no new position can be selected from the current position or if the result of the cost function for any accessible positions is below a threshold, the previous positions of the robot will be considered as test positions by means of again performing a loop over all stored positions. If from any previous position a new scanning position can be determined, the device will be set back to this position, and from there the new position is approached.

If from none of the previous positions of the robot a continuation is possible, either because the whole floor space area has already been covered or because the still open positions cause to high costs, the exploration process will be terminated.

Otherwise, it will be checked, if the test position, from where a new position has been found, is identical to the current position of the robot. While in this case, the new position can be approached directly with the robot head after turning in the respective direction, in all other cases the device first has to be set back to the respective test position. For this purpose, by evaluating the data field, the shortest route without obstacles within the already scanned area could be determined and the device could be moved along this route.

However, keeping in mind the low processing power of the robot, a more simple method for route planning has been implemented, characterized by the constraint that the robot can only move on straight lines between all previous robot positions. In order to avoid long detours, in case of a backtracking over more than one position, it is checked in the map for each position in between, if it can be skipped and if the robot possibly can be moved back directly from its current position to the position from which then the new scanning position is accessible.

Condition for a possible "short-cut" again is the fact that the robot only may traverse areas, which are marked in the map with the state 1 to avoid a collision with obstacles.

During the movement of the robot towards a new position, it can meet unexpected obstacles caused by shifted objects or because of odometry errors. In this case, the movement of the device is interrupted, the closer area is scanned to update the stored data, and then a new position is selected as described above.

By means of concatenating single sectors, areas with any contour and any kind of obstacle can be cleaned completely. Due to a certain overlapping of the sectors, some areas are cleaned several times, which additionally increases the cleaning effect and compensates possible inaccuracies of the position of the robot due to slipping.

III. SIMULATION- AND EXPERIMENTAL RESULTS

In order to examine the efficiency of the new exploration strategy, simulations in various environments have been performed under quasi real conditions. For this purpose, the software very closely adapts the behavior of a real robot, including odometry errors due to slipping. Most modules are identical to those of a real robot and only the sensor readings as well as the drive control are replaced by simulated input/output data.

Fig. 4a-4d show the cleaning progress of a mobile robot in a room with the dimensions of 3,6 × 2,6 m², furnished with a cupboard, a table and a chair. In these figures, the real contours of the room with the robot position are superimposed with the grid map according to the perception of the robot, assuming a slipping of 1%. According to the states in the map, the light gray color indicates the floor space already covered by the head, the darker dots mark possible new robot positions and the darkest color represents obstacles detected by the sensors.

When starting the robot, the first closer area always is a complete circle, followed by a special deterministic movement in order to cover the area around the starting point.

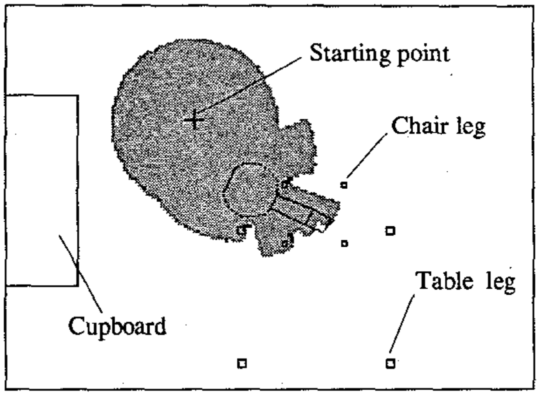

Fig.4a Scanned area during the third step of iteration

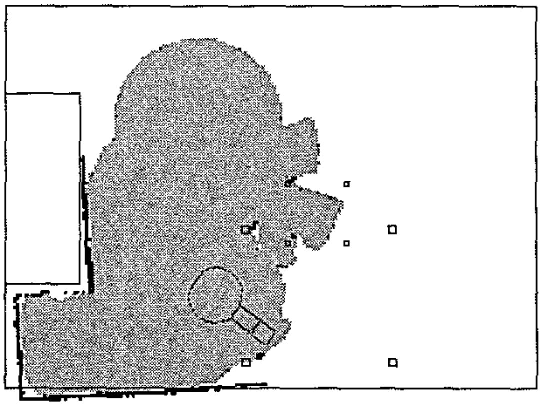

Fig.4b Scanned area during the 11th step of iteration

After this initial procedure, any further position of the robot is selected according to the algorithm described above. Fig. 4a depicts the situation during the third step of iteration with the robot currently cleaning the narrow region between the chair legs, while in the 11th step (fig. 4b) already a part of the outer border of the room has been detected. Fig. 4c shows the cleaning status during the scan of the closer area in step 23 and figure 4d the final result after the 41st step with an almost complete covering of the floor space.

In this example, the robot first performs a counter clock-wise turn and afterwards cleans the corners on the right side. This movement is solely determined by the selection of the positions dependent on the weighting factors in the cost function. In addition, a strong influence of the topology of the room and the starting point of the robot exists.

Nevertheless, with fixed weighting factors, the result of this selection strategy is very close to the optimum path, a human operator would choose, and it adapts automatically to any shape of the floor to be cleaned.

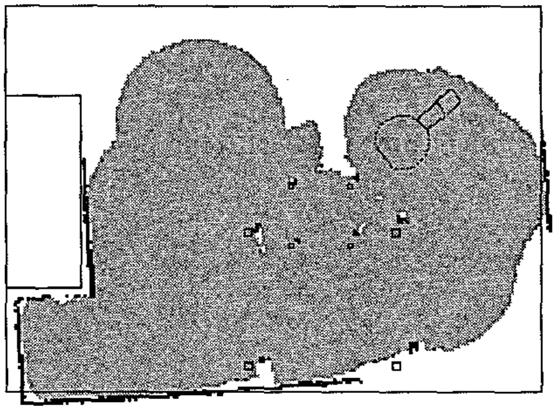

Fig.4c Scanned area during the 23rd step of iteration

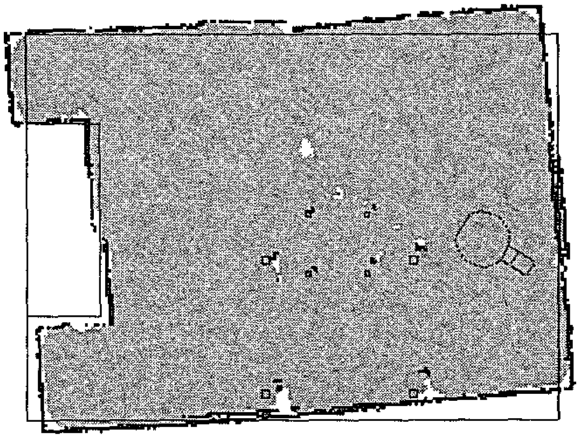

Fig.4d Finally scanned area after the 41st iteration step

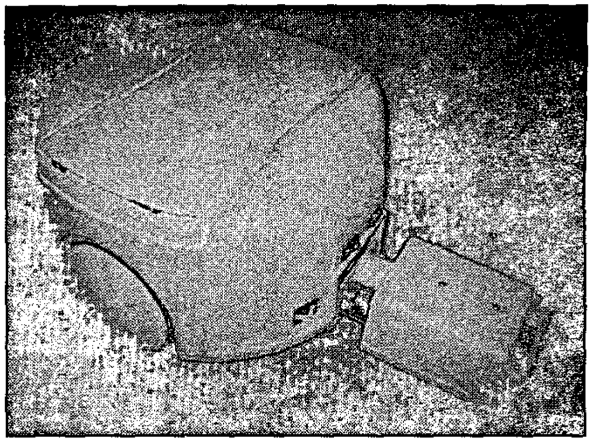

Fig.5 Prototype of vacuuming robot with retracted cleaning head

Although to localize the robot currently only odometry is used, the map represents the room contours quite precisely. Due to the pseudo random movement of the robot, which compensates to a certain degree for systematic slipping, the exploration method proves to be very reliable.

The insensibility with respect to odometry errors is further improved by the regular rotation of the robot, which allows a continuous update of the map in the vicinity of the robot by evaluating its sensor readings.

Since at each run a new map is recorded, which is needed mainly in order to avoid multiple cleaning of the same floor space and not for long-term self-navigation, the preciseness of localization is absolutely sufficient.

The software has already been successfully tested on a real cleaning robot prototype depicted in figure 5, exhibiting precisely controlled dc-motors, a contact sensor around the head as well as two IR distance sensors at both sides of the extensible arm. The device is equipped with a single board computer based on the well known MC68332 processor, which allows real time processing due to the low hardware requirements of the exploration algorithm.

Apart from the floor coverage and the necessity of a low complexity, the performance of an exploration algorithm is dependent on the floor space, which can be cleaned during a certain time period. Our experiments yield, that the robot needs an average time of about 15 seconds for the scanning of a closer area, including the time for approaching the respective next position. Thus, it can be deduced that the complete cleaning of a room as shown in fig. 4 is accomplished in about 10 minutes (41 × 15s), which means a high cleaning performance of up to 60 m^2/h in a typical home environment.

IV. CONCLUSIONS

A new exploration method for service robots is presented, which allows thoroughly covering a given floor space. Besides, the approach is fast and requires only low processing power provided by a standard micro controller, rendering possible the use with autonomous vacuum cleaners in private households.

REFERENCES

[1] E. Prassler et al., "A short history of cleaning robot", Autonomous Robots vol. 9, 2000, pp.211-226

[2] B. Rooks, "Robot reach the home floor", Industrial Robot, vol. 28, 2001, no. 1, pp.27-28

[3] First International Cleaning Robot Contest (2002), Lausanne, Switzerland jointly with IROS 2002

[4] P.W. Tse et al., "Design of a Navigation System for a Household Mobile Robot using Neural Networks", Proc. IEEE World Congress on Computational Intelligence, vol. 3, NY, USA, 1998, pp. 2151-2156

[5] I. Ulrich et al., "Autonomous vacuum cleaner", Robotics and Autonomous Systems 19, 1997, pp. 233-245

[6] D. Meizel, O. Lévêque, "General method for Localization of a mobile robot by means of a computable model of the environment", IEEE Trans. Robot. Automat., vol 18, 2002, pp. 966-971

[7] E. Bernabeu, J. Tornero, "Hough Transform for Distance Computation and Collision Avoidance", IEEE Trans. Robot. Automat., vol. 18, 2002, pp. 393-398