Field Patterns for the RoboCupJunior League? - A Car-Park Problem with LEGO Mindstorms Robots

Thomas Oelkers, Birgit Koch and Dietmar P.F. Möller
Universität Hamburg, Fachbereich Informatik,
Arbeitsbereich Technische Informatiksysteme,
Vogt-Kölln-Str. 30, 22527 Hamburg
E-Mail: {6oelkers, koch, dmoeller}@informatik.uni-hamburg.de

KEYWORDS

Mobile Autonomous Robots, RoboCup, Field Patterns, Self-Localization

ABSTRACT

In the RoboCupJunior league the robots on the soccer field may use the whole space of the field, but to get good results in playing soccer the critical element is that the robots should always know where they are located. Unfortunately a field with color tone is used where it is not possible to make an approximation of the position parallel to the goals as long as only light sensors are used. This paper investigates the use of different field patterns in solving a car-park problem for LEGO Mindstorms robots. Solutions based on guide lines, grid patterns, chessboards and color-tones are described, tested and compared to handle the problem of knowing the position and orientation of a mobile robot situated in a structured unstable environment.

INTRODUCTION

Today robots are used in many different scenarios. It is the goal of the scientists to improve the methods and techniques, which are useful to the society (e.g. mine cleaning robots, Mars-expedition robots, household robots). In order to advance the research in robotics, numerous competitions of different categories take place in the world, in which researchers present their own approaches. Examples of such competitions are disciplines in RoboCup. Each year small robots build for example by LEGO Mindstorms [Lego Mindstorms WWW] equipped with different sensors and motors can be admired in the RoboCupJunior league [RoboCupJunior WWW].
With LEGO Mindstorms robots different problems can be solved. Thus a student thesis [Oelkers 2002] evolved, in which a LEGO Mindstorms robot should accomplish the following tasks: First to drive into a car-park and second to look for a free space and to park in it. All this should work based on given landmarks on the floor, which have to be recognized by a photosensitive sensor.

THE ROBOTIC CONSTRUCTION KIT

The most important part of the LEGO Mindstorms Robotics Invention System is the RCX unit, a programmable LEGO component, based on a microprocessor of the Hitachi H8-family (H8/3292 processor). The programs for the RCX are written on a usual computer. They will be transfered via an infrared interface to the RCX, that also disposes of the following characteristics: a LCD display for the output of short and simple information, an integrated speaker to produce simple acoustic signals, an infrared interface for communication with other RCX units or a computer, four system timers and a writable memory of 32KB.

THE PARKING TASK

A robot constructed with LEGO Mindstorms should be built and programmed in such a way, that it can cope completely autonomously with the following tasks (Figure 1). The robot must

Proceedings 18th European Simulation Multiconference
Graham Horton (c) SCS Europe, 2004
ISBN 3-936150-35-4 (book) / ISBN 3-936150-36-2 (CD)

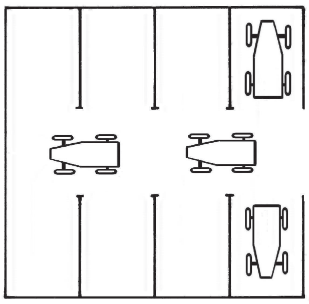

Figure 1: A typical situation in a car-park

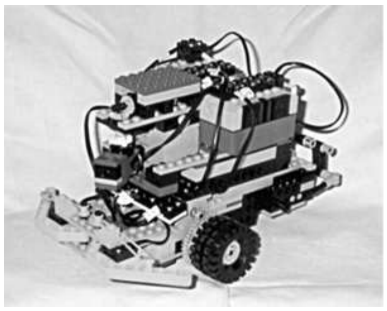

Figure 2: The robot for the parking task

been able to drive into a parking lot. Before driving in, the robot must stop and look, whether a parking lot is free. If a parking lot is occupied, the robot must recognize this and look for another one. If a parking lot is free, it must drive in correctly. If both parking lots are occupied, the robot must drive to the next parking row, where it has to proceed like before. If the robot is reaching the end of the parking lot, it must recognize this: it must not drive beyond this point. If the end of the parking lot is reached and no free parking lot was found, the robot must leave the parking lot. While driving the robot must not touch or damage another vehicle or drive over forbidden areas.

EQUIPMENT OF THE PARKING ROBOT

The parking robot (Figure 2) is a vehicle with three wheels. Two different motors activate two parallel wheels. A single wheel which is not powered serves for the support. The wheels can be differently arranged.
Contact sensors recognize physical contacts, they know only two conditions: pressed or not pressed. The robot is equipped with three of these sensors, one in the back and two in front, because the robot should be able to differentiate between left and right side.
Light sensors are active sensors, because they need energy from the RCX unit. Light sensors are not able to recognize different colors, they only measure the intensity of the light. Unfortunately, this type of sensor is easily influenced by other sources of light.
The **programming environment** used for the robot was leJOS ("LEGO Java operating system"), an implementation of a Java virtual machine for the RCX unit.

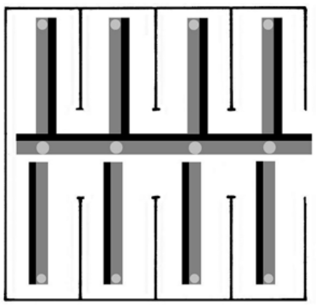

Figure 3: Leading lines for the robot

PROPOSALS FOR SOLUTIONS

In this paper a structured unstable environment is used that is suited to the needs of the robot. It is sufficient if the robot gets enough information about the environment to solve its task. For further information on environmental modulation see [Nehmzow 2000][Dudek and Jenkin 2000].

Guide Lines

The vehicle should drive on a grey line (Figure 3). Circles on the floor mark the individual landmarks, i.e. the individual parking spaces, that the robot must recognize. The robot should be limited in its freedom of movement as far as possible. Only those places which serve the purposes of the task should be reachable. In case of leaving the grey line, a control routine must be activated immediately, which leads the robot fast and easy back to the grey line. With this behavior, it is guaranteed that the vehicle drives straightforward and remains strictly on the way which leads to the individual destinations.
Problems

There are some problems, that occur with the lines approach:

- unevenness of the floor and thus wrongly noticed brightness
- because of too many "IF instructions" the measurements become more unreliable because they take place more irregularly
- landmarks for park row recognition are sometimes counted double or not at all
- reflectors are not always recognized

Advantages
To drive a vehicle on a given line can be compared with usual rail guidance. Without dirt on the guidelines, which would disturb the data taken up of the light sensor, it is very sure that the vehicle does not leave the way. The danger to drive over forbidden lines or bump against another parking vehicle is relatively small. Additionally the software is more or less easy to implement. The equipment with sensors is limited to a minimum. Thus, the vehicle can be guided quite simple. All tasks are actually fulfilled.
Disadvantages
As surely and reliably this approach of the solution works and as easily it is implemented, it is extremely inflexible. If two vehicles meet on the center line, it is not possible for them to avoid each other. The robot in this experiment has to leave the parking lot in order to make place for the other vehicle. Additionally the robot cannot drive around obstacles. The robot is practically joining with the line and cannot deviate from it. A possible solution would be to provide two additional leading lines which run alongside to the left and right of the main line. In order to avoid another vehicle, the robot can look for the guideline on the right side and evade to it, while the other vehicle evades to the other side. After this manoeuvre both robots go back to the main leading line.

Grid Pattern

Now the entire floor, on which the robot moves around, is filled with a grid (Figure 4), so that the soil consists of many squares equal in size. The idea is to use the individual small boxes as indication for coordinates. Each box stands for two values. The first value describes the distance to the left, the second value the distance to the

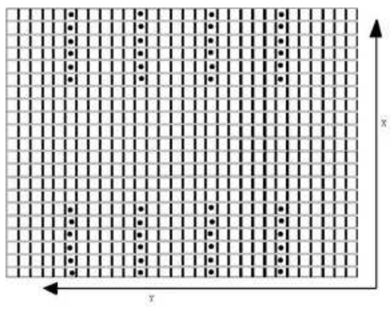

Figure 4: The grid on the floor with x- and y-axis

bottom edge of the field. If the robot counts each crossed line it always knows at which place (in which square) it is. This solution is substantially more flexible. It permits the robot to execute more manoeuvres. Additionally the lines serve the robot to navigate straightforward. In order to differentiate the horizontal lines from the vertical ones, they have different colors. The lines parallel to the x-axis are darker, the ones parallel to the y-axis are brighter than the floor.
Problems
There are some problems, that occur with the grid approach:

- lines are not always recognized reliably and therefore x- and y-values are not always correctly
- 90° turns with a sufficient result are very hard to implement; the robot turns too far or not far enough and therefore leaves its way

Advantages
The vehicle can move more freely and more flexibly than in the line approach. If the robot leaves a grid line it will immediately look for a new line by itself. Avoiding another vehicle is relatively simply. Special landmarks are not needed. Only a grid pattern on the floor and the knowledge of the starting point are necessary to navigate the robot. Only two different colors are used in order to distinguish the x- and y-coordinates. Thus, errors by unwanted false values are prevented completely.
Disadvantages
The software becomes more complex. If the robot makes an error counting the lines (which often happens during turns) it has no possibility to correct it. The robot looks on his x- and y-values to discover a parking lot. If the values do

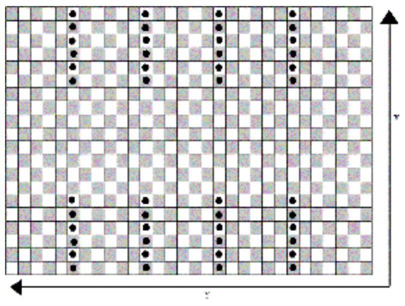
Figure 5: The chessboard on the floor

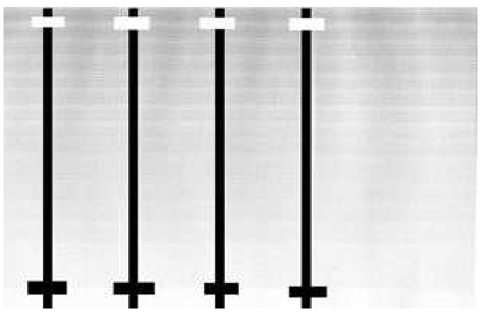
Figure 6: The color tone on the floor

not correspond with reality, the robot will stop in the wrong place. The further a park row is away, the more likely it is to have a miscount. The rotation is particularly incorrect. During avoidance of another vehicle, a robot must turn four times. Failures with each turn sum up, so at the end the result is rather dissatisfying. Driving parallel to the y-axis and driving parallel to the x-axis must be treated differently. Although the grid works with only two colors (grey and silver), in reality three colors are used, since the floor has the color white. Therefore the robot has to recognize three colors to be able to distinguish the floor from a silver or a grey line.

Chessboard

In this approach the robot should move over a black and white tiled field. The two colors selected for the tiles are not important, as long as the light sensor of the robot can clearly differentiate between them (Figure 5). With the help of the small boxes, the robot is able to move with the same flexibility like in the grid approach but with a better accuracy. The chessboard field seems to be very similar to the grid pattern but there are a few fundamental differences: The chessboard field operates with only two colors which makes the distinction safer and faster. In addition, a square is completely filled out with a color. There are no thin lines, only a special floor color. Hence the risk that a square is not recognized due to irregular measurements is reduced.
Problems
The following problems occur with the chessboard approach:

- the robot drives straight diagonal lines over the field instead of straightforward
- during a turn of 90° to the left or right very often incorrect results arise
- other vehicles are not always recognized reliably

Advantages
In the chessboard approach only two colors are used. Regardless of the direction the robot is driving, it can always use the same routine. A single method is sufficient to realize all rotations. A diagonal line (45° angle) is reliably possible.
Disadvantages
Rotations remain to be a problem. They are not always completed reliably. That is because of the fact that the robot is shifted a little bit when it is stopped before a rotation. Thus the radius that the light sensor pulls over the tiles can be shifted easily so that the number of color changes vary.

Color Tone

The fourth solution consists of a field with a color tone. As shown in Figure 6, the field is painted with a color tone parallel to the y-axis. The color tone runs from dark grey to white. Parallel to the x-axis four black lines are painted, which have an approximate thickness of 1.5 cm. They serve as park row marks and can also help the robot during the drive-in. At the end of the line another mark must exist, which highlights the end of the parking lot. These end marks are made with silver strips. The idea is the following: the robot needs only one sensor to measure the hue of the floor. As long as it is searching for a parking lot, it may drive only straightforward, parallel to the y-axis. With the help of the color tone, this is simply possible. The robot has to be programmed in such a way that it may drive only on a certain color brightness. The values depend on the field and can only be evaluated by tests. If a color which is not permitted is read the robot must correct its way immediately. The color lines, which mark the parking lot, are of lower intensity than the lowest value of the color tone. Therefore the lines can always be clearly identified. Thus a good orientation seems to become possible along the x-axis.

Problems

There are some problems, that occur with the color tone approach:

- the color tone is not continuous; sometimes brighter or darker values are taken on the same parallel
- the sensor data of the light sensor is quite unreliable, since it depends on the battery power

The main problem is the bad color tone on the floor. The robot does not really drive between two invisible leading lines. Sometimes it corrects the driving direction, while it does not behaves like this a few millimeters later. Because the interval of the permitted values is very small little deviations are fatal. Maybe this problem can be reduced if a color tone with a stronger gradation would be provided. The more clearly the gradations on the field can be recognized by the robot the more accurately it can move.

Advantages

A small number of landmarks is needed to mark the parking rows. The color tone alone gives the robot the information in which x-position it is. Inaccuracies of the measurements have only a small effect on the process. The best advantage is the simplicity of the algorithm. It requires only small adjustments and tests until the robot supplies satisfying results.

Disadvantages

Exact positioning along the y-axis is not possible. The positioning at the x-axis is possible but not as exact as in the two previous solutions. Compared to the other approaches the robot is more flexible while moving on the floor. In contrast to the other approaches in this solution avoiding is possible within the parking lot, but the danger to over-drive a line or to push a parking vehicle is very big. Avoiding another vehicle is still difficult to accomplish. However testing generates extremely unwanted results. Most likely the color tone must be of better quality.

Comparison Of The Different Methods Of Orientation

After adjustment and correction each method was tested 100 times counting the errors. An error occurred when the robot got off its actual line, a turn was dissatisfied, a marking was overlooked or counted twice and some other errors. On the basis of these results the different methods were evaluated. It is interesting that the simplest solution (the latter) showed the most reliable results. The more complex solutions had to cope with substantially higher error rates. Most errors that appeared with the first and the last solution could be repaired with small adjustments of the algorithm, so that they worked nearly free of errors. Unfortunately the grid pattern and the chessboard showed a substantially higher error rate. On the grid pattern the turns were very unreliable. Possibly another algorithm could give better results. However the selected light sensors did not meet the demands. The measured light values were unreliable and not very differentiated. The most significant problem on the chessboard was that the vehicle counted markings double or not at all. In this case a marking was a change between black and white. Furthermore the robot often got of the way. It drove straightforward but changed the trace sometimes, so that the x-coordinate was not correct any longer. This had consequences on the parking. The vehicle stopped too early or too late. Finally it can be said that the solution with the chessboard showed the highest number of errors. On the other hand it was very flexible. When the vehicle got off the way during the experimental phase of the guide line method it did not find the way again. On the chessboard and on the grid pattern the robot adjusted itself back to the right path again. The idea of the chessboard approach is certainly improvable. More exact positioning would be possible with a better control algorithm and better sensors.

CONCLUSION AND PERSPECTIVE

Each solution in robotics has its individual advantages and disadvantages. It is important to consider in which scenario a solution is needed. The main problem of the study was that the robot should drive autonomously without human help over the floor. The different solutions can be evaluated from different points of view. If one would create a real world parking system, then the line approach would be preferred although it is very inflexible. However it is very reliable and simple. There are only several places on which the vehicle may be located. A chessboard or a grid pattern would not add advantages. In the opposite these approaches are more error-prone. But in a real world parking lot errors are not acceptable. This is different on a soccer field: the robots may use the whole space of the field. For playing soccer an accurate location is less important than flexible driving. In this case the solutions of the grid pattern or chessboard are more suitable. Unfortunately in the RoboCupJunior league a field with color tone is used. It is not possible to make an approximation of the position parallel to the goals. On a chessboard this problem is not given. Both x- and y-coordinates can be determined. To avoid that errors sum up, special control places should be painted. If a robot crosses this field, it corrects its x and y-values. Other applications would be in the industry. Here robots are used to carry loads from place A to place B. If the floor would be tiled like a chessboard, then a transport to any place would be possible. In this case, a grid pattern or a chessboard would be quite conceivable.

References

[Dudek and Jenkin 2000] Dudek, G., Jenkin, M.: Computational Principles of Mobile Robotics. USA: Cambridge University Press (2000)

[Koch 2003] Koch, B.: Einsatz von Robotikbaukästen in der universitären Informatikausbildung am Fallbeispiel "Hamburger Robocup: Mobile autonome Roboter spielen Fußball". Diplomarbeit, Fachbereich Informatik, Universität Hamburg (2003)

[Lego Mindstorms WWW] The LEGO Mindstorms website. http://www.legomindstorms.com/

[Nehmzow 2000] Nehmzow, U.: Mobile Robotics: A Practical Introduction. London: Springer-Verlag (2000)

[Oelkers 2002] Oelkers, T.: Parksysteme mit Lego-Mindstorms. Studienarbeit, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Technische Informatiksysteme (2002)

[RoboCupJunior WWW] RoboCupJunior website: http://www.robocupjunior.org

[Seeboerger-Weichselbaum 2002] Seeboerger-Weichselbaum, M.: Mindstorms: Das Einsteigerseminar Java 2. Bonn: bhv (2002)

AUTHOR BIOGRAPHIES

THOMAS OELKERS studies Computer Science at the University of Hamburg. He wrote his studies thesis about Park systems with LEGO Mindstorms robots. Currently he writes his diploma thesis about fuzzy logic with LEGO Mindstorms robots. His E-Mail address is 6oelkers@informatik.uni-hamburg.de.

BIRGIT KOCH studied Computer Science at the Universities of Hildesheim, Balearic Islands and Hamburg. In 2003 she started working as research assistant at the Institute of Computer Engineering at the University of Hamburg. Her current research interests include mobile autonomous robots, RoboCup, socionic and gender. Her E-Mail address is koch@informatik.uni-hamburg.de and her website can be found at http://www.informatik.uni-hamburg.de/TIS/.

DIETMAR P.F. MÖLLER studied at the Universities of Mainz, Bonn and Bremen, where he studied Electrical and Control Engineering. In 1998 he was elected from the University of Hamburg as Full Professor and Chair of Computer Engineering. His E-Mail address is dmoeller@informatik.uni-hamburg.de and his website can be found at http://www.informatik.uni-hamburg.de/TIS/.