Annales UMCS Informatica AI 3 (2005) 263-271

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Experimental data driven robot for pattern classification

Krzysztof Sapiecha, Barbara Łukawska*, Paweł Paduch

Department of Computer Science, University of Technology,
Al. 1000-lecia Państwa Polskiego, 25-314 Kielce, Poland

Abstract

In this paper the problem whether or not a human being can plan a tour for a mobile robot with a camera so that all changes in the room are detected within a time limit is investigated. A tour simulator and a game based on it are developed. The plan of the winner of the game is used for a driving real robot. It is shown that this plan is two times shorter (as Java bytecode for Lego RCX) than the average one.

1. Introduction

Most today guarding systems use stationary cameras and mobile guards. However, while cameras are getting better and cheaper guards are still unreliable and very expensive. It is almost certain that in the near future mobile robots equipped with cameras will replace today guarding systems. In this new approach the robot will acquire information and also process it [1-5]. To do this effectively the robot should properly tour the area it guards and correctly classify patterns it sees.

Visibility-based problems of surveying, guarding, or searching have a long history in the area of computational optimization. Determining of minimum stationary cameras for guarding a region is the well-known art gallery problem [6]. The watchman problem [7] is to find the shortest tour along which one the mobile guard can see the entire region. If the region is unknown in advance, we are faced with the online watchman problem. All these problems are NP-hard [5].

In the paper an efficient method of planning a tour for a mobile robot in a room (including positions where pictures should be taken), so that all changes in the room are detected within a time limit is presented. The method is based on an experiment in which human intuition and perceptivity are involved. In section 2 the problem is formulated. Section 3 describes the method we use to solve the

*Corresponding author: *e-mail address*: b.lukawska@tu.kielce.pl

problem. Methodological details are given in section 4. In section 5 experimental results are presented. The paper ends with short conclusions.

2. Problem statement

A robot equipped with a camera tours a room. The robot moves forward and backward and also turns left and right. In every stop point the robot may take a picture of what "camera sees". There are some objects placed on the floor of the room. The objects may appear, disappear, move and be modified. When an object moves, two changes in the room should be observed: disappearance of the object and its appearance in a new place. Only a color and/or an orientation of the object may be modified.

Our analysis concerns only objects placed on the floor. Hence, walls and space above the robot are not taken into consideration. Each of the objects is bound to a floor square. We are interested only in coordinates of places and types of changes.

The robot can store (in its internal memory) a limited number of patterns (pictures compressed). It also can detect differences between two patterns. However, these problems are not investigated here and thus they won't be discussed later on. This work focuses on planning a tour of the robot in a room (including positions where pictures should be taken) so that all changes in the room are detected within a time limit.

If all parameters of the problem that is distribution of objects and their shapes and colors, camera range, lighting etc were known then the problem could probably be solved numerically. However, in general Shortest Rout Watchman Problem (SRWP) is NP-hard. Unknown conditions which robot may encounter in various points of the room (for example different lighting, unknown frequency and intensity of changes) make numerical solution of the problem more difficult than a solution of SRWP. Moreover, in real-word situations rapid numerical solution may not be available or simply impractical. Actually, not the shortest route but enough short (to meet time limit) rout is needed.

The aim of this research is to investigate to what extent in this very complex situation one may use human intuition and perceptivity, that is whether a human being can select probably not the best but still satisfactory solution or not.

3. Tour simulator

To answer the above question tour simulator and a game based on it were developed. The tour simulator (like flight simulator) creates virtual room where a mobile robot will work. In the game a player equipped with a map of the room uses the tour simulator to find the best route for the robot. In each place of the room the player sees exactly the same area as the robot will see. He decides which pictures are to be refreshed and what move is to be taken next. His task is

to plan the shortest route for the robot such that all changes in the room are detected as soon as possible. It is assumed that the number of players participating in the game (in parallel) is enough to get satisfactory result that is to find a tour along which one mobile robot can see the entire room within a time limit. Finally, the tour of the game winner is used for driving real robot.

Each game board (the room floor) is organized with two coordinates. X axis lays horizontally, heading right, Y axis – vertically, heading down. Then upper left corner has (0,0) coordinates. Objects are placed in the player path net mesh. The player moves across the board forward or backward (staying in the net corners) and turns 45 degrees left or right, looking at the pictures. If he stands in any point, looking in some direction for the first time (there is no required picture until now), the picture is taken, sent to the player and stored in the database simultaneously. If a picture has already been taken, it is retrieved from the database and sent to the player. It is possible to force the robot to take a new picture, too.

The game is implemented with the help of three software modules: *Game, Client* and *Base*. *Game* is a server connecting clients with the tour simulator. It also establishes connection with *Base* (using TCP/IP and XML) which manages the database. If the database exists the game can be started. The history of the game is logged. *Game* writes down the following information:

- start time and end time of the game and
- for each player:
  - join time and end time,
  - interval between subsequent moves which run by the start of the game,
  - positions that the player chooses i.e. coordinates x, y, z angles alfa and beta and the name of the board,
  - a signal whether a picture should be refreshed or not, and
  - changes found,
- a list of available boards.

Client is a program which connects a player to the *Game* server using TCP/IP. When the player wants to log in he has to give the nick-name, port number and host address. If the data is correct and the nick-name is unique the player may start the game. The host gives the list of available boards. The player may choose one of the boards and set up start position. The address, port, nick name and the start position are remembered for not to set them once again. The player gets the picture of a chosen position (if it exists) after "entry" on the board. Four buttons corresponding with the robot moves may be applied.

The game runs on 2D board (x,y) and one can only round the robot around the axis perpendicular to the board (Alfa). However, the communication takes into consideration full 3 dimensions (x,y,z) and two angles of camera views (Alfa and Beta). The option "refresh" is used to force taking a photo (when it is checked off only the most recent pictures are taken from the database).

Base manages the database, communicates with a robot and *Game*. The database is filled up with all necessary photos of the boards before the game is started (8 photos from each point of the boards – in directions 0, 45, 90, 135, 180, 225, 270, 315 degrees, made at the same time). Each set of photos is taken at different time. Hence, this reflects robot work manner.

The database containing photos runs on Oracle. *Base* opens a socket to communicate with *Game* and makes the connection to the database. Then *Base* and *Game* exchange messages and cooperate with each other in the loop.

In Fig. 1 the associations between the main actors of the game are shown.

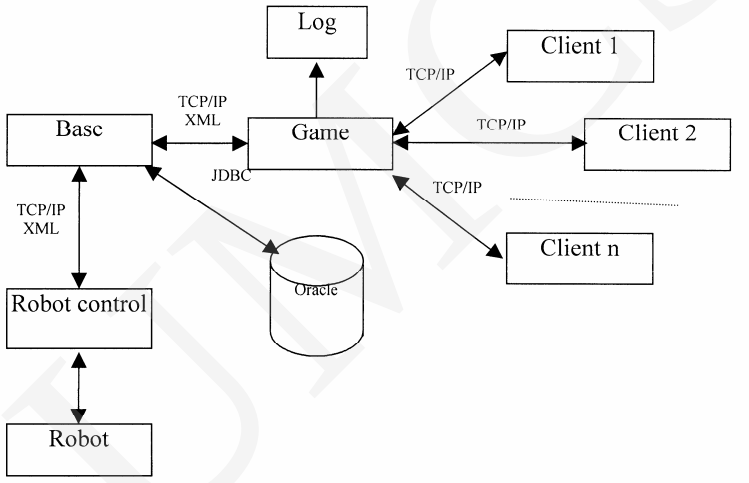


Fig. 1. Main actors of the game

4. Experiment

The aim of the experiment was to investigate to what extent a player equipped with a map of the room and the tour simulator would be able to choose a relatively short tour allowing him to see all required points of the room.

From the preliminary analysis it follows that the following factors can influence upon the length of the tour:

- type of changes
  - when an object disappears from the floor it is enough to inspect its place,
  - when an object appears then the whole admissible area in the room should be inspected.
- visibility, low visibility can harden inspection,
- density of objects in the room, the more objects the larger area should be inspected (the higher probability of interception).

A number of changes in the room should not influence on the tour of the robot. However, in this case a human factor should be taken into consideration. When the number of changes grows the players could make more mistakes.

The experiment was organized as a four-stage sequential decision procedure.

The first stage of the experiment was to familiarize the players with the game, its rules and usage and selection of proper time limits. The players did not know original placement of the objects on the board. They were able to choose freely a starting point, but the default starting point (upper left corner) was usually taken. Time for finding all the changes was not limited. Visibility was also unlimited.

The purpose of the second stage was to select reliable and competent players, so conditions of real scene were created. The following assumptions were changed (compared with the first stage):

- time for finding all the changes was limited,
- the players had maps with the initial state of the board,
- type of changes was known, but the time of changes and the number of them were unknown.

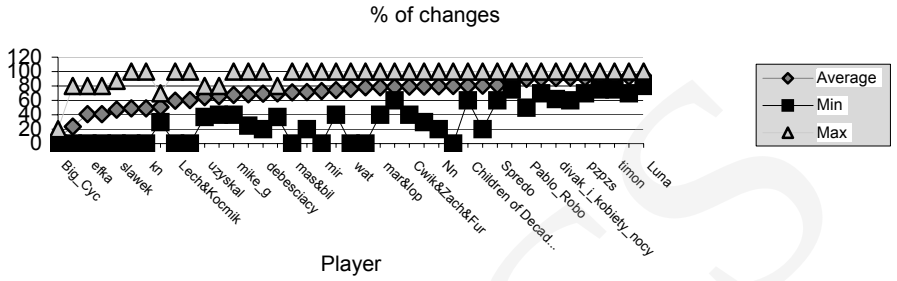
In the third stage of the experiment most successful tactics for driving the robot (most popular among the players and giving average best results in the second stage) within the selected groups of players were tested as regards the number of changes detected. Special cases concerning object placement (objects concentrated in the center or in the corner of the room or equally distributed around it) were taken into consideration. The aim of this stage was to cut out artifacts, if any. At the end of this stage the tactics were ranked according to the following criteria: number of changes discovered (first), percentage of mistakes (next) and number of movements (least).

The fourth stage of the experiment had to give an answer to the question which had motivated this research. For each placement of objects and each type of their changes the best tactic for driving the robot was chosen. Two types of changes, disappearing and appearing of objects, were examined as the most important. Visibility was limited (one, two or five units of length) to make a scene more realistic.

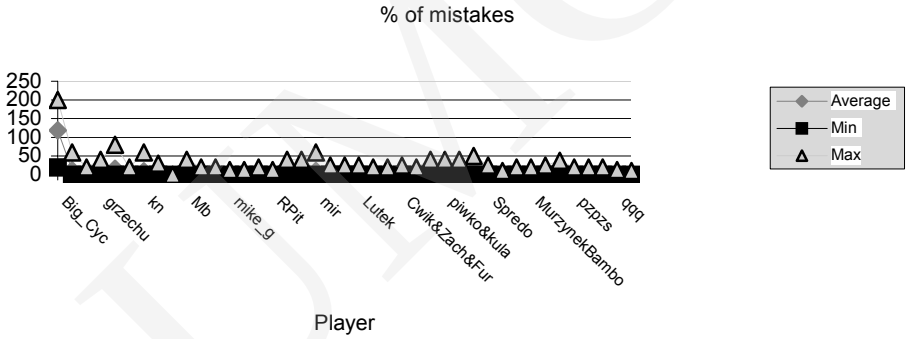
5. Experimental results

At the earliest stage of the experiment (the second stage) we had to check whether the task was trivial for the players or not. Positive answer would stop the research as useless. Figure 2 illustrates the quality of guarding in relevance to the player.

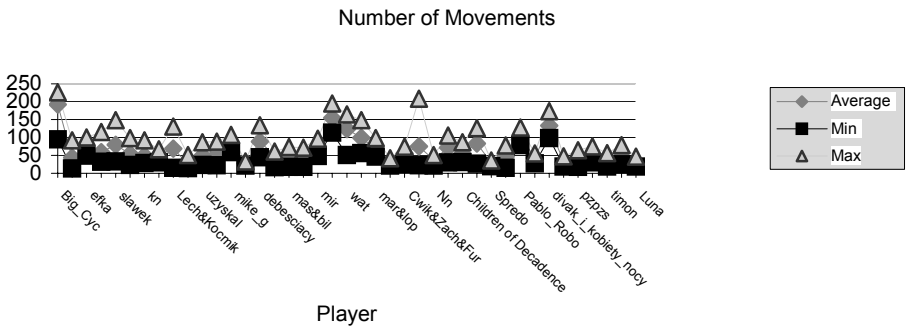
From Fig. 2a it is seen that the task was not trivial for the players and that their abilities to solve the task vary from about 40 up to about 100 percent (two worst results are rejected as false ones – see Figs. 2b and 2c for explanation).



(a)



(b)



(c)

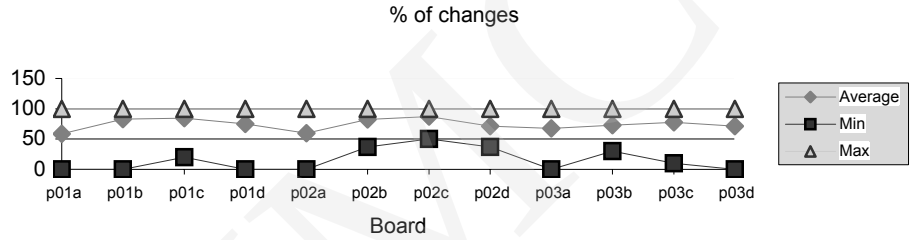
Fig. 2. Quality of guarding: number of changes discovered (a), mistakes (b), player activity (c)

The influence of the number of objects in the room and the type of their changes on the quality of guarding was investigated next. Table 1 describes the cases considered.

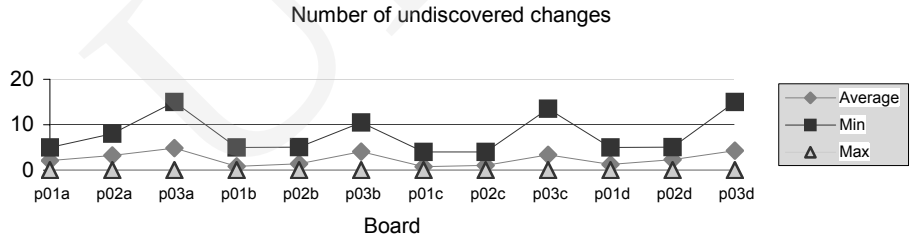
Table 1. Number of objects and types of changes

Number of objects	Disappearance	Appearance	Modification	All
10	P01a	P01b	p01c	p01d
15	P02a	P02b	p02c	p02d
20	P03a	P03b	p03c	p03d

Fig.3 shows that quality of guarding strongly varies depending on the type of changes in the room but it is much less susceptible to the number of objects to be guarded.



(a)



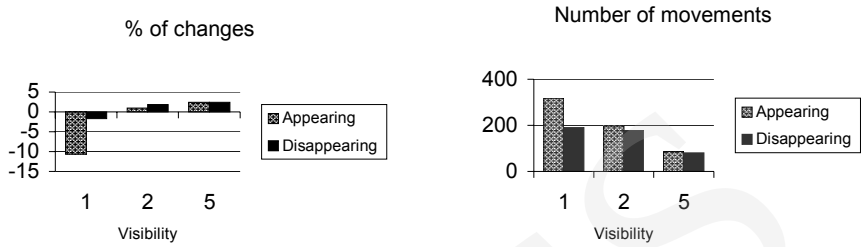
(b)

Fig. 3. Quality of guarding: discovered (a) and undiscovered (b) changes (percents)

Fig. 3b shows the number of undiscovered changes for each of the rooms. It was created assuming that 1 undiscovered change out of 5 decreases efficacy by 20%, but out of 10 only by 10%.

After obtaining some encouraging results we went further with evaluating capabilities of the players. In the fourth stage of the experiment the influence of visibility on the quality of guarding was investigated. Since the players behaved as expected the number of trials was limited.

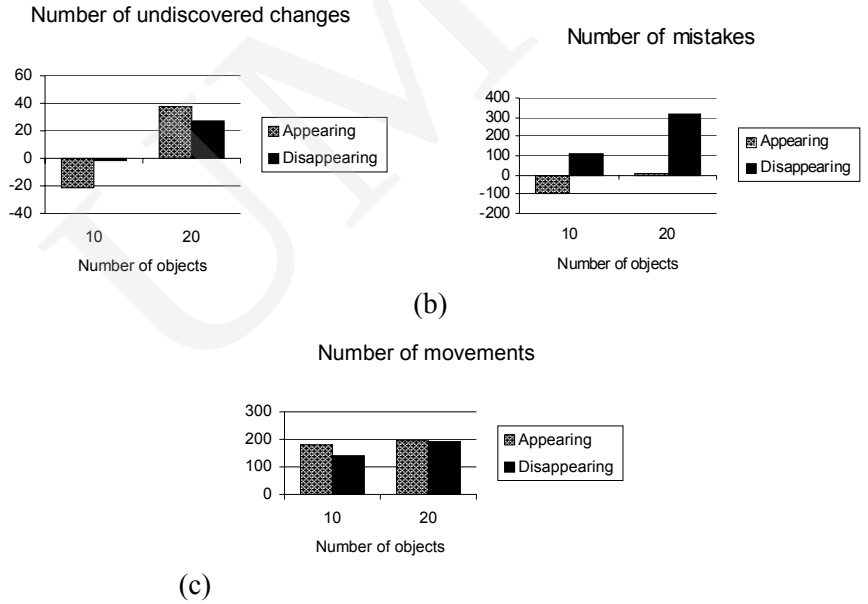
Fig. 4 shows how growing visibility upgrades quality of guarding when the number of objects is constant and equals 15.



(a) (b)

Fig. 4. Quality of guarding when visibility grows: discovered changes – relative (a), required activity (b)

Fig. 5 shows how growing number of objects diminishes quality of guarding when visibility is constant and equals 2.



(a) (b)

(c)

Fig. 5. Quality of guarding when the number of objects grows: number of undiscovered changes (a), mistakes – relative (b) and required activity (c)

6. Conclusions

The basic conclusion drawn from the experiment is positive. An individual having "a map" of the room and the tour simulator is able to plan a relatively short tour along which a mobile robot can see the entire room. The tour of the best player in our experiment was programmed. It was almost two times shorter (as Java bytecode for Lego RCX) than the average one (51.15kB against 90.36kB). Moreover, the game could be used not only for planning the route but

also for selecting effective planners. Only best players participated in the last stage of the experiment.

From the second stage of the experiment it follows that modification of objects is the easiest detectable kind of changes, next to appearance of objects. Disappearance is the hardest detectable kind of changes. When visibility is unlimited then density of objects in the room has low impact on quality of guarding. However, the more objects in the room, the more changes left unnoticed.

When the most successful tour planners worked with limited visibility in the fourth stage of the experiment it was observed that:

- detecting new objects required a little higher activity of the robot than detecting old ones,
- quality of guarding when only old objects might disappear was higher than when appearing of new objects was allowed,
- the more objects in a room the higher activity of a robot,
- the higher density of objects in a room the higher number of mistakes (human factor),
- the better visibility the lower activity of the robot and the higher quality of guarding.

In further research we are going to apply the above observations in developing an algorithm capable of automatic adjustment of robot tour to "on the fly" changes in a room.

References

[1] "Online Searching with an Autonomous Robot" – http://arxiv.org/PS_cache/cs/pdf/0404/0404036.pdf
[2] "Navigation Using Behavior-based and Path-planning Strategies" – http://palantir.swarthmore.edu/maxwell/classes/e28/S00/reports/addo-kim-silk-lab2/
[3] "The Polygon Exploration Problem I: A Competitive Strategy" – http://www.pi6.fernuni-hagen.de/publ/tr241.pdf
[4] "The Polygon Exploration Problem II: The Angle Hull" – http://www.pi6.fernuni-hagen.de/publ/tr245.pdf
[5] "Theoretical Robot Exploration" – http://www.cis.upenn.edu/~isleri/research/wpe/wpe.pdf
[6] „The Art. Galery Problem" – http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/97/Thierry/thierry507webprj/artgallery.html
[7] „Watchman's problem" – http://www.site.uottawa.ca/~jorge/openprob/Watchman/