

The problem of placement of visual landmarks

Anna Gorbenko
Department of Mathematics and
Mechanics
Ural State University
Ekaterinburg, Russia, 620083
Email: gorbenko.aa@gmail.com

Maxim Mornev
Department of Mathematics and
Mechanics
Ural State University
Ekaterinburg, Russia, 620083
Email: max.mornev@gmail.com

Vladimir Popov
Department of Mathematics and
Mechanics
Ural State University
Ekaterinburg, Russia, 620083
Email: Vladimir.Popov@usu.ru

Abstract—Many robotic problems are computationally hard. Implementation of various heuristics for solving such problems greatly complicates the development of efficient software for robotic systems. In recent years among developers of robotic software formed a direction of the development of individual solvers. Such solvers are designed for specific hard problems. It should be noted that developed specialized programming languages for robotic logic. These languages allow efficient scale logic circuits produced by the solver for large robotic complexes. On this basis it can be argued that now the main problem in this area consists in the developing efficient solvers themselves. In this paper we consider an approach to design of an efficient solver for the problem of placement of visual landmarks. In particular, we present a formalization of the problem of placement of visual landmarks for navigation of mobile robots. We show that this problem is NP-complete. Also we propose an approach to solve this problem. Our approach based on the construction of a logical model. In our construction we give an explicit polynomial reduction from the problem of placing of visual landmarks to the maximum satisfiability problem. Such approach provides effective solution of the considered problem using a high performance computing.

I. INTRODUCTION

Many robotic problems are computationally hard. In particular, we can mention planning problems, pattern recognition, pattern matching, localization problems, mapping problems, SLAM (simultaneous localization and mapping) and many others. Implementation of various heuristics for solving such problems greatly complicates the development of efficient software for robotic systems. In recent years among developers of robotic software formed a direction of the development of individual solvers. Such solvers are designed for specific hard problems [1] – [4]. It should be noted that developed specialized programming languages for robotic logic [5], [6]. These languages allow efficient scale logic circuits produced by the solver for large robotic complexes. On this basis it can be argued that now the main problem in this area consists in the developing efficient solvers themselves. In this paper we consider an approach to design of an efficient solver for the problem of placement of visual landmarks.

Visual navigation is extensively used in contemporary robotics (see e.g. [7] – [11]). In many cases methods of the visual navigation based on some algorithms of the selection of visual landmarks. Note that even for other approaches (e.g. reactive motion (see, in particular, [12] – [14]) or topological

navigation [15] – [19]) appliance of visual landmarks as an additional method allow significantly increase performance of a navigation system. We can use not only artificial landmarks [20] – [22] but also different objects from environment (see e.g. [23] – [30]).

Navigation systems based on landmarks are most simple and efficient method of orientation. An appliance of artificial landmarks provides a reliable way of autonomous operations. But for appliance of artificial landmarks we need either initial equipment of coverage area of the robot or the robot itself should have a function of self-installation of landmarks. In both cases decreasing of the number of landmarks have significantly influences on improvement of the robot performance. Another way consist in an implementation of natural landmarks. It is allows to avoid a cost of the landmarks installation. But in this case we need essentially more effective methods of visual information processing than for artificial marks (see e.g. [31], [32]). In particular, for search even a simple regularity we need to solve some NP-hard problem (see e.g. [33], [34]). When we do not have sufficiently fast algorithms for solving such problems, we use some self-learning navigation systems (see e.g. [25]). This leads to fast growth of landmarks data base and, finally, to dramatic drop of performance of on-board computer systems. Therefore, problem of minimizing of interest for both artificial and natural landmarks. In the first case we need to minimize the cost of installation of landmarks. In the second case we minimize an expense of computational resources for the new landmarks search and for the identification of available landmarks. A question about a minimization of the number of used landmarks can be formalized as some algorithmic problem where we need to find an explicit placement of landmarks. This placement must safeguard a navigation of mobile objects in a specified area.

II. PROBLEM DEFINITION

Usually, a problem of placement of visual landmarks in a three dimensional space can be reduced to such problem in a two dimensional space. Frequently, an appropriate reduction can be obtained using the fact that range of heights is too small in compare with horizontal coordinates (an underwater navigation) or by a sampling with a large step (an aerospace navigation). In both cases we can place a set of points, which corresponds to different heights in the one cell of

sampling. When the range of heights is too large and horizontal coordinates have a small variation (a navigation inside a skyscraper), usually, space is divided on horizontal layers lying in one plane. Thus, without loss of generality, we can consider a problem of placement of visual landmarks in the discrete space Z^2 , where Z is the set of integers. Also we identify every element x of Z^2 with a square with sides equal to one and the center in x . A set of points in Z^2 which of interest to navigation we denote by N . Note that N is not necessarily a connected area. For instance, we can be interested only in surface facilities and a part of the surrounding area can be covered by water. Let S be a set of points in Z^2 which permissible to placement of landmarks. It is natural to assume that we are dealing with some limited region R such that $N \subseteq R$, $S \subseteq R$. Since the deployment region R can contain obstacles or visual landmarks can be visible not from all points of space, it is natural to assign for each point of the set S its own field of vision which is defined by the function

$$F : S \rightarrow 2^R.$$

We can suppose that F is given by the sequence of pairs consisting of elements of S and corresponding subsets. We also consider some constant d which determines a minimal number of necessary landmarks. Note that the value of d , usually, does not exceed 4. In the form of a satisfiability problem the considering problem can be formulated as follows:

THE PROBLEM OF PLACEMENT OF VISUAL LANDMARKS (VL):

INSTANCE: A finite set R , $S \subseteq R$, $N \subseteq R$, $F : S \rightarrow 2^R$, a natural parameter k , and a natural constant d .

QUESTION: Is there $T \subseteq S$ such that $|T| \leq k$ and for all $y \in N$ there is $D \subseteq T$ such that

$$y \in F(x) \text{ for all } x \in D;$$

$$|D| \geq d?$$

III. COMPLEXITY OF VL

Theorem. VL is **NP**-complete.

Proof. Note that $T \subseteq S$. Therefore, the number of elements in T is limited by the number of elements in S . A size of $F(x)$ for $x \in T$ is limited by the size of N . It is evident that the value

$$\cup_{x \in T} F(x)$$

is computable in polynomial time from the number of elements in T and the size of $F(x)$. So, for the problem VL there is a polynomial algorithm of checking. Therefore, VL is in **NP**.

Now to prove the theorem it is sufficient to show the hardness of VL. We reduce the 3-set exact cover problem to VL. The 3-set exact cover problem is a well known **NP**-hard problem [35]. Initial data of this problem is the set

$$U = \{1, 2, \dots, n\}$$

and the set

$$\Sigma = \{X_i \mid 1 \leq i \leq r, X_i \subset U, |X_i| = 3\}.$$

In this problem we need to find out whether there is $\Pi \subseteq \Sigma$ such that for all $X, Y \in \Pi$ we have a following

$$X \cap Y = \emptyset,$$

$$\cup_{X \in \Pi} X = U.$$

When n is not divisible by three, the answer is trivially negative. So, without loss of generality, in the 3-set exact cover problem we can assume that n is divisible by three.

$$\text{Let } d = 1, k = \frac{n}{3},$$

$$N = \{((n+r-1)(2(r-1)+2i-1), 0) \mid i \in U\},$$

$$S = \{((n+r-1)(n+r-1+2(j-1)),$$

$$(n+r-1)(n+r-1)) \mid 1 \leq j \leq r\}.$$

For any point

$$S_j = ((n+r-1)(n+r-1+2(j-1)), (n+r-1)(n+r-1))$$

from S consider the triangle \mathcal{T}_j with vertices in this point and points

$$A_j = (2(n+r-1)(j-1), 0)$$

and

$$B_j = (2(n+r-1)(n+r+j-2), 0).$$

Suppose that for any m , $1 \leq m \leq n+r-1$, the segment with vertices

$$(2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2(m-1),$$

$$(n+r-1)(n+r-2))$$

and

$$(2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2m,$$

$$(n+r-1)(n+r-2)),$$

represents a tight space of zero width if and only if $m-r+j \notin X_j$. Assume that the signal propagates rectilinear and define $F(S_j)$ as a set of points x of triangle \mathcal{T}_j such that x is visible from S_j . As R we consider rectangle, the bottom left corner of which is located at the point $(0, 0)$ and a upper right corner of which is located at the point

$$(2(n+r-1)(n+3r-3), 2(n+r-1)(n+r-1)).$$

It is easy to see that values of R , S , N , F , k , d is defined correctly and their size polynomially depends from the initial data. Let \mathcal{T}'_j be a triangle with vertices S_j ,

$$A'_j = (2(n+r-1)(j-1) + (n+r-1)(n+r-2),$$

$$(n+r-1)(n+r-2)),$$

and

$$B'_j = (2(n+r-1)(j-1) + (n+r-1)(n+r),$$

$$(n+r-1)(n+r-2)).$$

Note that

$$A'_j \vec{S}_j = ((n+r-1)(n+r-1+2(j-1)) -$$

$$\begin{aligned}
& 2(n+r-1)(j-1), (n+r-1)(n+r-1)) = \\
& ((n+r-1)(n+r-1), (n+r-1)(n+r-1)), \\
& A'_j \vec{S}_j = ((n+r-1)(n+r-1+2(j-1)) - \\
& 2(n+r-1)(j-1) - (n+r-1)(n+r-2), \\
& (n+r-1)(n+r-1) - (n+r-1)(n+r-2)) = \\
& (n+r-1, n+r-1).
\end{aligned}$$

So, $A'_j \vec{S}_j = (n+r-1)A'_j \vec{S}_j$. Therefore,

$$A'_j \vec{S}_j \parallel A'_j \vec{S}_j.$$

In this case $A'_j \in A_j S_j$. Since

$$\begin{aligned}
& B'_j \vec{S}_j = ((n+r-1)(n+r-1+2(j-1)) - \\
& 2(n+r-1)(n+r+j-2), (n+r-1)(n+r-1)) = \\
& (-(n+r-1)(n+r-1), (n+r-1)(n+r-1)), \\
& B'_j \vec{S}_j = ((n+r-1)(n+r-1+2(j-1)) - \\
& 2(n+r-1)(j-1) - (n+r-1)(n+r), \\
& (n+r-1)(n+r-1) - (n+r-1)(n+r-2)) = \\
& (-n-r+1, n+r-1),
\end{aligned}$$

it is easy to see that $B'_j \in B_j S_j$. Clearly, in this case

$$A'_j B'_j \parallel A'_j B'_j.$$

Thus, triangles \mathcal{T}_j and \mathcal{T}'_j are similar with the similarity ratio

$$n+r-1.$$

It is easy to see that for all j segments with vertices

$$\begin{aligned}
& (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2(m-1), \\
& (n+r-1)(n+r-2))
\end{aligned}$$

and

$$\begin{aligned}
& (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2m, \\
& (n+r-1)(n+r-2)),
\end{aligned}$$

$1 \leq m \leq n+r-1$, give us a partition of the segment $A'_j B'_j$ into $n+r-1$ equal parts. It is evident that $A'_{j+1} = B'_j$ where $j < r$,

$$\begin{aligned}
A'_j &= (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2(m-1), \\
& (n+r-1)(n+r-2))
\end{aligned}$$

where $m = 1$ and

$$\begin{aligned}
B'_j &= (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2m, \\
& (n+r-1)(n+r-2))
\end{aligned}$$

where $m = n+r-1$. From this and from the similarity of triangles \mathcal{T}_j and \mathcal{T}'_j we obtain that points of the segments with vertices

$$(2(n+r-1)(j-1) + 2(m-1)(n+r-1), 0)$$

and

$$(2(n+r-1)(j-1) + 2m(n+r-1), 0)$$

is visible from the point S_j if and only if points of the segments with vertices

$$\begin{aligned}
& (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2(m-1), \\
& (n+r-1)(n+r-2))
\end{aligned}$$

and

$$\begin{aligned}
& (2(n+r-1)(j-1) + (n+r-1)(n+r-2) + 2m, \\
& (n+r-1)(n+r-2))
\end{aligned}$$

not form an obstacle. Therefore, the segments with vertices

$$(2(n+r-1)(j-1) + 2(m-1)(n+r-1), 0)$$

and

$$(2(n+r-1)(j-1) + 2m(n+r-1), 0),$$

is visible from the point S_j if and only if

$$m-r+j \in X_j.$$

Consider the point

$$N_i = ((n+r-1)(2(r-1) + 2i-1), 0).$$

It is easy to see that the point N_i belongs to the segment

$$\begin{aligned}
& [(2(n+r-1)(j-1) + 2(m-1)(n+r-1), 0); \\
& (2(n+r-1)(j-1) + 2m(n+r-1), 0)]
\end{aligned}$$

if and only if

$$\begin{aligned}
& 2(n+r-1)(j-1) + 2(m-1)(n+r-1) \leq \\
& (n+r-1)(2(r-1) + 2i-1) \leq \\
& 2(n+r-1)(j-1) + 2m(n+r-1).
\end{aligned}$$

Hence,

$$\begin{aligned}
& (n+r-1)(2(r-1) + 2i-1) - \\
& 2(n+r-1)(j-1) - 2(m-1)(n+r-1) \geq 0, \\
& 2(n+r-1)(j-1) + 2m(n+r-1) - \\
& (n+r-1)(2(r-1) + 2i-1) \geq 0.
\end{aligned}$$

Therefore,

$$\begin{aligned}
& 2(r-1) + 2i-1 - 2(j-1) - 2(m-1) = \\
& 2r-2+2i-1-2j+2-2m+2 = \\
& 2r+2i-2j-2m+1 \geq 0, \\
& 2(j-1) + 2m - 2(r-1) - 2i+1 = \\
& 2j-2+2m-2r+2-2i+1 = \\
& 2j+2m-2r-2i+1 \geq 0.
\end{aligned}$$

Let $i = m-r+j+a$. In this case we have

$$2a+1 \geq 0,$$

$$-2a + 1 \geq 0.$$

So, $a = 0$. Therefore, $i = m - r + j$. Clearly, N_i belongs to the segment

$$[(2(n+r-1)(j-1) + 2(m-1)(n+r-1), 0);$$

$$(2(n+r-1)(j-1) + 2m(n+r-1), 0)]$$

if and only if

$$i = m - r + j.$$

Therefore, the point N_i is visible from the point S_j if and only if $i \in X_j$. \square

IV. A LOGICAL MODEL FOR VL

Since the problem VL is **NP**-complete, there is no polynomial time algorithm for finding a solution for this problem. However, since the considered problem is of significant practical use, we need to find a fast algorithm for solving this problem.

Well known, many problems with practical applications belong to the class of computational complexity **NP**. Also all problems from the **NP** can be polynomially reduced to the problem of satisfiability of a Boolean function (SAT) (see e.g. [36]). This problem can be formulated as follows

BOOLEAN SATISFIABILITY PROBLEM IN CONJUNCTIVE NORMAL FORM (SAT):

INSTANCE: *A Boolean function*

$$g(x_1, x_2, \dots, x_n)$$

in conjunctive normal form.

QUESTION: *Is there an assignment of the set of variables such that*

$$g(x_1, x_2, \dots, x_n) = 1?$$

The satisfiability problem is a core problem in mathematical logic and computing theory. In practice, SAT is fundamental in solving many problems in automated reasoning, computer-aided design, computer-aided manufacturing, machine vision, database, robotics, integrated circuit design, computer architecture design, and computer network design. Traditional methods treat SAT as a discrete, constrained decision problem. Many optimization methods, parallel algorithms, and practical techniques have been developed for solving SAT. Recently, in the domain of development of fast algorithms to solving SAT was achieved a substantial progress (see, in particular, [37]). Most studies have focused on genetic algorithms and algorithms of local search. Note that significant attention is focused both on SAT and its optimization version MAXSAT (see e.g. [36]) which can be formulated as follows.

MAXIMUM SATISFIABILITY PROBLEM IN CONJUNCTIVE NORMAL FORM (MAXSAT):

INSTANCE: *A Boolean function*

$$g(x_1, x_2, \dots, x_n)$$

in conjunctive normal form.

QUESTION: *Is there an assignment of the set of variables such that in the function*

$$g(x_1, x_2, \dots, x_n)$$

the maximum number of clauses is true?

Recently, proposed several genetic algorithms [38] – [41]. Considered also hybrid algorithms based on combinations of genetic algorithms and local search algorithms [42]. Relatively high efficiency can be achieved for algorithms based solely on the local search. Of course, these algorithms run in exponential time in worst case. But they can relatively fast obtain a solution for many Boolean functions arising in practice. So, a reduction from hard problems to SAT and MAXSAT for its solving acquires a practical sense. For example, such approach was considered for hamiltonian path problem in [43], [44].

Consider a reduction the problem VL to the MAXSAT. For all i , $1 \leq i \leq n$, consider a set

$$M_i = \{p \mid b_i \in F(a_p)\}.$$

It is easy to see that the system of sets M_i , $1 \leq i \leq n$, can be constructed in polynomial time. Obviously, if for some i we have a following inequality $|M_i| < d$, then the solution for the problem VL is negative.

Thus, in further, without loss of generality, we assume that for all i it is true that $|M_i| \geq d$. Therefore, for all i we can consider the system

$$M_{i,1}, M_{i,2}, \dots$$

from

$$\frac{|M_i|(|M_i| - 1) \dots (|M_i| - d + 2)}{(d - 1)!}$$

pairwise different subsets of the set M_i each of which consists from $|M_i| - d + 1$ elements.

Directly from the definition of $M_{i,j}$ follows that if for some i and arbitrary j each of sets $M_{i,j}$ contains at least one number of a point of a landmark location, then from the point b_i is visible at least d landmarks.

We interpret $x_l = 1$ as presence of some landmark in the point with the number l . Hence, we obtain that

$$1 \leq j \leq \frac{|M_i|(|M_i|-1)\dots(|M_i|-d+2)}{(d-1)!} \quad (\bigvee_{l \in M_{i,j}} x_l)$$

is true if and only if from the point b_i is visible at least d landmarks.

Now we show that

$$\varphi = (\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq r \leq m+1, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1)\dots(|M_i|-d+2)}{(d-1)!}}} ((\bigvee_{l \in M_{i,j}} x_l) \vee s_r)) \wedge$$

$$(\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq r \leq m+1, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1)\dots(|M_i|-d+2)}{(d-1)!}}} ((\bigvee_{l \in M_{i,j}} x_l) \vee \neg s_r)) \wedge$$

$$\left(\bigwedge_{1 \leq t \leq m} \neg x_t \right)$$

provides a reduction from VL to MAXSAT.

Note that in φ the total number of clauses is equal to

$$2n(m+1) \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!} + m.$$

It is easy to see if there is a required placement of landmarks, then the Boolean function

$$\left(\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!}}} \bigvee_{l \in M_{i,j}} x_l \right) \wedge$$

$$\left(\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!}}} \bigvee_{l \in M_{i,j}} x_l \right)$$

is satisfiable. This obviously implies a satisfiability of the Boolean function

$$\left(\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq r \leq m+1, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!}}} \left(\bigvee_{l \in M_{i,j}} x_l \right) \vee s_r \right) \wedge$$

$$\left(\bigwedge_{\substack{1 \leq i \leq n, \\ 1 \leq r \leq m+1, \\ 1 \leq j \leq \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!}}} \left(\bigvee_{l \in M_{i,j}} x_l \right) \vee \neg s_r \right).$$

If there is required placement of landmarks, then in the Boolean function φ at least

$$2n(m+1) \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!}$$

clauses is true.

Suppose, that there is no required placement of landmarks. Then there is i and j such that the Boolean function

$$\bigvee_{l \in M_{i,j}} x_l$$

is false. Therefore for all r either

$$\left(\bigvee_{l \in M_{i,j}} x_l \right) \vee s_r$$

or

$$\left(\bigvee_{l \in M_{i,j}} x_l \right) \vee \neg s_r$$

is false. Thus, in the Boolean function φ no more then

$$2n(m+1) \frac{|M_i|(|M_i|-1) \dots (|M_i|-d+2)}{(d-1)!} - 1$$

clauses is true. Therefore, we obtain a reduction from VL to MAXSAT.

V. CONCLUSION AND EXPERIMENTAL RESULTS

In previous section we obtain an implicit reduction from VL to MAXSAT.

New algorithms allow ordinary desktop computers to solve boolean functions in conjunctive normal form, which has more than 10000 conjuncts (see e.g. [43]). There is a well known site on which posted solvers for SAT [45]. Currently on the site published 16 implementations of algorithms for solving SAT. They are divided into two main classes: stochastic local search algorithms and algorithms improved exhaustive search. All solvers allow the conventional format for recording DIMACS Boolean function in conjunctive normal form and solve the corresponding problem [46]. In addition to the solvers the site also represented a large set of test problems in the format of DIMACS. This set includes a randomly generated problems of satisfiability.

For the computational experiments we used heterogeneous cluster based on three clusters:

- The cluster of Ural State University (8 computational nodes, Linux, processor Intel Pentium IV 2.40GHz);
- The cluster umt of Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences (256 computational nodes, Linux, processor Xeon 3.00GHz) [47] (see also [48]);
- The cluster um64 of Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences (124 computational nodes, Linux, dual-core processor AMD Opteron 2.6GHz bi-processors) [47] (see also [48]).

In our experiments we used own genetic algorithm MSAT and two standard solvers [45] (fgrasp and posit). Computational experiments were carried out on standard tests [45] and tests, obtained by special generators creating a natural data for the problem VL and for a number of others robotics problems.

The total was carried out 14 runs of cluster in the format of 100 nodes on 20 hours. For summarizing final statistics were selected 200 tests. Chosen tests was counterated by three solvers in 5 modes with constraints on the limit of time

- no restriction;
- 10 seconds;
- 100 seconds;
- 500 seconds;
- 1000 seconds.

We found that all three solvers have about the same performance. The best average velocity is showed by fgrasp. A slightly lower average velocity is showed by posit. Genetic algorithm showed a worst average velocity. From other hand we found that algorithms of local search are more resource demanding then the genetic algorithm. In particular, on some tests fgrasp could not finish the execution.

From our experiments we obtain an important property of genetic algorithm. Algorithms of local search showing relatively smooth results on several tests but genetic algorithm has a significant difference (from 26 seconds to 12 hours) in time during the test execution. The total trend can be described as follows. In many cases run time ranges from few seconds

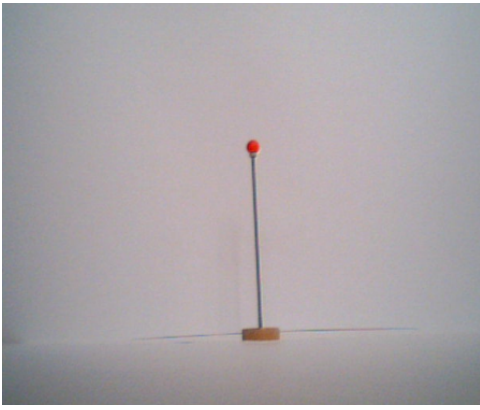


Fig. 1. An artificial landmark.



Fig. 2. Semi-artificial landmarks.

to few minutes. However, on a small set of Boolean functions a run time of genetic algorithm increased to 10-12 hours.

Our experiments have shown that our approach can be used for design of an efficient solver for the problem of placement of visual landmarks. Further advantages we can get by improving our genetic algorithm.

The first author developed a software package for processing video data to compute three-dimensional coordinates of objects [49]. This software package is designed for visual navigation on landmarks. To date, an intelligent system allowing the use of artificial (specially designed) and semi-artificial (located in a special way but not specifically designed) landmarks of various types implemented (e.g. Fig. 1 and 2).

Module of visual navigation based on landmarks used as part of the onboard control system of various modifications of robots Kuzma-I (e.g. Fig. 3 and 4) and Kuzma-II (e.g. Fig. 5 and 6). The onboard control system with module of visual navigation based on landmarks of a modifications of Kuzma-II (Fig. 5) was demonstrated at the International Exhibition INNOPROM – 2010 (15.07.2010 – 17.07.2010).

For our experiments on intelligent control systems, we use

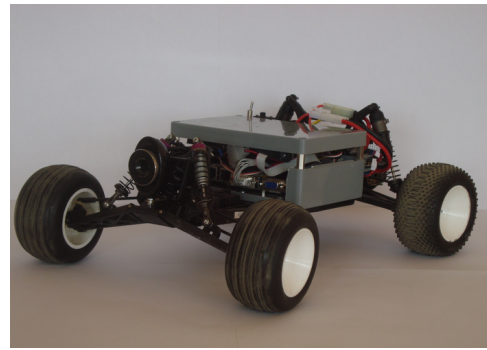


Fig. 3. Robot Kuzma-I. Design of this robot based on the well-known RC cars. From RC-CAR AT-10ES Thunder Tiger [50] we use only the four wheel chassis, the high torque DC-MOTOR and a steering servo. The DC-MOTOR drives the chassis and a steering servo controls the direction. The electronic system based on SSC-32 microcontroller. Onboard computer based on a motherboard with x86 compatible processor AMD Geode LX600 for embedded systems. The robot is equipped with USB web camera Live! Cam Video IM Pro (VF0410) [52].

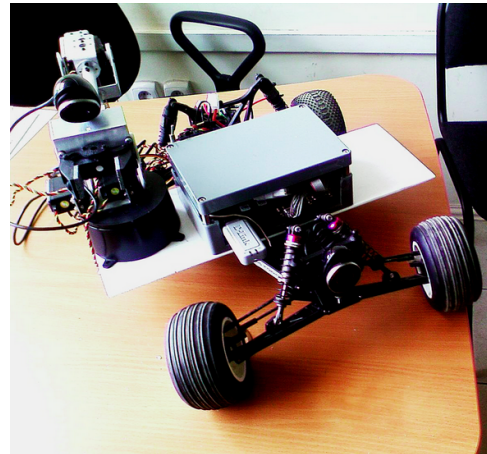


Fig. 4. Another modification of Kuzma-I. The robot is equipped with modified Lynxmotion [51] robotic arm and 2 x USB web camera Live! Cam Video IM Pro (VF0410).

a testbed composed of these mobile robots and a stationary monitoring system. In particular, we study different algorithms of visual navigation based on landmarks. In general, good results of robotic experiments do not guarantee high efficiency of algorithms. Perhaps experiments are conducted in too simple environments. Our theoretical results help us to select appropriate methods as well as testbeds to demonstrate them. Some of our robots are able to add their own landmarks (e.g. Fig. 4 and 5). They use a wireless connection to a supercomputer to run the solver for VL to plan their actions.

ACKNOWLEDGMENT

The work was partially supported by Grant of President of the Russian Federation MD-1687.2008.9 and Analytical Departmental Program "Developing the scientific potential of high school" 2.1.1/1775.



Fig. 5. Robot Kuzma-II. Design of this robot based on the well-known Johnny 5 Robot [53]. By utilizing heavy duty polypropylene and rubber tracks with durable ABS molded sprockets the robot has excellent traction. It includes two 12vdc 50:1 gear head motors and the Sabertooth 2 x 5 R/C motor controller. The electronic system based on SSC-32 microcontroller. Onboard computer of this robot is Asus Eee PC 1000HE. The robot is equipped with modified Lynxmotion robotic arm with wrist rotate and USB web camera Live! Cam Video IM Pro (VF0410).

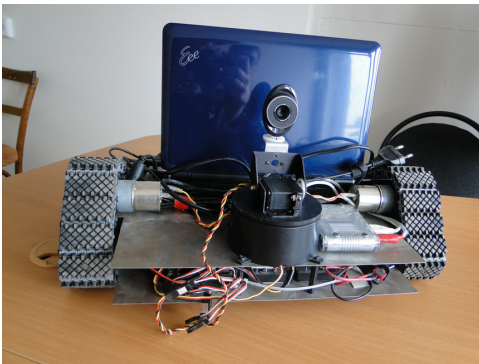


Fig. 6. Another modification of Kuzma-II. The robot is equipped with a 2 DOF robotic camera (USB web camera Live! Cam Video IM Pro (VF0410)).

REFERENCES

- [1] Rus D., Vona M. *Crystalline Robots: Self-reconfiguration with Unit-compressible Modules*. Autonomous Robots. 2001. Vol. 10(1). P. 107–124.
- [2] Ünsal C., Khosla P.K. *A Multi-layered Planner for Self-Reconfiguration of a Uniform Group of I-Cube Modules*. IEEE International Conference on Intelligent Robots and Systems. 2001. Vol. 1. P. 598–605.
- [3] Christensen D., Ostergaard E., Lund H.H. *Metamodule control for the atron self-reconfigurable robotic system*. Proceedings of the The 8th Conference on Intelligent Autonomous Systems. 2004. P. 685–692.
- [4] Dewey D., Srinivasa S.S., Ashley-Rollman M.P., De Rosa M., Pillai P., Mowry T.C., Campbell J.D., Goldstein S.C. *Generalizing Metamodules to Simplify Planning in Modular Robotic Systems*. Proceedings of IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems. 2008. P. 1338–1345.
- [5] Ashley-Rollman M., Goldstein S., Lee P., Mowry T., Pillai P. *Meld: A declarative approach to programming ensembles*. Proceedings of the IEEE International Conference on Robots and Systems. 2007. P. 2794–2800.
- [6] Charron-Bost B., Delporte-Gallet C., Fauconnier H. *Local and temporal predicates in distributed systems*. ACM Transactions on Programming Languages and Systems. 1995. Vol. 17(1). P. 157–179.
- [7] Lowe D. *Object Recognition from Local Scale-Invariant Features*, ICCV. 1999. P. 1150–1157.
- [8] Dudek G., Jugessur D. *Robust Place Recognition using Local Appearance based Methods*. Proceedings of 2000 IEEE International Conference on Robotics and Automation. 2000. P. 1030–1035.
- [9] Carneiro G., Jepson A.D. *Multi-scale Phase-based Local Features*. CVPR. 2003. P. 736–743.
- [10] Yang G., Hou Z.-G., Liang Z. *Distributed visual navigation based on neural Q-learning for a mobile robot*. International Journal of Vehicle Autonomous Systems. 2006. Vol. 4(2-4). P. 225–238.
- [11] *Visual navigation system for a mobile robot having capabilities of regenerating of hidden images*. United States Patent 4887223.
- [12] Graf B. *Reactive navigation of an intelligent robotic walking aid*. Proceedings of the 2001 IEEE International Workshop on Robot Human Interaction. 2001. P. 353–358.
- [13] Pal P.K., Kar A. *Sonar-Based Mobile Robot Navigation Through Supervised Learning on a Neural Net*. Autonomous Robots. 1996. Vol. 3. P. 355–374.
- [14] Saint-Bauzel L., Pasqui V., Monteil I. *A reactive robotized interface for lower limb rehabilitation: clinical results*. IEEE Transactions on Robotics. 2009. Vol. 25(3). P. 583–592.
- [15] Angeli A., Filliat D., Doncieux S., Meyer J.-A. *Visual topological SLAM and global localization*. Proceedings of the International Conference on Robotics and Automation. 2009. P. 1–6.
- [16] Angeli A., Filliat D., Doncieux S., Meyer J.-A. *Incremental vision-based topological SLAM*. Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems. 2008. P. 1–6.
- [17] Filliat D. *Interactive learning of visual topological navigation*. Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems. 2008. P. 1–7.
- [18] Goedemé T., Nuttin M., Tuytelaars T., Van Gool L. *Omnidirectional Vision Based Topological Navigation*. International Journal of Computer Vision. 2007. Vol. 74(3). P. 219–236.
- [19] Vale A., Isabel Ribeiro M. *Environment Mapping as a Topological Representation*. Proceedings of the 11th International Conference on Advanced Robotics. 2003. P. 1–6.
- [20] Kazumi O., Hidenori T., Takanori E., Takeshi T., Shigenori O. *Navigation using Artificial Landmark without using Coordinate System*. Nippon Robotto Gakkai Gakujutsu Koenkai Yokoshu. 2003. Vol. 21. P. 3J11.
- [21] Samuelsson M. *Artificial landmark navigation of an autonomous robot*. Master Thesis. Örebro University, Department of technology, SE-70182. Örebro, Sweden, 2005.
- [22] Hellmann I., Siemiatkowska B. *Artificial landmark navigation system*. International symposium on intelligent robotic systems N 9. 2001. P. 219–228.
- [23] Åstrand B., Baerveldt A.-J. *An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control*. Autonomous Robots. 2002. Vol. 13. P. 21–35.
- [24] Hagrais H., Callaghan V., Colley M. *Prototyping design and learning in outdoor mobile robots operating in unstructured outdoor environments*. IEEE International Robotics and Automation Magazine. 2001. Vol. 8(3). P. 53–69.
- [25] Hagrais H., Colley M., Callaghan V. *Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture*. Autonomous Robots. 2002. Vol. 13. P. 37–52.
- [26] Jung H.C. *Visual Navigation for a Mobile Robot Using Landmarks*. Advanced Robotics. 1994. Vol. 9(4). P. 429–442.
- [27] Gilg A., Schmidt G. *Landmark-oriented visual navigation of a mobile robot*. IEEE International Symposium on Industrial Electronics. 1993. P. 257–262.
- [28] Zhu Z., Oskiper T., Samarasekera S., Kumar R. *Precise visual navigation using multi-stereo vision and landmark matching*. Proceedings of the SPIE. 2007. Vol. 6561. P. 656108.
- [29] Murrieta-Cid R., Parra C., Devy M. *Visual navigation in natural environments: from range and color data to a landmark-based model*. Autonomous Robots. 2002. V. 13(2). P. 143–168.
- [30] Sala P., Sim R., Shokoufandeh A., Dickinson S. *Landmark Selection for Vision-Based Navigation*. IEEE Trans. on Robotics. 2006. V. 22(2). P. 334–349.
- [31] Basri R., Rivlin E. *Localization and Homing Using Combinations of Model Views*. AI. 1995. Vol. 78(1-2). P. 327–354.
- [32] Wilkes D., Dickinson S., Rivlin E., Basri R. *Navigation Based on a Network of 2D Images*. ICPR-A. 1994. P. 373–378.

- [33] Popov V. *The approximate period problem for DNA alphabet*. Theoretical Computer Science. 2003. Vol. 304. P. 443–447.
- [34] Popov V. *The Approximate Period Problem*. IAENG International Journal of Computer Science. 2009. Vol. 36(4). P. 268–274.
- [35] Garey M.R., Johnson D.S. *Computers and Intractability. A Guide to the Theory of NP-completeness*. W. H. Freeman, California, 1979.
- [36] Papadimitriou C.H. *Computational Complexity*. Addison-Wesley Publishing Company, Reading/Menlo Park, NY, 1994.
- [37] Gu J., Purdom P., Franco J., Wah B. *Algorithms for the Satisfiability (SAT) Problem: A Survey*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 1996. P. 19–152.
- [38] Fleurent J. *Genetic algorithms and hybrids for graph coloring*. Annals of Operations Research. 1996. Vol. 63. P. 437–461.
- [39] Hao J., Dorne R. *A new population-based method for satisfiability problems*. Proceedings of 11th European Conference on Artificial Intelligence. 1994. P. 135–139.
- [40] Jong K., Spears W. *Using genetic algorithms to solve np-complete problems*. Proceedings of the International Conference on Genetic Algorithms. 1989. P. 124–132.
- [41] Voorn R., Dastani M., Marchiori E. *Finding simplest pattern structures using genetic programming*. Proceedings of the Genetic and Evolutionary Computation Conference. 2001. P. 3–10.
- [42] Hao J., Lardeux F., Saubion F. *A hybrid genetic algorithm for the satisfiability problem*. Proceedings of the 1st International Workshop on Heuristics. 2002. P. 102–109.
- [43] Iwama K., Miyazaki S. *SAR-variable complexity of hard combinatorial problems*. IFIP Trans. A Comput. Sci. Tech. 1994. Vol. I. P. 253–258.
- [44] Plotnikov A.D. *A Logical Model of HCP*. International Journal of Mathematics and Mathematical Sciences. 2001. Vol. 26(11).
- [45] <http://people.cs.ubc.ca/~hoos/SATLIB/index-ubc.html>
- [46] <http://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/satformat.ps>
- [47] http://parallel.imm.uran.ru/mvc_now/hardware/supercomp.htm
- [48] <http://parallel.uran.ru/node/6>
- [49] Gorbenko A. *Software for processing video data to compute three-dimensional coordinates of objects*. Bachelor Thesis, Department of Mathematics and Mechanics, Ural State University, Ekaterinburg, 2009. (in russian)
- [50] <http://www.tiger.com.tw/>
- [51] <http://www.lynxmotion.com/>
- [52] [http://support.creative.com/Products/ProductDetails.aspx?%20catID=218&CatName=Web+Cameras&subCatID=%20846&subCatName=Live!+Cam+Series&prodID=16904&prodName=Live!+Cam+Video+IM+Pro+\(VF0410\)&bTopTwenty=1&VARSET=prodfaq:PRODFAQ_16904,VARSET=CategoryID:218](http://support.creative.com/Products/ProductDetails.aspx?%20catID=218&CatName=Web+Cameras&subCatID=%20846&subCatName=Live!+Cam+Series&prodID=16904&prodName=Live!+Cam+Video+IM+Pro+(VF0410)&bTopTwenty=1&VARSET=prodfaq:PRODFAQ_16904,VARSET=CategoryID:218)
- [53] <http://www.lynxmotion.com/c-103-johnny-5.aspx>