

---

**APPLIED COMPUTER SYSTEMS**

---

**LIETIŠKĀS DATORSISTĒMAS****APPLICATION OF MAPPING METHODS FOR SOLVING  
NAVIGATION TASKS OF AUTONOMOUS INTELLIGENT  
SYSTEM****KARTĒŠANAS METOŽU PIELIETOJUMS AUTONOMAS  
INTELIĢENTAS SISTĒMAS NAVIGĀCIJAS UZDEVUMU  
RISINĀŠANAI**

**Laura Sudraba**, *Riga Technical University,*  
*Meza 1/4, Riga, LV-1048, assistant,*  
*lsudraba@gmail.com*

**Agris Nikitenko**, *Riga Technical University,*  
*Meza 1/4, Riga, LV-1048, lecturer Dr.sc.ing.,*  
*agris@cs.rtu.lv*

*Intelligent systems, Autonomous systems, Automatic mapping, Occupancy grid, Intelligent robotic system*

**1. Introduction**

Artificial intelligence is one of the youngest branches of modern science. Its main sphere of studies is problem solving and knowledge representation. During the last few decades plenty of methods, technologies and approaches in both these directions developed, allowing to solve increasing range of problems. Methods of artificial intelligence are applied practically in all economic sectors, but particularly widely in medicine, military sphere, manufacturing, businesses, etc.

Intelligent systems in the context of this paper fall into two groups – autonomous and supervised. The autonomous systems unlike the supervised ones can operate using their own experience without human (supervisor) intervention or help [1]. The development of autonomous systems faces with wide range of problems concerning decision making, knowledge acquisition and representation because these systems have to acquire, organize knowledge and make decisions on their own. Therefore there are necessarily situations in when decision has to be made on incomplete knowledge and limited information about the system's environment [1]. As regard knowledge representation the autonomous systems are

specific because they link units of the knowledge unaided and unassisted by humans (supervisors) or other artificial systems. So the development of the autonomous systems is usually more complex than the development of supervised systems or systems with limited autonomy. The development becomes even more complex if the system is developed for operation in sophisticated environments which are highly dynamic, hardly predictable, includes larger number of elements and links connecting them, etc [1, 2, 3].

In the field of robotics one of the most important features, without which in most domains robot cannot be autonomous, is its ability to navigate through a certain territory without any assistance. A class of problems that is closely related with navigation is area mapping. The result of area mapping is a map of robot's environment that may be used for further purposes.

In the following chapters the most important mapping methods that are used for robot navigation are shortly described. The methods were analyzed in order to select the most appropriate for autonomous robotic system which is driven by knowledge-based intelligent and adaptive control system. We were looking for a method that would:

- Have been implemented previously by many other researchers.
- Require no initial knowledge of the environment.
- Have no restrictions on choice of mapping strategy or initial position.
- Have means for incomplete knowledge representation.
- Have means for moving object representation or temporarily occupied space indication on a map.
- Have a representation that would be easy for a non-specialist to understand.
- Be GIS compatible.
- Be platform independent.

These criterions were due to the robotic system and its multiple environments built using MS Robotic Studio. The robotic system is developed within the project num. ZP2006-04, funded by Riga Technical University.

## **2. Studied methods**

### ***2.1. Mapping using Landmarks***

Landmarks are objects that are derived from the environment. Examples of natural landmarks are rocks and boulders, trees, anthills and animal burrows. Examples of artificial landmarks are milestones, kilometer posts, signposts, junctions, monuments, furniture, corners, columns, pointing signs. A type of landmarks called beacons can transmit a signal like radio waves that is receivable for a robot. Landmark mapping simply means recording the positions of landmarks relative to one another. Localization and navigation using landmarks is based on sequential landmark identification. Depending on the task landmarks can mark out reference-points for localization or obstacles that have to be considered during the path planning. A small number of all landmarks can reference to the goal. Landmarks are not expected to differentiate one from another in a context of the same task, because landmarks are used in a joint context. Normally more than one landmark has to be in sight of the robot. Typically the robot does computations according to the mutual placement of landmarks sensed from a distance and never coming close to the observed landmarks. Localization is carried out by estimating the distance to all of the “visible” landmarks. A standard

localization method is to measure angles between landmark positions and to compute the pose from intersections of the circles that represent the possible positions of the robot [4].

***Advantages of mapping using landmarks:***

- Landmarks provide means for determining both robot's position and its orientation.
- Any objects can be used as landmarks, if those objects are alike and the robot is able to recognize them in the environment.
- Landmarks can be deployed in the environment in advance or an agent can place landmarks during the exploration in order to achieve better navigation in future.
- Beacons when used for localization can provide additional information for the robot.
- The minimum of landmarks required for self localization is one landmark when the initial position and odometry of the robot is known. It is proved by researchers at Vienna University of Technology [5].

***Disadvantages of mapping using landmarks:***

- Landmark objects must have the same appearance and that appearance must be known to the robot. This problem is particularly hard to overcome when using natural landmarks.
- Recognizing landmarks among other acquired data is a challenging problem because of noisy percepts caused by random objects, the appearance of landmarks varying from one observation to another and variations due to illumination change [6].
- Impermanence is characteristic to landmarks. This property makes landmarks unsafe for long term navigation.
- Better localization and path planning requires more landmarks. More landmarks require more memory space and higher computational effort and time accordingly.
- Experiments conducted in [4] show that varying landmark configurations influence significantly a quality of the generated control signals.
- Ill-considered algorithms for landmark and robot localization and noisy sensor data lead to significant errors.
- The robot's initial positioning is normally done manually and an additional technique like dead-reckoning is used to support landmark based navigation. Also, re-calibration is done manually when the mobile robot gets lost. This is not convenient for real-world applications. It is necessary to find a feasible way to initialize the position of a mobile robot automatically [7].

## ***2.2. Mapping using Waypoints***

Waypoints are relatively static environment objects that do not change their appearance and position for a long period- several years, decades and even centuries, are easy to be noticed and recognized in the environment. Traditional human used waypoints are mountains, waterways, oasis, buildings, highways and railways and other reference objects. Typical waypoints on waters are buoys, lighthouses, anchorages and harbors. Typical waypoints in the city are important buildings, skyscrapers, churches, supermarkets, factories, stadiums, bridges, tunnels and parks. There are waypoints for practical considerations made by industries, e.g. radio masts, beacons and satellites.

Nowadays the conception of waypoints has reached the abstraction level which minimizes the requests for visible environmental features. For example, artificial airways also known as highways-in-the-sky has no visible form at all. Those are series of waypoints used to define invisible navigation routes that a pilot navigates through. Abstract waypoints are maintained by radio masts on the ground and satellite based technology of Global Positioning System.

Generally, a single waypoint is a set of coordinates which indicates an important point in the environment. Waypoints are defined in a 2D space by longitude and latitude. Some air navigation systems incorporate the third coordinate of height. The fourth coordinate of time is added in case an outer space waypoint is to be specified.

Waypoints are named this way because waypoints are used to "point out the way" to the goal from the robot's initial position. When navigating through waypoints the robot has an initial instruction about the actions that has to be taken when a specific waypoint is reached. Actions may imply some modification of the robot's motion, like a direction, speed, acceleration or height when navigating an aircraft. Typically the modification leads to reaching the next waypoint.

***Advantages of mapping using waypoints:***

- Waypoints are constant even when placed in dynamic environments.
- Waypoints are typically big. This feature makes waypoints recognizable using cheaper sensor systems.
- There is no need for explicit waypoint coordinates at the very beginning of the task. In [8] waypoints are specified as goal regions whose dimensions are specified at runtime.
- Practical solutions incorporate waypoints indicated with names or identification numbers instead of coordinates. Objects can have properties e.g. colors, sizes, strategic importance, distance to the goal, that can be used to describe them or constrain subgroups of objects of an issue. Those properties can be passed to a robot associating them with the coordinates of the waypoints. Relations between waypoints can be set. Examples of relations include *near, far, right of, left of, in between* etc [9].
- Experiments have shown that relatively long distances can be traveled by a robot using waypoints for navigation [8]. As a part of an end-to-end system test, the mission planner generated a sequence of waypoints marking out a 6 km sun-synchronous circuit that the navigator was able to achieve over a period of 24 hours.

***Disadvantages of mapping using waypoints:***

- Errors in a GPS signal, like high frequency noise and long-term drift come in many forms and arise from a variety of different hard-to-tack-down sources [10].
- The most common solution for solving the problem of GPS noise is to augment the GPS with other sensors and employ Kalman filter to optimally combine all of those sensor inputs. The necessity for Kalman filtering benefits to the complexity of the task solution.

### ***2.3. Simultaneous Localization and Mapping- SLAM***

*SLAM* is a process by which a mobile robot when placed in an unknown location of an unknown environment can incrementally build a consistent map of the environment while simultaneously determining its location within this map [11]. Observations are made between robot and environment landmark locations. The true locations are never known or measured

directly. Much of the error between estimated and the true landmark locations is common between landmarks. In turn, this implies that the errors in landmark location estimates are highly correlated. Practically this means that the relative location between any two landmarks may be known with high accuracy, even when the absolute location of a single landmark is quite uncertain.

Solution to the probabilistic SLAM problem involve finding an appropriate representation for both the observation model and motion model that allows efficient, consistent and over time correlating computation of the prior and posterior distributions of landmark locations.

***Advantages of SLAM:***

- In *SLAM*, both the trajectory of the platform and the location of all landmarks are estimated online without the need for any a priori information of its location.
- The solution of the *SLAM* problem improves with the passage of time, because the combined mapping and localization problem, once formulated as a single estimation problem is convergent [11].

***Disadvantages of SLAM:***

- SLAM requires a deep understanding of probability theory, statistics, recursion and system dynamics. The most common representation for a solution is a state space with additive Gaussian noise, leading to the use of the extended Kalman filter to make the solution complete. Alternative representations of the robot's motion model leads to the use of the Rao-Blackwellized particle filter, or FastSLAM algorithm to solve the SLAM problem.
- SLAM is especially fragile to incorrect association of observations to landmarks.
- [12] showed that even when the robot is stationary and no process noise acts on it, the joint SLAM solution is guaranteed to be inconsistent. Furthermore, situation studies showed that a moving robot with process noise exhibits a similar type behavior. However, the time required for the map to become visibly inconsistent is of the order of several hundred time steps, which is longer than most experimental runs presented in literature.

## ***2.4. Configuration Space Mapping***

To understand the Configuration Space paradigm suppose that at robot arm (e.g. SCARA robot) has  $k$  degrees of freedom. Every state or configuration if the robot can be described with  $k$  real values:  $q_1, \dots, q_k$  The  $k$  values can be regarded as a point in a  $k$ -dimensional space called the configuration space  $C$  of the robot. Consider the robot arm moving in an environment where the workspace contains known obstacles. The goal of path planning is to find a path in the physical space from the initial position of the arm to the goal position, avoiding collisions with the obstacles. This is a difficult problem to visualize and solve in the physical space, particularly as  $k$  grows large. But in configuration space the problem is straightforward, if we define the *configuration space obstacle*  $O$  as the subspace of  $C$  where the robot arm bumps into something. We can now compute the free space  $F=C-O$  in which the robot can move safely [13].

***Advantages of Configuration Space Mapping***

- Configuration space representation allows a high degree of accuracy in the representation of environment, though the quality of data is sensor accuracy dependent.

- There are some path planning methods that employ configuration space and are extremely fast when used in sparse environments. Examples of those methods are Construction of Visibility Graphs and Construction of Voronoi diagrams [13].

#### *Disadvantages of Configuration Space Mapping*

- The solution to a navigation problem in a configuration space is always a trade between the safety and the optimality of the solution. Some path planning methods tends to find paths that take the robot as close to obstacles as possible. Such path planning is optimal in terms of the length of the solution path. But all sense of safety, in terms of staying at a reasonable distance from obstacles, is sacrificed for this optimality.
- The common approach of regarding the robot as a simple point in the configuration space leads to significant errors and threats to safety of the solution.
- Since the accuracy of data is sensor accuracy dependant, in some cases relevant data can be entered into a configuration space only when a robot is close to an obstacle, so that the obstacle is "clearly seen". This creates a safety drawback of configuration space mapping.
- The geometry of the obstacles in the real world has to be known explicitly in order to apply configuration space mapping.

## **2.5. Topological Mapping**

Formally, a topological representation is a graph that specifies two things: nodes and edges between those nodes. Insofar as a topological representation is intended for the use of a mobile robot, nodes are used to denote areas in the world and edges are used to denote adjacency of pairs of nodes. The nodes in the topological map graph specify waypoints in the real world. Typical waypoints used for topological mapping are corners, midpoints in space, passage crossings, dead ends, turning points and doorways. Edges that connect the nodes in the topological graph map indicate that environment has the traversability feature between the waypoints which correspond to nodes connected by an edge. An edge between two nodes in the graph means that the route between two waypoints in the real environment is obstacle free and thus traversable for the robot. An edge between nodes also means that the robot can traverse from one node to the other without traversal of any other intermediary node. The nodes in the topological map are not a direct representation of the area for the mapped region, and the edges do not reveal the distance between the regions. Topological representation contains only structural information about the robot's environment. The chief motivation of the topological approach is that the environment may contain important nongeometric features- features that have no geometric relevance but are useful for localization [13].

Topological maps are suitable for navigation tasks which involve traveling long distances and do not require accurate control of the robot position along the path. Typically, it corresponds to commands such as "follow the corridor" or "turn left". To adopt this navigation strategy the traditional metric map is extremely inadequate [14]. Therefore topological maps are the most common maps used for navigation in offices.

#### *Advantages of Topological Mapping*

- Topological approaches avoid direct measurement of geometric qualities, instead concentrating on environmental characteristics that are most relevant to the robot [13].
- Topological map offers a lot of alternative routes to a path planner.

- Two or more topological maps are easy to merge together producing a more complete map.
- Topological mapping requires a small amount of memory space comparison to more expressive maps like grid-based maps.

#### *Disadvantages of Topological Mapping*

- Topological map is over specified in the perspective of the path-planner and involves difficulties when searching for the shortest route.
- Topological maps lack expressiveness because they not contain accurate metric information and relevant details such as small obstacles. The more the topological representation diverges from the geometrical representation, the more expressiveness for describing a robot position and rout replanning is lost.
- Topological mapping is appropriate only for a very narrow range of real world applications which requires no geometry. Topological navigation fails in most outdoor environments.
- Topological navigation is difficult to implement in dynamic environments, because edges of the topological map graph are intended to indicate obstacle free routes The topological map may contain faulty edges that indicate an obstacle free environment when there are closed doors in the pathways.

## **2.6. Mapping using Occupancy Grid**

The occupancy grid mapping is the most popular mapping method used in praxis. The main idea of the occupancy grid mapping is a grid which overlays the entire robot's environment at the height of interest. Each discrete cell has a state associated to it. The state records the properties of the corresponding space in the environment. The appropriate set of possible states is chosen according to the task, the capabilities of robot's sensor system and specifics of the environment. It is up to the researcher how many different states there are and how they are assigned to particular cells in the grid.

When the robot has just started mapping, each cell is a set to the initial state "unknown". Whenever a free space or an obstacle is encountered, this information is entered in the grid data structure, so cells can either be "free" or contain "obstacle". In addition the states "preliminary free" and "preliminary obstacle" are used to deal with sensor error and noise. These temporal states can later be either confirmed or changed when the robot passes within close proximity through that cell. The algorithm proposed by [15] stops when the map generation is completed that is, when no more preliminary states are in the grid data structure. Other cell states can be introduced if necessary.

Cells in an occupancy grid may have values indicating the robot's beliefs about the occupancy of a particular cell. The cell value may be assigned to zero, if the robot has not yet had any perceptions of the cell. It may be changed to one when the robot percepts an obstacle in the cell. Alternatively the robot may obtain a noisy perception and assign the cell value somewhere between zero and one when it is not 'sure' about the presence of an obstacle.

Graphical representation of this mapping method is a grid with miscellaneous shaded or miscellaneous colored cells. An adopted representation is darker shading for cells more likely to be obstacles (black for cells with known obstacles) and lighter for cells more likely to be free (white for known free space).

#### ***Advantages of Mapping using Occupancy Grid***

- An occupancy grid can be used for effective path planning. [13] shows that cells in an occupancy grid can be labeled with the total distance to the goal. A robot completes the distance moving from the cells with a larger label value to the cells with smaller label value. In [8] an obstacle detection algorithm generated a traversability grid map. Each cell on the grid had three associated metrics: the slope, roughness and "step height". Those metrics were used to determine the overall traversability of different paths.
- A grid is a discrete representation. The robot can work on a selected region (not necessary a rectangular territory) in the grid in order to make computation process faster.
- The space occupancy grid representation does not restrict the robot on the strategy it uses for the exploration of the environment. In a perfect case the path planner will be able to extract data supporting the subsequent exploration from the grid built while the preceding exploration.
- The termination criterion for the mapping algorithm is defined explicitly as each cell of the grid having some state assigned to it.
- Vector field histograms can be used on occupancy grids for obstacle avoidance [16].
- [17] has shown how to use grids for spatial representation allowing fusing information from different types of sensors. This type of grids is called the evidence grid and has spurred from the occupancy grid approach.
- Thanks to the discrete data representation, it is very easy to enhance this method in order to use as complimentary part of *Geographic Information System (GIS)* which allows to reason about the map using spatial querying technology.

#### ***Disadvantages of Mapping using Occupancy Grid***

- The precision of robot's sensors has influence on the resolution of the obtained grid. Poor precision of the sensors leads to a poor quality of the map.
- The method is appropriate for local environment mapping only. The maximal range of the robot's sensor system must be considered. Only those environmental objects that are in the range of active sensors can be mapped. Objects mapped into the grid must have known coordinates in space that are inside the boundary of the grid. There is no technique obtaining occupancy data from video camera. A robot cannot add to the cell information that church towers, constellations, horizon or other large object can be seen from it.
- The robot have to spend a lot of time exploring its environment in order to build a truly global map, because it has to visit the space it has not visited before. When the global map is complete, the values of the cells visited early may already be obsolete.
- The size of the map in the robot's memory grows as the explored environment grows. It is often hard to do an *a priori* estimation about the space required by the occupancy grid. It is particularly pressing problem, when a grid map with a high resolution must be achieved.
- There is no standard representation for cell states in the space occupancy grid. The lack of a rises the problem of inconsistency among grid representations among different robot's made by different researchers. It is common that a robot cannot use an occupancy grid that is mapped by another robot.

## ***2.7. Line Segment based Mapping***

Line segment based mapping is based on distance measurement data retrieval and integration. The robot starts mapping with obtaining a set of environment scans. Line segments approximate the points returned by laser range scanner. Two sequential scans are selected from the obtained set and a scan matching is performed. According to [18] scan matching is a process of finding the rotation angle and translation that superimposes at least one angle in a scan to an equal angle in another scan. The rotation angle and translation describes the dissimilarity of two scans, or the translation that must be applied to one of the scans so that it would conform to the other scan. The mean translation of all scans affects the time that is required to build a complete map. Greater mean translation will require a smaller number of integrations, but it also will require more time to carry out the scan matching. When the robot has finished scan matching it proceeds with partial map integration which is carried out similar to the scan matching process. A partial map according to [18] is the result of the integration of two scans, of a scan and a partial map, or two partial maps. Thus, a scan is a special case of a partial map. Both scans and partial maps are collection of line segments. The difference is that the line segments in a scan are ordered (clockwise or counter-clockwise) while they are not in a partial map.

### ***Advantages of Line Segment based Mapping***

- No odometric knowledge about robot's position is necessary to build a global geometric map.
- Line segment based maps are more precise than those which incorporate odometric data in map building process. Odometric data are not as reliable because odometry suffers from errors of the sensors, the control mechanism and the computation. Line segment based mapping is not affected with the control mechanism generated errors.
- Line segments are easy to be extracted from range data automatically [18].
- Line segments can be represented with few numbers and produce maps that are easy to use and require a small storage space [18].
- Increased flexibility in data collection, since range data can be collected by a single or multiple robots in one or multiple sessions [18] with any time interval between sessions.

### ***Disadvantages of Line Segment based Mapping***

- The method restricts robot by the way it explores the environment. The scans must be ordered clockwise or counter-clockwise in order to be suitable for scan-matching.
- Scan acquisition always involves with sensor error. Scan matching always involves transformation error. Scan integration always involves integration error. Because of scanning, calculations and matching errors, the scans might not align exactly. To produce the output map matching chains of the line-segments in transformed scans are replaced by a polyline that approximates the matching chains [18] and increase the error. This leads to spurious lines, which are a typical error in line segment based maps.
- Only two scans or partial maps can be matched or integrated at a time. There is no technique for simultaneous integration of more than two scans or partial maps at a time.
- Heuristics are required to speed up the scan matching process, because finding the possible translations is a difficult combinatorial problem. Heuristics are not always easy to determine and to implement and they must be provided by human.

- In case when a map is built by multiple robots, integration has to be accomplished on two levels. First each individual robot must accomplish scan matching. Second partial maps acquired by individual robots must be mutually integrated. Partial map integration requires more time than scan matching does. This drawback makes line segmented mapping incongruous with real time mapping.

## ***2.8. Symbolically annotated 3D Mapping- SATD***

Symbolically annotated 3D map (SATD map) is a special case of a map used particularly by a RHINO plan-based controller [19]. An SATD map consists of a set of object models and a set of structural elements. An object model has a name and consists of a specification of its category (desk, shelf, table,...), a geometric model, the position of the center of the object and its orientation, and a set of attribute-value pairs of the form {<category, desk>,<color, red>, ...}. Object models can be retrieved through relational symbolic queries such as “the desk in room A-120”. To provide the necessary information SATD maps store models of the task-relevant objects such as desks and shelves. The task-relevant objects themselves are represented as hierarchically structured 3D models, which enables RHINO to identify the part of the robot's camera image that corresponds to the taskable top of a desk. SATD enables RHINO to perform tasks such as “get a book form the shelf in room A-120 and bring it to Wolfram’s desk.”

Structural elements, such as smoking areas or rooms, represent regions of the environment. With each region is associated a set of doorways through which the region can be entered and left. Aspects of the environment structure are represented explicitly. This enables RHINO to adapt its behavior to the different part in the environment (different driving strategies in offices and hallways, etc.) and detect problems such as closed doors.

### ***Advantages of Symbolically annotated 3D Mapping***

- SATD representation enables the RHINO robot recognize closed doors and other temporal environmental obstacles.
- SATD maps can be used to generate occupancy grid maps for robot self-localization. These grid maps are generated by making a projection of the 3D map at a given height (e.g., the height of the robot’s sonar sensor). These maps are used to estimate the robot’s position with an average accuracy of 5-10 cm. RHINO uses SATD maps for interpreting location descriptions of objects to be searched. Queries may be applied to SATD maps much like queries are applied to data bases.

### ***Disadvantages of Symbolically annotated 3D Mapping***






- SATD is limited in use, because RHINO are the only controller that has the means for SATD mapping.
- As a consequence there is a little research done and not enough literature available on this mapping method.

## **3. Conclusions**

It was necessary to select one– the most appropriate method- for a robotic system developed for the scientific project funded by Riga Technical University. The method was

selected according to the criteria described explicitly in the introduction. According to the findings and disadvantages the project group selected methods called "Mapping using occupancy grid". The most significant advantage of this particular method is that it may be a complementary part of GIS that is used to interact with the robotic system's user. In addition to it, the interaction between user and the system is defined in terms of goals and results. Occupancy grid allows defining system's goals using the set theory and appropriate simple mathematical expressions while other methods are hard to use for explicit goal definition.

For instance, let us assume the following cells to be defined in terms of sets:

	Cell state "Unknown"	set $N = \{n_i\}$
	Cell state "Preliminary free"	set $SB = \{sb_i\}$
	Cell state "Free"	set $B = \{b_i\}$
	Cell state "Preliminary obstacle"	set $SK = \{sk_i\}$
	Cell state "Obstacle"	set $S = \{s_i\}$

Using the described sets, it is possible to define goals and results via appropriate set expressions and operations:

#### *Mapping*

Available data :  $N = \{n_i\}$   
 Goal:  $X = \{x\}, \forall x \in (B \vee S \vee SB \vee SK)$  OR  
 $X = \{x\}, \forall x \notin N$

The goal is a set X, where every  $x_i$  belongs to the set of known cells or does not belong to unknown cells.

At the beginning user can mark the necessary territory thus defining the dimensions of the N and X. The system starts to map the territory and fills out the set X while the goal conditions are met.

#### *Area monitoring*

Available data :  $N = \{n_i\}$  - if the map is not available  
 $X = \{x\}, \forall x \in (B \vee S \vee SB \vee SK)$  Or  
 $X = \{x\}, \forall x \notin N$  - if the map is known  
 Goal:  $X = X_i \cap X_{i-1}$

The goal is to determine differences between observation  $X_i$  and  $X_{i-1}$ . If the acquired set X meets some certain criterion then appropriate action may be executed. This approach allows moving object detection in the robot's environment.

While the defined sets of cells may be complemented by additional cell types, it is possible to use the given fundamental goal definitions for more complex tasks. For example, the area monitoring goal definition may be used for area security purposes while the mapping goal for territory marking (chemical pollution, etc...). Thereby the selected method is very easy to use and enhance to meet requirements of a certain application.

## References

1. S.Russell, P.Norvig, „*Artificial Intelligence – A Modern Approach. 2nd edition*”// Upper Saddle River, New Jersey, USA, Pearson Educations Inc., 2003, 1080 p., ISBN 0-13-080392-2.
2. A.D.Cvirkun, „*Structure of complex system*” // Moscow, Soviet Radio, 1975, 198 p. (A. Д. Цвиркун, Структура сложных систем. Москва, Советское радио, 1975, 198с.)
3. V.V.Druzinin, D.S.Kontorov, „*Systemtechnique*” // Moscow, Radio and communications, 1985, 200 p.(В.В. Дружинин, Д.С. Конторов, Системотехника. Москва, Радио и Связь, 1985, 200 с.)
4. Burschaka D. et. al. „*Optimal Landmark Configuration for a Vision-Based Control of Mobile Robots*” // in proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei, Taiwan., 2003 ([http://www.cs.jhu.edu/CIRL/publications/pdf/icra2003\\_dg.pdf](http://www.cs.jhu.edu/CIRL/publications/pdf/icra2003_dg.pdf)).
5. Bais, A.; Sablatnig, R.; Gu, J. „*Single Landmark based self-localization of mobile robots*” // in proceeding of the 3rd Canadian Conference on Computer and Robot Vision, 2006, 67 p. (<http://dx.doi.org/10.1109/CRV.2006.67>)
6. Takeuchi Y. et.al. „*Visual Learning for Landmark Recognition*” // in proceeding of IUW 1997, (<http://www.cs.cmu.edu/~takeuchi/iuw97/iuw97.html>).
7. Hu H., Gu D. „*Landmark- based Navigation of Industrial Moblie Robot*” // International Journal of Industry Robot, Vol. 27, No. 6, 2000, 458p – 467p (<http://cswww.essex.ac.uk/staff/hhu/Papers/Journal-IR27-6.pdf>)
8. Urmson C., Dias M.B. „*Stereo Vision Based Navigation for Sun-Synchronous Exploration*” // web article - <http://www.frc.ri.cmu.edu/sunsync/publications/02icra.stereo.draft.pdf>, cited on 2007.09.23.
9. Brenner M. et.al. “*Mediating Between Qualitative and Qauntitative Representations for Task-Oriented Human-Robot Interaction*” // Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, 2072p -2077p. ([http://www.cs.bham.ac.uk/~nah/bibtex\\_/papers/brenneretal07ijcai.pdf](http://www.cs.bham.ac.uk/~nah/bibtex_/papers/brenneretal07ijcai.pdf))
10. Bruch M.H. et.al. “*Accurate Waypoint Navigation Using Non- differential GPS*” // AUVSI Unmanned Systems 2002, Lake Buena Vista, FL, 2002 ( [http://www.spawar.navy.mil/robots/pubs/auvsi02\\_mprs.pdf](http://www.spawar.navy.mil/robots/pubs/auvsi02_mprs.pdf)).
11. Durrant-Whyte H., Bailey T. “*Simultaneous Localization and Mapping: Part I*” // web article - <http://automation.tkk.fi/edu/as84145/SLAM%20Tutorial%20Part%20I.pdf>, cited on 2007.09.24.
12. Julier J.S., Uhlmann K.J. “*A Counter Example to the Theory of Simultaneous Localization and Map Building*” // web article - [http://www.ait.nrl.navy.mil/3dvmel/papers/cp\\_ICRA01.pdf](http://www.ait.nrl.navy.mil/3dvmel/papers/cp_ICRA01.pdf) cited on 2007.08.30.
13. Siegwart R., Nourbakhsh R.I. “*Introduction to Autonomous Mobile Robots*” // 2006, London: The MIT Press.- 2004.- 322 pp.
14. Winters N. et al. “*Visual Attention-based Robot Navigation using Information Sampling*” // in proceedings of IEEE/RSJ Intl. Conference on Intelligent Robots and Systems IROS2001, Maui, Hawaii, 2001 (<http://vislab.isr.ist.utl.pt/publications/01-iros-info.pdf>)
15. Bräunl T. „*Embedded Robotics*” // Leipzig, Germany, Springer, 2006.– 458 p.
16. Borenstein J., Koren Y. „*The vector field histogram - fast obstacle avoidance for Mobile Robots*” // IEEE Journal of Robotics and Automation. - 1991. Issue 7- 278-288 pp.
17. Yamauchi B. et. al. “*Mobile Robot Exploration and Map-Building with Continous Localization*” // Iweb article - <http://citeseer.ist.psu.edu/cache/papers/cs/1491/http:SzzSzwwww.aic.nrl.navy.milzSz~yamauchizSzpaperszSzcompressedzSzcra98.pdf/yamauchi98mobile.pdf>, cited on 2007.09.24.
18. Amigoni F. et.al. “*Building Segment-Based Maps Without Pose Information*” // web article - <http://www.airlab.elet.polimi.it/index.php/airlab/content/download/893/7297/file/pij12.pdf>, cited on 2007.09.26.
19. Beetz M. “*Plan-Based Control of Robotic Agents*”//Berlin, Germany, Springerlink, 2002., 196 p.

**Sudraba L., Nikitenko A. Kartēšanas metožu pielietojums autonomas inteligēntas sistēmas navigācijas uzdevumu risināšanai**

Šis raksts dod ieguldījumu autonomu inteligēntu sistēmu problēmsfērā, kura pēdējos gados piedzīvo ļoti strauju attīstību. Sistēmas autonomija ir ļoti cieši saistīta ar tās spēju pieņemt lēmumus un darboties bez ārējas palīdzības. Parasti autonomiem robotiem ir nepieciešama spēja kustēties, lai sasniegtu uzstādītos mērķus. Mērķi var būt saistīti ar kādu vietu atrašanu telpā un to sasniegšanu pārvietojoties lielos attālumos. Līdz ar to robotizētai sistēmai jāizpilda noteikti uzdevumi, kuri ir cieši saistīti ar navigāciju. Piemēram, apkārtnes kartēšana, pašlokalizācija, ceļa plānošana un telpas pāriešana ir navigācijas uzdevumi. Raksts ir veltīts metodēm, kuras aplūko šos uzdevumus no autonomas kartes sastādīšanas perspektīvas. Autori ir izpētījuši astoņas kartes sastādīšanas metodes, kuras ir visbiežāk pielietotās autonomas navigācijas uzdevumu risinājumos, lai izvēlētos vienu vispiemērotāko autonomam robotam. Šīs metodes iekļauj robežzīmju izmantošanu, virzienpunktu izmantošanu un citas. Raksts dod detalizētāku noteikto kartēšanas mērķu specifikāciju, īsi apraksta pētītās kartēšanas metodes un apkopo specifiskas metodes piemērotību specifisku mērķu sasniegšanai.

**Sudraba L., Nikitenko A. Application of mapping methods for solving navigation tasks of autonomous intelligent system**

This paper contributes to the domain of autonomous intelligent systems that experiences a very rapid advancement in the recent years. System's autonomy is very closely related with its ability to make decisions and act without any outer assistance. Usually autonomous robots require an ability to move in order to achieve the given goals. Those goals may be related to estimating certain locations in space and reaching those by traveling long distances. Therefore the robotic system has to perform certain tasks that are closely related with navigation. For instance, area mapping, self localization, path planning and space traversal are navigation tasks. The paper is devoted to the methods that address those tasks from the autonomous map building perspective. The authors have studied eight map building method, that are most commonly used in autonomous navigation task solutions, in order to select the most appropriate one for an autonomous robot. These methods include using landmarks, using waypoint and other. The paper gives a more detailed specification of the prescribed mapping goals, shortly describes the studied mapping methods and summarizes the adequacy of a specific method for a specific goal achievement.

**Судраба Л., Никитенко А. Применение методов картирования для решения задач навигации автономных систем.**

Эта работа даёт вклад в проблематику автономных интелегентных систем, которая переживает очень быстрое развитие в последние года. Автономия системы тесно связана с её способностью принимать решения и действовать без внешней помощи. Обычно автономные роботы требуют способность двигаться чтобы достичь установленные цели. Эти цели могут быть в связи с установлением определённых местонахождений в пространстве и достижения их путешествуя длинные дистанции. Связи с этим роботизированная система должна выполнять определённые задачи которые тесно связаны с навигацией. На пример, картирование площади, само-локализация, планирование маршрута и пересечение пространства являются задачами навигации. Эта работа посвящена методам, которые рассматривают эти задачи с точки зрения построения карты. Авторы изучили восемь методов построения карты, наиболее использованных для решения навигационных задач, чтобы выбрать самую пригодную для автономного робота. Эти методы включают использование межевых знаков, использование указателей поворота и другие. Публикация даёт более подробную спецификацию выдвинутых целей картирования, вкратце описывает изучение методы картирования и обобщает пригодность специфических методов для достижения специфических целей.