ISSN 0361-7688, Programming and Computer Software, 2019, Vol. 45, No. 8, pp. 557–569. © Pleiades Publishing, Ltd., 2019.

Software Advances using n-agents Wireless Communication Integration for Optimization of Surrounding Recognition and Robotic Group Dead Reckoning

M. Ivanov[a],*, O. Sergiyenko[a],, V. Tyrsa[b],***, L. Lindner[a],****, J. C. Rodriguez-Quiñonez[b],*****, W. Flores-Fuentes[b],******, M. Rivas-Lopez[a], D. Hernández-Balbuena[b],*******, and J. I. Nieto Hipólito[c],**********

[a]*Universidad Autonoma de Baja California (UABC), Instituto de Ingeniería,*
Mexicali, B.C., Mexico

[b]*Universidad Autonoma de Baja California (UABC), Facultad de Ingeniería,*
Mexicali, B.C., Mexico

[c]*Universidad Autonoma de Baja California (UABC), Facultad de Ingenieria, Arquitectura y Diseño,*
Ensenada, B.C., Mexico

**e-mail: ivanovm@uabc.edu.mx*
***e-mail: srgnk@uabc.edu.mx*
****e-mail: vtyrsa@uabc.edu.mx*
*****e-mail: lindner.lars@uabc.edu.mx*
******e-mail: julio.rodriguez81@uabc.edu.mx*
*******e-mail: flores.wendy@uabc.edu.mx*
********e-mail: dhernan@uabc.edu.mx*
*********e-mail: jnieto@uabc.edu.mx*

Received August 13, 2019; revised October 16, 2019; accepted October 16, 2019

Abstract—Nowadays artificial intelligence and swarm robotics become wide spread and take their approach in civil tasks. The main purpose of the article is to show the influence of common knowledge about surroundings sharing in the robotic group navigation problem by implementing the data transferring within the group. Methodology provided in article reviews a set of tasks implementation of which improves the results of robotic group navigation. The main questions for the research are the problems of robotics vision, path planning, data storing and data exchange. Article describes the structure of real-time laser technical vision system as the main environment-sensing tool for robots. The vision system uses dynamic triangulation principle. Article provides examples of obtained data, distance-based methods for resolution and speed control. According to the data obtained by provided vision system were decided to use matrix-based approach for robots path planning, it inflows the tasks of surroundings discretization, and trajectory approximation. Two network structure types for data transferring are compared. Authors are proposing a methodology for dynamic network forming based on leader changing system. For the confirmation of theory were developed an application of robotic group modeling. Obtained results show that common knowledge sharing between robots in-group can significantly decrease individual trajectories length.

DOI: 10.1134/S0361768819080139

1. INTRODUCTION

The successful use of mobile robots in deferent fields of civil industries brings researchers to the task of robotic group effective control. As an examples can be mentioned object detection in different types of environment, use a group as distributed sensor system for chemical analysis, etc.

Generalizing all the tasks, we can say that robotic group collaboration in a densely cluttered terrain is one of the core problems in swarm robotics. In such

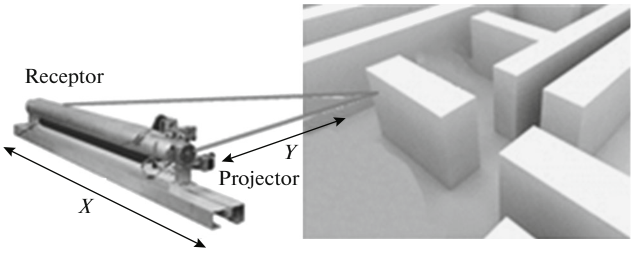


Fig. 1. Technical Vision System.

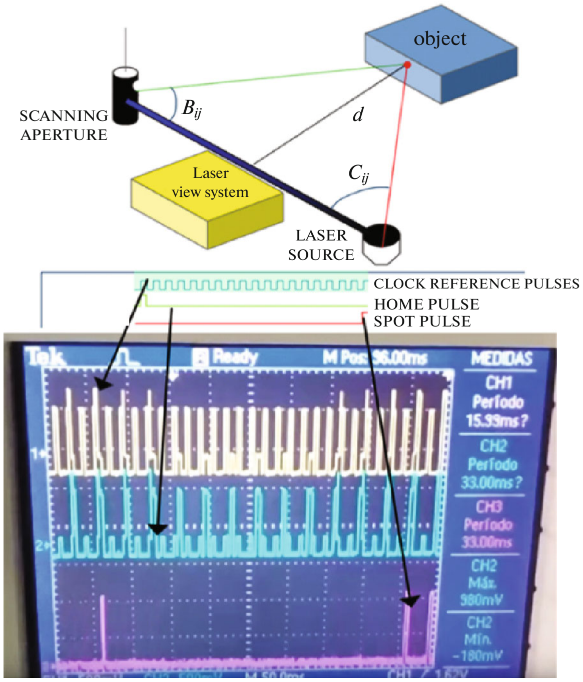


Fig. 2. Dynamic triangulation Principle of laser scanning TVS.

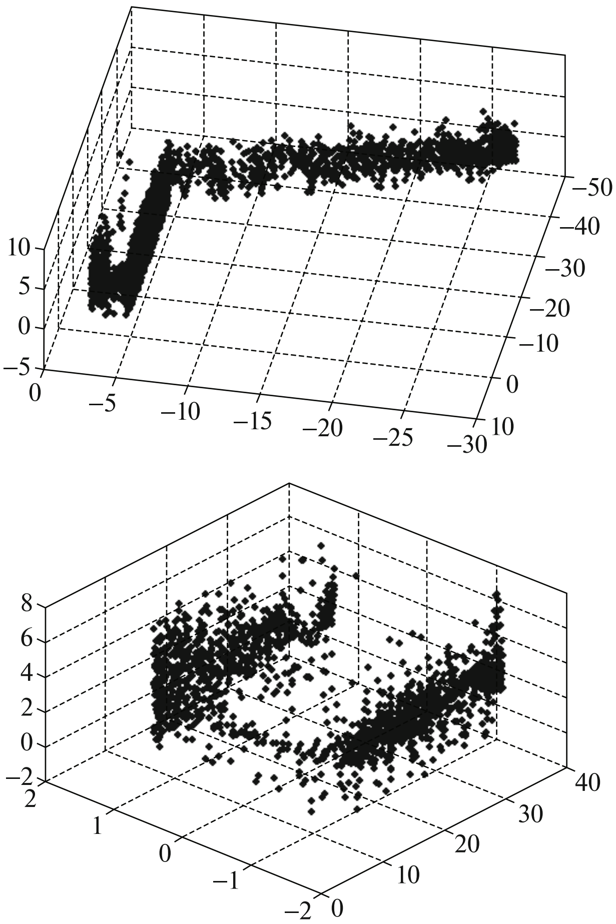


Fig. 3. Examples of scanned obstacles.

type of surroundings group is usually distributed in the area. Such distribution gives them an ability to perform tasks of real time dynamic events detection in surroundings, searching for objects of interest, etc. in a better way than, for example, a sensor network. This is achievable because each individual robot can cover larger part of the area by patrolling rather than staying in static. This means that the group can perform monitoring of an environment with fewer amount of sensors. For achieving automatization inside the group, while performing such tasks, robots must operate with the predefined rules of collective behavior.

Each individual robot within the group is a device with a specific set of features (sensing, communication, movement and computational processing) and factors limiting them. We consider using a homogeneous group of robots ([1], [2]) that are able to perform different simple tasks independently. In addition, robots must be able to move to their goal in the environment without losing contact with the group members along the way. That is why were allocated three types of problems that need to be solved simultaneously in real-time:

1. Obstacle detection and localization. The success of a mobile group of robots in an unstructured environment is only possible if the robots has ability to adjust to variable factors in the environment. These changes, which include the presence of new obstacles, can be detected using sensory systems like laser technical vision systems, cameras etc. Obstacle detection serves as an initial state of motion planning to predict possible routes.

2. Group communication process ([3], [4]). The need for data transfer can occur in the robotic teamwork in the case where it is necessary to make a collective decision, the lack of its own data to make an individual decision or as well as to inform other robots of the occurrence of abrupt changes in the environment or conditions.

3. Motion planning or collective navigation ([5], [6], [7], [8]) for robotic group in cluttered terrain implements the task of developing the potential trajectories through an unknown obstacle field from a given initial position to a desired goal position.

So the main contribution of the present paper to review the possible solutions of this problems and by implementing them into the model understand the influence of data exchange on the robotic group interaction with environment during the task of mapping [9] and goal localization.

This paper organized as follows: Section 2 describes obstacle detection and digitalization. Section 3 reveals the principles of dynamic data transferring network

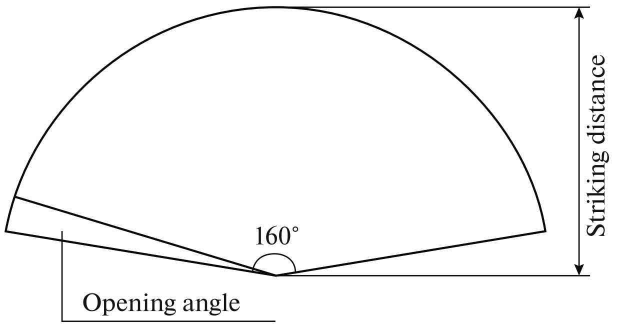


Fig. 4. Field of view scanning arc.

forming. Section 4 covers the aspects and methods of motion planning optimization. Section 5 describes basic modeling results considering group of three robots.

2. TECHNICAL VISION SYSTEM

The most common vision systems are using approaches based on CCD or CMOS ([10], [11], [12], [13]), more expensive equipment such as Time-of-Flight [14] cameras or some laser based [15]. Most of them are good for object recognition, path planning, etc. However, in case of more complicated conditions, like low light or a sharp landscape change and so on, vision system should satisfy these circumstances. Reviewing approach is based on laser technical vision system (TVS) [16] (Fig. 1). This system in comparison to others has wide opening field of view angle (up to 160°) and perfect matches for operation in completely dark environment, which is completely impossible for camera systems.

It uses method called dynamic triangulation [17]. System calculates Cartesian coordinates according to two detected angles B_{ij} and C_{ij} (in this particular case ij are horizontal and vertical scanning steps) of laser instantly highlighted point and fixed distance a between a projector and a receptor. In obtained triangle (Fig. 2) angle B_{ij} is calculated as simple ratio of two counters codes: number of clock pulses between two home pulses and in interval 'home pulse – spot pulse' (1).

$$B_{ij} = \frac{2\pi N_A}{N_{2\pi}}, \qquad (1)$$

where N_A is the number of reference pulses when laser rays are detected by the stop sensor and $N_{2\pi}$ is the number of clock reference pulses when the 45° mirror completes a 360° turn detected by the zero sensor.

To calculate x, y and z coordinates the next equations are used (2-5):

Table 1. Comparison of scanning angles

				Resolution (points/m)				
				9	10	11	12	13
		Striking distance	Length of arc	Opening angle (deg)				
Striking distance range	Critical	0.358	1.000	17.783	16.004	14.549	13.337	12.311
		0.500	1.396	12.732	11.459	10.417	9.549	8.815
		1.000	2.793	6.366	5.730	5.209	4.775	4.407
	Optimal	1.500	4.189	4.244	3.820	3.472	3.183	2.938
		2.000	5.585	3.183	2.865	2.604	2.387	2.204
		2.500	6.981	2.546	2.292	2.083	1.910	1.763
		3.000	8.378	2.122	1.910	1.736	1.592	1.469
	Effective	3.500	9.774	1.819	1.637	1.488	1.364	1.259
		4.000	11.170	1.592	1.432	1.302	1.194	1.102
		4.500	12.566	1.415	1.273	1.157	1.061	0.979
		5.000	13.963	1.273	1.146	1.042	0.955	0.881
		Average opening angle for width	Low	12.294	11.064	10.059	9.220	8.511
			Medium	3.024	2.722	2.474	2.268	2.094
			High	1.525	1.372	1.247	1.206	1.181

Table 2. Rules for resolution and speed control

Striking distance (SD) (m)	Radius type (linguistic variable)	Opening angles (deg.)	Resolution (linguistic variable)
SD ≤ 1	Critical	5.209	Low
1 < SD ≤ 3	Optimal	2.474	Medium
3 < SD ≤ 5	Effective	1.247	High

$$x_{ij} = a\frac{\sin B_{ij} \cdot \sin C_{ij} \cdot \cos \sum_{j=1}^{j} \beta_j}{\sin[180° - (B_{ij} + C_{ij})]}, \tag{2}$$

$$y_{ij} = a\left(\frac{1}{2} - \frac{\sin B_{ij} \cdot \sin C_{ij}}{\sin[180° - (B_{ij} + C_{ij})]}\right) \quad \text{at} \quad B_{ij} \le 90°, \tag{3}$$

$$y_{ij} = -a\left(\frac{1}{2} + \frac{\sin B_{ij} \cdot \sin C_{ij}}{\sin[180° - (B_{ij} + C_{ij})]}\right) \quad \text{at} \quad B_{ij} \ge 90°, \tag{4}$$

$$z_{ij} = a\frac{\sin B_{ij} \cdot \sin C_{ij} \cdot \cos \sum_{j=1}^{j} \beta_j}{\sin[180° - (B_{ij} + C_{ij})]}, \tag{5}$$

The detected obstacle or moreover surroundings for the robot is presented as a mesh consisted of points (Cartesian coordinates) on the surface (Fig. 3).

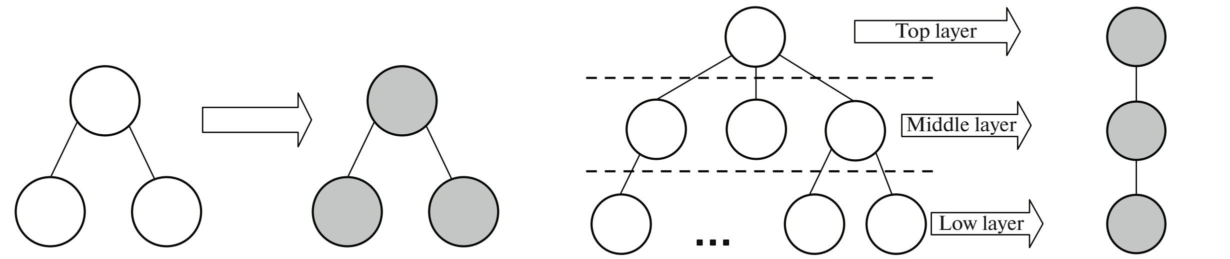
TVS is able to give a highly detailed information about the detected obstacle. In the task of recognition or classification it is considered as a benefit. During the motion planning for robot to avoid collision the information about an obstacle must contain general data (width, depth, height and large convexities of its surface which can obstruct the movement). In other words, the obstacle should be represented for robot as a simple 3D figure with low resolution. This is done by calculating the equivalents of opening angles that are needed on the different striking distances in order to maintain the same resolution of the received TVS image.

The [18] recommends three different angles 14.5636°, 5.5455°, 1.9091°. They represent three types of resolution presented as linguistic variables: "Low", "Medium" and "High". Field of view of the TVS can be presented as an arc (Fig. 4). Difference between horizontal steps we will call "Opening angle" and the radius of arc is "Striking distance" (distance to the detected object).

According to these assumptions will be calculated opening angles needed to reach the specific resolution (amount of point on arc) for different striking distances (Table 1).

The range of striking distances is separated into three parts: "Effective" where there is no need for robot to drop down its speed to avoid collision; "Optimal" the reaction depends on the maneuver needed to be done; "Critical" robot needs to slow down or even stop.

As can be seen, average angle for the "Critical" range in the end of the distance will give a low resolution that is not enough for critical obstacle detection. In this case, it is necessary to increase the resolution. That is why the limit value of angle for "critical" range is taken. The set of angles becomes next: 5.209°, 2.474°, 1.247°.



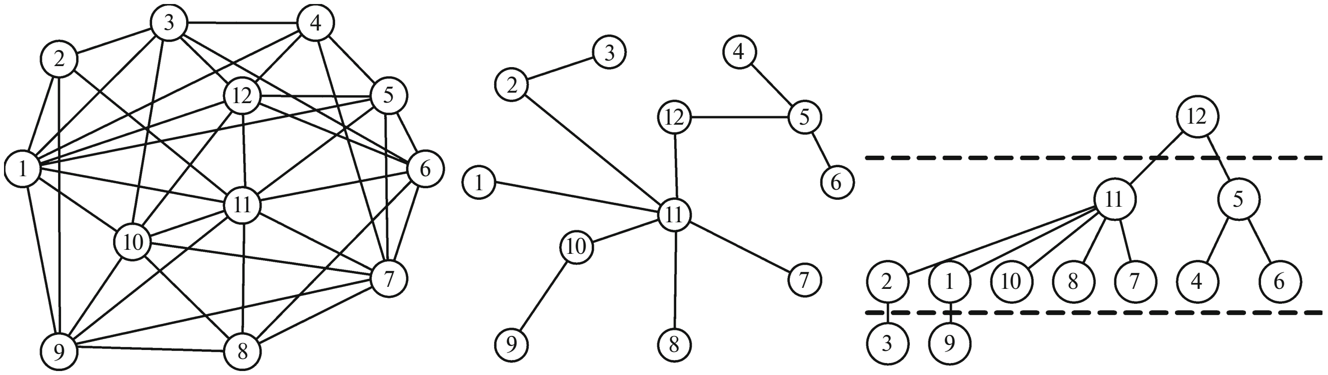
(a) General model (b) Current case

(a) Information exchange with centralized management

(a) General model (b) Current case

(b) Information exchange using strategies of centralized hierarchical control

Fig. 5. Models of data transferring.

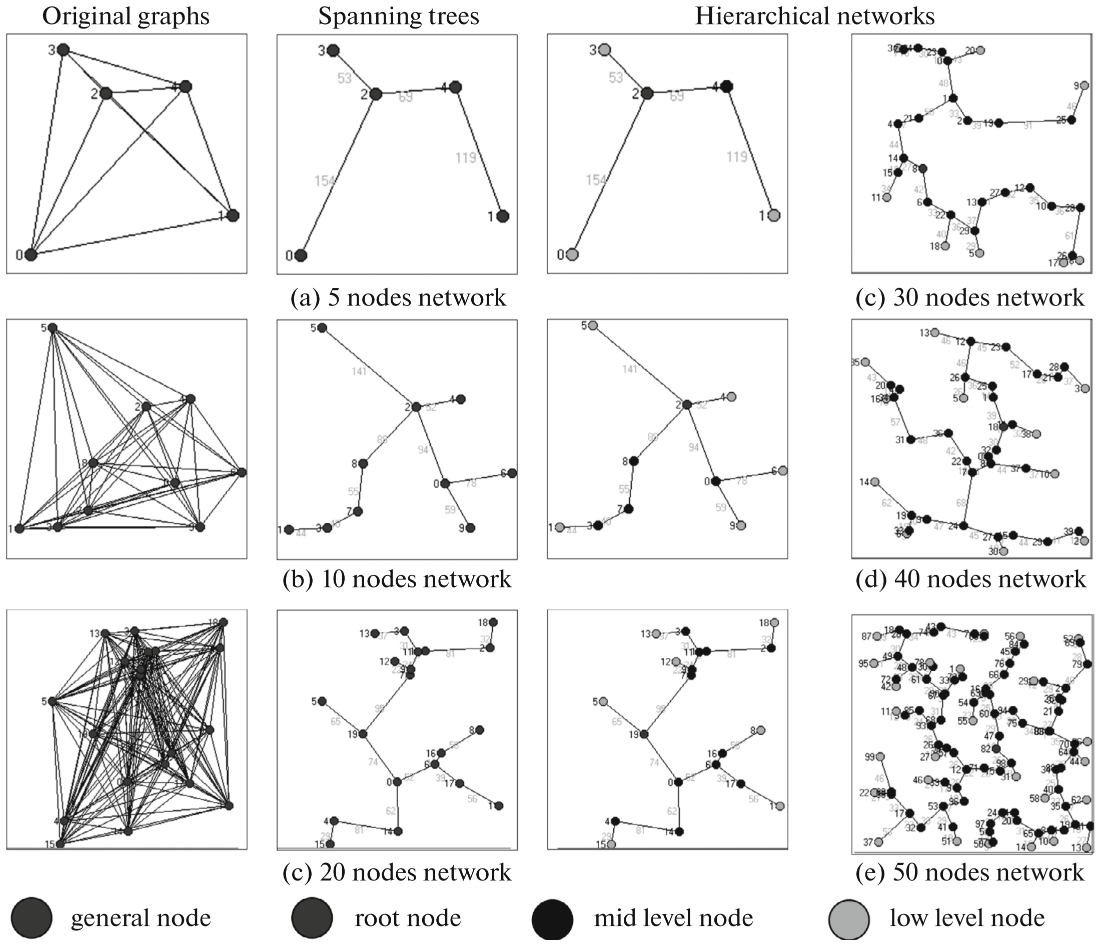


(a) Fully connected network graph (b) Spanning tree (c) Hierarchical network

Fig. 6. Hierarchical network forming.

Using the obtained data can be performed a set of rules for robots speed control and resolution stabilization during the movement (Table 2).

Implementing variable frequency of data storing (Change of opening angles) will help to leave conventionally equal level of obstacles detalization in the memory of single robot. Moreover, this discretization process allows performing data exchange within the group to update the common knowledge base more often than with raw data from TVS.



(a) 5 nodes network (c) 30 nodes network

(b) 10 nodes network (d) 40 nodes network

(c) 20 nodes network (e) 50 nodes network

Fig. 7. Calculated networks.

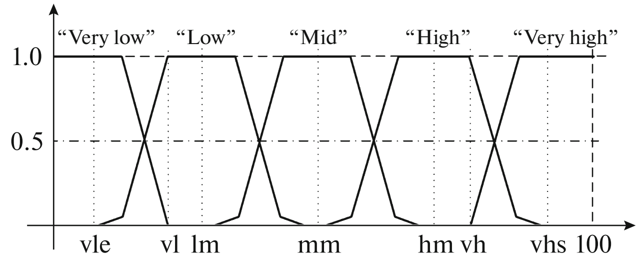


Fig. 8. Membership functions.

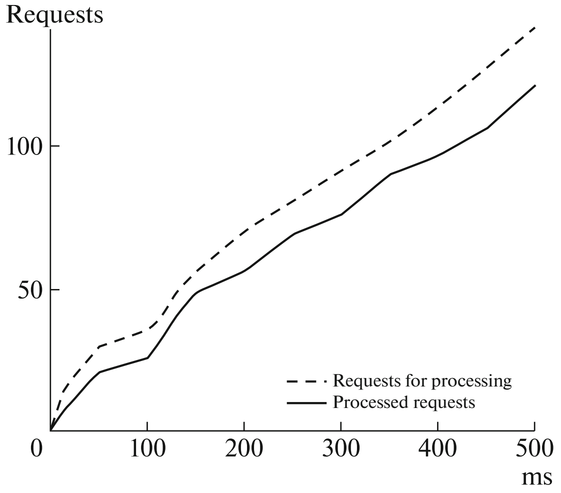


Fig. 9. Total amount of sent and processed requests.

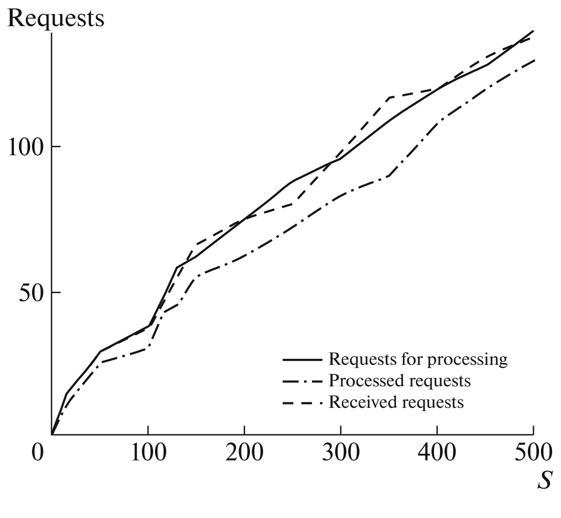


Fig. 10. Total amount of sent, received and processed requests with signal loss using centralized hierarchical control.

Implementation of the variable angle for data storing will help to maintain the equal level of obstacle detalization in memory of single robot for different striking distances. Such surrounding discretization provides a reduced datasets. It allows to make path planning faster (robots have to analyze less data) and performing more often the data exchange within the group, with the aim to upgrade the common knowledge database. That is why on this stage occur the need of data exchange network forming for group.

3. DYNAMIC NETWORKS

Article considers two models of data transferring: information exchange with centralized management (Fig. 5a) and strategies of centralized hierarchical control (Fig. 5b). On the figure presented solution for general case and for the reviewed model with the group of three robots.

3.1. Network Forming for Robotic Swarm

Consider a general case of swarm can be proposed a method of network forming based on creating a spanning tree. Algorithm consists of seven steps and includes the use of classical approaches. Steps of dynamic network forming for robotic swarm are next:

1. Build a fully-connected network graph (Fig. 6a);

2. Use the Kruskal algorithm to build the minimum spanning tree (Fig. 6b);

3. In the obtained tree, use the Floyd-Warshall algorithm to receive the list of all possible routes in the network;

4. Calculate the average route length for each node;

5. Select the node with the lowest average length and configure it as a high level node;

6. Nodes with "one side" connection configured as Low level nodes;

7. Other nodes configured as mid-level nodes (Fig. 6c);

3.2. Leader based Method

According to the complexity of Kruskal and Floyd-Warshall algorithms in group with small amount of robots would be more rational to use a role distribution methodology.

One of the models that describe the locally interacting robots organization is static swarm [5]. It is characterized by the absence of a given control center and is some kind of a fixed network – a set of agents. The basic properties of static swarm are activity, local interactions and functional heterogeneity. That is why will be reviewed the method of role distribution based on the task of selecting a leader.

Under the term "Leader" we will understand the central node of data exchange (robot for a short period of time will become server to store and marge data).

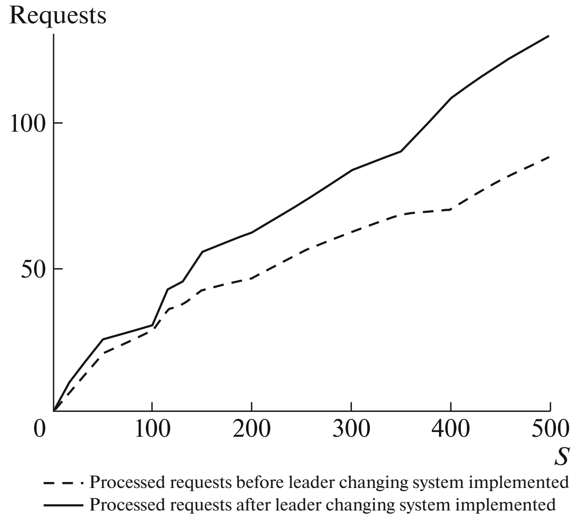


Fig. 11. Comparison of the number of processed requests before and after leader changing system implementation.

For choosing a leader robots will be using a voting process.

Each robot can be described as set of parameters

$$R = (I, L, E, N). \tag{6}$$

Where I – identifier of robot, L – identifier of voted leader, E – evaluation of the leader L (amount of voices that have to be given for a leader), N – list of connections available for robot (its neighbors).

The voting process on the initial step goes the following way: each robot evaluates his neighbors for the role of leader according to the set of previously defined its characteristics; each of these characteristics has their own weight; using the membership function robot selects the neighbor with the highest value.

For the vote value will be implemented a linguistic variable e = "evaluation of robot". Its value is based on the scale of M = {"very low", "low", "medium", "high", "very high"} or it can have a digit equivalent $M = \{1, 2, 3, 4, 5\}$. After voting process many alternatives for E will be generated, so it will have next form:

$$E = \{e_1, e_2, \ldots, e_n\}, \tag{7}$$

where e_i – alternative "candidate" at $i = 1..n$ and n is amount of visible neighbors.

For robot evaluation are offered following characteristics:

1) surroundings: c_1 = "the number of neighbors evaluable for candidate";

2) territorial: c_2 = "the distances to the neighbors or levels of signals";

3) status: c_3 = "the physical state of the robot".

Each of these characteristics are estimated by a voting robot for each of its neighbors:

$$C_i = \{c_{i1}, c_{i2}, \ldots, c_{ik}\}, \tag{8}$$

where – characteristic value of i-th "candidate" at $j = 1..k$.

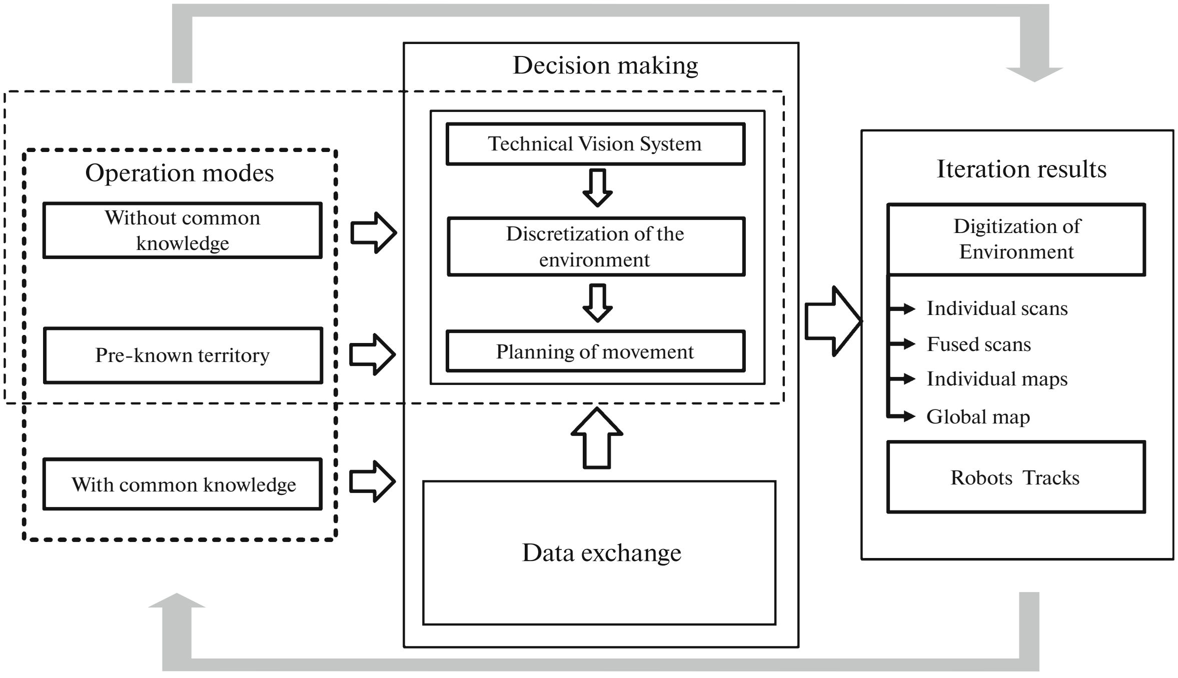


Fig. 12. System structure.

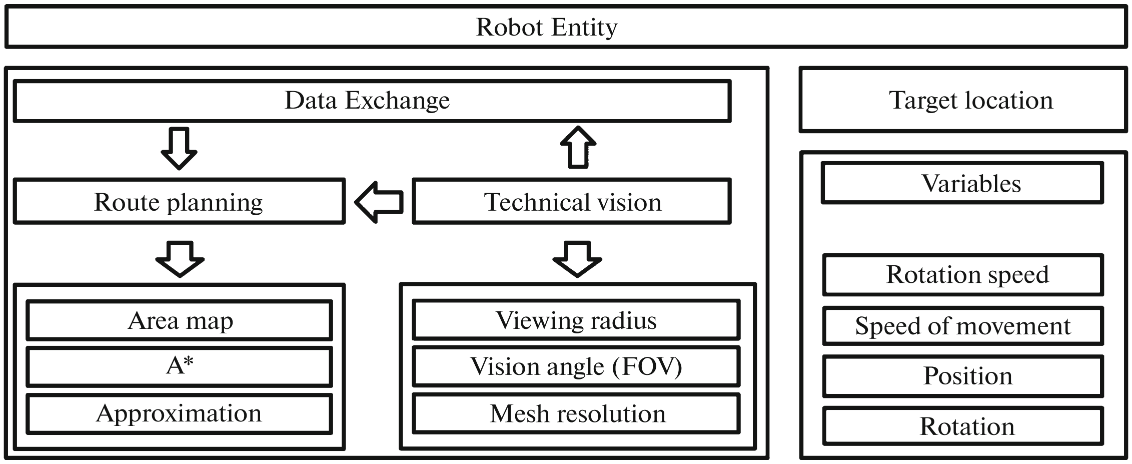


Fig. 13. Robot entity structure.

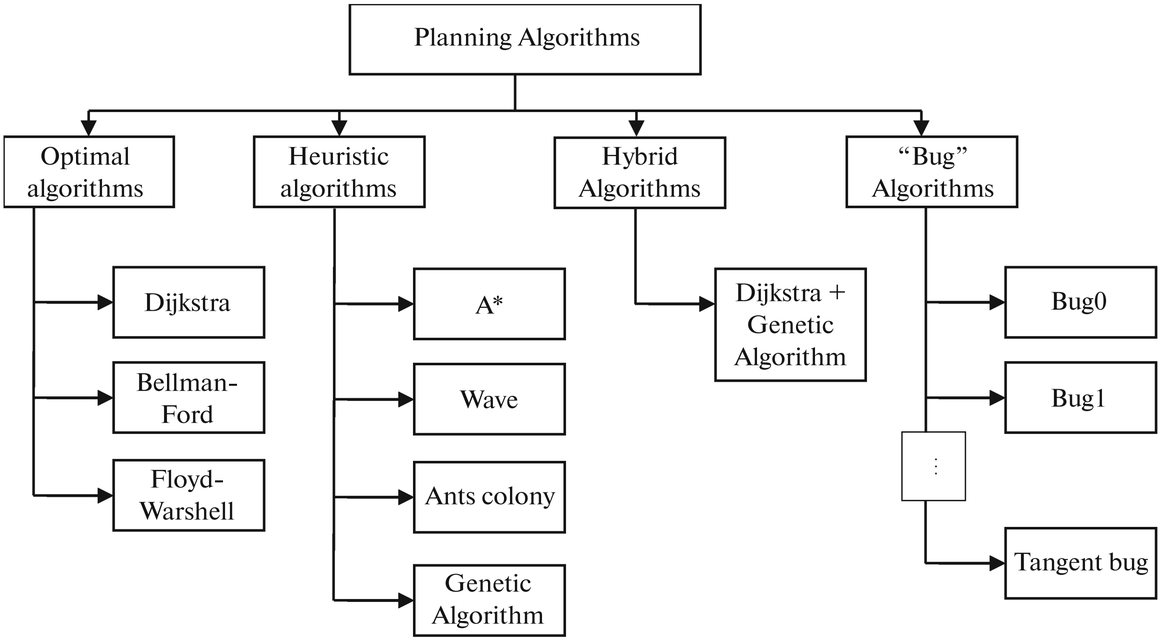


Fig. 14. Dynamic Path Planning Algorithm.

Each of the characteristics has its weight:

$$W = \{w_1, w_2, \ldots, w_k\}, \tag{9}$$

where – the j-th characteristics weighting,

Evaluation of the i-th neighbors uses the following formula:

$$e_i = \sum_{j=1}^{k} w_j c_{ij}. \tag{10}$$

To determine the value of linguistic variable we use three types of membership functions (11–13), where a general view represented on Fig. 8.

$$f_{vl}(e_i) = \begin{Bmatrix} 1, x < vle \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{e_i - vle}{vl - vle}\right), vle \le x \le vl \\ 0, x > vl \end{Bmatrix}, \tag{11}$$

where – the threshold to which the membership function is equal "1"; – a threshold, after which the membership function is equal "0";

$$f_{vh}(e_i) = \begin{Bmatrix} 1, e_i > vhs \\ \frac{1}{2} + \frac{1}{2}\cos\left(\frac{e_i - vhs}{vhs - vh}\pi\right), vh \le e_i \le vhs \\ 0, e_i < vh \end{Bmatrix}, \tag{12}$$

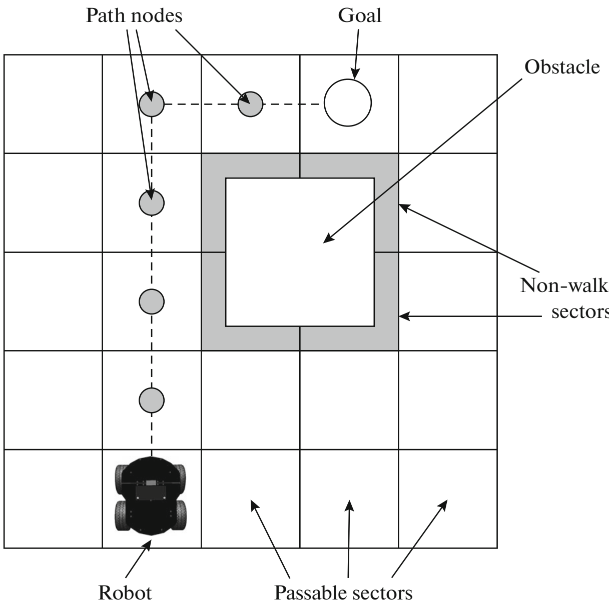


Fig. 15. Environment simplification.

where vhe – threshold, after which the membership function is equal to "1"; vh – a threshold, after which the membership function is equal "0";

$$f_{gb}(e_i) = \frac{1}{1 + \left|\frac{e_i - c}{a}\right|^{2b}}, \qquad (13)$$

where c – mid of a membership function; a – value at which $f_{gb}(c + a) = 1$ and $f_{gb}(c - a) = 1$; b – the value of function smooth regulation.

For obtaining the results, we will review leader changing method implementation on centralized hierarchical control network with five nodes (robots) in it. Initially robotic group works based on centralized control model where they have a central node for data margin. On Fig. 9 it is represented the total requests for processing and amount of requests that were processed (shows the sum from all available robots). There difference between series is explained by the data loss and duplicates of requests.

Data loss in such tasks causes additional load on path planning modules (unknown parts of terrain cause addition recalculations of rout) that also increase the time needed to solve main tasks (terrain mapping, object search, etc.). According to this should be implemented structural changes in to data transferring network.

Implementation of leader changing system gives a possibility for robots that lost connection to the central node to send their obtained data using neighbors, compensating by this all losses (Fig. 10).

Figure 11 compares the processed requests before and after using the leader changing system.

Presented results of our approach help to see that for each individual robot in-group common data (knowledge) becomes evaluable more frequently. To see the positive influence of such data exchange and formation of common knowledge base needs to be used a robotic group in some cluttered environment.

4. MODELING SYSTEM

4.1. Software Structure

To prove the theoretical basis of presented questions was used framework for the simulation and robotic group collaboration. Presented framework has been developed in Unity 5 [19], it is a multiplatform engine provided with different features and tools.

Framework is developed using programming language C# in MonoDevelop integrated development environment (IDE) for Windows 10.

Developed system (Fig. 12) has three operating modes "Without common knowledge", "Pre-known territory", "With common knowledge". First two use

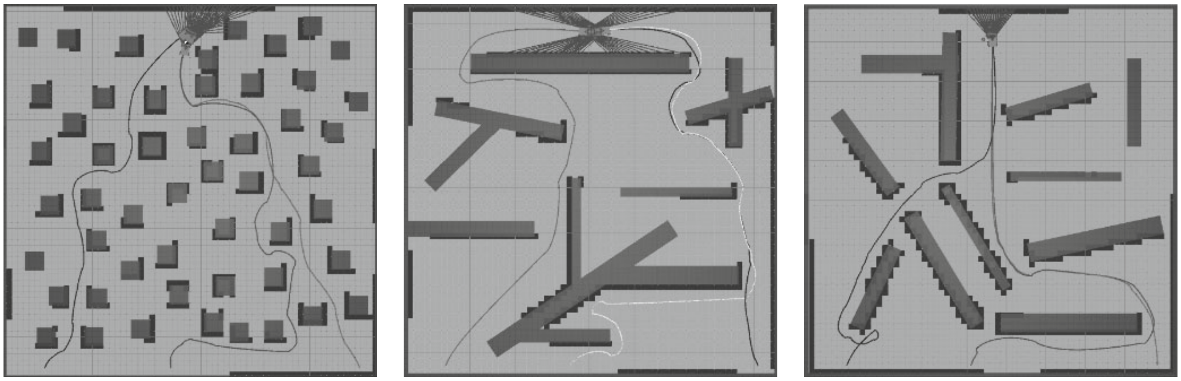


Fig. 16. Scenes examples.

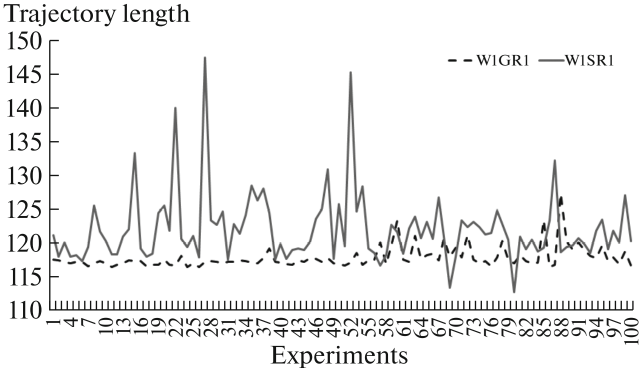


Fig. 17. Trajectories length for scene# 1, robot #1.

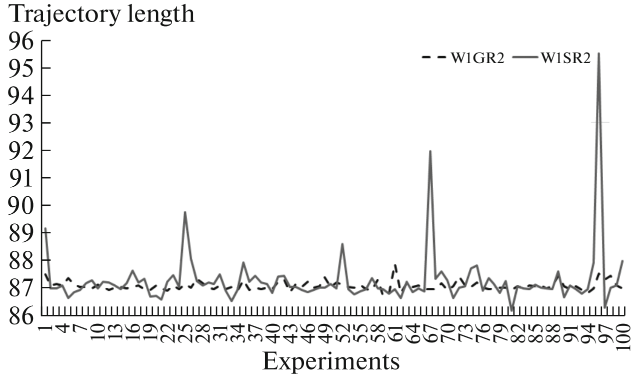


Fig. 18. Trajectories length for scene# 1, robot #2.

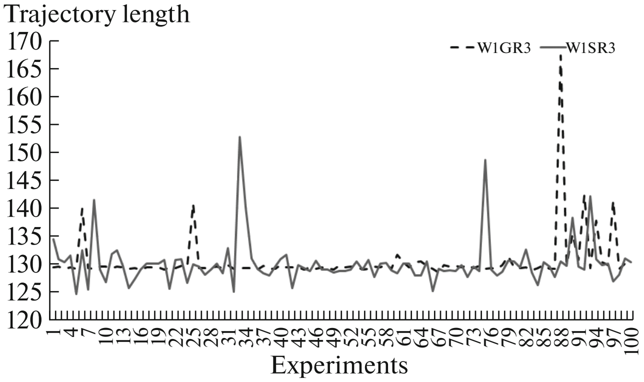


Fig. 19. Trajectories length for scene# 1, robot #3.

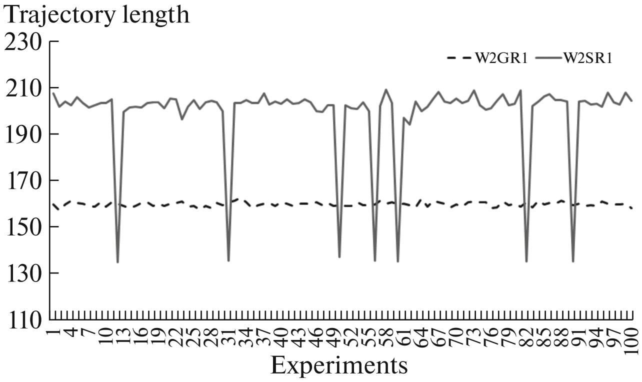


Fig. 20. Trajectories length for scene# 2, robot #1.

only part of decision-making system for path planning and obstacle avoidance. The third one implements full stack of decision-making process. In the end, system returns data about environment and state of each robot in moment of time.

4.2. Motion Planning

Most of the research reviews motion planning in the frames of path building such as [5] where authors represented an approach that uses motion primitive libraries. Other examples are [6], representing an attempt to implement animal motion for robot behavior, or [20] suggested an algorithm of collision free trajectory for robots. The articles cover some aspects of certain subtasks and widely describe special cases of behavior in a group of robots. However, no one has considered the task from the point of view of the interconnected global approach. It must include the correlation of robots' technical vision systems with communication and navigation rules in defined group of n-agents.

Table 3. Motion planning comparing results

	Scene #1	Scene #2	Scene #3
Robot 1	0.035	0.213	0.19
Robot 2	0.002	0.0042	0.012
Robot 3	0.13	0.0025	0.018
Total	0.066	0.098	0.1

In mathematics, there are well-developed algorithms (Fig. 14) for finding the way in an unknown or partially known environment (Optimal and heuristic algorithms). For this purpose, discrete mathematics (graph theory) and linear programming are usually used. Tasks of the shortest path search in the graph are known and studied (for example, Dijkstra's, Floyd-Warshell's, Prim's, Kruskal's, algorithms, etc. [21], [22]).

According to such matrix-based structure will be considered to use A* algorithm [23] with two-step post processing for dead reckoning.

Based on this logic, the agent moves in the direction of the target as long as no obstacle is to be

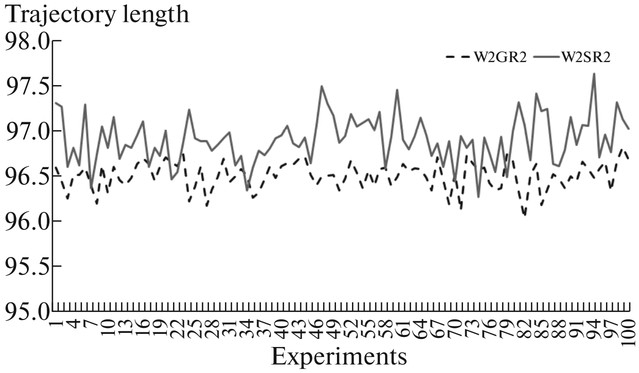


Fig. 21. Trajectories length for scene# 2, robot #2.

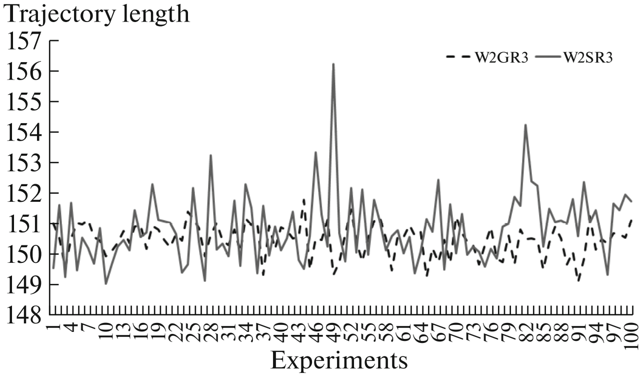


Fig. 22. Trajectories length for scene# 2, robot# 3.

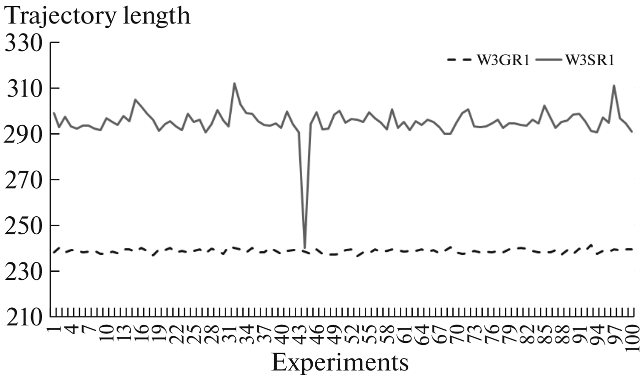


Fig. 23. Trajectories length for scene# 3, robot #1.

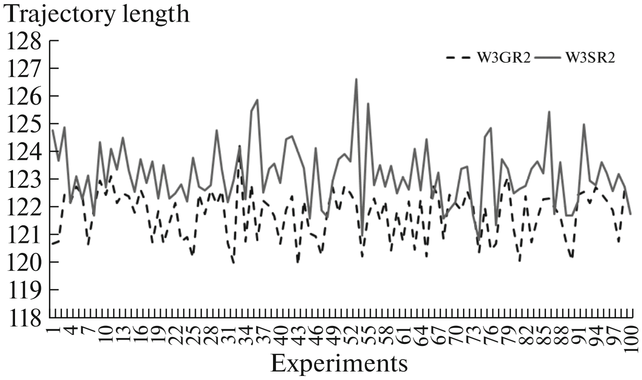


Fig. 24. Trajectories length for scene# 3, robot #2.

detected. After the presence of obstacles is detected by TVS in real scene robots initialize clear A* path planning. Points that are causing the necessity of changing the vector of movement are separated from the full path, forming the graph of movement. Such points will be called the break points. To improve the smoothness of the trajectory used the approximation – the second step of the post. For approximation used Bezier method with four points polygons ([24], [25], [26]). Thereafter, the agent starts to move to the first of the break points, for which the heuristic evaluation is minimal. During its movement the robot calculates new break points after each new warning of TVS (obstacle detection), and moves to them until the heuristic evaluation of the distance to goal stops decreasing (reaches a local minimum).

The above-mentioned TVS represents the surroundings as a cloud of points. Therefore, for path planning it is possible to separate terrain in small areas (cells) to decrease the amount of nodes needed for faster dead reckoning (Fig. 15).

5. EXPERIMENTAL RESULTS

The experimental results include the modeling of three scenes (Fig. 16). Each of them consider the group of three robots (n = 3) with possibility to communicate with each other (share obtained knowledge) and use laser-based technical vision system for environment interaction.

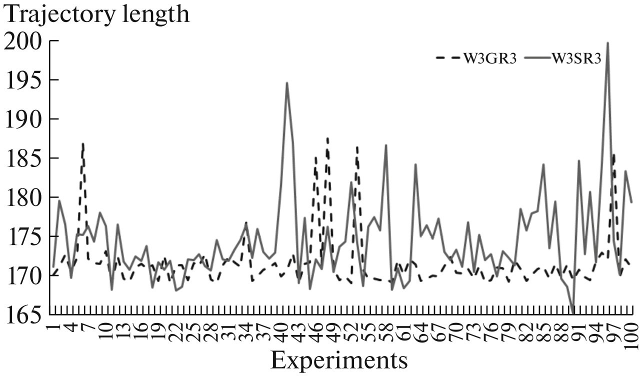


Fig. 25. Trajectories length for scene# 3, robot #3.

Experiments consider next scenario: for the group of three robots is given unknown terrain, each of the robots has the same common target to reach and one individual secondary point; first robots start to move to their individual targets, after reaching it, robots continue movement to common target; when all robots reach the common target scenario stops.

Obtained results are shown at Figs. 17–25. Series on figures uses the next marking like W1GR1 or W1SR1, here W{scene number} corresponds the scene number, G – goes for group movement (with common knowledge), in case of S as single movement (no common knowledge), and R{robot identifier} is robot number. The summary of the modeling presented in Table 3.

Table shows the results of the averaged distances for each individual robot in case of common knowledge comparing to the single robot movement. Result are shown that implementing path planning with common knowledge base can decrease the route of robots up to 21.3%.

6. CONCLUSIONS

This article offers an original solution that improves group routing in unknown terrain and collaboration between robots in-group. Proposed the dynamic data-transferring network forming models for the robotic swarm (spanning three-based model) and group by using leader-changing system. Communication within the group using voting process (leader-changing) shows the improvements in data exchange process.

According to the modeling results implementation of leader-changing system based on voting process and common knowledge base improves the path planning performance. Implemented approach creates continuous trajectories during the movement. The implemented method of resolution stabilization gives an additional option of the detalization adjustment according to specifics of the environment or current task. The use of fuzzy logic in various subtasks makes the behavior of robots more variable and allows for stable functioning of the group at lower energy costs. According to the obtained results individual trajectories length have improvements up to 21.3%.

ACKNOWLEDGMENTS

We want to extend our gratitude to the Universidad Autónoma de Baja California, Instituto de Ingeniería and the CONACYT for providing the resources that made this research possible.

REFERENCES

1. Eskridge, B., Valle, E., and Schlupp, I., Emergence of leadership within a homogeneous group, *PLoS One*, 2015, vol. **10**, no. 7.
2. Pshikhopov, V., Medvedev, M., Kolesnikov, A., Fedorenko, R., and Boris, G., Decentralized control of a group of homogeneous vehicles in obstructed environment, *J. Control Sci. Eng.*, vol. 2016, art. ID 7192371.
3. Sergiyenko, O., Ivanov, M., Tyrsa, V., Rivas-Lopez, M., Hernandez-Balbuena, D., Flores-Fuentes, W., Rodriguez-Quinonez, J.C., Nieto-Hipolito, J.I., Hernandez, W., and Tchernykh, A., Data transferring model determination in robotic group, *Rob. Auton. Syst.*, 2016, vol. **83**, pp. 251–260.
4. Sergiyenko, O.Yu., Ivanov, M.V., Kartashov, V.M., Tyrsa, V.V., Hernandez-Balbuena, D., and Nieto-Hipolito, J.I., "Transferring model in robotic group, *Proc. 25th IEEE Int. Symp. on Industrial Electronics (ISIE)*, Santa Clara, 2016.
5. Grymin, D.J., Neas, C.B., and Farhood, M., "A hierarchical approach for primitive based motion planning and control of autonomous vehicles, *Rob. Auton. Syst.*, 2014, vol. **62**, no. 2, pp. 214–228.
6. Kovacs, B., Szayer, G., Tajti, F., Burdelis, M., and Korondi, P., A novel potential field method for path planning of mobile robots by adapting animal motion attributes, *Rob. Auton. Syst.*, 2016, vol. **82**, pp. 24–34.
7. Bobkov, V.A., Ron'shin, Y.I., Kudryashov, A.P., and Mashentsev, V.Y., 3D SLAM from stereoimages, *Program. Comput. Software*, vol. **40**, no. 4, pp. 159–165.
8. Bobkov, V.A., Kudryashov, A.P., and Mel'man, S.V., On the recovery of motion of dynamic objects from stereo images, *Program. Comput. Software*, vol. **44**, no. 3, pp. 148–158.
9. Kamaev, A.N., Sukhenko, V.A., and Karmanov, D.A., Constructing and visualizing three-dimensional sea bottom models to test AUV machine vision systems, Program. Comput. Software, vol. **43**, no. 3, pp. 184–195.
10. Vilao, C.O., Perico, D.H., Silva, I.J., Homem, T.P.D., Tonidandel, F., and Bianchi, R.A.C., A single camera vision system for a humanoid robot, *Proc. Joint Conf. on Robotics: SBRLARS Robotics Symp. and Robocontrol*, Sao Carlos, 2014.
11. Katalinic, B., Gryaznov, N., and Lopota, A., Computer vision for mobile on-ground robotics, *Proc. Eng.*, 2015, vol. **100**, pp. 1376–1380.
12. Achtelik, M.C. and Scaramuzza, D., Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments, *IEEE Rob. Autom. Mag.*, 2014, vol. **21**, no. 3, pp. 26–40.
13. Pashchenko, N.F., Zipa, K.S., and Ignatenko, A.V., An algorithm for the visualization of stereo images simultaneously captured with different exposures, *Program. Comput. Software*, 2017, vol. **43**, no. 4, pp. 250–257.
14. Alenya Ribas, G., Foix Salmeron, S., and Torras Genis, C., ToF cameras for active vision in robotics, *Sens. Actuators, A*, 2014, vol. **218**, pp. 10–22.
15. Mikhaylichenko, A.A. and Kleshchenkov, A.B., Approach to non-contact measurement of geometric parameters of large-sized objects, *Program. Comput. Software*, 2018, vol. **44**, no. 4, pp. 271–277.

16. Basaca-Preciado, L.C., Sergiyenko, O.Y., Rodriguez-Quinonez, J.C., Garcia, X., Tyrsa, V.V., Rivas-Lopez, M., Hernandez-Balbuena, D., Mercorelli, P., Podrygalo, M., Gurko, A., Tabakova, I., and Starostenko, O., Optical 3D laser measurement system for navigation of autonomous mobile robot, *Opt. Lasers Eng.*, 2014, vol. **54**, pp. 159–169.

17. Sergiyenko, O., Hernandez, W., Tyrsa, V., Cruz, L.D., Starostenko, O., and Pena-Cabrera, M., Remote sensor for spatial measurements by using optical scanning, *MDPI Sens.*, 2009, vol. **9**, no. 7, pp. 5477–5492.

18. Garcia, X., Sergiyenko, O., Tyrsa, V., Rivas-Lopez, M., D. Hernandez-Balbuena, Rodriguez-Quinonez, J.C., Basaca-Preciado, L.C., and Mercorelli, P., Optimization of 3D laser scanning speed by use of combined variable step, *Opt. Lasers Eng.*, 2014, vol. **54**, pp. 141–151.

19. Hocking, J., *Unity in Action: Multiplatform Game Development in C# with Unity 5*, New York: Manning, 2015.

20. Ali, A.A., Rashid, A.T., Frasca, M., and Fortuna, L., An algorithm for multi-robot collision-free navigation based on shortest distance, *Rob. Auton. Syst.*, 2016, vol. **75**, pp. 119–128.

21. Trianni, V., Tuci, E., Ampatzis, C., and Dorigo, M., Evolutionary swarm robotics: a theoretical and methodological itinerary from individual neurocontrollers to collective behaviors, in *The Horizons of Evolutionary Robotics*, Cambridge, Mass: MIT Press, 2014.

22. Vincent, R., Morisset, B., Agno, A., Eriksen, M., and Ortiz, C., Centibots: large-scale autonomous robotic search and rescue experiment, *Proc. 2nd Int. Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems*, Red Hook, NY: Curran Assoc., 2008.

23. Munoz, P., Maria, R.-M.D., and Barrero, D.F., Unified framework for path-planning and task-planning for autonomous robots, *Rob. Auton. Syst.*, 2016, vol. **82**, pp. 1–14.

24. Bezier, P.E., *How Renault Uses Numerical Control for Car Body Design and Tooling*, Detroit: Soc. Autom. Eng., 1968.

25. Han, L., Yashiro, H., Nejad, T., Do, Q., and Mita, S., Bezier curve based path planning for autonomous vehicle in urban environment, *Proc. IEEE Intelligent Vehicles Symp.*, La Jolla, CA, 2010.

26. Kawabata, K., Ma, L., Xue, J., Zhu, C., and Zheng, N., A path generation for automated vehicle based on Bezier curve and viapoints, *Rob. Auton. Syst.*, 2015, vol. **74**, pp. 243–252.