Advances in Robotics Research, *Vol. 2, No. 3 (2018) 219–236*
DOI: https://doi.org/10.12989/arr.2018.2.3.219

Controlling robot formations by means of spatial reasoning based on rough mereology

Lukasz Zmudzinski*, Lech Polkowski[a] and Piotr Artiemjew[b]

Faculty of Mathematics and Computer Science University of Warmia and Mazury in Olsztyn Sloneczna 54, 10-710 Olsztyn, Poland

(Received July 25, 2018, Revised October 5, 2018, Accepted November 5, 2018)

Abstract. This research focuses on controlling robots and their formations using rough mereology as a means for spatial reasoning. The authors present the state of the art theory behind path planning, robot cooperation domains and ways of creating robot formations. Furthermore, the theory behind Rough Mereology as a way of implementing mereological potential field based path creation and navigation for single and multiple robots is described. An implementation of the algorithm is shown in simulation using RoboSim simulator. Five formations are tested (Line, Rhomboid, Snake, Circle, Cross) along with three decision systems (First In, Leader First, Horde Mode) as compared to other methods.

Keywords: rough mereology; path planning; robot teams; potential fields

1. Introduction

Investigations into problems of multiple–robot systems extend not only the single robot problem studies, but also they allow for a study of a plethora of new problems resulting from collective behavior of robots. In this paper the autors propose a way of introducing Rough Mereology in the field of mobile robotics teams after successfully implementing the methods for single robots. The method uses spatial reasoning (by the notion of a part from Rough Mereology) to implement geometric path planning in a known environment, for mobile robots. The authors test, if proposed methods give valid results and are adequate for implementation for teams of robots. Each of these robots can be regarded as an intelligent mobile agent and those agents have to communicate, cooperate, divide tasks, allocate them, plan collision–free trajectories for agents and implement them, i.e., navigate together to achieve the cooperative goal. Passing from a single robot to teams of robots can be motivated also by pragmatic reasons as seen in Cao *et al.* (1997):

1. Tasks for robots can be too complex for a single robot or many robots can do the task easier at a lesser cost,

*Corresponding author, E-mail: lukasz@zmudzinski.me
[a]Professor, E-mail: lech.polkowski@uwm.edu.pl
[b]Ph.D., E-mail: piotr.artiemjew@uwm.edu.pl

http://www.techno-press.org/?journal=arr&subpage=7 ISSN: 2287-4976(Print), 2287-4984(Online)

2. Teams can perform the task more reliably with a smaller margin of error,
3. Study of robot teams can provide insight into problems of social sciences like cognitive theories, biology, ethology, organization, as well as provide new solutions to problems of artificial intelligence and help to work out new techniques and heuristics.

In Cao *et al.* (1997), authors single as exemplary the following task domains which have stimulated the developments in multiple–robot systems.

- TRAFFIC CONTROL. Robots moving in a common environment face problems of collision–avoidance (resource conflicts, traffic rules, priorities, path planning as a geometric problem in configuration space, collision–avoidance behaviors).
- COOPERATIVE MANIPULATION. Cooperative task performing involves box–pushing (communication), large objects manipulation (problems with mutual robot visibility).
- FORAGING. Consists of a group of robots finding and picking objects (food) dispersed in the environment.

From those research domains, Cao *et al.* (1997) extract five main research "axes":

1. **Group Architecture**. It features problems of *centralization / decentralization*: whether control is performed by a dedicated robot or there is no such agent. Decentralization is divided into *distributed* or *hierarchical*. In particular, control may be implemented by means of some "leaders" i.e., agents on which others are locally oriented. Decentralized architectures often lead to "emergent behavior" or "self–organization".
2. **Resource Conflict**. It does arise when many robots request usage of certain means like communication channels, space including paths or objects to manipulate (e.g., food). The most studied problem has been space sharing centered on *path planning*, *collision–avoidance* and *deadlock–avoidance*. To solve those problems, robots are provided with a priori chosen routes and traffic rules as well as with deadlock avoiding communication means, Caloud *et al.* (1990).
3. **Emergence of Cooperation**. A study of mechanisms leading to cooperation has led to two principal types of collective behavior: *eusocial behavior* and *cooperative behavior*, McFarland (1994). Eusocial behavior is characteristic for lower biological organisms like ants, bees etc. leading to cooperative behavior through interactions of individual efforts, stimulated often by biochemical mechanisms. Cooperative behavior is characteristic for higher vertebrates and consists in conscious desire to join efforts in order to achieve better utility; simulations of emergent cooperation were done with robots bound to cooperate for e.g. surviving with minimal energy, McFarland (1994).
4. **Learning**. The problem of learning that is central for machine intelligence in Mitchell (1998) has been addressed in multiple–robot systems with recourse to reinforcement learning as in Matric (1994) in behavior–based multiple–robotic systems. Also, genetic programming was applied in learning predator avoidance by herding by Werner and Dyer (1992) and in learning flocking behavior as in Reynolds (1992).

5. **Geometric Problems**. This topic is central to our work as it does involve multiple–robot path planning, formation maintaining, moving to formations, pattern generation.
6. **Simulation versus Experiment**. As discussed in the context of multiple–robot systems issue is the simulation/real experiment dilemma. Some authors argue that simulation ignores most difficult problems related to perception and actuation, hence it does over–simplify problems and produces over–successful systems which could be impossible to render in real conditions - Brooks (1991). On the other hand, according to many researchers - *”it is unlikely that truly large–scale collective behavior involving hundreds or thousands of real robots will be feasible at any time in near future. This, cooperative mobile robot researchers have used a variety of techniques to simulate perception”*, Cao *et al.* (1997). In Matric (1994), among others, it is advocated to make simulations as prototypes and an experiment with a small number of real robots as a proof–of–concept demonstration and this approach seems the most sensible and it is taken in our work.

The most interesting part for our work is the **Geometric Problems** research ”axe”. This comes from the fact, that we use geometric problems for controlling robot formations by means of spatial reasoning based on rough mereology:

- **Path planning** involves many specialized strategies; some of them propose initial individual paths for each robot, often straight lines to the goal, with strategies for obstacle negotiating and conflict resolution by either negotiations between robots or by a supervising agent.
- **Formation** problem as well as Marching Problem require of robots to move into a prescribed formation and march to the goal maintaining the formation. Quoting Cao *et al.* (1997): *”The Formation problem seems very difficult, e.g., no published work has yet given a distributed ”circle forming” algorithms that guarantees the robots will actually end up in a circle”.*

1.1 Potential fields for planning and navigation

For our work we use potential field approach for path and formation planning, but there are several other methods, that were presented in the literature that are characteristic to the robotic state of the art. These methods include:

- **Behavior–based aproach**, based often on the ”boid” algorithms that imitate simple bird groups behaviors like: collision–avoidance, velocity matching, flock centering and geometric positions, Reynolds (1987). Such behaviors are described in this work, where teams of robots can adapt to the changing environment and change their formations based on surroundings.
- **Potential field aproach**, defines interaction forces among robots forcing them to keep at desired distance from another. Virtual leaders are moving reference points for robots to control the group movement and maintain group geometry. In this approach, the already mentioned by us biological behaviors of swarms like: avoidance of close neighbors, keeping distance to the group, velocity matching are encoded by means of artificial local potentials defined as functions of relative distances between pairs of neighbors are implemented; control forces are then defined as negative gradients of the sum of potentials affecting a given robot. By their action,

robots are driven to minimum of the global potential function; local potentials can be designed as to correspond to a given geometry of the group.

- **Metric aproach**, where geometric formation relies on a direct usage of using a metric described by various parameters: threshold, robot diameters, path length ratio, average position error or percentages regarding formation time.

It is worth noticing, that the potential field method is considered for current research for various robotic problems such as behaviors of pedestrians in a lane in Wei *et al.* (2015) where potential field callular automaton is applied to the study of avoidance and following behaviors or path planning in Ahmed *et al.* (2015) where a modification of the classical attracive-repulsive Coulomb-style potential is applied, or Orozco-Rosas *et al.* (2015) where bacterial genetic programming is the tool in studying navigation problems of robots..

1.2 Potential fields

Classical methodology of potential fields works with integrable force field given by formulas of Coulomb or Newton which prescribe force at a given point as inversely proportional to the squared distance from the target. In consequence the potential is inversely proportional to the distance from the target. The basic property of the potential is that its density (=force) increases in the direction toward the target. We observe this property in our construction. We start in our construction with a rough inclusion – a similarity measure formalized in the theory of rough mereology. Its basic construct is a *rough inclusion*. Rough inclusion is a ternary relation $\mu : U \times U \times [0,1]$ and its primitive formulas of the form $\mu(x, y, r)$ read *'the object x is a part to object y to a degree of at least r'*. A rough inclusion can be used to induce a distance function as well as primitive relations of elementary geometry: betweenness and nearness.

Geometry induced by means of a rough inclusion can be used to define a generalized potential field: the force field in this construction can be interpreted as the density of squares that fill the workspace and the potential is the integral of the density. We present now the details of this construction, see also Ośmiałowski (2010), Ośmiałowski (2009).

1.3 Potential field construction

The potential field generation methods called **Square Fill Algorithm** introduced by Ośmiałowski (2009) was presented with some alternative modifications in Polkowski *et al.* (2018), Zmudzinski and Artiemjew (2017) and Gnys (2017) as follows:

1. Define initial values:
   - Set **current distance** to the goal: $d = 0$,
   - Set algorithm **direction** to *clockwise*,
2. Create an empty queue Q

$$Q = \emptyset \tag{1}$$

3. Add to queue Q the first **potential field** $p(x, y, d)$, where x, y describe the location of the field and d represents the **current distance** to the goal

$$Q \cup \{p(x, y, d)\} \tag{2}$$

4. Enumerate through Q,
5. If $(\{p_k(x, y, d)\} \cap F) \vee (\{p_k(x, y, d)\} \cap C)$ where $p_k(x, y, d)$ is the current potential field, F is a set of already created potential fields and C is a set of collision objects, then remove the current field $p_k(x, y, d)$ from Q and go back to point **4**,
6. Add the current potential field $p_k(x, y, d)$ to the created potential fields set F

$$F \cup \{p_k(x, y, d)\} \tag{3}$$

7. Increase the **current distance** to the goal

$$d = d(p_k) + 0.01 \tag{4}$$

8. Define neighbours depending on the current **direction**:
   - *clockwise* as N

$$N = \left\{ \begin{array}{l} p_0 = p(x-d, y, d), \\ p_1 = p(x-d, y+d, d), \\ p_2 = p(x, y+d, d), \\ p_3 = p(x+d, y+d, d), \\ p_4 = p(x+d, y, d), \\ p_5 = p(x+d, y-d, d), \\ p_6 = p(x, y-d, d), \\ p_8 = p(x-d, y-d, d) \end{array} \right\} \tag{5}$$

   - *anticlockwise* as N'

$$N' = \left\{ \begin{array}{l} p_0 = p(x-d, y-d, d), \\ p_1 = p(x, y-d, d), \\ p_2 = p(x+d, y-d, d), \\ p_3 = p(x+d, y, d), \\ p_4 = p(x+d, y+d, d), \\ p_5 = p(x, y+d, d), \\ p_6 = p(x-d, y+d, d), \\ p_8 = p(x-d, y, d) \end{array} \right\} \tag{6}$$

9. Add neighbours to queue Q depending on current direction:
   - If **direction** is *clockwise* then: $Q \cup N$,
   - If **direction** is *anticlockwise* then: $Q \cup N'$,
10. Change the current **direction** to the opposite,
11. Remove the current potential field $p_k(x, y, d)$ from the queue Q,
12. if $Q(p) = \emptyset$ then finish, else go to **4**.

1.4 Planning for multirobot systems

Planning for multiple robots is based on planing methodology for a single robot, with necessary modifications, see Hwang and Ahuja (1992) or Latombe (1991) for surveys. As with a single robot, approaches to planning for multi-robots can be divided into *centralized* or *decoupled*. Centralized approach makes works by finding a path in a complex configuration space describing the whole system provides *complete* planners which always find a path for the system if it exists; however, this comes at the cost of exponential complexity: the problem of planning for rectangular robots in the rectangular workspace is **P**–space complete Hopcroft *et al.* (1984). Planners of polynomial complexity based on cell decomposition, for disc shaped robots in polygonal obstacle world were described in Schwartz and Sharir (1983).

Variants of the method of potential fields were applied as well in centralized planning: a potential field approach in which centralization was introduced as a local optimization problem. In Barraquand and Latombe (1990), a randomized path planner, a potential field based planner with random fluctuations escaping local minima was described. Such methods require the use of geometric methods for describing the potential fields generated by robots and obstacles. Similar methodologies can be seen in Alonso-Mora *et al.* (2016) where authors use that approach for distributed multi-robot formation control.

In the area of decoupled planning, the problem is to merge plans for individual robots into one general plan for the whole system; here, the idea of *prioritization* was put forth Erdmann and Lozano–Perez (1986). A decoupled planner based on a scheduling technique was described in O'Donnel and Lozano–Perez (1989). Liu *et al.* (1989) described a decoupled approach in which conflicts among paths were resolved by means of a Petri net. An approach resting on an incremental merging of plans for individual robots on the basis of current information about the merged plan and actions of remaining robots is presented in Alami *et al.* (1995). Some schemes in decoupled planning rely on using traffic rules Grossman (1988). High complexity of centralized planning and incompleteness of decoupled planning prompted research in the between area and the idea of separate *roadmaps* for robots emerged in which separate roadmaps are combined into a global roadmap Valle and Hutchinson (1999). A graph–based approach to roadmap coordination was presented in Svestka and Overmars (1998) leading to a probabilistically complete planner.

We now describe in short some of these works in order to give a more detailed view into solutions adopted. Prioritized planning in Erdmann and Lozano–Perez (1986) is a compromise between demands of centralized planning and computational feasibility: in this approach, motions are planned one object at a time, thus central planning becomes a sequence of autonomous planning operations. However, the lower the priority of an object, the more heavy demands are placed on its plan to avoid collisions with earlier objects. Also, completeness is not assured and the goal may not be reached. The implementation of this idea requires a construction of space–time configuration which reflects the state of the system at each moment of time; planning for an object involves motion on that space. Building of the configuration space is facilitated by assumption that obstacles are translation–invariant with regard to shape; thus, translation of an obstacle in the real world is associated with translation of the configuration space obstacle representation. Search for collision–free paths is performed by examining segment paths meeting adjacent slices, with endpoints in vertices of obstacles, see visibility graphs; segments intersecting obstacles at non–vertex points are ignored and remaining segments are joined into paths. A prioritization method which maximizes the number of robots able to move from start to goal in the straight line is proposed in Buckley (1989).

A stochastic approach in Barraquand and Latombe (1990) makes use of a configuration space discretized into a grid of n–dimensional rectangloids whose resolution conforms to resolution of the bitmap representation of the workspace as to ensure that small movement in the grid is reflected in a small movement of robots. The goal assignment is achieved by the idea of a *control point*: control points on robots are assigned for their goal positions and the system reaches the goal when each control point has reached its goal position. With each control point p, a potential function is associated: $U_p : W_{free} \to R^+$ sending each point $q \in W_{free}$ in the free workspace W_{free} into a real positive value. Those potentials are combined in the configuration space into a global potential function $U : q \in C_{free} \to G(\{U_p(c(p, q)) : p \in \{control\ points\})$ where $c(p, q)$ is the coordinate set of p at a configuration space coordinates q. Authors have worked out techniques that ensure that the potential U has the only global minimum at the goal and local minima resulting from combining by means of G of control point potentials are easy to be bypassed.

The algorithm described by the authors, begins with the initial configuration point q_{init} and proceeds by following the negative gradient of U. The gradient following motion stops when a local minimum q_{loc} is reached; if it is the minimum, the algorithm stops. Otherwise, it does execute a series of random motions each of them followed by the gradient following motion until a new local minimum q'_{loc} is detected; then, q_{loc} and q'_{loc} are connected with an edge. This graph is incrementally built until the global minimum is reached and then the graph provides a path to the goal. The gradient motion is organized on the "best–first" search basis: neighboring grid cells are searched for the lowest value; in high dimensions the number of neighbors may be prohibitive and then the search is random; a variant of depth–first search is also discussed. The random motion applied is modeled as the Brownian motion (the Wiener–Levy stochastic process) which assures that the planner is probabilistically complete (i.e., it reaches the goal with probability 1).

A view of a planner for many robots as a concurrent system with constraints on concurrency is presented in Liu *et al.* (1989). The case studied is of two planar robots circular in shape in presence of obstacles. The configuration space is represented as a quadtree with nodes marked with respect to meeting obstacles: yes, no and partial. A two level hierarchical planner is used to build a Petri net representing conflicting situations. The higher level planner plans paths for robots by searching the quadtree through obstacle–free nodes, and the lower level planner solves the conflict problem. The higher level planner finds the shortest path obstacle–free to the goal for each robot. The lower level planner examines the paths and finds nodes where paths intersect for which it finds neighboring nodes on paths excluding other path (avoidance nodes). By means of avoidance nodes, conflicts can be resolved. A means for conflict resolution is provided by a Petri net whose sequence of firing transition is such that it prevents robots from simultaneously entering the same node on their paths.

A more general problem of mission planning for multiple vehicles and concurrent goals is addressed in Brumitt *et al.* (2001) where a distributed planner is introduced in the context of a dynamic allocation of goals to autonomous vehicles. Planning is effected in the environment of a mission grammar MG: $m \to M(r, g)|m{\wedge}m|m{\vee}m|m \Rightarrow m|$(m); $r \to R_i|r{\wedge}r|r{\vee}r|$(r); $g \to G_j|g{\wedge}g|g{\vee}g|g \Rightarrow g|$(g) where $a \Rightarrow b$ means "a followed by b", $a \wedge b$ means "a and b", $a \vee b$ means "a or b", R_i means "robot i", G_i means "goal i", $M(r, g)$ means "move robot r to goal g". For instance, the expression $M((R_1 \wedge R_2), G_1 \Rightarrow G_2)$ means that robots 1 and 2 are to go to goal 1 and then to goal 2.

This simple grammar expressions are examined by the mission planner and parsed into sequences of executable commands and planning of paths for them uses D^* search algorithm.

Finally, we return to already mentioned paper by Švestka and Overmars, in which a coordinated approach to decoupled planning is developed. For each robot, a probabilistic roadmap, see sect. 2, is constructed. Then, roadmaps for individual robots are combined into a roadmap for the system. The global roadmap is constructed as a graph structure called a super–graph. In constructing a super–graph, few steps are taken. Robots are modeled as copies of a single robot a. For this robot, given a workspace, a configuration space is built and then paths in the free space are found for a. This gives a roadmap for a as a graph (V, E). Then, a coordinated roadmap is constructed for the group of robots $a_1, a_2, ..., a_n$ by defining coordinated paths as n–tuples $(p_1, ..., p_n)$ where p_i is a path for the robot a_i and paths p_i, p_j are collision free for $i \neq j$. Authors consider such coordinated paths on which only one robot moves at a time, therefore the robot movement is sequential. The super–graph is then constructed as a graph whose nodes are n–tuples $(x_1, x_2, ..., x_n)$ of collision–free nodes in (V, E) and edges are coordinated paths joining two nodes. The search for a path for the system is performed on the super–graph; authors provide a retraction of the configuration space onto the super–graph in order to refer the former to the latter. A number of technical operations on a super–graph is available, improving the performance, smoothing the paths etc.

2. Rough mereology

Mereology introduced as the formal system of reasoning by Stanisław Leśniewski replaces the notion of an *element* basic for set theory with the notion of a *part* Leśniewski (1982). It is therefore suited best for reasoning about solids, figures etc. as relations among them can often be expressed by means of parts not by elements, consider an example: 'The circle is a part of the disc' but not 'The circle is an element of the disc'. Mereology is founded on the notion of a part, formally rendered by the formula $\pi(x, y)$ which reads: *'x is a part of y'*. The relation π is supposed to satisfy the following requirements: (1) If $pi(x, y)$ then $x \neq y$ (2) If $\pi(x, y)$ and $\pi(y, z)$ then $\pi(x, z)$. Hence, $\pi(x, y)$ means that x is a proper part of y which can be visualized by saying that x is *'smaller than y and inside y'*. The relation of being possibly the whole of y is expressed by the notion of a *Part*, Π, defined as

$$\Pi(x, y) \ if \ and \ only \ if \ \pi(x, y) \ or \ x = y. \tag{7}$$

Rough mereology which is concerned with the relation of cutting a piece off y but neither being any part of y nor the whole of y, requires for its expression a new language in which one can express degrees to which y is in x, and this language is many-valued logic over the interval $[0, 1]$, see Hájek Hajek (1998). The basic notion is that of *rough inclusion* $\mu(x, y, r)$ which reads *'x is a part of y to a degree of at least r'*. Rough mereology was proposed by Polkowski cf. Polkowski (2011), Polkowski and Skowron (1996), Polkowski and Skowron (1994), Polkowski (2017). As with the part relation, rough inclusions are subject to requirements which are:

1. $\mu(x, y, 1)$ if and only if $\Pi(x, y)$. This means that a rough inclusion can be defined only on a universe on which the mereological structure is set.
2. If $\mu(x, y, 1)$ and $\mu(z, x, r)$ then $\mu(z, y, r)$. This is the monotonicity condition: the 'bigger' y cuts from z at least the fraction cut by 'smaller' x.

3. If $\mu(x, y, r)$ and $s < r$ then $\mu(x, y, s)$. This is the condition which justifies the phrase '... is a part to a degree at least ...'.

Rough mereological geometry Spatial reasoning requires as well a formal system in which to carry out reasoning. Such system was constructed by Polkowski cf. Polkowski (2011) on basis of ideas and results by Tarski Tarski (1959) and van Benthem van Bethem (1983). This system which renders the Euclidean geometry of finite-dimensional spaces is based on two relations: equidistance and betweenness. Equidistance captures the metric aspect of geometry while betweenness responds for the affine properties. Equidistance relation denoted $Eq(x, y, u, z)$ (or, as a congruence: $xy \equiv uz$) means that the distance from x to y is equal to the distance from u to z (pairs x, y and u, z are equidistant) is subject to requirements:

- Eq–REFLEXIVITY: $Eq(x, y, y, x)$.
- Eq–IDENTITY: If $Eq(x, y, z, z)$ then $x = y$.
- Eq–TRANSITIVITY: If $Eq(x, y, u, z)$ and $Eq(x, y, v, w)$ then $Eq(u, z, v, w)$.

Betweenness relation $B(x, y, z)$ (y is between x and z) is required to satisfy the requirements:

- B–IDENTITY: If $B(x, y, x)$ then $x = y$.
- B–PASCH AXIOM:
  If $B(x, u, z)$ and $B(y, v, z)$ then there is some a such that $B(u, a, y)$ and $B(v, a, x)$.
- B–CONTINUITY: Let $\phi(x)$ and $\psi(y)$ be first–order formulas in which objects a, b do not occur as free, and, x is not free in $\psi(y)$ and y is not free in $\phi(x)$. If there is a such that for each pair x, y [from $\phi(x)$and $\psi(y)$ it follows that $B(a, x, y)$], then there is b such that for each pair x, y [from $\phi(x)\ textrmand\ \psi(y)$ it follows that $B(b, x, y)$].
- B–LOWER DIMENSION: For some triple a, b, c [not $B(a, b, c)$ and not $B(b, c, a)$ and not $B(c, a, b)$].

For our purpose, we need to exploit the notion of betweenness in combination with rough inclusions. To this end, we introduce the notion of the *mereological distance*, $\kappa(x, y)$ cf. Polkowski (2011). We need some auxiliary notions

$$\mu^+(x, y) = r\ if\ and\ only\ if\ \mu(x, y, r)\ and\ not\ \mu(x, y, s)\ for\ any\ s > r. \tag{8}$$

Hence, $\mu^+(x, y)$ denotes the largest degree to which x is a part of y. We assume that this value exists. We are able now to define the distance κ.

$$\kappa(x, y) = min\{\mu^+(x, y), \mu^+(y, x)\}. \tag{9}$$

Let us observe that $\kappa(x, y) = 1$ means that $x = y$, contrary to the Euclidean distance whose value in that case would be zero. On the other hand, $\kappa(x, y) = 0$ means that x, y can only touch themselves on their boundaries. This will become more clear when we introduce an example of a rough inclusion suitable for solids, e.g., rendering mobile robots in the plane or 3-D space. We denote with the symbol

$V^n(x)$ the n-dimensional volume in the n-dimensional space. Then, we define the rough inclusion $\mu_V(x, y, r)$ for bounded measurable sets x, y in the n-space

$$\mu_V(x, y, r)\ if\ and\ only\ if\ \frac{V^n(x \cap y)}{V^n(x)} \geq r. \tag{10}$$

It is easy to see that the rough inclusion $\mu_V(x, y, r)$ does satisfy requirements (1)-(3) for rough inclusions. In order to include betweenness in our discussion, we first recall the notion of *nearness relation*, $N(x, y, z)$ in symbols, which reads *'x is nearer y than z'* due to van Benthem van Bethem (1983)

$$N(x, y, z)\ if\ and\ only\ if\ \kappa(x, y)\ \geq\ \kappa(z, y). \tag{11}$$

The distance between x and y is greater than the distance between z and y, i.e., in mereological terms, it is nearer from x to y than from z to y.

In terms of nearness, we define after van Benthem (loc.cit.) the relation of betweenness in the sense of van Benthem $VB(x, y, z)$ which reads *'x is between y and z'*

$$VB(x, y, z)\ if\ and\ only\ if\ for\ each\ w:\ either\ x = w\ or\ N(x, y, w)\ or\ N(x, z, w). \tag{12}$$

In plain wording: $VB(x, y, z)$ means that given any w, either $x = w$ or x is nearer to one of y, z than w. Betweenness in that sense can be applied to characterize mutual robot positions among robots in a team of robots. To that end, we assume that our robots are modeled as planar rectangles positioned regularly, i.e., with sides parallel to coordinate axes. For robots a, b, we define their *extent*, $ext(a, b)$, as the minimal rectangle containing a and b and positioned regularly. The main result in this section is the following

Theorem 1 *Given robots a, b modeled as planar disks enveloped in rectangular safety regions A and B, the extent $ext(A, B)$ is between A and B, i.e., the formula $VB(ext(A, B), A, B)$ is true.*

For completeness' sake, we recall the proof, see Polkowski (2011) Ch. 6.10.

Proof 1 *As linear stretching or contracting along an axis does not change the area relations, it is sufficient to consider two unit squares A, B of which A has (0,0) as one of vertices whereas B has (a,b) with $a, b > 1$ as the lower left vertex (both squares are regularly positioned). Then the distance κ between the extent $ext(A, B)$ and either of A, B is $\frac{1}{(a+)(b+1)}$.*

For a rectangle $R : [0, x] \times [0, y]$ with $x \in (a, a + 1), y \in (b, b + 1)$, we have that,

$$\kappa(R, A) = \frac{(x - a)(y - b)}{xy} = \kappa(R, B). \tag{13}$$

For $\phi(x, y) = \frac{(x-a)(y-b)}{xy}$, we find that

$$\frac{\partial \phi}{\partial x} = \frac{a}{x^2} \cdot (1 - \frac{b}{y}) > 0, \tag{14}$$

and, similarly, $\frac{\partial \phi}{\partial y} > 0$, i.e., ϕ is increasing in x, y reaching the maximum when R becomes the extent of A, B.

An analogous reasoning takes care of the case when R has some (c,d) with $c, d > 0$ as the lower left vertex.

We use this result by saying for robots a, b, c enveloped in safety regions A, B, C that the robot c is *r-between* robots a and b in case c is contained in the extent $ext(A, B)$.

For a team $T = \{R_1, R_2, ..., R_k\}$ of planar mobile robots, we say that they make a *formation* whenever they are endowed with a betweeness relation on T.

3. Formation control

Although it is impossible to draw a separating line between formerly discused topics and formation control, we find it convenient to separately mention a few works on the specific theme of formation control.

The aim of works in that area is to maintain formations by controlling relative position and orientation of robots in a team while moving the group as a whole Toibero *et al.* (2008). This subject involves assignment of feasible formations, moving into formation, maintaining a formation and changing formations.

Desai *et al.* (2001) represent formations by means of graphs with nodes representing robots and edges representing relations leader–follower. In their approach there is an unique "global" leader and many local leader–follower pairs corresponding to edges in the graph. The graph is represented by means of an adjacency matrix and transition from one formation to the other in this context requires a transition matrix with entries -1, +1, where the former value means that the edge in the graph is removed, the latter value means that the edge is added. Control laws express the state of the follower robot in terms of the state parameters of the leader robot, parameters being the linear and angular velocities of the axle center of each robot and state variables being the distance and angle between the robots.

Toibero *et al.* (2008) discuss the same problem of mobile formation control in presence of obstacles with objectives: (i) to place robots in their desired positions in the formation before staring the leader robot navigation; (ii) reduce large formation errors; (iii) avoid static obstacles while maintaining the given formation geometry. Authors propose a hybrid approach, in which the continuous formation controller is augmented with an orientation controller for each robot follower, along with the presence of switching supervisor responsible for indicating the active controller at a specific moment. For obstacle avoidance, the contour–following strategy by the leader robot is applied.

Lawton *et al.* (2003) propose three behavior–based strategies for control of a formation. In their approach, robot positions are estimated by a combination of dead reckoning and an overhead camera system. The behavior–based strategy applied is the coupled dynamics. A pattern is defined as the set of hand positions of robots, and the optimized criterion is the error of formation coordination. Coupled dynamics formation control is the behavior which compels the robot knowing relative positions and velocities of its two neighbors in communication ring to keep up with them. Coupled dynamics formation control with passivity–based interrobot damping is like the former except it does dispense with information about relative velocities, compensating the lack of those with additional damping factors to prevent robot oscillations. Saturation control variant adds upper bounds on the force and torque in dynamics equations of robots.

A vision–based formation control approach is proposed and presented in Das *et al.* (2002). The goal of this research is to develop a method for bottom–up buildup of a controller for formation control from simple controllers and estimators. The described idea resembles that of behavior–based architecture yet differs in more formal control–oriented approach. Simple controllers are responsible for obstacle avoidance, collision recovery, target pursuing, while complex controllers are responsible for formation maintenance. A sequential composition of controllers is aimed at maintaining or changing formations along a given trajectory. The main sensing ability is performed by an omnidirectional camera mounted on each robot. The paradigm implemented is that of the leader-followers; there is the lead robot whose movement determines the group movement and other robots refer to the leader for their positions and orientations. This is represented via a graph as in Desai *et al.* (2001). Leader–follower and leader–obstacle controls are provided to ensure local formation maintenance and obstacle avoidance by the follower with distance ensurance to the leader. Strategy for changing formations is guided by robot camera readings and the presence of the leader in their images.

A leaderless strategy for formation control based on the notion of a Virtual Structure is described in Lewis and Tan (1997). In defining a Virtual Structure, authors are motivated by a rigid body definition as a finite system of point masses any two of which are kept at a constant distance. Transferring this image to the robot world, authors call a Virtual Structure a system of stationary points with respect to a reference frame moving through space and such that distances among points are enforced by a control system. The idea poses itself: given a Virtual Structure, of n points, with position vectors in the reference frame p_i^R, $i = 1, 2, ..., n$, and a mapping $T_{R,W}$ from reference to world frame with $T_{R,W}(p_i^R) = p_i^W$ consider n robots with position vectors in the world reference frame r_i^W: $i = 1, 2, ..., n$; declare that robots are in a formation if at each point of the trajectory we have $r_i^W = T_{R,W}(p_i^R)$. In other words: robots mimic in a sense movement of the Virtual Structure being guided to respective points images in the world frame.

Recently, much attention is devoted to swarm robotics, i.e., large teams of robotic agents and some distinct approaches are proposed for the task of controlling robots in order, e.g., to keep them in prescribed patterns or formations. Oikawa *et al.* (2015) propose a mobile software agents system for keeping robots in a team in certain prescribed positions. Their approach is rooted in ant algorithms theory and applies a combination of acting by ant agents and pheromone agents. Xu *et al.* (2014) describe a behavior-based approach to robot control. Biological-based paradigms involved in studies of robotic swarms control involve membrane computing Paun (2010), seeFlorea and Buiu (2017).

We propose an approach based on ideas of rough mereology Polkowski (2011) based on the notion of a part to a degree. A rough mereological approach to formation buildup was originated in Polkowski and Osmialowski (2008), Polkowski and Osmialowski (2010) which is extended in this work in the following sections.

4. Testing in simulation

We performed testing using an open-source project called **RoboSim** RoboSim. It is a Python 3.6 based solution for emulating intelligent agent (including robotic) behavior in randomly generated environments - each map is created using the Random Walk Algorithm Revesz (1990). The library is using *pygame* PyGame to present realtime results, which were passed to the spatial reasoning algorithm that was based on Rough Mereology. An example of the approach can be seen in Fig. 1, where a group of robots in the romboid formation is traveling from the starting point to the goal through a narrow passage using the **First In** decision system as aggregated from images taken in **RoboSim**.

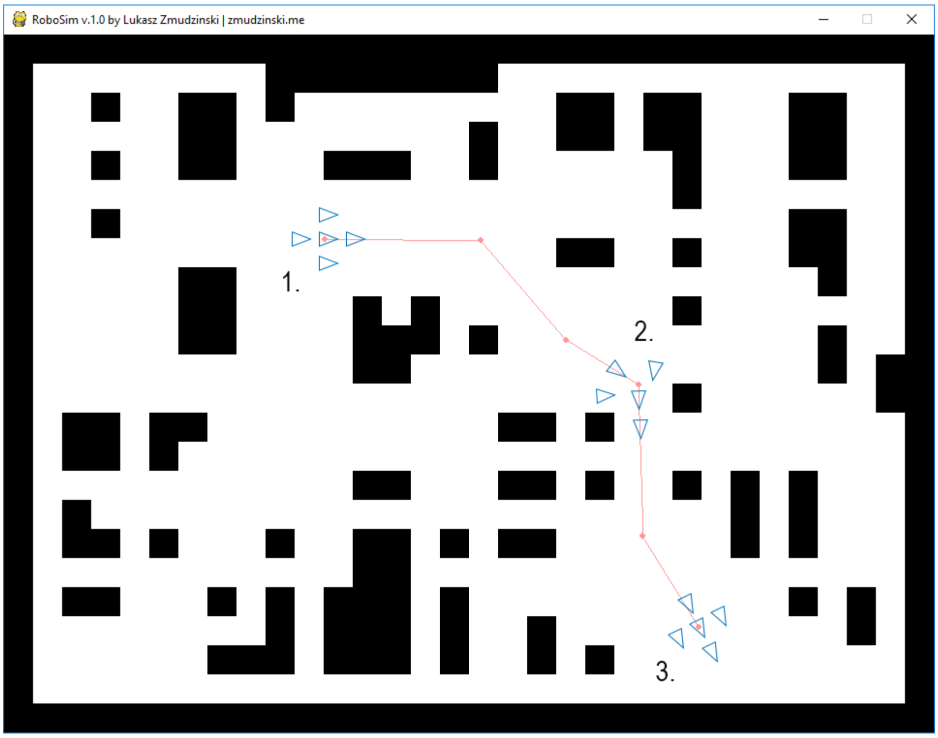

Fig. 1 Representation of a full path coverage by a team of intelligent agents that are using the rhomboid formation. The aggregated data is presented for three points of the robot journey: point 1 represents the starting position of the team, point 2 the First In decision system for narrow passages and finally point 3 is the target goal of the path algorithm

4.1 Path creation

The path was created using the provided Rough Mereology algorithm, see section 3.2. We have used the default version of the **Square Fill Algorithm** without later alterations and smoothing as seen in Fig. 2.

The path was connected with the central robot, which performed the role of the team leader. Other robots in the formation were following the main robot at specified distances between each team member (up to a degree of error). Using such approach gave us the possibility of performing less calculations. In case the main robot would be unavailable due to external reasons, a new robot would take command and calculate the path to the goal. Moreover, using the system in simulation allowed the authors to take shortcuts in probability distribution of incorrect sensor readings, which is present in current algorithm tests.

4.2 Robot formations

The formations we created were parameter based (metric), where each robot was checking the distance to another, designated robot in the formation. The data was taken from x and y positions of the intelligent agents in the simulation environment, updated every frame of the *pygame* engine. The

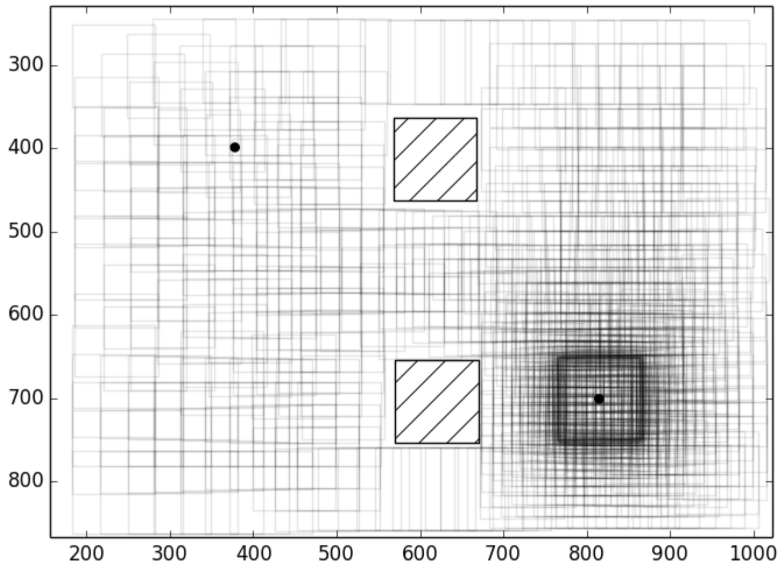

Fig. 2 Square Fill Algorithm potential field example with marked start and goal positions and two obstacles. Due to the fact, that the potential field is created as a gradient, you can observer that the closer to the goal, the more potential fields are present

data was used to calculate the $heading$ of the robot and preffered distances to other units. We have tested several robot formations as seen on Fig. 3:

- **Line formation** - with robots moving as a straight line, having the main unit placed in the middle of the line,
- **Rhomboid formation** - with robot team shaped as a rhombus, with the main unit placed in the center,
- **Snake formation** - immitating the movement of a snake, with the main unit placed in the "head",
- **Circle formation** - with robot team shaped as a circle, with the main unit placed in the center,
- **Cross formation** - shaped as a cross, with the main unit in the center.

4.2.1 Decision systems

Of course using such formations, required specified decision systems for places, where such formation wouldn't be possible eg. narrow passages. We have created three simple cases, to test how they would work in simulation:

- **First In** decision system, which was based on the distance of each robot to the tunnel entrance. If a robot was closer to the entrance, it went in first into the tunnel, with other robots following

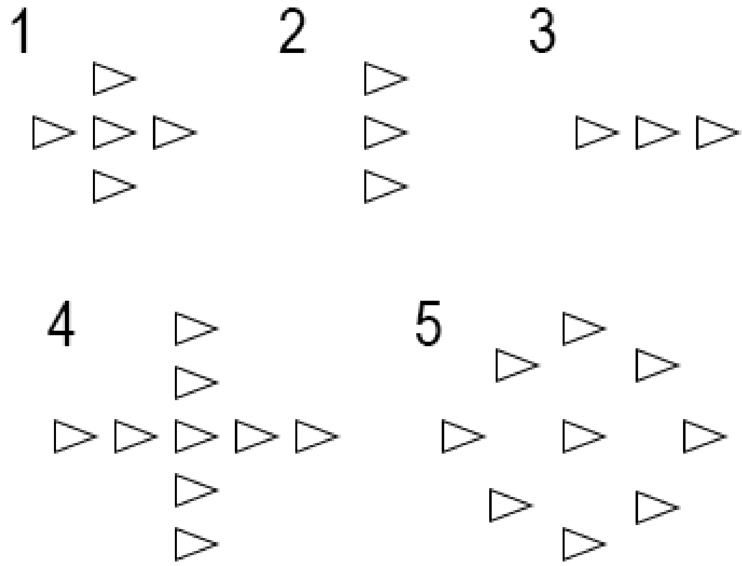

Fig. 3 Tested formations where, 1 stands for rhomboid, 2 for line, 3 for snake, 4 for cross and 5 for circle formation

in the same manner. The robot formation was changed to **Snake Formation** as they navigated to the end of the narrow passing,switching back to their original formation afterwards.

- **Leader First** decision system prioritized the leader robot. As all the other robots were already following the main unit, it was obvious to change the formation, with the same robot still remaining at lead. In testing it was denoted, that the solution wasn't efficient, because the amount of movement needed to bring the leader forward in more complicated formations was too high.
- **Horde Mode** decision system, which was based on swarms, where each robot tried to push itself first on the built path to the goal. This model was prone to error and some robots got stuck in some tunnels, which was presented as a bad outcome. In simpler tunnels, the decision system has proven better in two things: calculation time and amount of time needed to get through obstacles.

5. Conclusions

In this work we have presented the problem of path planning and navigation for teams of robots. Moreover, we have presented a working in simulation solution, using the state of the art Rough Mereology methods for calculating potential fields and robot formations that solve presented problems. We describe the domains of robot team control from literature with specified ”axes” taken from research task domains, from which the **Geometric Problem** suits the best for Rough Mereology spatial reasoning. Moreover, we describe the Rough Mereology theory that was used to build the rough mereology potential field, which is used for path building for single and multiple robots.

The proposed algorithms were tested using the Python 3.6 based RoboSim simulator with the following robot formations: line, rhomboid, snake, circle and cross. Moreover each team of robots had a built-in decision system, that was responsible for formation changes during transport through narrow passages. The authors created three decision systems, that consisted of **First In**, **Leader First** and **Horde Mode** variants. The experiments show, that some connections of formations – decision systems worked well, e.g., Rhomboid and Horde Mode, while others failed to deliver satisfying results like Cross formation with Leader First decision system.

In our future work, we plan on transferring the simulation code to mobile robots for navigating through populated university buildings, to test possible uses of the algorithms as guide robots for the University visitors.

References

Ahmed, A., Abdalla, T. and Abed A. (2015), “Path planning of mobile robot by using modified optimized potential field method”, *Int. J. Comput. Appl.*, **113**(4).

Alami, R., Robert, F., Ingrand, F. and Suzuki, S. (1995), “Multi-robot cooperation through incremental plan merging”, *Proceedings of the IEEE ICRA*, Nagoya, Japan, May.

Alonso-Mora, J., Montijano, E., Schwager, M. and Rus, D. (2016), “Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus”, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May.

Barraquand, J. and Latombe, J.C. (1990), “A Monte-Carlo algorithm for path planning with many degrees of freedom”, *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, U.S.A., May.

Brooks, R.A. (1991), “Intelligence without reason”, *Proceedings of the IJCAI*, Sydney, New South Wales, Australia, August.

Brumitt, B., Stentz, A. and Hebert, M. (2001), “CMU UGV group: Autonomous driving with concurrent goals and multiple vehicles: Mission planning and architecture”, *Autonom. Robot.*, **11**, 103-115.

Buckley, S.J. (1989), “Fast motion planning for multiple moving robots”, *Robot. Automat*., **1**, 322-326.

Caloud, P., Choi, W., Latombe, J.C., Pape, L.C. and Yin, M. (1990), “Indoor automation with many mobile robots”, *Proceedings of the IEEE/RSJ IROS*, Ibaraki, Japan, July.

Cao Uny, Y., Fukunaga, A.S. and Kahng, A.B. (1997), “Cooperative mobile robotics: Antecedents and directions”, *Autonom. Robot.*, **4**, 7-27.

Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J. and Taylor, C.J. (2002), “A vision-based formation control framework”, *IEEE Trans. Robot. Automat.*, **18**(5), 813-825.

Desai, J.P., Ostrowski, J.P. and Kumar, V. (2001), “Modeling and control of formations of nonholonomic mobile robots”, *IEEE Trans. Robot. Autom.*, **17**(6), 905-908.

Erdmann, M. and Lozano-Perez, T. (1986), “On multiple moving objects”, *Proceedings of the IEEE ICRA*.

Florea, A.G. and Buiu, C. (2017), *Membrane Computing for Distributed Control of Robotic Swarms: Emerging Research and Opportunities*, IGI Global, Hershey, Pennsylvania, U.S.A.

Gnys, P. (2017), *Mereogeometry Based Approach for Behavioral Robotics*, *Rough Sets*, IJCRS 2017, Lecture

Notes in Computer Science, Springer, Cham, **10314**.

Grossman, D.D. (1988), "Traffic control of multiple robot vehicles", *IEEE J. Robot. Automat.*, **4**(5), 491-497.

Hajek, P. (1998), *Metamathematics of Fuzzy Logic*, Kluwer, Dordrecht, the Netherlands.

Hopcroft, J., Schwartz, J.T. and Sharir, M. (1984), "On the complexity of motion planning for multiple independent objects: P-space hardness of the warehouseman's problem", *Int. J. Robot. Res.*, **3**(4), 76-88.

Hwang, Y. and Ahuja, N. (1992), "Gross motion planning-a survey", *ACM Comput. Surv.*, **24**(3), 219-291.

Kube, C.R. and Zhang, H. (1996), "The use of perceptual cues in multi-robot box-pushing", *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, U.S.A., April.

Latombe, J.C. (1991), *Robot Motion Planning*, Kluwer Acad. Publ., Boston, U.S.A.

Lawton, J.R.T., Beard, R.W. and Young, B.J. (2003), "A decentralized approach to formation maneuvers", *IEEE Trans. Robot. Autom.*, **19**(6), 933-941.

La Valle, S.M. and Hutchinson, S.A. (1999), "Multiple robot motion planning under independent objectives", *IEEE Trans. Robot. Autom*.

Leśniewski, S. (1916), *Foundations of General Set Theory (in Polish)*, Moscow, Russia, (1982), *See Also: On the Foundations of Mathematics*, Topoi 2, 7-5.

Lewis, M.A. and Tan, K.H. (1997), "High precision formation control of mobile robots using virtual structures", *Autonom. Robot.*, **4**(4), 387-403.

Liu, Y.H., Kuroda, S., Naniwa, T., Noborio, H. and Arimoto, S. (1989), "A practical algorithm for planning collision-free coordinated motion of multiple mobile robots", *Proceedings of the IEEE ICRA*, Scottdale, U.S.A., May.

Mataric, M. (1994), "Interaction and intelligent behavior", Ph.D. Dissertation, MIT, U.S.A.

McFarland, D. (1994), "Towards robot cooperation", *Proceedings of the Simulation of Adaptive Behavior*.

Mitchell, T. (1998), *Machine Intelligence*, Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A.

O'Donnell, P.A. and Lozano-Perez, T. (1989), "Deadlock-free and collision-free coordination of two robotic manipulators", *Proceedings of the IEEE ICRA*, Scottsdale, Arizona, U.S.A., May.

Oikawa, R., Tahimoto, M. and Kambayashi, Y. (2015), "Distributed formation control for swarm robots using mobile agents", *Proceedings of the 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*, Timisoara, Romania, May.

Orozco-Rosas, U., Montiel, O. and Sepulveda, R. (2015), "Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation", *Int. J. Adv. Robot. Syst.*, **12**(7), 81.

O'smiałowski, P. (2009), "On path planning for mobile robots: Introducing the mereological potential field method in the framework of mereological spatial reasoning", *J. Automat. Mob. Robot. Intellig. Syst.*, **3**(2), 24-33.

Ośmiałowski, P. (2010), "Spatial reasoning based on rough mereology in planning and navigation problems of mobile autonomous robots", Ph.D. Dissertation, Polish-Japanese Institute of IT, Warszawa, Poland.

Paun, G. (2010), "A quick introduction to membrane computing", *J. Log. Algebr. Program.*, **79**(6), 291-294.

Polkowski, L. (2011), *Approximate Reasoning by Parts. An Introduction to Rough Mereology*, IRSL Vol. 20, Springer-Verlag Berlin-Heidelberg.

Polkowski, L. and Ośmiałowski, P. (2008), *Spatial Reasoning with Applications to Mobile Robotics*, *Mobile Robots Motion Planning*, New Challenges, I-Tech Vienna, 433-453.

Polkowski, L. and Ośmiałowski, P. (2010), *Navigation for Mobile Autonomous Robots and Their Formations: An Application of Spatial Reasoning Induced from Rough Mereological Geometry*, *Mobile Robots Navigation*, In Tech Zagreb, 329-354.

Polkowski L. and Skowron, A. (1996), “Rough mereology: A new paradigm for approximate reasoning”, *Int. J. Approx. Reason.*, **15**(4), 333-365.

Polkowski, L. and Skowron, A. (1994), “Rough mereology”, *Proceedings of the ISMIS 1994*, LNCS, Springer-Verlag, Berlin, Germany.

Polkowski, L. (2017), *Rough Sets, Rough Mereology and Uncertainty*, *Thriving Rough Sets*, Stuśdies in Computational Intelligence, Springer, Cham.

Polkowski, L., Zmudzinski, L. and Artiemjew, P. (2018), “Robot navigation and path planning by means of rough mereology”, *Proceedings of the IEEE International Conference on Robotic Computing*, Laguna Hills, California, U.S.A., January-February.

PyGame Project <https://www.pygame.org/>.

Revesz, P. (1990), *Random Walk in Random and Non Random Environments*, World Scientific, Singapore.

Reynolds, C. (1992), “An evolved, vision-based behavioral model of coordinated group motion”, *Proceedings of the Simulation of Adaptive Behavior*, **2**, 384-392.

Reynolds, C. (1987), “Flocks, herds and schools. A distributed behavioral model”, *Comput. Graph.*, **21**(4), 25-34.

RoboSim Project <https://pypi.org/project/robosim/>.

Schwartz, J.T. and Sharir, M. (1983), “On the “piano movers” problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal obstacles”, *Int. J. Robot. Res.*, **2**(3), 46-75.

Švestka, P. and Overmars, M.H. (1998), “Coordinated path planning for multiple robots”, *Robot. Autonom. Syst.*, **23**(3), 125-152.

Tarski, A. (1959), *What is Elementary Geometry?*, *The Axiomatic Method with Special Reference to Geometry and Physics*, Studies in Logic and Foundations of Mathematics, North-Holland, Amsterdam, the Netherlands.

Toibero, J.M., Roberti, F., Carelli, R. and Fiorini, P. (2008), *Formation Control for Non-Holonomic Mobile Robots: A Hybrid Approach*, In: Lazinica, A., *Recent Advances in Multi-Robot Systems*, I-Tech Education and Publishing, Vienna, Austria.

Van Benthem, J. (1983), *The Logic of Time*, Reidel, Dordrecht, the Netherlands.

Vidal, T., Ghallab, M. and Alami, R. (1996), “Incremental mission allocation to a large team of robots”, *Proceedings IEEE International Conference on Robotics and Automation*, Minneapolis, U.S.A., April.

Wei, G., Xiaolu, W. and Xiaoping, Z. (2015), “Lane formation in pedestrian counterflows driven by a potential field considering following and avoidance behaviours”, *Phys. A: Statis. Mech. Its Appl.*, **432**, 87-101.

Werner, G. and Dyer, M. (1992), “Evolution of herding behavior in artificial animals”, *Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior*, **2**, 393.

Xu, D., Zhang, X., Zhu, Z., Chen, C. and Yang, P. (2014), “Behavior-based formation control of swarm robots”, *Math. Prob. Eng.*, **2014**, 13.

Żmudzinski, L. and Artiemjew, P. (2017), “Path planning based on potential fields from rough mereology”, *Proceedings of the International Joint Conference on Rough Sets*, Olsztyn, Poland, Lecture Notes in Computer Science (LNCS), **10314**, 158-168, Springer, Heidelberg, Germany.

HM