Chapter 8

Reasoning by Rough Mereology in Problems of Behavioral Robotics

In Ch. 6, we have developed basic notions and propositions of rough mereogeometry and rough mereotopology. We have stressed that by its nature, rough mereology does address collective concepts, relations among which are expressed by partial containment rendered as the predicate of a part to a degree. Behavioral robotics falls into this province, as usually robots as well as obstacles and other environmental objects are modeled as figures or solids. In this chapter, we discuss planning and navigation problems for mobile autonomous robots and their formations. In particular, we give a formal definition of a robot formation based on the betweenness relation, cf., Ch. 6., sect. 10. First, we introduce the subject of planning in robotics.

0.1 Planning of robot motion

Planning is concerned with setting a trajectory for a robot endowed with some sensing devices which allow it to perceive the environment in order to reach by the robot a goal in the environment at the same time bypassing obstacles.

Planning methods, cf., e.g., Choset et al. [17], vary depending on the robot abilities, features of the environment and chosen methodology. Among them are simple geometric methods designed for a robot endowed with sensors detecting obstacles, e.g., touch sensors or range sensors and able to detect distance between any pair of points. These methods are called ‘contour following’, as for such a robot, the idea can be implemented of moving to goal in a straight line segment and in case of meeting with an obstacle to bypass it by circumnavigating its boundary until the straight line to goal is encountered anew.

In this class belong so called ‘bug algorithms’ like BUG 1 algorithm due to Lumelsky and Stepanov [35], and its modification, the ‘tangent bug’ planner, Kamon et al. [27], cf., Choset et al. [17], in which the robot performs a heuristic search of A^* type, see, e.g., Russell and Norvig [52] or Choset et al. [17] with the heuristic function $h(x) = \rho(x, O) + \rho(O, goal)$ where x is the current position of the robot, and the point O is selected as an end–point of the continuity interval of ρ – the distance function, whose values are bound by a constant R.When the distance measured by range sensors exceeds R

the value of ρ is set to infinity. The graph of ρ against the position x exhibits then discontinuities and continuity intervals clearly outline boundaries of obstacles, hence, the idea of selecting O as a boundary continuity point. Minimization of h leads to optimization of the chosen safe trajectory.

To stay with geometric methods, we proceed with *Voronoï diagrams.* Most often, obstacles are modeled as two–dimensional polygons, and then the Voronoï diagram V is a 1–dimensional set consisting of points which are at equal distance from two closest to them obstacles, i.e., $x \in V$ if and only if $\rho(x, W_i) = \rho(x, W_j) \leq \rho(x, W_k)$ for $k \neq i, j$. Clearly, navigating the robot along V keeps it at safe distance from obstacles, hence, the planner based on the idea of the Voronoï diagram transports the robot from the starting point to the nearest point on V and then along V to the point in V nearest to the goal, see Choset et al. [17].

Another geometric idea is implemented in *visibility graphs*, see Latombe [32] and Li and Canny [34]. Vertices of obstacles, again, modeled as 2–dimensional polygons, constitute nodes of the graph. Two distinct nodes are joined by an edge if and only if two corresponding vertices can be connected by a straight line segment avoiding any contact with any of obstacles except for these two points. For particular start and goal points $start$, $goal$, these points are added to the graph as nodes and also connected by edges with other nodes when the straight line visibility condition is satisfied.

A further development is provided by *silhouette methods*, see Canny [15]. It exploits the *sweeping algorithm* which consists in moving the *sweeping line* of the form, e.g., $\{x_1\} \times R$ by increasing x_1; at each position, the end points of segments in the line obtained by intersection with obstacles constitute the *silhouette*, or, the *Canny roadmap.*

In some cases, not only the aim is to transport the robot from start to goal, but to cover all free space, e.g., when one wants the robot to paint the floor. One says in such cases of the *coverage problem.* Then, a method exploited is the *cell decomposition.* One represents the free space as a union of *cells* and builds on cells as nodes of a graph, the *adjacency graph* joining two nodes–cells with an edge if and only if the cells share a boundary.

There are various implementations of the idea, e.g., *trapezoidal decomposition*, see de Berg et al. [10]. This implementation looks at each vertex v of an obstacle for half–lines going 'up' and 'down' and registering points on them of intersection with obstacles or boundaries of working space. This defines a decomposition into cells.

As 'exact' planning algorithms are infeasible in some environments, see Schwartz and Sharir [54], Kavraki [28], or Canny [15] approximate planners based on probabilistic sampling were developed, see, e.g., Barraquand et al. [6], Kavraki et al. [29].

When many queries about paths are intended, then it is useful to build a probabilistic roadmap planner, see Kavraki et al. [29]. It is built on a standard planner, say P, which explores the possibility of a path between points in the working space and builds a metric on it. Nodes of the graph are configurations sampled from possible ones and roadmaps are built incrementally; any time a configuration, say c, is sampled, k nearest neighbors already sampled are selected and the planner P checks whether there is a safe path between c and any neighbor, adding an edge to the graph for each pair with a safe path.

Planning is necessarily coupled with *localization*, in order to plan one should know the robot position. To this effect, bayesian filtering, in particular Kalman filtering is applied, see Kalman [26], cf. Choset et al. [17].

A method referring to physical inspiration is the *potential field* method, see Khatib [30]. A potential field is composed of attractive potentials for goals and repulsive potentials for obstacles.

An example may be taken as the quadratic potential function

$$U_{attractive}(x) = \frac{1}{2} \cdot ||x - x_{goal}||^2 \tag{0.1}$$

which induces the gradient

$$\nabla U_{attractive}(x) = x - x_{goal} \tag{0.2}$$

which assures that the force (the gradient) exerted on the robot is greater when the robot is far from the goal and diminishes to zero as the robot is approaching the goal.

A repulsive potential should have opposite properties: it should exert a force tending to ∞ with the distance to the obstacle reaching 0. Denoting the distance from a point x to the closest obstacle with $s(x)$, the repulsive potential can be defined as in

$$U_{repulsive}(x) = \frac{1}{2} \cdot [\frac{1}{s(x)}] \tag{0.3}$$

with the gradient

$$\nabla U_{repulsive}(x) = -\frac{1}{s(x)^2} \cdot \nabla s(x) \tag{0.4}$$

The global potential function U is the sum of the attractive and repulsive parts:

$$U(x) = U_{attractive}(x) + U_{repulsive}(x)$$

Given U, the robot performs a well–known *gradient descent* : it does follow the direction of the gradient in small steps : the $(i+1)$–th position is given from the i-th position and the gradient therein as

$$x_{i+1} = x_i + \xi_i \cdot \nabla U(x_i) \tag{0.5}$$

Potential fields method suffers from the local minima problem immanent to gradient descent method: though the absolute minimum of 0 is achieved by the potential function at the goal, yet superposition of many fields from obstacles and goals induces local minima and saddle points typical to many–

dimensional landscape. A way out of local minima can be found by means of small random perturbations, see Barraquand et al. [6].

This closes our very brief glimpse at planning methods showing their variety. For a discussion of an underlying architecture, see Brooks [12]. One can ask, what rough mereology can add to this repertoire? We have proposed a new variant of potential field method, based on rough inclusion technique, see Ośsmiałowski [41] and Polkowski and Ośmiałowski [48], [49].

0.2 Potential fields from rough inclusions

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 refer to mereogeometry of Ch. 6, sect. 10, and we recall the rough inclusion

$$\mu(x, y, r) \Leftrightarrow \frac{||x \cap y||}{||x||} \tag{0.6}$$

where $||x||$ is the area of the region x. In our construction of the potential field, region will be squares: this corresponds with the robots used which are disc–shaped Roomba (a trademark of iRobot, Inc.) robots, so they can be represented by squares circumscribed on them.

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 Ośmiałowski [41], Polkowski and Ośmiałowski [49].

We construct the potential field by a discrete construction. The idea is to fill the free workspace of a robot with squares of fixed size in such a way that the density of the square field (measured, e.g., as the number of squares intersecting the disc of a given radius r centered at the target) increases toward the target.

To ensure this property, we fix a real number – the `field growth step` in the interval $(0, \text{square edge length})$; in our exemplary case the parameter `field growth step` is set to 0.01.

The collection of squares grows recursively with the distance from the target by adding to a given square in the $(k+1)--th$ step all squares obtained from it by translating it by $k \times$ `field growth step` (with respect to Euclidean distance) in basic eight directions: N, S, W, E, NE, NW, SE, SW (in the implementation of this idea, the *floodfill algorithm* with a queue has been used, see Ośmiałowski [41]. Once the square field is constructed, the path for a robot from a given starting point toward the target is searched for.

The idea of this search consists in finding a sequence of *way–points* which delineate the path to the target. Way–points are found recursively as centroids of unions of squares mereologically closest to the square of the recently found way–point. We recall, see Ch. 6, that the mereological distance between squares x, y is defined by means of

$$k(x,y) = min\{\max\{r,s\} : \mu(x,y,r), \mu(y,x,s)\} \tag{0.7}$$

We also remind that the mereological distance $k(x,y)$ takes on the value 1 when $x = y$ and the minimal value of 0 means that $x \cap y \subseteq Bd(x) \cap Bd(y)$. In order do define a "potential" of the rough mereological field, let us consider how many generations of squares will be centered within the distance r from the target. Clearly, we have

$$d + 2d + ... + kd \leq r \tag{0.8}$$

where d is the field growth step, k is the number of generations. Hence,

$$k^2 d \leq \frac{k(k+1)}{2} d \leq r \tag{0.9}$$

and thus

$$k \leq (\frac{r}{d})^{\frac{1}{2}} \tag{0.10}$$

The potential $V(r)$ can be taken as $\sim r^{\frac{1}{2}}$. The force field $F(r)$ is the negative gradient of $V(r)$,

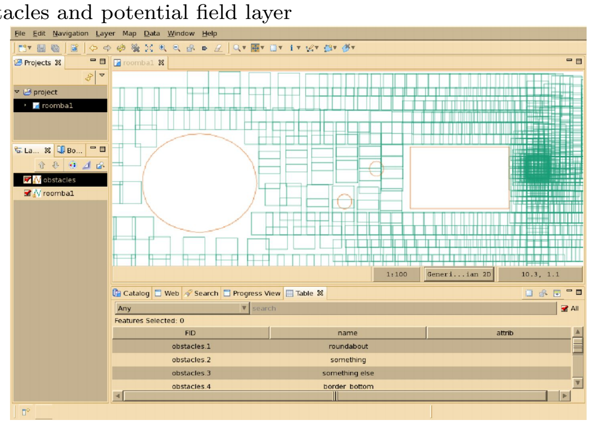
$$F(r) = -\frac{d}{dr}V(r) \sim -\frac{1}{r^{\frac{1}{2}}} \tag{0.11}$$

Hence, the force decreases with the distance r from the target slower than traditional Coulomb force. It has advantages of slowing the robot down when it is closing on the target. Parameters of this procedure are: the `field growth step` set to 0.01, and the size of squares which in our case is 1.5 times the diameter of the Roomba robot.

The path planner designed in this way, accepts target point coordinates and provides list of way–points from given robot position to the goal. To do its job, it needs a map of static obstacles that a robot should avoid while approaching target point. A robot and a target should both lay within the area delimited by surrounding static obstacles that form borders of robot workspace. There can be other static obstacles within the area, all marked on the provided map. After the path is proposed a robot is lead through the path until it reaches given target. If a robot cannot move towards goal position for some longer time (e.g., it keeps on hitting other robot reaching its target or any other unknown non–static obstacle), new path is proposed.

We tested our planner device running simulations in which we have had a model of Roomba robot traveling inside artificial workspace. Real Roomba robots are round and therefore easy to model, however they do not provide many useful sensor devices (except bumpers which we were using to implement lower–level reaction for hitting unexpected obstacles). Also odometry of Roomba robots is unreliable, Tribelhorn and Dodds [59], hence, we assume that simulated robots are equipped with a global positioning system. A map of an environment as used in simulations along with a potential field generated for a given goal is is shown in Fig. 0.1.

Fig. 0.1 Obstacles and potential field layer



A robot should follow the path proposed by planner by going from one area centroid to another until the goal is reached. The proposed path is marked on the map, see Fig. 0.2, 0.2.

Fig. 0.2 Stage simulator: iRobot Roomba robots starting to a goal

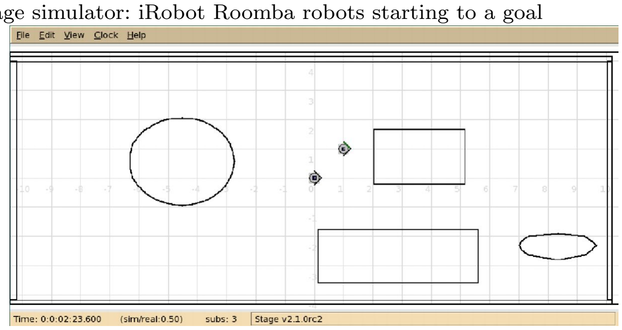
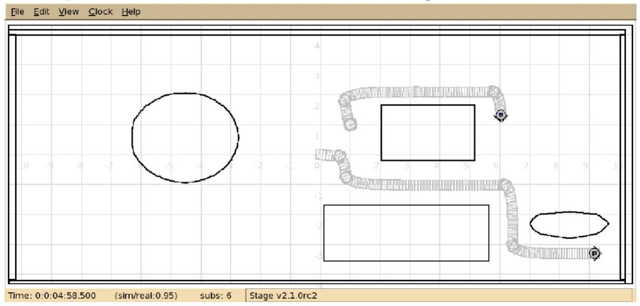


Fig. 0.3 Planned paths of Roomba robots to their targets



To perform re–planning of the path, usually the planner should repeat the planning routine. Using our method, only second stage of planning routine is done during replanning as potential field is computed only once (unless the database is updated with new obstacles). Searching for a path within already computed potential field is computationally cheap as it is limited to database lookup operations (therefore speed of database communication is critical if this method is intended to be working fast).

0.3 Planning for teams of robots

Both theoretical interests, see Walter [61], [62] as well as practical motivations, have driven the attention of researchers in robotics toward problems related to teams of robots. From purely intellectual point of view, this opens a new venue for solving problems of cooperation, communication, task–sharing and division, and planning non–collision paths for robots.

According to Cao et al. [16], robot teams can provide a useful playground for studies of cognitive theories, biology, ethology, organization and management. They can also lead to new solutions to problems of artificial intelligence. Passing from a single robot to teams of robots can be motivated also by pragmatic reasons, Cao et al. [16], as tasks for robots can be too complex for a single robot, or many robots can do the task easier at a lesser cost, or many robots can perform the task more reliably. Practical studies along these lines were concerned with moving large objects of irregular shapes by groups of robots, see Kube and Zhang [31], search and rescue, see Jennings et al. [25], formations of planetary outposts of mobile robots, see Huntsberger et al. [23]. Simulations of systems a few robots were studied, e.g., in CEBOT, see Fukuda and Nakagawa [21], ACTRESS, see Asama et al. [3], GOFER, see Caloud et al. [14], cf., the ALLIANCE architecture in Parker [46].

In Cao et al. [16], main research directions in this area were systematized, among them, Geometric Problems, involving multiple–robot path planning, formation maintaining, moving to formations, marching, pattern generation. To be more specific, 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. The choice here is between prioritized planning which takes one robot at a time according to some priority scheme, see Erdmann and Lozano–Perez [19] and path coordination method which plans paths by scheduling configuration space resources. Problems of cooperation and negotiations are discussed in Naffin and Suthname [39] and Parker [45].

The Formation Problem as well as Marching Problem require of robots to move into a prescribed formation and march to the goal maintaining the formation. Various have been solutions adopted for these problems. A study of the concept of a robot team was initially based on a perception of animal behavior like herding, swarming, flocking or schooling. Work by Reynolds [51] brought forth an approach to flocking in groups of birds called 'boids' based on simple behaviors like: collision–avoidance, velocity matching, flock centering, where each bird senses its neighbors only, and geometric positions of birds are not specified. A similar position was adopted in Mataric [36], [37] who studied flocking in wheeled robots induced by simple behaviors: wandering, homing, following, avoidance, aggregation, dispersion, cf., [20], [38]. See Wilson [63] for a deep study in sociobiology and Agah [1], Agah and Bekey [2], and Bekey [8] for robotic counterparts.

These observations brought forth some basic principles of *behavioral approach*, see, e.g., Balch and Arkin [5]: it is vital to keep all robots within a certain distance from one another (e.g., to ensure mutual visibility), to move away when the distance becomes too close (to avoid congestion, collision, or resource conflict), to adapt own movement to movement of neighbors (e.g., by adjusting velocity of motion), to orient oneself on a leader, or a specific location, e.g., the gravity center of the group.

Balch and Arkin, op.cit., implement these principles by imposing geometric constraints on individual robots: each of them is to keep a precisely specified geometric position. They investigate four types of formations: *line* where robots move line–abreast, *column* where robots travel in an *Indian file*, *diamond*, *wedge*. Each robot in a formation is given its ID number. Maintaining position in a formation is provided by two processes: *detect–formation–position* detects the current position of a robot on basis of sensory data, *maintain–formation* produces motor commands to move robot to the proper position. They evaluate three *referencing techniques* for computing a proper position of a robot in the formation: *unit–center–reference*, *leader–reference*, *neighbor–reference*.

In *unit–center–reference* each robot computes the coordinates of the center of the formation and keeps position with respect to it.

In *leader–referencing*, one robot is designated as the leader whose position is irrelevant and other robots are keeping the formation.

In *neighbor–reference* each robot keeps position with respect to one specified neighbor. The local coordinate system in which positions are computed is at each step given by the unit center and the line from it through the next navigation way–point. The parameter *spacing* of the procedure *detect–formation–position* is used to keep distances between robots.

This scheme sets demands on sensory and motor capabilities of robots. Authors propose using dead reckoning, or GPS, or direct perception of neighbors as means of sensory determination of a position in the formation. One can say that Balch and Arkin, op. cit., are proponents of the *geometric approach* to formations: it uses *referencing* techniques; reference is made either to the team center or to the team leader, or to a specified neighbor in a coordinate system given by the position of the team center or the leader along with the orientation given by the nearest navigation point; positions are determined, e.g., with the help of GPS or dead reckoning. Spacing between robots is imposed and robots keep their geometric positions. In this approach, complex behaviors of a team of robots result from an interaction of primitive behaviors organized by sensory and motor components of robots. Dorigo et al. [18] situate themselves with their approach in the realm of self–organization and swarm intelligence, see Bonabeau et al. [11], Payton et al. [47], and evolutionary computing, see Baeck et al. [4]. Authors study s–bots: robots of limited capabilities, with respect to their coordinated behaviors. S–bots have ability to connect one to another by physical links therefore making rigid formations and the objective of the authors is to study coordinated movement of robots in a formation. The method of evaluating performance of the complex controller is by a genetic process. Coordinated behavior of a team results as a complex product of individual behaviors.The important issue raised in this work is stressing the importance of ability of a formation to change shape or to adapt to changing environment and pointing to the problem of dependence of complex behaviors of a team on the set of individual behaviors of robots in that team. Current status of the field of self–configurable robots, a variation on the theme of coupled formations can be found in Stoy et al. [57].

The potential field methodology has also been extended to teams of robots. A good example of this approach is provided in Leonard and Fiorelli [33] which combines potential field approach with the virtual leader–reference approach. Potentials define interaction forces among robots forcing them to keep at desired distance one 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; control forces are then defined as negative gradients of the sum of potentials affecting a given robot. By their action, robots are driven to the absolute minimum of the global potential function; local potentials can be designed as to correspond to a given geometry of the group.

In addition to local neighbor–pair potentials, virtual leaders, i.e., moving reference beacons are added, each of them generating its own potential field with the aim of manipulating the group, directing it or herding robots into a group. Authors discuss motions like schooling and flocking. Schooling is a maneuver in which a steady group translation occurs, and flocking takes place when robots circle a stationary point. For instance, authors demonstrate a stationary movement of a group of 6 robots forming vertices of a hexagonal lattice.

An approach to controlling of a group of robots using the leader idea is presented in Shao et al. [55]. The leader–follower paradigm means that each robot in a group has a neighbor assigned as its leader whom it follows with prescribed distance and eventual other parameter values. For a group on N robots, authors propose to express the group structure in the form of a tree in which pairs of the form parent–child are pairs the leader, the follower, encoded in the usual form of an adjacency matrix; another matrix is the parameter matrix: authors discuss four parameters for each robot: distance, orientation error, angle and the Boolean attribute Presence meaning visibility of the leader. Authors show some patterns: a *hexagon*, a *diamond* of twelve robots, a *column* (line), a *wedge*. An interesting fact is that authors raise the problem of changing patterns.

Another method for forming a geometric formation relies on a direct usage of a metric, see, e.g., Sugihara and Suzuki [58]: given a threshold δ, and a parameter D – the circle diameter, for each robot M in a team, its farthest neighbor M_1 and the nearest neighbor M_2

1. *If* $\rho(M, M_1) > D$, *then* M *moves toward* M_1;

2. *If* $\rho(M, M_1) < D - \delta$, *then* M *moves away from* M_1.

   These two steps assure that the diameter of the set of robots in each cross-section is about D. Finally

3. *If* $D - \delta < \rho(M, M_1) < D$, *then* M *moves away from* M_2.

By this method, robots are arranged on an approximation to a circle of diameter D. This procedure is performed iteratively and in each iteration robots move sequentially.

In Schneider et al. [53] authors attempt a systematic discussion of metrics for formation navigation. They use the term ‘metric’ in order to denote some criterion of performance evaluation for a group of robots. Authors assume a group of identical robots, communicating among themselves, with ability to sense the environment and one another. Among some metrics of that type authors mention

1. *Path length ratio: the ratio of the average path length by robots in a group to the straight–line distance to the goal*;
2. *Average position error: average displacement from the correct position in the group during the run*;
3. *Percentage of time out of formation*;
4. *As an additional measure, time to convergence, i.e., time needed for the group to assume a given pattern is added.*

Authors define a formation of robots in strict geometric terms, characterizing a formation by means of a finite set of segments and angles between them, such that

1. *Uniform dispersion is secured: all neighboring robots keep the same distance d with maximum error ε*;
2. *Shape is proper: each robot keeps its position within an error ε*;
3. *Orientation: the angles are kept within the error ε_a.*

In order to keep a formation, potential fields are used. In addition to the goal potential and the attractive potentials of obstacles, each robot in a group exerts attractive and repulsive forces on other robots. All these potentials sum up for each robot inducing the directing forces.

Planning paths for multiple robots adapts and modifies planing methodology for a single robot, see Hwang and Ahuja [24] or Latombe [32] for surveys. The *centralized* approach, finding a path in a complex configuration space describing the 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, see Hopcroft et al. [22]. Schwartz and Sharir [54] describe planners of polynomial complexity based on cell decomposition for disc shaped robots in polygonal obstacle world. Variants of the method of potential field, were applied as well in centralized planning, see Barraquand and Latombe [7]; they proposed a randomized path planner based on a a potential field induced by goals with random fluctuations for escaping local minima. 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 in Erdmann and Lozano–Perez [19].

High complexity of centralized planning and incompleteness of decoupled planning prompted research in the area between the two and the idea of separate *roadmaps* for robots emerged in which separate roadmaps are combined into a global roadmap, see La Valle and Hutchinson [60]. A general problem of mission planning for multiple vehicles and concurrent goals is addressed in Brumitt et al. [13] 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*:

1. $m \to M(r, g)$—$m \wedge m$—$m \vee m$—$m \Rightarrow m$—(m);

2. $r \to R_i$—$r \wedge r$—$r \vee r$—(r);

3. $g \to G_j$—$g \wedge g$—$g \vee g$—$g \Rightarrow g$—(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. These simple grammar expressions are examined by the mission planner and parsed into sequences of executable commands and planning of paths for them uses the D^* search algorithm, see Choset et al. [17].

So now again the recurrent question: what rough mereology can do for robot teams in terms of planning and navigation? We may observe that one can hardly find in literature a formal definition of what a robot formation is, independent of the context, e.g., a metric. Rather, formations are defined by setting constraints on individual robot either absolute or relative to a leader, or a neighbor. A definition absolute in a sense, abstracted from metric context, can be useful. Also, rigidity of constraints, e.g., necessity of keeping distances and angles, rids the team of flexibility, necessary when, e.g., obstacles force the team to change the formation in order to pass, e.g., a bottleneck. Thus, we strive for a definition and conditions for a formations which on one hand would secure its maintenance through manoeuvering to goal, and, on the other hand would permit a flexible behavior, e.g., bypassing an obstacle whereas keeping the formation. We propose a solution based on mereogeometry of Ch. 6.

0.4 Rough mereological approach to robot formations

We again resort to mereogeometry, see Ch. 6, sect. 10, Ośsmiałowski [41], and Polkowski and Ośsmiałowski [48], [49]. We recall that on the basis of the rough inclusion μ, and mereological distance κ defined as

$$\kappa(X,Y) = min\{max\ r, max\ s:\ \mu(X,Y,r), \mu(Y,X,s)\} \quad (0.12)$$

geometric predicates of *nearness* and *betweenness*, see Ch. 6, sect. 10, are redefined in the mereological frame.

The relation N of *nearness* proposed by Van Benthem [9] is defined in mereological context as

$$N(X,U,V) \text{ if and only if } \kappa(X,U) > \kappa(V,U) \quad (0.13)$$

Here, $N(X,U,V)$ means that X is closer to U than V is to U.

The *betweenness* relation T_B, see Van Benthem [9], is defined as

$$T_B(Z,U,V) \text{ if and only if [for each } W\ (Z=W) \text{ or } N(Z,U,W) \text{ or } N(Z,V,W)] \quad (0.14)$$

The principal example bearing on our approach to robot control deals with rectangles in 2D space regularly positioned, i.e., having edges parallel to coordinate axes. We model robots (which are represented in the plane as discs of the same radius in 2D space) by means of their safety regions about robots; those regions are modeled as rectangles circumscribed on robots. One of advantages of this representation is that safety regions can be always implemented as regularly positioned rectangles circumscribed on discs representing robots.

Given two robots a, b as discs of same radii, and their safety regions as circumscribed regularly positioned rectangles A, B, we search for a proper choice of a region X containing A, and B with the property that a robot C contained in X can be said to be between A and B. For two (possibly but not necessarily) disjoint rectangles A, B, we define the *extent*, $ext(A,B)$ of A and B as the smallest rectangle containing the union $A \cup B$. Then we have the claim, obviously true by definition of T_B, see Ch. 6, sect. 10

Proposition 1. *We consider a context in which objects are rectangles positioned regularly, i.e., having edges parallel to axes in R^2. The measure μ is μ^G, see Ch. 6, sect. 6. In this setting, given two disjoint rectangles C, D, the only object between C and D in the sense of the predicate T_B is the* extent *$ext(C,D)$ of C, D, , i.e., the minimal rectangle containing the union $C \cup D$.*

For details of the exposition which we give now, please consult Ośmiałowski [42], [43], Polkowski and Ośmiałowski [49], Ośmiałowski and Polkowski [44]. The notion of betweenness along with Proposition 1 permits to define the notion of betweenness for robots. Recall that we represent the disc–shaped Roomba robots by means of safety squares around them, regularly placed, i.e., with sides parallel to coordinate axes.

For robots a, b, c, we say that a robot b is *between robots a and c*, in symbols

$$(\text{between } b\ a\ c) \quad (0.15)$$

in case the rectangle $ext(b)$ is contained in the extent of rectangles $ext(a)$, $ext(c)$, i.e.

$$\mu_0(ext(b), ext(ext(a), ext(c)), 1) \tag{0.16}$$

i.e., see Ch. 6, sect. 10, $ext(b) \subseteq ext(ext(a), ext(b))$.

This allows as well for a generalization of the notion of betweenness to the notion of *partial betweenness* which models in a more realistic manner spatial relations among a, b, c; we say in this case that robot b is *between robots a and c to a degree of at least r*, in symbols,

$$(\text{between–deg}r\ b\ a\ c\ ) \tag{0.17}$$

if and only if

$$\mu_0(ext(b), ext[ext(a), ext(c)], r) \tag{0.18}$$

i.e., $\frac{||ext(b) \cap ext(ext(a), ext(c))||}{||ext(b)||} \geq r$.

For a team of robots, $T(r_1, r_2, ..., r_n) = \{r_1, r_2, ..., r_n\}$, an *ideal formation IF* on $T(r_1, r_2, ..., r_n)$ is a betweenness relation (between...) on the set $T(r_1, r_2, ..., r_n)$ of robots.

In implementations, ideal formations are represented as lists of expressions of the form

$$(\text{between } r_0\ r_1\ r_2) \tag{0.19}$$

indicating that the object r_0 is between r_1, r_2, for all such triples, along with a list of expressions of the form

$$(\text{not–between } r_0\ r_1\ r_2) \tag{0.20}$$

indicating triples which are not in the given betweenness relation.

To account for dynamic nature of the real world, in which due to sensory perception inadequacies, dynamic nature of the environment etc., we allow for some deviations from ideal formations by allowing that the robot which is between two neighbors can be between them to a degree in the sense of (0.17). This leads to the notion of a real formation.

For a team of robots, $T(r_1, r_2, ..., r_n) = \{r_1, r_2, ..., r_n\}$, a *real formation RF* on $T(r_1, r_2, ..., r_n)$ is a betweenness to degree relation (between–deg ....) on the set $T(r_1, r_2, ..., r_n)$ of robots.

In practice, real formations will be given as a list of expressions of the form,

$$(\text{between–deg } \delta\ r_0\ r_1\ r_2), \tag{0.21}$$

indicating that the object r_0 is to degree of δ in the extent of r_1, r_2, for all triples in the relation (between–deg ....), along with a list of expressions of the form,

$$(\text{not–between } r_0\ r_1\ r_2), \tag{0.22}$$

indicating triples which are not in the given betweenness relation.

Description of formations, as proposed above, can be a list of relation instances of large cardinality, cf., examples below. The problem can be posed of finding a minimal set of instances sufficient for describing a given formation, i.e., implying the full list of instances of the relation (between...). This problem turns out to be NP–hard, see Ośmiałowski and Polkowski [44].

Proposition 2. *The problem of finding a minimal description of a formation is NP–hard.*

Proof. (Ośmiałowski and Polkowski [44]) We construct an *information system*, see Ch. 4, *Formations* as a triple (U, A, f) where U is a set of objects, A is a set of attributes and f is a value assignment, i.e., a mapping $f : A \times U \to V$, where V is a set of possible values of attributes in A on objects in U. For a formation F, with robots $r_1, ..., r_n$ we let $U = T(r_1, ..., r_n)$, a team of robots; $A = \{[r_k, r_l, r_m] : r_k, r_l, r_m \text{ pairwise distinct robots}\}$. For a given formation F of robots $r_1, ..., r_n$, the value assignment f is defined as follows,

$$f([r_k, r_l, r_m], r_i) = \begin{cases} 1 \text{ in case } r_i = r_l \text{ and (between } r_l\ r_k\ r_l) \\ \frac{1}{2} \text{ in case } r_i = r_l \textit{ or } r_i = r_m \text{ and (between } r_l\ r_k\ r_m) \\ 0 \text{ in case } r_i \ \neq r_l\, r_k\, r_m \end{cases} \quad (0.23)$$

The system $Formations$ describes the formation F.

Clearly, reducts of the system $Formations$ provide a complete description of the formation F and correspond to minimal descriptions of the formation. As shown by Skowron and Rauszer [56] the problem of finding a minimum size reduct of a given information system is NP–hard □

To describe formations we have proposed a language derived from LISP–like s–expressions: a formation is a list in LISP meaning with some restrictions that formulates our language. We will call elements of the list the objects. Typically, LISP lists are hierarchical structures that can be traversed using recursive algorithms. We restrict that top–level list (a root of whole structure) contains only two elements where the first element is always a formation identifier (a name). For instance

Example 1. (formation1 (some_predicate $param1$ ... $paramN$))

For each object on a list (and for a formation as a whole) an extent can be derived and in facts, in most cases only extents of those objects are considered. We have defined two possible types of objects

1. *Identifier: robot or formation name (where formation name can only occur at top–level list as the first element)*;

2. *Predicate: a list in LISP meaning where first element is the name of given predicate and other elements are parameters; number and types of parameters depend on given predicate.*

Minimal formation should contain at least one robot. For example

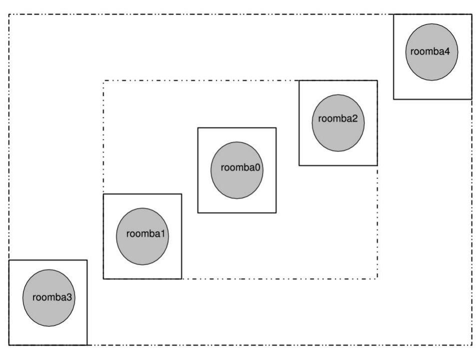
Example 2. (formation2 roomba0)

To help understand how predicates are evaluated, we need to explain how extents are used for computing relations between objects. Suppose we have three robots ($roomba0$, $roomba1$, $roomba2$) with $roomba0$ between $roomba1$ and $roomba2$ (so the *between* predicate is fulfilled). We can draw an extent of this situation as the smallest rectangle containing the union $roomba1 \cup roomba2$ oriented as a regular rectangle, i.e., with edges parallel to coordinate axes. This extent can be embedded into bigger structure: it can be treated as an object that can be given as a parameter to predicate of higher level in the list hierarchy. For example:

Example 3. (formation3 (between (between roomba0 roomba1 roomba2) roomba3 roomba4))

We can easily find more than one situation of robots that fulfill this example description. That is one of the features of our approach: one s–expression can describe many situations. This however makes very hard to find minimal s–expression that would describe already given arrangement of robots formation (as stated earlier in this chapter , the problem is NP–hard). An exemplary s–description is shown in Fig. 0.4.

Fig. 0.4 Formation described by an s–expression: *(formation3 (between (between roomba0 roomba1 roomba2) roomba3 roomba4))*



Typical formation description may look like below, see Ośmiałowski [42], [43], Polkowski and Ośmiałowski [49], [50]

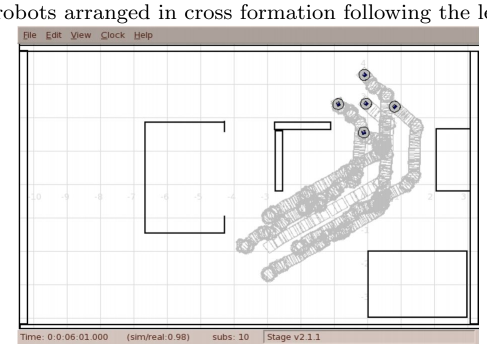
Example 4. (cross
    (set

```
            (max–dist 0.25 roomba0 (between roomba0 roomba1 roomba2))
            (max–dist 0.25 roomba0 (between roomba0 roomba3 roomba4))
            (not–between roomba1 roomba3 roomba4)
            (not–between roomba2 roomba3 roomba4)
            (not–between roomba3 roomba1 roomba2)
            (not–between roomba4 roomba1 roomba2)
        )
    )
```

This is a description of a formation of five Roomba robots arranged in a cross shape. The *max–dist* relation is used to bound formation in space by keeping all robots close one to another.

The final stage of planning is in checking its soundness by navigating robots in an environment with obstacles. We show results of navigating with a team of robots in the initial formation of cross–shape in a crowded environment, see Fig. 0.5. In order to bypass a narrow avenue between an obstacle and the border of the environment, the formation changes to a line, see Fig. 0.6, see Ośmiałowski [40], [43].

Fig. 0.5 Trails of robots arranged in cross formation following the leader



After the line was formed and robots passed through the passage, the line formation can be restored to the initial cross–shaped formation, see Figs. 0.7, 0.8. These behaviors witness the flexibility of our definition of a robot formation: first, robots can change formation, next, as the definition of a formation is relational, without metric constraints on robots,the formation can manage an obstacle without losing the prescribed formation (though, this feature is not illustrated in figures in this chapter).

Fig. 0.6 Trails of robots moving to their positions in the line formation

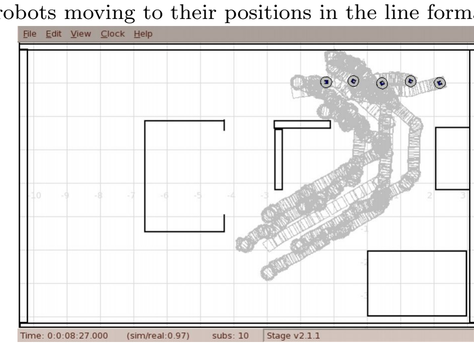


Fig. 0.7 Trails of robots moving in the line formation through the passage

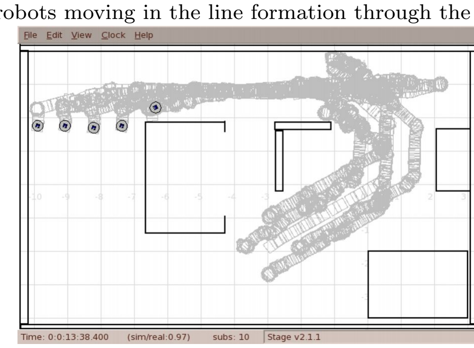
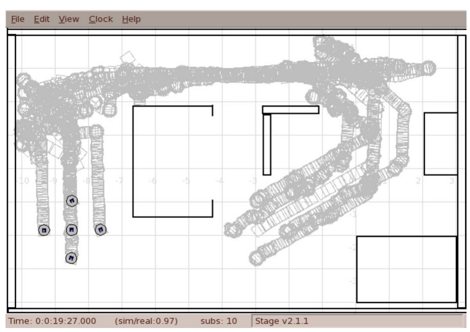


Fig. 0.8 Trails of robots in the restored cross formation in the free workspace after passing through the passage



References

1. Agah A. (1996) Robot teams, human workgroups and animal sociobiology. A review of research on natural and artificial multi–agent autonomous systems. Advanced Robotics 10, pp 523–545
2. Agah A., Bekey G. A. (1997) Tropism based cognition: A novel software architecture for agents in colonies. Journal of Experimental and Theoretical Artificial Intelligence 9(2-3), pp 393–404
3. Asama H., Matsumoto A., Ishida Y. (1989) Design of an autonomous and distributed robot system: ACTRESS. In: Proceedings IEEE/RSJ IROS, pp 283–290
4. Baeck T., Fogel D., Michalewicz Z. (1997) Evolutionary Computation 1: Basic algorithms and operators. Taylor and Francis, New York
5. Balch T., Arkin R. C. (1998) Behavior–based formation control for multiagent robot teams. IEEE Transactions on Robotics and Automation 14(12)
6. Barraquand J., Kavraki L.E., Latombe J.–C., Li T.–Y., Motvani R., Raghavan P. (1997) A random sampling scheme for robot path planning. Intern. J. Robotic Research 16(6), pp 759–774
7. Barraquand J., Latombe J. C. (1990) A Monte–Carlo algorithm for path planning with many degrees of freedom. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp 1712–1717
8. Bekey G. A. (2005) Autonomous Robots. From Biological Inspiration to Implementation and Control. MIT Press, Cambridge, MA
9. Van Benthem J. (1983) The Logic of Time. D. Reidel, Dordrecht
10. de Berg M., van Kreveld M., Overmars M. (1997) Computational Geometry: Algorithms and Applications. Springer Verlag, Berlin
11. Bonabeau E., Dorigo M., Theraulaz G. (1999) Swarm Intelligence. From Natural to Artificial Systems. Oxford University Press, Oxford, UK
12. Brooks R. (1999) Cambrian Intelligence: The Early History of the New AI. MIT Press, Cambridge, MA
13. Brumitt B., Stentz A., Hebert M. (2001) The CMU UGV Group. Autonomous driving with concurrent goals and multiple vehicles: Mission planning and architecture. Autonomous Robots 11(2), pp 103–115
14. Caloud P., Choi W., Latombe J.-C., Pape Le C., Yin M. (1990) Indoor automation with many mobile robots. In: Proceedings IEEE/RSJ IROS, pp 67–72
15. Canny J. F.(1988) The Complexity of Robot Motion Planning. MIT Press, Cambridge, MA
16. Cao Y. Uny, Fukunaga A. S., Kahng A. B. (1997) Cooperative mobile robotics: Antecedents and directions. Autonomous Robots 4, pp 7–27
17. Choset H., Lynch K. M., Hutchinson S., Kantor G., Burgard W., Kavraki L. E., Thrun S. (2005) Principles of Robot Motion. Theory, Algorithms, and Implementations. MIT Press, Cambridge, MA

18. Dorigo M., Trianni V., Sahin E., Gross R., Labella Th., Baldassare G., Nolfi S., Deneubourg J.–L., Mondada F., Floreano D., Gambardella L. (2004) Evolving self–organizing behaviors for a Swarm–Bot. Autonomous Robots 17, pp 223–245
19. Erdmann M., Lozano–Perez T. (1986) On multiple moving objects. In: Proceedings IEEE ICRA, pp 1419–1424
20. Fredslund J., Matarić M. (2002) A general algorithm for robot formation using local sensing and minimal communication. IEEE Transactions on Robotics and Automation 18(5), pp 837–846
21. Fukuda T., Nakagawa S. (1987) A dynamically reconfigurable robotic system (concept of a system and optimal configurations). In: International Conference on Industrial Electronics, Control, and Instrumentation, pp 588–595
22. Hopcroft J., Schwartz J. T., Sharir M. (1984) On the complexity of motion planning for multiple independent objects: P–space hardness of the warehouseman's problem. International Journal of Robotic Research 3(4), pp 76–88
23. Huntsberger T., Stroupe A., Aghazarian H., Garrett M., Younse P., Powell M. (2007) TRESSA: Teamed robots for exploration and science of steep areas. Journal of Field Robotics 24 (11/12), pp 1015–1031
24. Hwang Y., Ahuja N. (1992) Gross motion planning - A survey. ACM Computing Surv. 24(3), pp 219–291
25. Jennings J.S., Whelan G., Evans W.F. (2001) Cooperative search and rescue with a team of mobile robots. In: SPIE 4364, Aerosense, Orlando, Fla
26. Kalman R. (1960) A new approach to linear filtering and prediction problem. Transactions of the ASME. Journal of Basic Engineering 82, pp 35–45
27. Kamon I., Rimon E., Rivlin E. (1998) A range–sensor based navigation algorithm. Intern. J. Robotic Research 17(9), pp 934–953
28. Kavraki L.E. (1995) Random Networks in Configuration Space for Fast Planning. PhD Thesis. Stanford University
29. Kavraki L.E., Švestka P., Latombe J.–C., Overmars M. (1996) Probabilistic roadmaps for path planning in high–dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4), pp 566–580
30. Khatib O. (1986) Real–time obstacle avoidance for manipulators and mobile robots. Intern. J. Robotic Research 5, pp 90–98
31. Kube C.R., Zhang H. (1996) The use of perceptual cues in multi–robot box–pushing. In: Proceedings IEEE Intern. Conference on Robotics and Automation, pp 2085–2090
32. Latombe J. C. (1991) Robot Motion Planning. Kluwer Academic Publishers, Boston
33. Leonard N. E., Fiorelli E. (2001) Virtual leaders, artificial potentials and coordinated control of groups. In: Proceedings of the 40th IEEE Conference on Decision and Control, pp 2968–2973
34. Li Z., Canny J.F. (1993) Nonholonomic Motion Planning. Kluwer Academic Publishers, Boston
35. Lumelsky V., Stepanov A. (1987) Path planning strategies for point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica 2, pp 403–430
36. Matarić M. (1993) Kin recognition, similarity and group behavior. In: Proceedings of 15th Annual Cognitive Society Conference. Lawrence Erlbaum Assoc., pp 705–710
37. Matarić M. (1994) Interaction and Intelligent Behavior. PhD Dissertation. MIT EECS Dept.
38. Matarić M. (1997) Behavior–based control: Examples from navigation, learning, and group behavior. Journal of Experimental and Theoretical Artificial Intelligence 9(2,3), pp 323–336
39. Naffin D., Sukhatme G. (2004) Negotiated formations. In: Proceedings of the Intern. Conference on Intelligent Autonomous Systems, pp 181–190
40. Ośmiałowski P. (2007) Player and Stage at PJIIT Robotics Laboratory. Journal of Automation, Mobile Robotics and Intelligent Systems (JAMRIS) 2, pp 21–28

41. Ośmiałowski P. (2009) On path planning for mobile robots: Introducing the mereological potential field method in the framework of mereological spatial reasoning. Journal of Automation, Mobile Robotics and Intelligent Systems (JAMRIS) 3(2), pp 24–33
42. Ośmiałowski P. (2009) A Case of Planning and Programming of a Concurrent Behavior: Planning and Navigating with Formations of Robots. In: Proceedings of CSP 2009. Concurrency, Specification, Programming, Kraków, Warsaw University Press
43. Ośmiałowski P. (2010) L.Polkowski (supervisor). Spatial reasoning based on rough mereology in planning and navigation problems of autonomous mobile robots. PhD dissertation. Polish–Japanese Institute of IT, Warszawa
44. Ośmiałowski P., Polkowski L. (2009) Spatial Reasoning Based on Rough Mereology: Path Planning Problem for Autonomous Mobile Robots. Transactions on Rough Sets XII. Lecture Notes in Computer Science 6190, Springer Verlag, Berlin, pp 143–169
45. Parker L. E. (1997) Cooperative motion control for multi–target observations. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1591–1598
46. Parker L. E. (1998) ALLIANCE: An architecture for fault tolerant multirobot cooperation. IEEE Transactions on Robotics and Automation 14(2), pp 220–240
47. Payton D., Estkowski R., Howard M. (2002) Progress in pheromone robotics. In: Proceedings of the 7th Intern. Conference on Intelligent Autonomous Systems, Marina del Rey, CA
48. Polkowski L., Ośmiałowski P. (2008) Spatial reasoning with applications to mobile robotics. In: Jing Xing–Jian (Ed.)(2008) Mobile Robots Motion Planning. New Challenges. I-Tech, Vienna, pp 433–453
49. Polkowski L., Ośmiałowski P. (2008) A framework for multi–agent mobile robotics: Spatial reasoning based on rough mereology in Player/Stage system. Lecture Notes in Artificial Intelligence 5306. Springer Verlag, Berlin, pp 142–149
50. Polkowski L., Ośmiałowski P. (2010) Navigation for mobile autonomous robots and their formations: An application of spatial reasoning induced from rough mereological geometry. In: Barrera, A. (Ed.) (2010) Mobile Robots Navigation. InTech, Zagreb, pp 329–354
51. Reynolds C. (1987) Flocks, herds and schools. A distributed behavioral model. Comput. Graph. 21(4), pp 25–34
52. Russell S. J., Norvig P. (2009) Artificial Intelligence: Modern Approach. 3rd ed. Prentice Hall, Upper Saddle River, NJ
53. Schneider F., Wildermuth D., Kräusling A. (2005) Discussion of exemplary metrics for multi–robot systems for formation navigation. Advanced Robotic Systems, pp 345–353
54. Schwartz J. T., Sharir M. (1983) On the piano movers' problem II. General techniques for computing topological properties of real algebraic manifolds. Advances in Applied Mathematics 4, pp 298–351
55. Shao J., Xie G., Yu J., Wang L. (2005) Leader–following formation control of multiple mobile robots. In: Proceedings of the 2005 IEEE Intern. Symposium on Intelligent Control, pp 808–813
56. Skowron A., Rauszer C. (1992) The discernibility matrices and functions in decision systems. In: Słowiński, R. (Ed.) (1992) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dordrecht, pp 311–362
57. Stoy K., Brandt D., Christensen D. J. (2010) Self–Reconfigurable Robots. The MIT Press, Cambridge, MA
58. Sugihara K., Suzuki I. (1990) Distributed motion coordination of multiple mobile robots. In: Proceedings 5th IEEE Intern. Symposium on Intelligent Control, pp 138–143
59. Tribelhorn B., Dodds Z. (2007) Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education. In: IEEE International Conference on Robotics and Automation ICRA 2007, Roma, Italy, pp 1393–1399
60. La Valle S. M., Hutchinson S. A. (1996) Optimal motion planning for multiple robots having independent goals. In: Proceedings IEEE International Conference on Robotics and Automation, pp 2847–2852

61. Walter W. G. (1950) An imitation of life. Scientific American 182, pp 42–45
62. Walter W. G. (1953) The Living Brain. Norton, New York
63. Wilson E. O. (2000) Sociobiology: The New Synthesis. (25th Anniversary Ed.). Belknap Press of Harvard University Press, Cambridge, MA