We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Time-Invariant Motion Planner in Discretized C-Spacetime for MRS

Fabio M. Marchese
Università degli Studi di Milano-Bicocca
Italy

1. Introduction

In this work, a new class of planners for MRS is introduced: Time-invariant Motion Planners, a class of planners that operate indifferently in forward or in backward planning-time direction. Thanks to the specular symmetry (with respect to the timeline) of the motion operators, the planning algorithm can operate both in top-down way (from the goal to the starting pose) or vice versa bottom-up (from the starting pose to the goal) addressing different types of problems.

The planner underlying mechanism is an artificial field over a lattice (CAs), where the robots are shrunk to points subjected to attractive and repulsive forces (Lagrangian mechanics). Building a regular manifold of potential values and following its minimum valleys, a trajectory in the spacetime is extracted, corresponding to a robot movement (geometrization of the motion). The potential manifold is constructed on the base of the motion operators. These are the atomic (non interruptible) moves over the space and the time lattice and the set of all of them represents the entire kinematics of a robot. Every robot has its own set and there are contemporarily robots with different kinematics. The manifold emerges from the interaction of the set of operators, the world model and the representation of the robots' shapes. Using a discretized C-Spacetime, the definition of velocity of a robot becomes an intrinsical (geometrical) property emerging from the interaction between the motion operators and the spacetime.

It is fundamental for the correctness of the planning to take care of the actual robot occupancy during an atomic move to avoid collisions with other robots/obstacles. It derives the definition of *Motion Silhouette*, a conceptual evolution of the *Sweeping Silhouette* (2002), which is itself an evolution of the Obstacles Enlargement concept by Lozano-Pérez in 1983. To avoid the problem of the swapping of two robots, it is important to take in consideration of the well-known Shannon's Theorem in the discretization phase of the C-Spacetime and consequently in the definition of *Motion Silhouette*.

2. Multi robots systems motion planning

The basic problem is how to make a flock of robots to navigate coordinately to achieve a common task. The robot(s) navigation can be shortly described as follow (question, process, result):

1. Where have I been?   Map making   ➢ World Representation
2. Where am I?   Localization   ➢ Robot(s) Pose(s)
3. Where am I going?   Task/Mission planning   ➢ Goal pose(s)
4. What's the best way there? Path/Motion planning   ➢ Trajectory/Movement
5. How am I going to get there? Path/Movement exec. ➢ Moves/Motion Commands

Therefore, Multi Robots Systems Motion Planning is a phase of the overall MRS Coordinated Navigation problem.

Many approaches have been proposed to solve the Path/Motion Planning problem for single and multiple robots in the last thirty years. A model-based solution has been proposed since 1979 (Lozano-Pérez & Wesley, 1979, Lozano-Pérez, 1983) where a geometrical description of the environment is given.

To address the problem in a dynamical world, some authors proposed the Artificial Potential Fields Methods. Khatib first proposed a method for the real-time collision avoidance problem of a manipulator in a continuous space (Khatib, 1986). Jahanbin and Fallside introduced a wave propagation algorithm in the Configuration Space (C-Space) on discrete maps (*Distance Transform,* Jahanbin & Fallside, 1988). In the '90s, Barraquand et al. used the Numerical Potential Field Technique over the C-Space to build a generalized Voronoi Diagram (Barraquand et al., 1992). Zelinsky extended the *Distance Transform* to the *Path Transform* (Zelinsky, 1994). Marchese in 1996 first introduced the Cellular Automata paradigm in robot path planning problem (Marchese, 1996) for non-holonomic rototranslating robots. Tzionas et al. in (Tzionas et al., 1997) described a VLSI implementation for a CA based algorithm for diamond-shaped translating holonomic robot in a static environment. For multiple robot motion planning, in (Warren, 1990) the coordination of robots is solved using a discretized 3D C-Spacetime (2D Workspace plus Time) for translating robots with same shapes (only square and circle). In (LaValle & Hutchinson, 1998) the authors apply the concepts of the Game Theory and multi-objective optimization to the centralized and decoupled planning. A solution in the C-Spacetime is proposed in (Bennewitz et al., 2001), where the authors use a decoupled and prioritized path planning in which they repeatedly reorder the robots to try to find a solution. It can be proven that these approaches are not complete.

3. MRS motion planning in a discretized world

A MRS is a set (a flock, a team) of robots with a common task. Robots not sharing the same task are insulated or own to different MRSs with different tasks. The condivision of a task is an important issue: it implies that the robots access to the same resources.

In particular, MRS Motion Planning is a concurrent task, where the shared resource is the common workspace.

In this work, we want to design a motion planner for a set of heterogeneous mobile robots, in order to determine the motions of mobile agents and able to avoid collisions with (statical or dynamical with a designed movement) obstacles and with other robots.

We have adopted Cellular Automata as formalism for merging a grid model of the world (Occupancy Grid) with the C-Spacetime of multiple robots and Artificial Potential Fields Methods, with the purpose to give a simple and fast solution.

3.1 Prioritized planning

While the path planning problem for a single robot has a polynomial complexity, in 1979 Reif established that the problem for a team of robots is PSPACE-hard (Reif, 1979) which implies it is NP-hard. Canny later established that the problem lies in the PSPACE and therefore the general motion planning problem is PSPACE-complete (Canny, 1988). Even a Warehouseman's problem on a discrete 2D grid is PSPACE-hard (Culberson, 1998).

In general, for a N rigid robots problem the number of dimensions of the C-Spacetime would be: $C_{ST} = C^1 \times C^2 \times \cdots \times C^N \times T$ where C^i is the C-Space of the ith robot. If the robots are moving on a 2D surface (manifold), the C-Space of a single robot is $\mathbb{R}^2 \times SO(2)$, thus the overall C-Spacetime has $3N + 1$ dimensions. Even for small MRS, the cardinality of the space makes the problem untreatable. Therefore it is necessary to reduce the number of dimensions, adopting the Prioritized Planning technique (a description in LaValle, 2006).

It is a case of Decoupled Planning for multiple robots, where the interaction robot to robot is ignored in the first phase of motion design. Then the interactions are taken into account to constrain the options available. The problem arises when no option remains, because this approach is not reversible, thus losing the completeness. The typical example is shown in Fig. 1. The two robots have to exchange their positions in the corridor. The red one has the highest priority and has to pass first, occluding the lateral space that would be useful for the green robot to overtake the red one.

Nevertheless, the Prioritized Planning is very practical and solves most of the situations.

In the prioritized approach, an order of robot planning (*priority*) is given, starting to plan with the high-priority robot first. Robots with lower priority view the higher-priority robots as moving obstacles with designed trajectories.
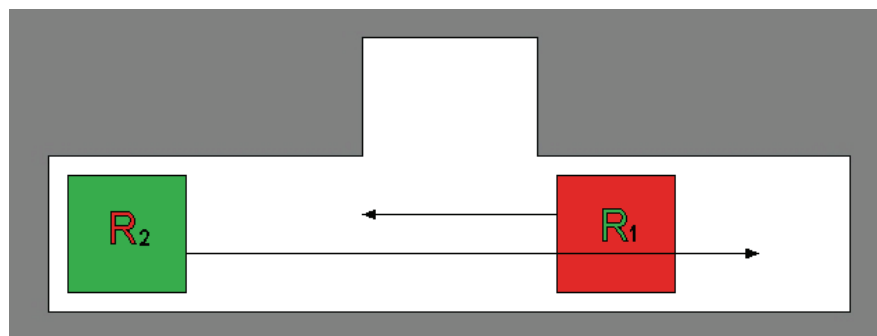

Fig. 1. Priority planning counterexample: red robot moves first

In Fig. 2 is shown an example of the approach in the Spacetime: the blue object is a static obstacle, while two robots (red and green) turn around it, leaving a helicoidal temporal trace.

The planning phases are:
1. Establish an order of priority for the robots.
2. Plan the motion for the robot with the highest priority not yet planned (single robot motion planning).
3. Using the Coordination Space, select one collision-free movement from the set of all movements found.
4. Trace the robot (mark the configurations as not available) in the Coordination Space (the robot becomes a dynamical obstacle for all the other robots with lower priority).
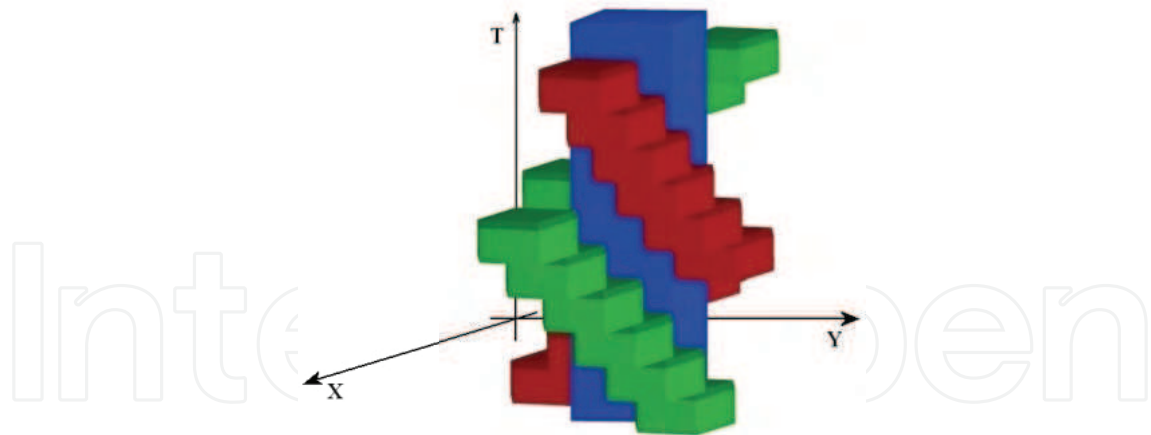5. Goto step 2 until the robot with the lowest priority has been planned.

Fig. 2. Spatiotemporal traces of three objects: a static obstacle (blue) and two robots (red and green)

If no collision-free movement can be found for a robot, many recovery strategies can be adopted. The simpler is to eliminate the robot from the space and from the problem. Another strategy consists in considering the robot as a static object standing in the starting pose (as the blue object in Fig. 2). A more complex strategy is to let the robot at the starting pose, with the final goal to stay in that pose, but reducing its priority. In this case, the robot can move away if another robot (with a higher priority) has to pass in that pose, and then it get back to the initial pose.

The Motion Planning components needed for a well-posed problem are:
-    World description
-    MRS description: shapes & kinematics of all the robot
-    Problem description: starting events (timed poses) and goal events of all the robots.
In this case, we use a discretized description of all the components.

3.2 Discrete world representation

The world representation consists in a Regular Decomposition map, i.e. a grid of cells, marked as full or empty (Occupancy grid). Because of the temporal evolution of a dynamical world, even the world map is a 2D workspace with a discrete time axis (Fig. 3).
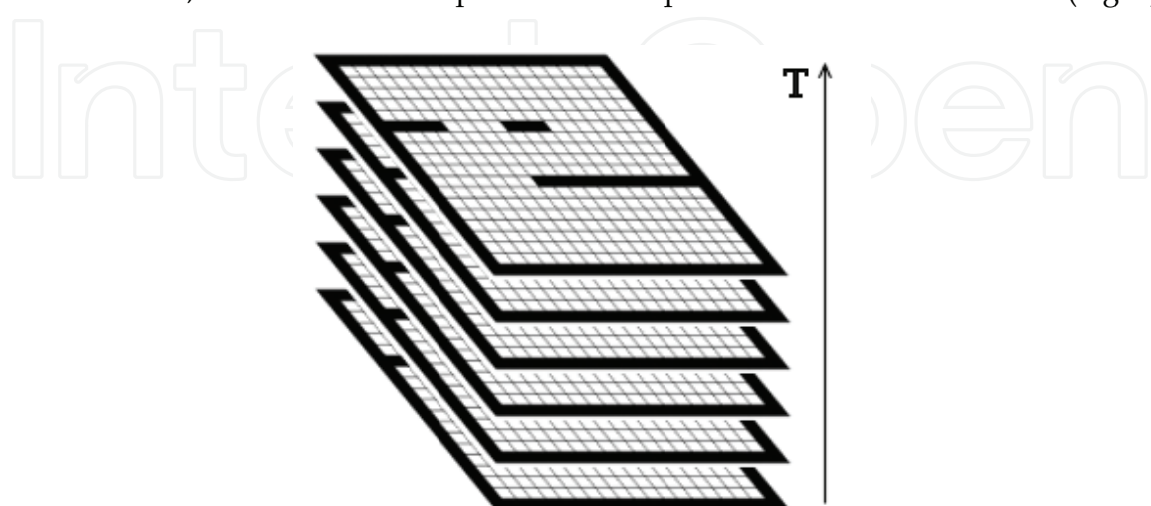

Fig. 3. Spatiotemporal world representation

From this representation derives the most important data structure: the C-Spacetime which, in our case, coincides with the Coordination Space. It is a 4D discretized space composed by the 2D workspace, the orientation axis and the time axis, and it is fundamental for the prioritized planning.

3.3 Robot discrete representation: the motion silhouette

In Regular Decomposition world models, the robot is often represented as a point (usually the robot cinematic center) as in the Lagrangian mechanics, a point moving from one free cell to a neighbor free one. To take into account of its real extension, the well-known technique of enlarging the obstacles by a given quantity has been considered.

Lozano-Pérez et al. at the end of the '70s first introduced this method using the robot maximum radius and approximating its shape to a cylinder with the consequence of losing a great amount of space around the obstacles and a loss of executable trajectories. Then they improved it using an anisotropic enlargement (Lozano-Pérez & Wesley, 1983), i.e. using a different obstacles enlargement for each robot orientation, to solve (only partially) the problem for robots with asymmetric shapes: counter-examples (fig.4.c) can be found in which the robot still collides with obstacles (a peg in the example) due to the sliding of its silhouette between two consecutive poses. In 2002 we proposed a different and more precise approach to address this problem (Marchese, 2002), introducing the *Sweeping Silhouette* defined as the whole space covered during the movement between two consecutive poses (Fig.4.d).

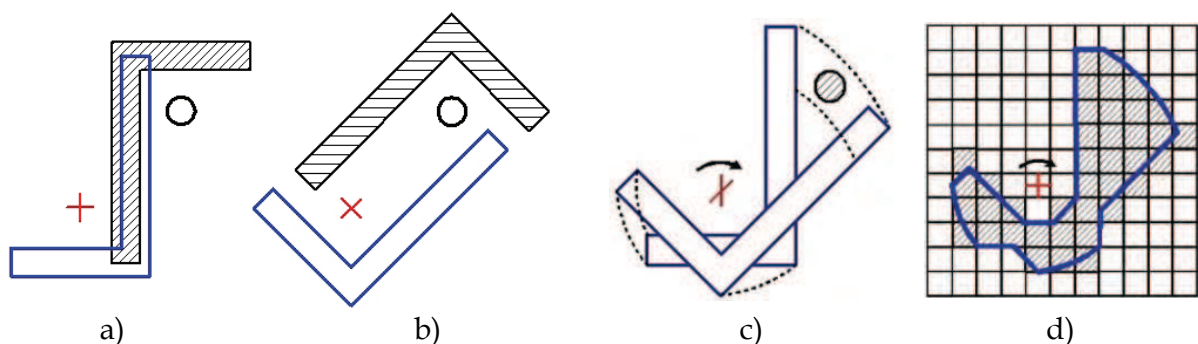a)   b)   c)   d)

Fig. 4. Silhouette sweeping: a-b) expanded obstacle (hatched) for two robot poses (white); c) a counter-example due to a coarse discretization of the orientation; d) *Sweeping Silhouette* (hatched cells) obtained sweeping the robot silhouette between the two poses

The *Sweeping Silhouette* is not sufficient in a spatiotemporal representation: it is necessary a finer representation that takes into account of the position of the robot in every time slice during an atomic movement. The problem is similar to the previous one that brought to define the *Sweeping Silhouette*: a coarse discretization of the time axis could result in undesired effects. For example, two thin and fast robots could swap their positions between two time ticks, passing one through each other. In the same way, a small robot could pass through a thin wall (tunneling).

The necessity of a finer modeling of the motion carries to the definition of a new feature: the *Motion Silhouette*. It is a stack (a sequence) of silhouettes along the time, modeling an atomic move of the robot (e.g. in Fig. 6). Every move has a corresponding *motion silhouette*, or the *motion silhouette* is a conceptual extension of the move where the shape and the physical size of the robot are considered.
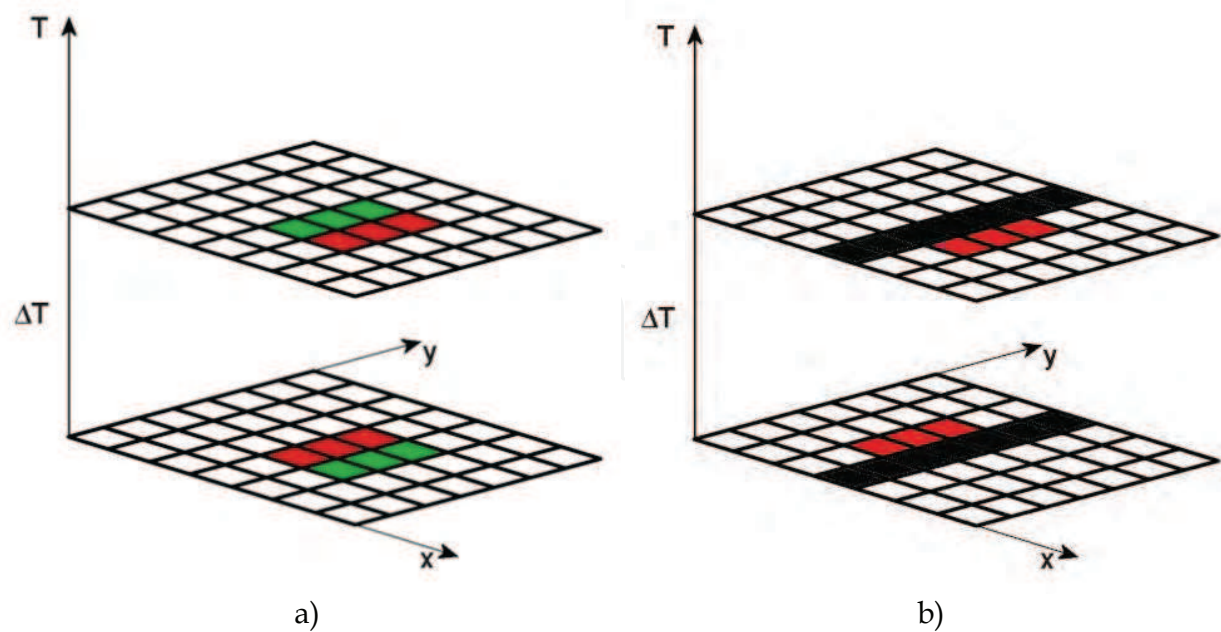
a)   b)

Fig. 5. Examples of undesired effects: a) robots swapping (red move (+1, 0, 0, ΔT), green move (-1, 0, 0, ΔT)); b) robot passing through the wall (red move (+2, 0, 0, ΔT))

To be consistent with the Shannon's Theorem it is sufficient to apply a sampling of the time at a twice frequency, i.e., the timeline must have a time unit half of the other timelines. In particular, the twice sampling has to be applied in the space where the collisions are detected: the Coordination Space. This guarantees an adequate representation of all the obstacles along the time, static (e.g., walls) and dynamical (e.g., other robots, opening doors, etc.). This assumption also ensures to avoid the problem of Fig. 5 of robots tunneling the walls or other robot through.
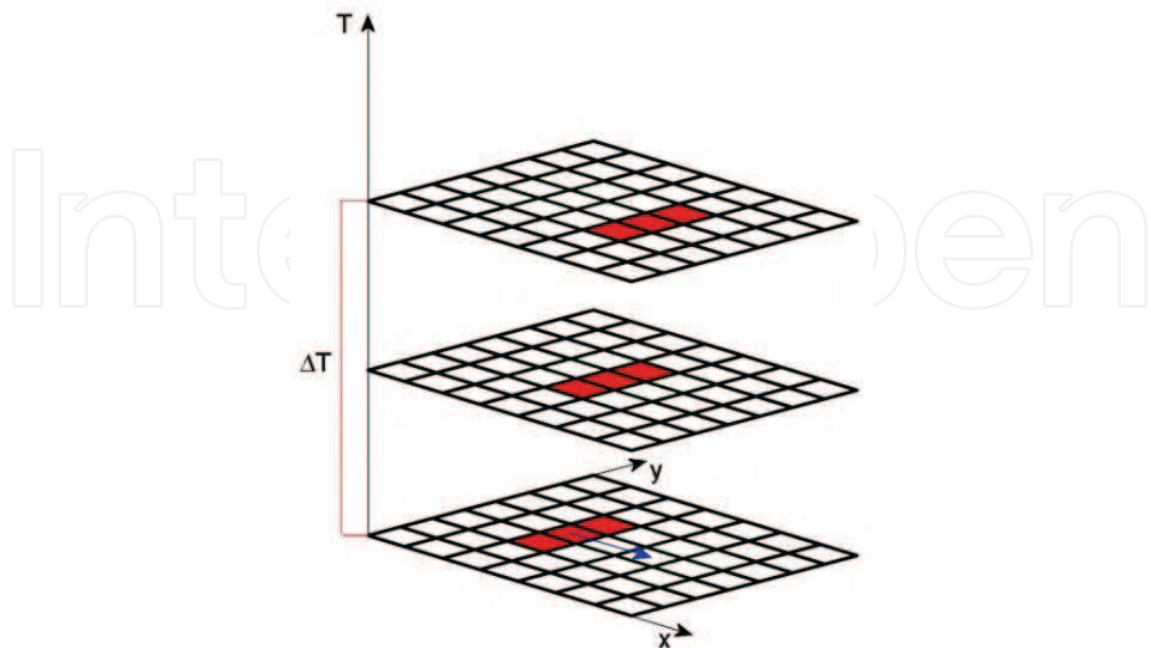

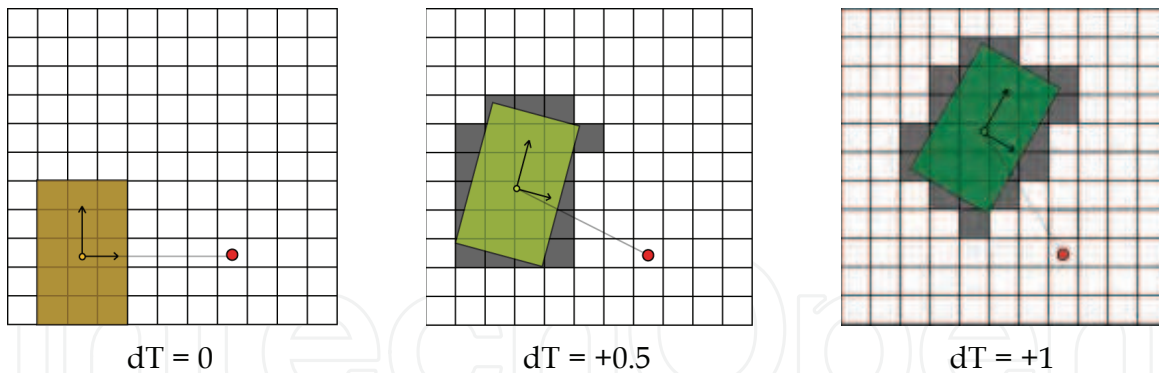Fig. 6. The *Motion Silhouette* for the translational move (+2, 0, 0, ΔT))

Fig. 7. The *Motion Silhouette* for a rototranslation

3.4 Robot discrete kinematics: the spatiotemporal discrete move

To define properly the behavior of a robot in a discretized world is mandatory to define accurately its kinematics. A discretized kinematics is a set of atomic moves defined on the base of single moves along the coordinate axes.

In a discretized spatiotemporal space Z^4 for a robot moving on a 2D manifold, the definition of *spatiotemporal move* is the 4-tuple: $(\Delta x, \Delta y, \Delta\theta, \Delta t)$, where Δ is a finite variation, $(\Delta x, \Delta y) \in Z^2$, $\Delta\theta \in S^1$, $\Delta t \in Z$ and with the obvious constraint $\Delta t > 0$ (an example in Fig. 8). This definition has two main interpretations:

- $(\Delta x, \Delta y, \Delta\theta)$ are finite increments of the spatial coordinates during the finite time interval Δt. It entails the following space metrics Δs:

$$\text{move} \cong \frac{(\Delta x, \Delta y, \Delta\theta)}{\Delta t} \quad \Rightarrow \quad \Delta s^2 = \Delta x^2 + \Delta y^2 + r^2 \cdot \Delta\theta^2$$

- $(\Delta x, \Delta y, \Delta\theta, \Delta t)$ are finite increments of the spatiotemporal coordinates, inducing the following metrics:

$$\text{move} \cong (\Delta x, \Delta y, \Delta\theta, \Delta t) \Rightarrow \Delta S^2 = \Delta x^2 + \Delta y^2 + r^2 \cdot \Delta\theta^2 + v_\tau^2 \cdot \Delta t^2 = \Delta s^2 + v_\tau^2 \cdot \Delta t^2$$

Where ΔS is the "distance" between two events of the spacetime, r is a dimensional constant, v_τ is the "speed" of spontaneous translation along the time axis.

It is easier to see it if considering a robot standing in a place (a static object), then the *temporal speed* results to be:

$$\text{Move} \cong (0, 0, 0, \Delta t) \qquad \Delta S^2 = v_\tau^2 \cdot \Delta t^2 \qquad |v_\tau| = \sqrt{\Delta S^2 / \Delta t^2}$$

Because of this definition of *spatiotemporal move*, the movement of a rigid body becomes a trajectory in the spacetime executed by means of a sequence of finite moves, and where the necessity to indicate the speed disappears (it becomes an intrinsic factor), thus we have the concept of geometrization of the movements.

The speed is computed as usual, but it is a rational value:

$$\Delta s^2 = \Delta x^2 + \Delta y^2 + r^2 \cdot \Delta\theta^2 \qquad |v| = \left|\frac{\Delta s}{\Delta t}\right| = \sqrt{\frac{\Delta x^2 + \Delta y^2 + r^2 \cdot \Delta\theta^2}{\Delta t^2}} = \sqrt{v_x^2 + v_y^2 + r^2\omega^2}$$

For example, $(+2, 0, 0, +1)$ is a move with double speed (x direction) or $(+1, 0, 0, +2)$ represent a move at a half speed with respect to normalized units.
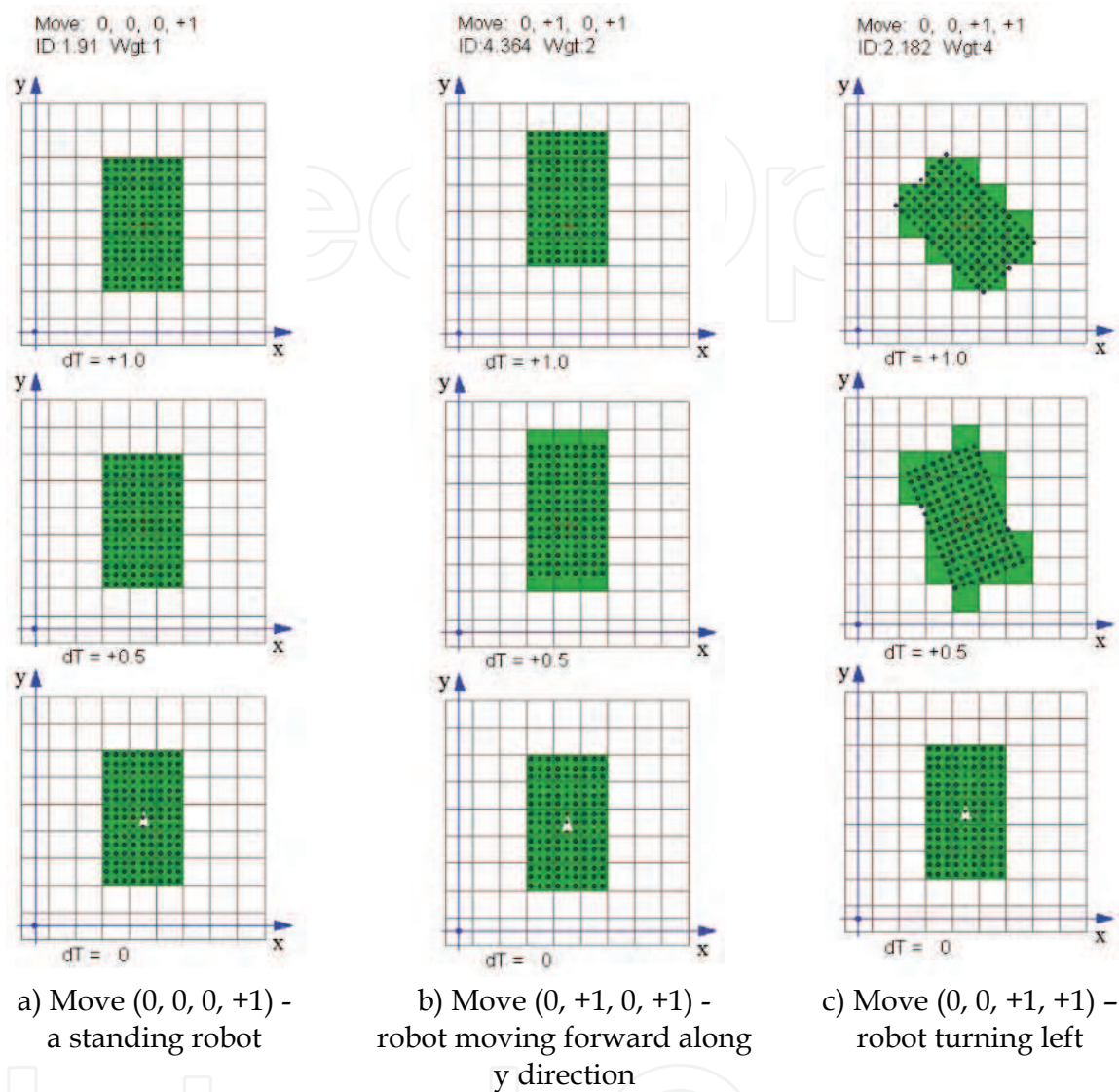

a) Move (0, 0, 0, +1) - a standing robot   b) Move (0, +1, 0, +1) - robot moving forward along y direction   c) Move (0, 0, +1, +1) – robot turning left

Fig. 8. An example of three moves for a rectangular robot

3.5 T-invariant planning and dual motion planning problem

First we have to define the *move dualization* operation. It is thought to simplify the top-down planning (from the goal pose to the starting pose). Every move of the kinematics is dualized inverting the sign of each component of the *spatiotemporal move* (see Fig. 9).

$$\text{Move} \cong (\Delta x, \Delta y, \Delta\theta, \Delta t) \quad \Rightarrow \quad \text{dualMove} \cong (-\Delta x, -\Delta y, -\Delta\theta, -\Delta t)$$

The dual kinematics is the set of all the dual moves computed from the original kinematics. Using the dual moves, it is easy to plan a movement from the goal to the start. It is a top-down planning or backward planning, but using the dual moves it becomes a forward planning (bottom-up) where the "bottom" is the goal and the "up" is the starting pose. Thus it is easy to solve the dual problem as it would be a forward planning.

a) dualMove (0, 0, 0, -1) -
a standing robot

b) Move (0, -1, 0, -1) -
backward move along y & t
direction

c) Move (0, 0, -1, -1) –
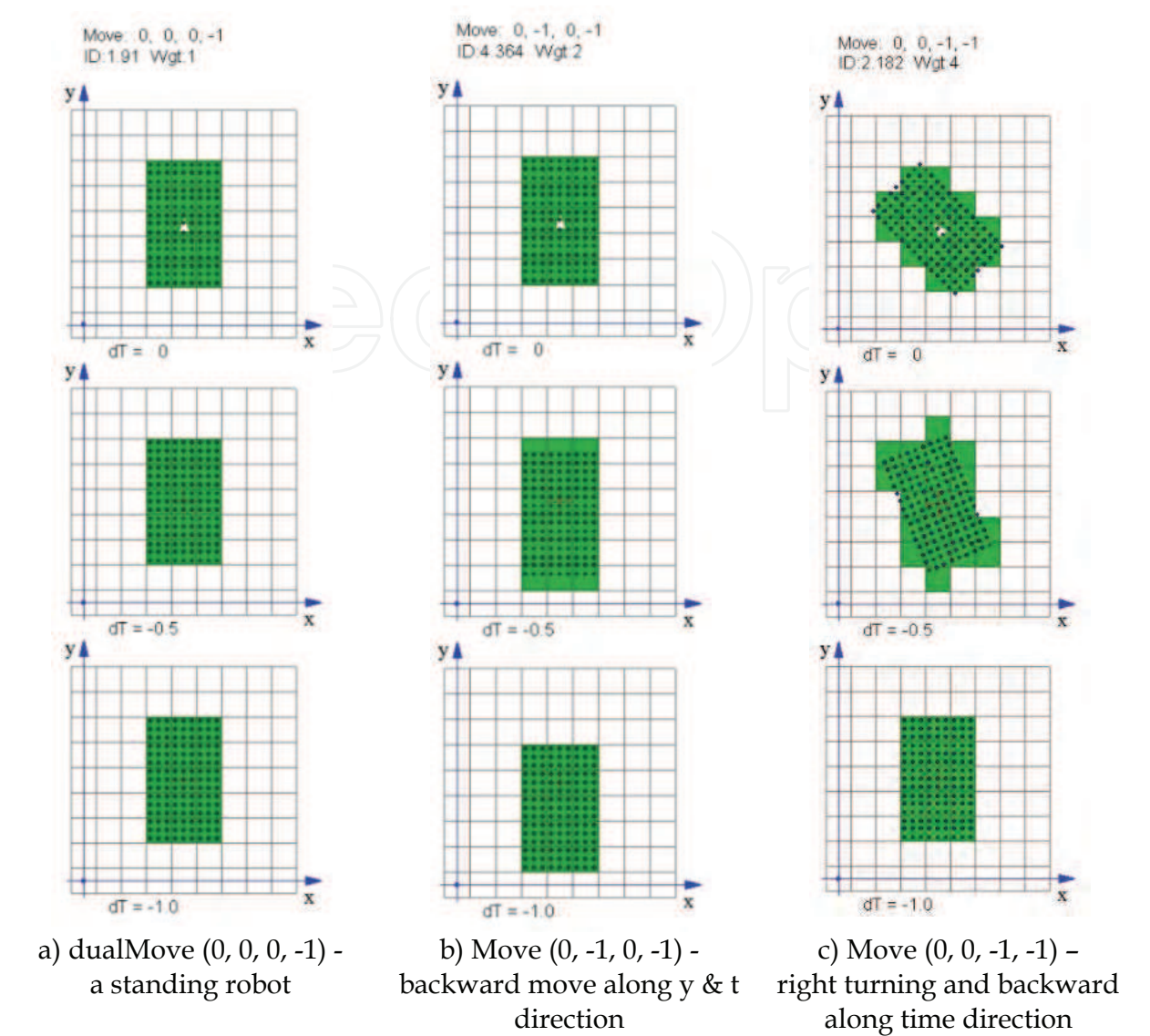right turning and backward
along time direction

Fig. 9. Examples of the dual moves of Fig. 8

Exchanging the start with the goal and dualizing the two sets of moves we have the dual problem (Fig. 10). To be able to solve even this problem we should relax a constraint: we must consider that moves with $\Delta t < 0$ have to be admissible. It could seem quite strange that a body could run backward in the past, but it is only a logical operation of remapping the original problem to the dual problem. In any case, we do not intend to affirm that the robot could navigate back in the Time! Rather, it would seem as a kind of retrograde motion, but in the real spacetime any robot will always move forward in the Time.

If we relax the constraint $\Delta t > 0$, and we admit any *spatiotemporal move* (Δx, Δy, $\Delta\theta$, Δt) with any sign of delta, the set of moves is said to be closed under the dualization operation. Thus the Moves set and the dualMoves set belongs to the same superset that it will be called *dMoves set*.

The definition of the *dMoves set* makes any motion planning problem and any dual motion planning problem the same problem embedded in the same space: the C-Spacetime. They are both solved with the same algorithm, unifying them in a unique problem: *the motion planning problem over the C-Spacetime.*

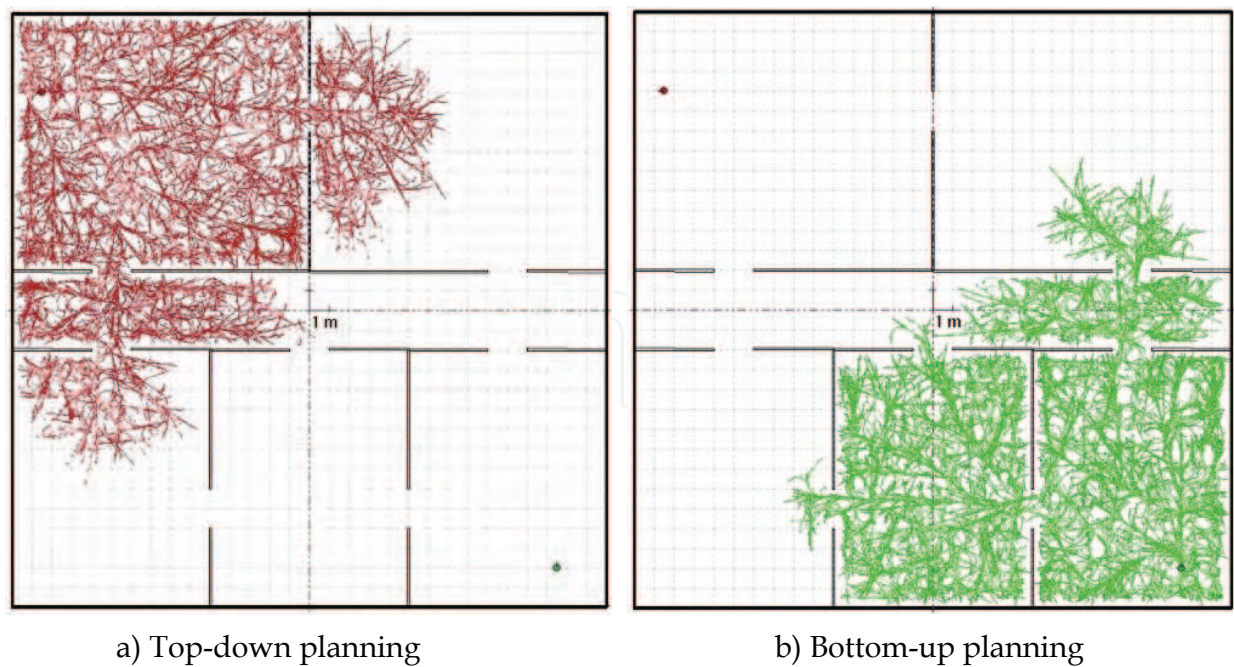a) Top-down planning   b) Bottom-up planning

Fig. 10. Examples of search trees in path planning, from the goal (red tree) and from the start (green tree)

It is a Time-invariant Motion Planner. Because the top-down and the bottom-up planning are both solved by the same algorithm and thus are perfectly equivalent, it has the property of T-invariance concerning the planning time (not the robot time). It means that the solution of a problem is invariant with respect to the starting time of the planning, but even if we invert the time axis direction the solution found is the same. It is possible to plan a movement from the goal to the starting pose or, vice versa, from the starting pose to the goal and we will find the same motion.
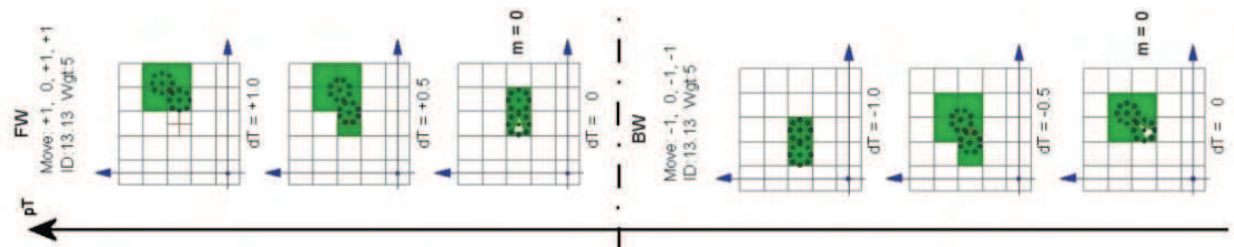
Fig. 11. Time specularity of the dMove operator with respect to the dot-dashed axis

The time-invariance emerges from the property of temporal specularity of the dMove operators with respect to the planning time (pT). In Fig. 11 an example is shown for a simple robot (only two cells size). The dMove (+1, 0, +1, +1) and the dual move (-1, 0, -1, -1) are specular with respect to the dot-dashed axis, unless a spatial shift (because the dMove are always invariant with respect to any space displacement, this shift is not relevant). The meaning is: during the planning, the dMoves are applied from right to left (starting with m = 0 grid). The dMoves are specular with respect to the planning order (time) of application, while are complementary with respect to the time T (the real time of motion of the robot). For the latter consideration, the composition of the two operators generates an identity mapping (applying both in sequence, the robot get back to the original event).

3.6 Attraction Space

The Attraction Space is the substrate on which a discrete representation of a potential function is built (C-Potential function U(q)) in the metaphor of the Artificial Potential Fields. The C-Potential function generates a potential bowl (Fig. 12) in the free space (avoiding and surrounding the obstacles) with a global minimum in the goal cell that attracts the robot with a force -grad(U). If it is correctly defined (it is always possible), no local minima are generated and there is only one global minimum.
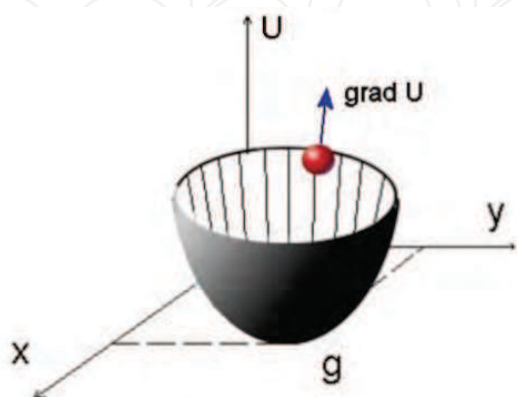
Fig. 12. Potential bowl

The potential value represents the integer "distance" of a cell c from the goal cell along the shortest collision-free path, or more simply it is the cost to reach the goal from the cell c. To evaluate the entire path cost, every basic robot movement (spatiotemporal move) has its own positive weight. Even the "move" along the time axis has its cost.

It is easy to demonstrate two main properties: the termination of the propagation of the potential values through the C-Spacetime and the absence of local minima. The later property ensures that the robot does not enter in obstacles concavities and stall (unless the goal is inside the concavity). In Fig. 13.b, it is shown a simplified example of potential
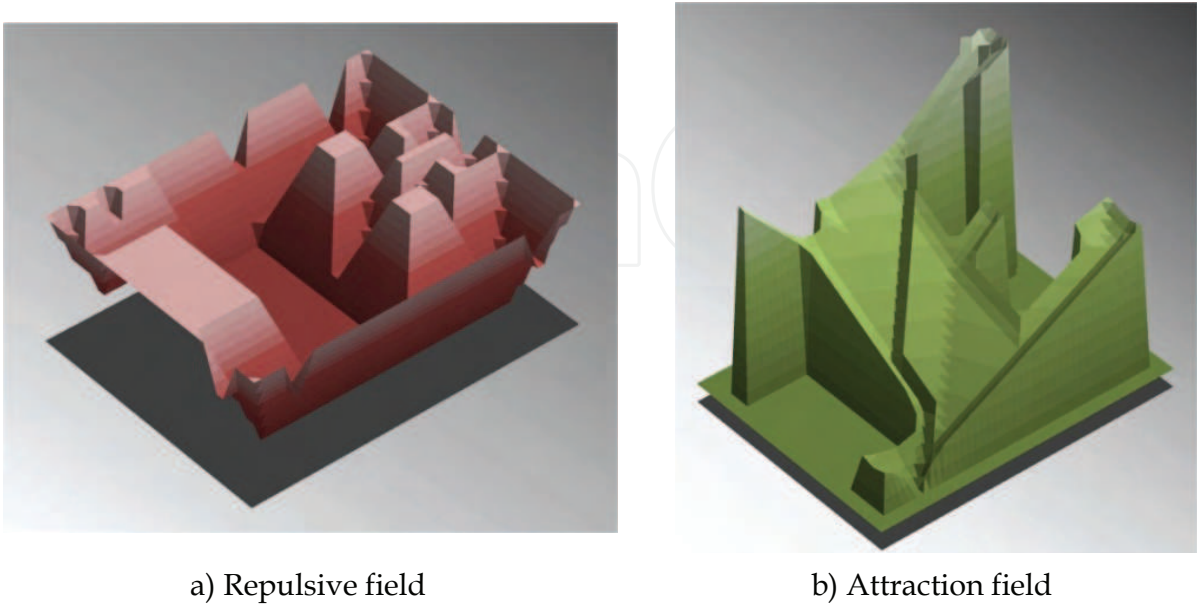
a) Repulsive field   b) Attraction field

Fig. 13. Potentials fields

surface generated from the goal and surrounding the obstacles. The real surface is embedded in a 5D discrete space ($Z^2 \times S^1 \times Z \times Z$), where also the Time is represented. Every single robot has its own Attraction Space, which is computed depending on the obstacles distribution and the temporal traces of the robots with higher priorities.

3.7 Coordination space or repulsive space

It is a unique space for all the MRS, where the interaction robot-robot and robot-obstacle are modeled. By the way, it is the C-Spacetime and it represents the evolution in the time of the environment. Another metaphor for this space is the Repulsive Space. In this space, repulsive forces are generated by the obstacles to take the robot away from them. It is very useful for the coordination of the robots in a prioritized planning: after the planner has computed the movement for a robot, its passing points (configurations) are marked as unavailable in the Coordination Space. The remaining robots will plan their movements only in the free space, avoiding any unavailable configuration. As stated before, to

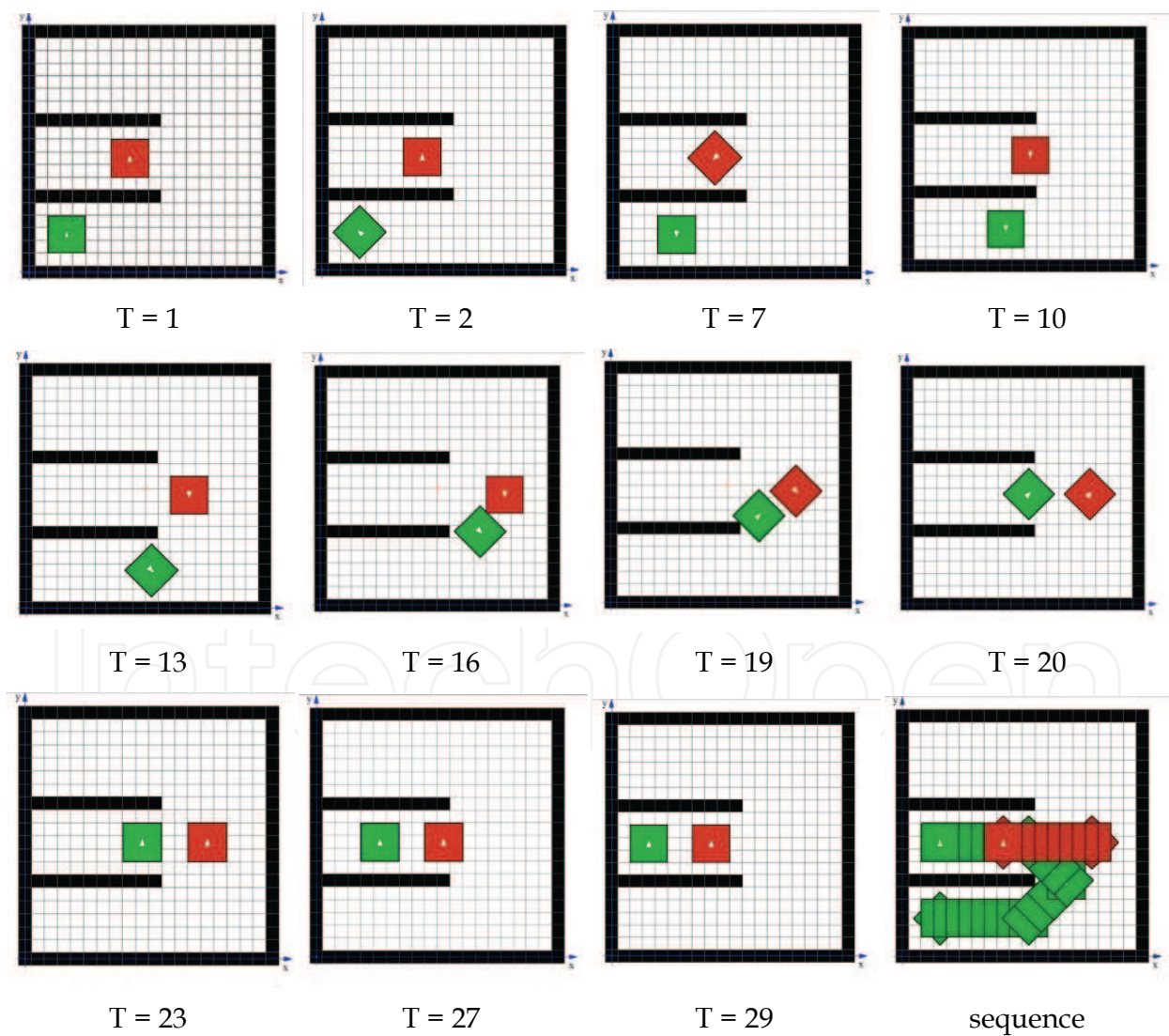| T = 1	T = 2	T = 7	T = 10
T = 13	T = 16	T = 19	T = 20
T = 23	T = 27	T = 29	sequence

Fig. 14. Example with two robots with same shapes and kinematics (priority order: green, red)

guarantee the Shannon's Theorem, the time unit must be half of the time unit of the Attraction Space. A simplified version of repulsive potential surface is shown in Fig. 13.a (the repulsive force is maximum over the obstacles); the real surface is embedded in a 5D discrete space ($Z^2 \times S^1 \times Z \times Z$).
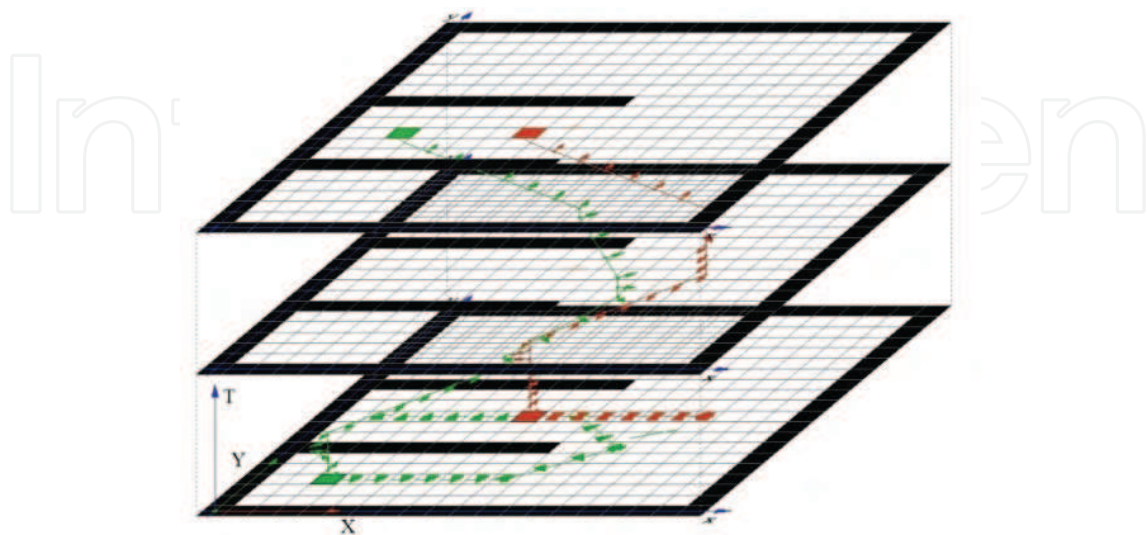

Fig. 15. Spacetime representation of the robots movements of Fig. 14

4. Experimental results

In the example of Fig. 14, two robots with particular kinematics move in a simple environment. The target for the green one (GRobot) is to get inside of a corridor closed by the red RRobot. The RRobot has only to rest in the same place. Thus the GRobot having a higher priority, forces the red one to get outside, enters inside, and then lets the RRobot to return to the original pose. The kinematics is quite strange: the two robots can rest, turn counterclockwise or move aside on their left. Kinematics definition for GRobot and RRobot: {(0, 0, 0, +1), (0, 0, +1, +1), (0, +1, 0, +1)}. Even with a quite limited kinematics, a solution is found.

In Fig. 15, it is shown the Spacetime and the movements of the robots (colored lines). The Spacetime should be 4D: to reduce it to 3D, the robot orientations have been represented with arrows. The robot trajectories are represented on the plane T = 0 (orthogonal projection of the movements along the time direction).

In the example of Fig. 16, three O-ring robots move one inside the others (matrioska). The robots have different shapes and sizes, but all have the same kinematics: {(0, 0, 0, +1), (0, 0, +1, +1)}. The difficulty of this example is the coordination of the movements of the three robots to avoid the collision one with each others: thus the green one start first to let enough space to the red one to move without to collide with it. The red robot also has to move to let the space to the blue robot. At the end of the motions, they all stack against the pegs.

In the example of figure Fig. 18, there are two robot with different shapes and kinematics. GRobot kinematics: {(0, 0, 0, +1), (+1, 0, 0, +1), (−1, 0, 0, +1)} (forward and backward translations).

RRobot kinematics: {(0, 0, 0, +1), (0, 0, +1, +1), (0, 0, −1, +1)} (clockwise and counterclockwise rotations).

To let the GRobot to pass, the RRobot has to turn until the vertical orientation and then it has to stay fixed till the green one is close to it. When the GRobot has overtaken the red one, the latter can complete the motion.

5. Conclusion

An important result is the following: the potential field over the spatiotemporal domain is not conservative (not irrotational). The spatiotemporal lines connecting the starting event and the goal event over the spacetime are not equivalent: not every geometrical solution (movement) is admissible. The dependency arises from the interaction between robots with finite size shapes (it would be conservative for robots smaller then a cell).

The overall algorithm works with any polygonal obstacles, with multiple robots, with any type of kinematics (holonomic, non-holonomic, omnidirectional, car-like, etc.) and any shapes (non-connected, with convexities, concavities and holes). It plans the optimal motions for robots (even one inside another), facing different types of problems without stalling inside the concavities.
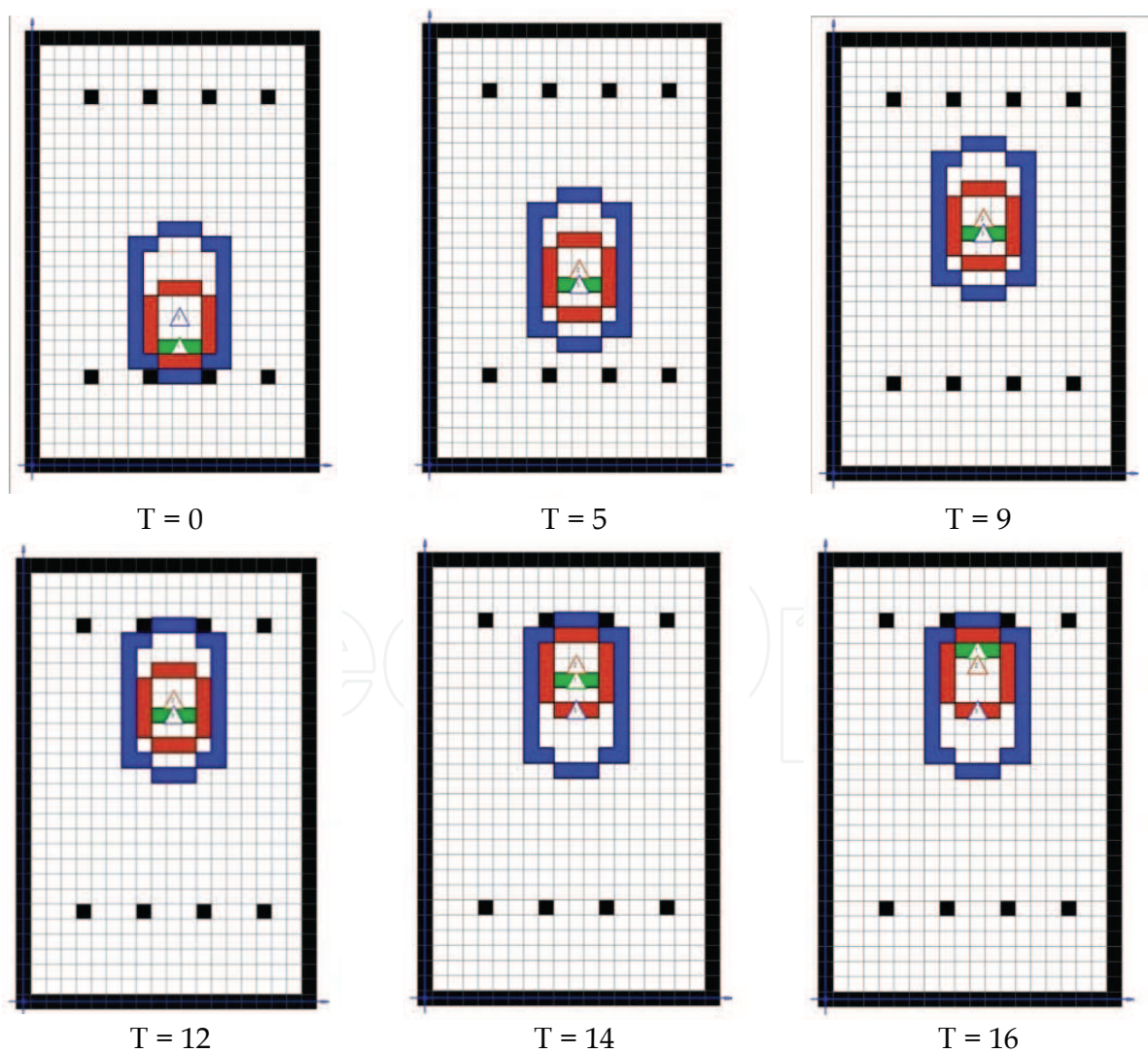

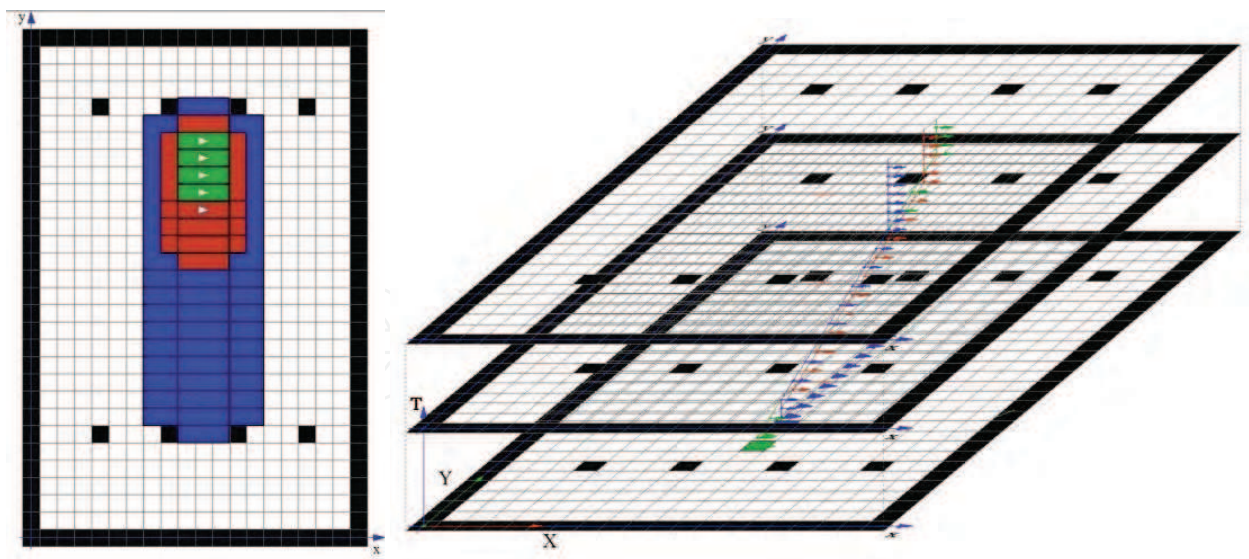Fig. 16. Matrioska example (priority order: green, red, blue)

Fig. 17. Silhouettes sequences and Spacetime of example of Fig. 16

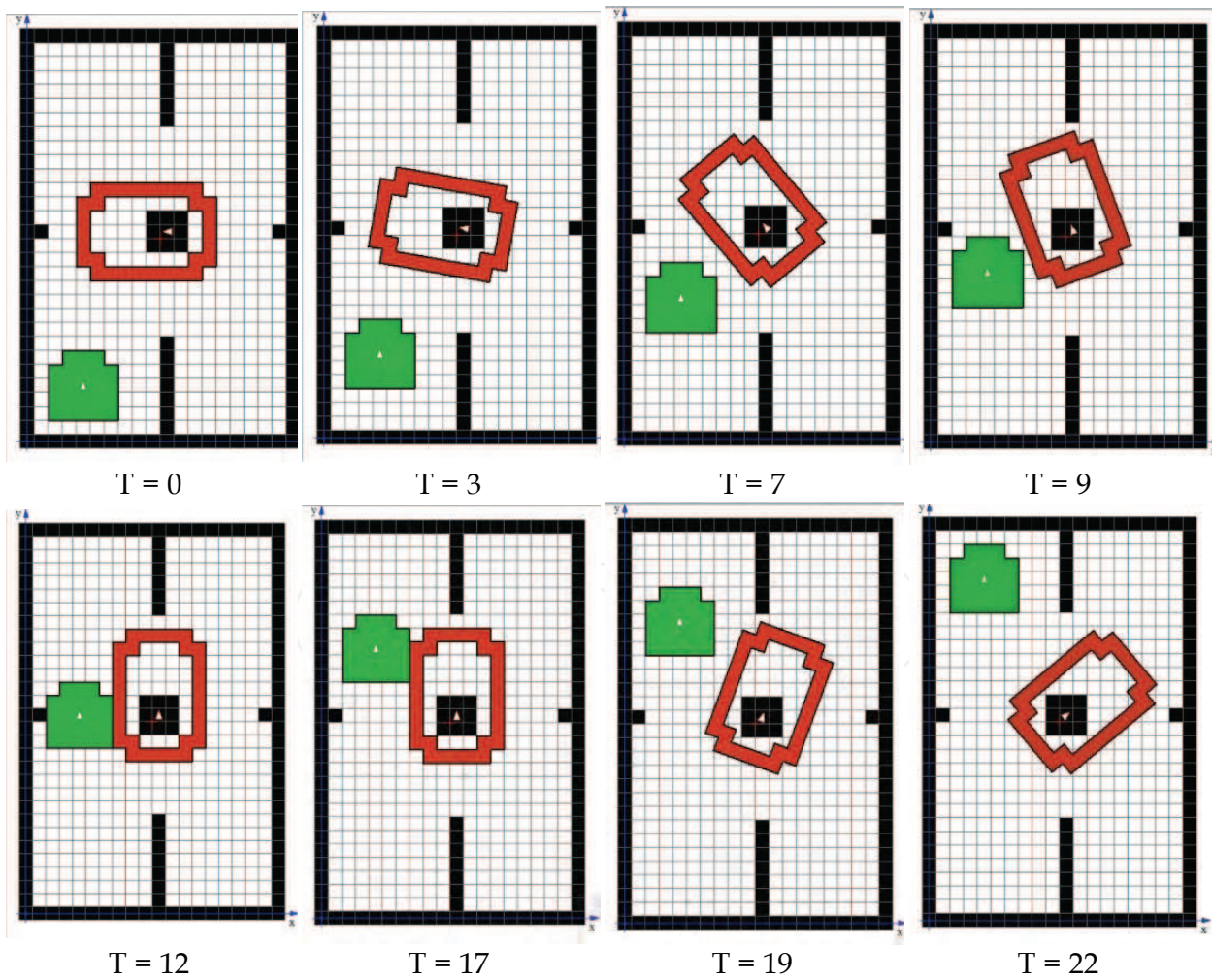| T = 0 | T = 3 | T = 7 | T = 9 |

| T = 12 | T = 17 | T = 19 | T = 22 |

Fig. 18. Example with robots with completely different shapes and kinematics (priority order: green, red)
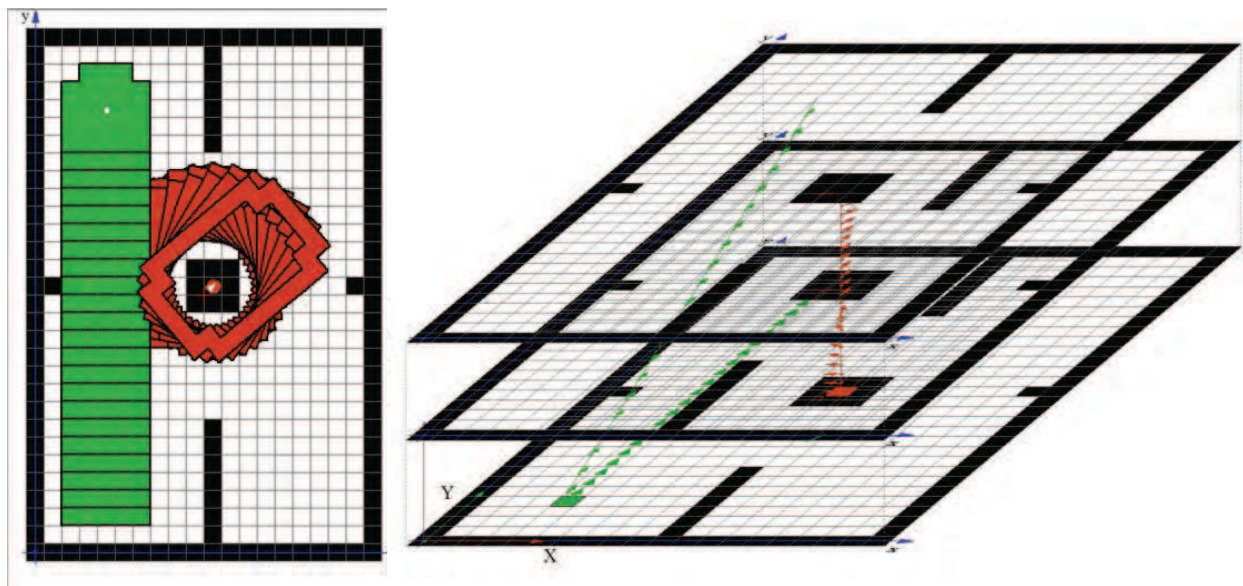
Fig. 19. Silhouettes sequences and Spacetime of example of Fig. 18

In this work, only the solutions satisfying the Dirichlet (or first-type) boundary condition are searched. In other words, we search the motion connecting a starting event with a goal event in a 4D Spacetime (initial and final conditions) in the case of a single robot. For MRS, a vector of starting events (one for each robot) and a vector of goal events are specified. Many other problems can be studied, for example specifying a range of starting time (interval) and a range of arrival time for each robot, relaxing the temporal constraints and searching the best time of arrival (or leaving time).
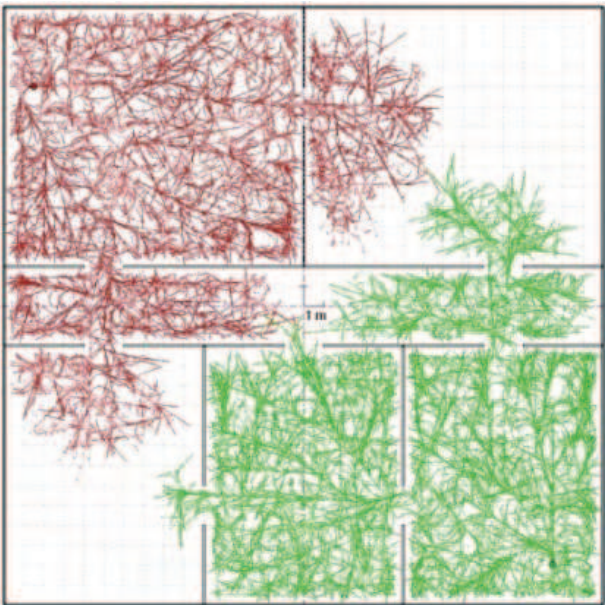

Fig. 20. Search trees growing one against the other (bidirectional planning)

A future development will regard the simultaneous growing of the search trees, one from the goal and the other from the start, as in Fig. 20, intersecting somewhere in the middle of the search space (bidirectional planning). The task is to reduce the planning time, but still maintaining the optimality of the solutions found.

6. References

Barraquand, J.; Langlois, B. & Latombe, J. C. (1992). Numerical potential field techniques for robot path planning, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 22, No. 2, pp. 224-241, ISSN 0018-9472

Bennewitz, M.; Burgard, W. & Thrun, S. (2001). Optimizing schedules for prioritized path planning of multi-robot systems, *Proceedings of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 271-276, Seoul (Korea), May 2001

Canny, J. F. (1988). *The Complexity of Robot Motion Planning*, MIT Press, ISBN 978-0-262-03136-3, Cambridge, MA

Culberson, J. C. (1998). Sokoban is pspace-complete. *Proceedings of the International Conference on Fun with Algorithms (FUN98)*, pp. 65-76, Waterloo, Ontario, Canada, June 1998, Carleton-Scientific

Erdmann, M. & Lozano-Perez, T. (1987). On multiple moving obstacles, *Algorithmica*, Vol. 2, No. 1-4, pp. 477-521, ISSN 0178-4617, Springer-Verlag, New York, NY, USA

Jahanbin, M. R. & Fallside, F. (1988). Path planning using a wave simulation technique in the configuration space, In: *Artificial Intelligence in Engineering: Robotics and Processes*, J. Gero S. (Ed.), Computational Mechanics Inc., ISBN 0-931215-98-6, Billerica, MA, USA

Kathib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. of Robotics Research*, Vol. 5, No. 1, pp. 90-98, ISSN 0278-3649

Lamound, J. P. (1998). Robot Motion Planning and Control, *Lectures Notes in Control and Information Sciences*, No. 229, Springer-Verlag, ISBN 3-540-76219-1, New York, NY, USA

Latombe, J. C. (1991). *Robot Motion* Planning, Kluwer Academic Publishers, ISBN 079239206X, Boston, MA

LaValle, S. M. (2006). *Planning Algorithms*, Cambridge University Press, ISBN 0521862051, New York, NY, USA

LaValle, S. M. & Hutchinson, S. A. (1998). Optimal motion planning for multiple robots having independent goals, *IEEE Trans. on Robotics and Automation*, Vol. 14, No. 6, (December 1998), pp. 912-925, ISSN 1042-296X

Lozano-Pérez, T. & Wesley, M. A. (1979). An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, *Communications of the ACM*, Vol. 22, No. 10, (October 1979), pp. 560-570, ISSN 0001-0782

Lozano-Pérez, T. (1983). Spatial Planning: A Configuration Space Approach, *IEEE Trans. on Computers*, Vol. C-32, No. 2, (February 1983), pp. 108-120, ISSN 0018-9340

Marchese, F, (1996). Cellular automata in robot path planning, *Proc. of the 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT)*, pp. 116-125, ISBN 0-8186-7695-7, Kaiserslautern (G), October 1996, IEEE Computer Society

Marchese, F. M. (2002). A Path-Planner for Mobile Robots of Generic Shape with Multi-Layered Cellular Automata, In: *Cellular Automata - 5th Int. Conf. on Cellular Automata for Research and Industry (ACRI 2002)*, Bandini, S.; Chopard, B.; Tomassini, M., (Eds.), LNCS, Vol. 2493, (October 2002), pp. 178-189, Springer-Verlag, Berlin/Heidelberg, Germany, ISBN 978-3-540-44304-9, ISSN 0302-9743

Marchese, F. M. (2008). Spatiotemporal MCA Approach for the Motion Coordination of Heterogeneous MRS, In: *Recent Advances in Multi Robot Systems*, Lazinica, A. (Ed.), pp. 123-138, Tech Ed. and Publ., ISBN 978-3 902613-24-0

Murphy, R. R. (2000). *An Introduction to AI Robotics*, MIT Press, ISBN 978-0-262-13383-8, Cambridge, MA, USA

Reif, J. H. (1979). Complexity of the mover's problem and generalizations, *Proc. of 20th Annual Symposium on Foundations of Computer Science (FOCS 1979)*, pp. 421-427, October 1979, IEEE Computer Society, ISSN 0272-5428, Washington, DC, USA

Sheu, P. C. Y. & Xeu, Q. (1993). Intelligent Robotic Planning Systems, *World Scientific Series in Robotics and Automated Systems*, World Scientific Publishing, ISBN 9810207581, River Edge, NJ, USA

Tzionas, P. G.; Thanailakis, A. & Tsalides, P. G. (1997). Collision-free path planning for a diamond-shaped robot using two-dimensional cellular automata, *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 2, pp. 237-250, ISSN 1042-296X

Warren, C. (1990). Multiple robot path coordination using artificial potential fields, *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pp. 500-505, ISBN 0-8186-9061-5, Cincinnati, OH, USA, May 1990

Zelinsky, A. (1994). Using Path Transforms to Guide the Search for Findpath in 2D, *Int. J. of Robotics Research*, Vol. 13, No. 4, pp. 315-325

Multi-Robot Systems, Trends and Development

Edited by Dr Toshiyuki Yasuda

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Fabio Marchese (2011). Time-Invariant Motion Planner in Discretized C-Spacetime for MRS, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: http://www.intechopen.com/books/multi-robot-systems-trends-and-development/time-invariant-motion-planner-in-discretized-c-spacetime-for-mrs

INTECH
open science | open minds