An Experimental Comparison of Path Planning Techniques Applied to Micro-sized Magnetic Agents

Stefano Scheggi[1] and Sarthak Misra[1, 2]

[1]Surgical Robotics Laboratory, Department of Biomechanical Engineering, MIRA-Institute for Biomedical Technology and Technical Medicine, University of Twente, The Netherlands

[2]Department of Biomedical Engineering, University of Groningen and University Medical Centre Groningen, The Netherlands

Email: {s.scheggi,s.misra}@utwente.nl

Abstract—**Micro-sized agents can be used in applications such as microassembly, micromanipulation, and minimally invasive surgeries. Magnetic agents such as paramagnetic microparticles can be controlled to deliver pharmaceutical agents to difficult-to-access regions within the human body. In order to autonomously move these microparticles toward a target/goal area, an obstacle-free path must be computed using path planning algorithms. Several path planning algorithms have been developed in the literature, however, to the best of our knowledge, only few have been employed in an experimental scenario. In this paper we perform an experimental comparison of six path planning algorithms when applied to the motion control of paramagnetic microparticles. Among the families of deterministic and probabilistic path planners we select the ones that we consider the most fundamental, such as: A* with quadtrees, A* with uniform grids, D* Lite, Artificial Potential Field, Probabilistic Roadmap and Rapidly-exploring Random Tree. We consider a 2D environment made by both dynamic and static obstacles. Four scenarios are evaluated. Three metrics such as computation time, length of the trajectory performed by the microparticle, and time to reach the goal are used to compare the planners. Experimental results reveal equivalence between almost all the considered planners in terms of trajectory length and completion time. Concerning the computation time, A* with quadtrees and Artificial Potential Field achieve the best performances.**

I. INTRODUCTION

Micro-sized agents, such as self-propelled microjets [1], MagnetoSperm [2], magnetotactic bacteria [3], paramagnetic microparticles [4], Janus micromotors [5], and sperm-driven Micro-Bio-Robots [6] (see Fig. 1), can be used to carry out limited minimally invasive surgeries in difficult-to-access regions within the human body. Paramagnetic nanoparticles and microparticles could be used as contrast agents and localized drug delivery inside the human body. They have an average diameter of 100 μm and can be controlled by externally applying a magnetic field gradient thus eliminating the need to carry an on-board power source [7].

For certain applications it is necessary not only to control the micro-sized agent, but also to make it avoid collisions with the environment. Path planners compute obstacle-free paths
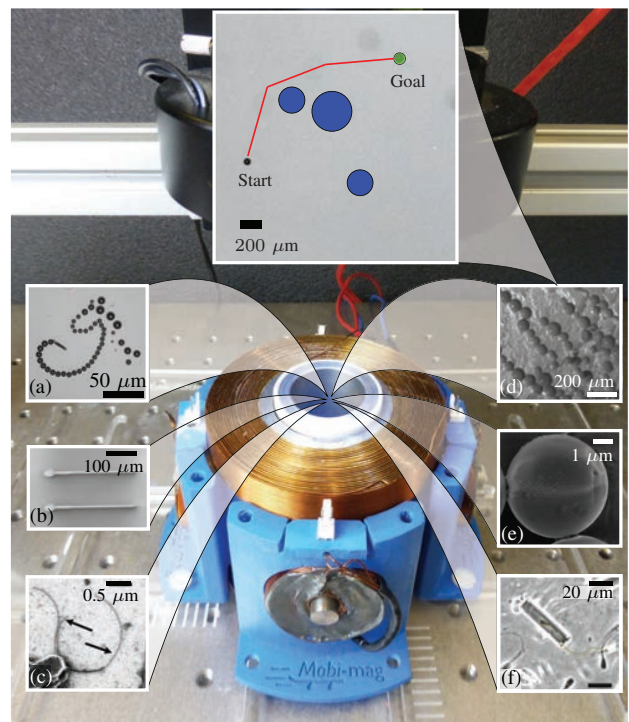
Fig. 1. The Mobi-Mag magnetic system used for the motion control of several micro-sized agents: (a) Self-propelled microjets (b) MagnetoSperm (c) Magnetotactic bacteria (d) Paramagnetic microparticles (e) Janus micromotors, and (f) Sperm-driven micro-bio-robots [1]-[6]. In order to autonomously move toward a target, these magnetic agents need an obstacle-free path which can be provided by a path planner. In this work we focus on paramagnetic microparticles. The Mobi-Mag system is composed by six coils (four on the sides, one on the top, and one on the bottom) fixed in a frame around a reservoir with the liquids. The microscope objective is positioned above the reservoir. Magnetic forces are exerted on the microparticle to control its position and reach a target/goal area. (Top inset) The image captured by the camera shows an obstacle-free path (red) generated by a Probabilistic Roadmap planner. The path is used to move the paramagnetic microparticle (black) toward the goal area (green) while avoiding dynamic and static obstacles (blue).

in order to move the microparticle from its initial position to the goal region while avoiding collisions with static and dynamic objects. Although several pathfinding algorithms have been developed in recent years, only few have been applied

to magnetic agents in an experimental scenario. In [4] the authors implemented a 2D path planner based on the A* algorithm [8]. The path was calculated based on the initial positions of the microparticle and of the obstacles; collision avoidance was based on the Artificial Potential Field (APF) approach [9]. In [10] a 3D probabilistic-based path planning method for static obstacles was developed. The motion planner explicitly considered uncertainty in the motion of the microparticle and maximized the probability to avoid obstacle collisions and reach the target. The proposed motion planner was successfully tested in a simulated 3D environment with static obstacles.

Since path planners have been rarely implemented for micro/nano applications, in this paper we perform a comparison between six path planners, when applied to paramagnetic microparticles in a real-world 2D dynamic environment (see Fig. 1). For each planner we report possible advantages and drawbacks. Among the families of *deterministic* and *probabilistic* planners (the former use a deterministic decomposition of the configuration space, the latter use a random sampling of it) we select the ones that we consider the most fundamental, such as: A* [8], D* Lite [11], Artificial Potential Field (APF) [9], Probabilistic Roadmap (PRM) [12], [13], and Rapidly-exploring Random Tree (RRT) [14] (see Fig. 2). Regarding the A* algorithm, it is used with both uniform grids and quadtrees. The reasons to investigate deterministic path planning are the following: (*i*) the simplicity to implement these algorithms; (*ii*) the space analyzed which does not have a huge size to require a very complex algorithm to calculate the path. On the contrary, probabilistic path planners provide solutions to problems involving vast, high-dimensional configuration spaces that would be intractable using deterministic approaches. The proposed planners are evaluated in four experimental scenarios, including static and dynamic obstacles and environments where the information available concerning the environment is incomplete. Three metrics such as computation time, length of the trajectory traveled by the microparticle, and time to reach the goal area are used to compare the planners. The results of this paper should help future users of planning approach applied to paramagnetic microparticles to choose the correct techniques. It is worth noting that, although we evaluate the path planners with paramagnetic microparticles, analogous results can be obtained with other micro-sized agents having motion equations similar to that of microparticles, i.e., hydrogel micro-grippers [15], [16], and metallic micro-grippers [17]. The motivation to use paramagnetic microparticles is related to their reduced size which allows us to evaluate the planners in more challenging scenarios.

The rest of the paper is organized as follows. Sect. II reviews the dynamic model of paramagnetic microparticles. Sect. III describes the techniques used to partition the configuration space. In Sect. IV the path planning algorithms used in this paper are reviewed, while in Sect. V they are evaluated via experiments and results are given and discussed. Finally, in Sect. VI we summarize the main contributions of the paper, and we discuss possible avenues for future research.
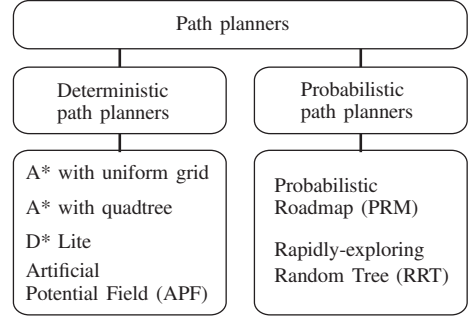

Fig. 2. Path planners can be classified into deterministic and probabilistic planners. Deterministic planners use a deterministic decomposition of the configuration space. They are simple to implement and suitable if the space analyzed does not have a huge size to require a very complex algorithm to calculate the path. On the contrary, probabilistic path planners use a random sampling of the configuration space and provide solutions to problems involving vast, high-dimensional configuration spaces that would be intractable using deterministic approaches.

II. MODELING MICROPARTICLE DYNAMICS AND CONTROL

In this section we describe the 3D dynamic model of paramagnetic microparticles. Let $\mathbf{p} = [x, \ y, \ z]^T \in \mathbb{R}^{3 \times 1}$ be the position of a microparticle in the 3D space. The state of the microparticle is defined as $\mathbf{x} = [x, \ y, \ z, \ v_x, \ v_y, \ v_z]^T \in \mathbb{R}^{6 \times 1}$ where $v_x, \ v_y, \ v_z$ represent the velocity of microparticle. Microparticles move under the influence of a magnetic force, a drag force, and a buoyancy force. We can control the magnetic force by applying currents to the magnets surrounding the microparticle's workspace. Let $\mathbf{f}(\mathbf{p}) = [F_x, \ F_y, \ F_z]^T \in \mathbb{R}^{3 \times 1}$ be the magnetic force and let n be the number of electromagnets within the magnetic system, the magnetic force equals [2], [4], [5]:

$$\mathbf{f}(\mathbf{p}) = \frac{4}{3\mu_0(1 + \chi_m)} \pi r_p^3 \chi_m \ \nabla(\mathbf{I}_c^T \widetilde{\mathbf{B}}^T(\mathbf{p}) \widetilde{\mathbf{B}}(\mathbf{p}) \mathbf{I}_c), \quad (1)$$

where χ_m is the magnetic susceptibility constant, μ_0 is the vacuum permeability, $\widetilde{\mathbf{B}} \in \mathbb{R}^{3 \times n}$ is a matrix defining the magnetic field that depends on the position where the magnetic force is measured, $\mathbf{I}_c \in \mathbb{R}^{n \times 1}$ is the vector of the applied current, and r_p is the radius of the microparticle. To compute currents for any desired magnetic force, a current-force map that inverts Eq. (1) is used. Since the gradient of the magnetic field is almost a constant in the workspace, we can write the magnetic force as $\mathbf{f}$. The drag force acting on the microparticle is $\mathbf{f}_d = [kv_x, \ kv_y, \ kv_z]^T \in \mathbb{R}^{3 \times 1}$, where $k = -6\pi\eta r_p$ is a constant. We define the net buoyancy force acting on the microparticle as $F_b = V(\rho_b - \rho_f)g$ where V, ρ_b, ρ_f, and g are the volume and density of the microparticle, the density of fluid, and the acceleration due to gravity, respectively. The continuous-time dynamics of the paramagnetic microparticle can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\tilde{\mathbf{f}}, \quad (2)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{E}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{k}{m}\mathbf{E}_{3 \times 3} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \frac{1}{m}\mathbf{E}_{3 \times 3} \end{bmatrix},$$

$\tilde{\mathbf{f}} = \mathbf{f} + F_b[0, 0, -1]^T$, $\mathbf{0}_{3 \times 3} \in \mathbb{R}^{3 \times 3}$ is a zero matrix, $\mathbf{E}_{3 \times 3} \in \mathbb{R}^{3 \times 3}$ is the identity matrix, and m is the mass of the microparticle. From Eq. (2), it is worth noting that the
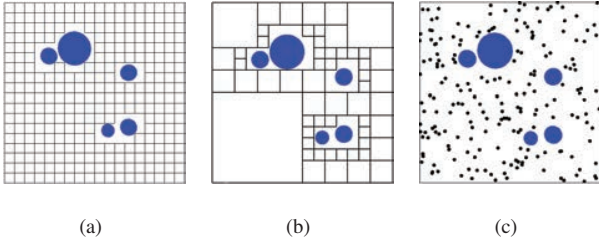
Fig. 3. Space partition of a 1024×1024 units area via: (a) uniform grid, (b) quadtree, (c) random sampling. Obstacles are depicted in blue. The uniform grid divides the area in 324 obstacle-free quadrants with side of 60 units. The quadtree generates 94 obstacle-free quadrants (the minimal side of the quadrants is 60 units). Note: in (c) the area is randomly sampled with 200 samples using a uniform distribution.

microparticle is a holonomic system. The microparticle has no kinematic constraints in its motion, on the contrary it has dynamic constraints on its velocities and forces.

In order to properly understand the planning algorithms used in this work, in the next section we describe how they represent the environment in a way that they can easily manage it.

III. PARTITION OF THE CONFIGURATION SPACE

Path planners partition the world into a structure which reduces the number of dimensions that a planner has to deal with. The elements of such a structure are the states of the planner and they are used to generate obstacle-free paths. Let $\mathcal{C}$ be the configuration space, i.e. the space of all possible displacements of the agent and let $\mathcal{F}$ be the free space (a collision-free subset of $\mathcal{C}$). In this work we consider three methods to represent $\mathcal{C}$ and consequently $\mathcal{F}$: (i) uniform grids, (ii) quadtrees, and (iii) random sampling. We implement A*, APF and D* Lite algorithms with uniform grids (Fig. 3 (a)); A* algorithm is implemented also with quadtrees (Fig. 3 (b)); PRMs and RRTs use a random sampling of the configuration space (Fig. 3 (c)). A uniform grid represents the simplest way to partition the space. Regular grids perform better than quadtrees in terms of space decomposition and neighbor search, however quadtrees usually partition the space in less cells than uniform grids (see Fig. 3 (a)-(b)). Quadtrees have also the advantage that the larger blocks allow for a faster search and the representation is easy to store. This properties result very useful in relation to path planning algorithms since less states need to be explored.

Uniform grids and quadtrees decompose the environment into quadrants. Uniform grids decompose the 2D space in a matrix of equal-size blocks, see Fig. 3(a). For uniform grids, inserting a new point or searching for the neighbors of a given quadrant are very fast and trivial operations.

A quadtree [18] is a tree data structure in which each internal node has exactly four children. Quadtrees are based on the principle of regular decomposition and they can be used to partition a 2D space by recursively subdividing it into four quadrants or regions. Each child represents one of the four quadrants in which the original space was divided. The algorithm stops when the tree reaches a maximal depth or, equivalently, when the area of the quadrants is below a given threshold. At each step, if the new node contains or intersects obstacles, it is marked as full. On the contrary, if it does not contain or intersect obstacles, it is not divided and it is marked as free (see Fig. 3(b)). Inserting a 2D object into a quadtree

requires to explore the tree from the root. At each step, the algorithm checks if the object intersects the current node. The proposed approach is recursively applied until a leaf node is reached. A similar approach is performed to search for the neighbors of a given node.

The probabilistic planners considered in this work use a random sampling of the configuration space $\mathcal{C}$, i.e. a new sample is created by choosing random values for all its degrees of freedom and added to the graph (PRM) or to the tree (RRT) if it is collision-free (see Fig. 3(c)).

IV. PLANNING ALGORITHMS

In this section we briefly report an overview of the planners used in this work.

A. Deterministic planners

A* [8] is a standard algorithm which finds a least-cost path from a given initial state to the goal state. A* has several advantages: (i) it finds a path from the start to the goal if it exists; (ii) it finds an optimal path as long as the heuristic is admissible; (iii) it makes the most efficient use of the heuristic, i.e., no search that use the same heuristic function to find optimal paths examines fewer states than A*. A drawback of A* is that, on a large map, thousands of states might be stored, which can require a lot of memory. Moreover, A* is most inefficient in determining that no path is possible between the start and goal locations. In this case A* examines all the possible locations of the map.

Several algorithms have been developed in order to speed up the performance of the standard A* during the exploration and replanning phases by reusing the information from previous searches. In this work we consider the D* Lite algorithm [11]. D* Lite planner is based on the Lifelong Planning A* (LPA*) [19] to mimic D* [20]. D* runs similarly to A* on the initial run, however, as the unit moves from start to finish, D* is able to quickly recalculate the best path from that unit's position to the finish, much faster than A*. D*, however, is considered to be complex. D* Lite [11] uses LPA* to mimic D*. On the initial run, D* Lite runs very similarly to A*. As the unit moves from start to finish, D* Lite uses LPA* to very quickly recalculate the best path from that unit's position to the finish. D* Lite is considered much simpler than D* and it is at least as efficient as D* [11].

Different from A* and D* Lite, Artificial Potential Field (APF) method is based on attractive potential field due to the target and repulsive potential field due to the obstacles of the world [9]. The sum of these two potential gives us the current potential of the micro-sized agent. The agent moves by following the negative gradient of the potential energy function. We use two different attraction function: conic and quadratic. The quadratic formula gives us distance-dependent replacement vectors and it is used when the microparticle is far from the goal. On the other hand, the conic formula is not linear with respect to square of the distance and is good for near-the-goal cases. The repulsive potential is calculated according to each obstacle and the sum of them gives us the total repulsive function. APFs are not computationally expensive, however they are not optimal (it is not guaranteed that they find the best solution) nor complete (it is not
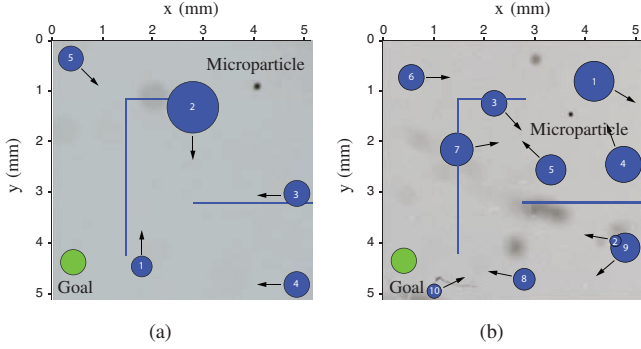
Fig. 4. The experimental scenarios contain a goal region (green) and virtual dynamic and static obstacles (blue). The goal region is located at $(0.4, 4.35)$ mm and has a radius of 0.25 mm. The initial position of the microparticle is shown in black. Dynamic obstacles are represented as circles while static obstacles are depicted as line segments. The black arrows represent the velocity directions of the dynamic obstacles. The size of the environment is about 5.1×5.1 mm^2. The objective is to move the microparticle from its actual position to the goal region while avoiding obstacles. (a) We consider 5 dynamic obstacles having speed of 0.25, 0.25, 0.25, 0.25, and 0.35 mm/s, respectively for obstacle 1, ..., 5. The initial position of the paramagnetic microparticle is $(4.1, 0.72)$ mm. (b) 10 dynamic obstacles are considered. The speed of the obstacles are 0.12, 0.38, 0.27, 0.42, 0.41, 0.26, 0.16, 0.18, 0.39, and 0.42 mm/s, respectively for obstacle 1, ..., 10. The initial position of the microparticle is $(3.7, 1.45)$ mm.

guaranteed that they find the best solution or a goal state if one exists due to local minima).

While a deterministic planner would be overwhelmed by the prohibitive cost of computing an exact representation of the free space $\mathcal{F}$, a probabilistic one builds only a simplified representation of $\mathcal{F}$. In the next subsection we describe the two probabilistic planners used in this paper.

B. Probabilistic planners

Probabilistic planners represent a class of methods of remarkable efficiency, especially in problems involving high-dimensional configuration spaces. The downside of the probabilistic methods is that they are only probabilistically complete, i.e., the probability of finding a solution to the planning problem when one exists tends to 1 as the execution time tends to infinity. This means that, if no solution exists, the algorithm will run indefinitely. In practice, a maximum number of iterations is enforced so as to guarantee its termination.

PRM planners generate an extremely simplified representation of $\mathcal{F}$, called a probabilistic roadmap. A roadmap is a graph $\mathcal{G}$ whose nodes are configurations sampled from $\mathcal{F}$ according to a suitable probability measure and whose edges are simple collision-free paths, e.g., straight-line segments, between sampled configurations. The PRM approach consists of two phases: a *preprocessing* phase and a *query* phase. In the preprocessing phase a roadmap graph $\mathcal{G}$ is constructed by sampling $\mathcal{F}$ and adding the nodes to $\mathcal{G}$. In the query phase, the start and goal configurations are connected to $\mathcal{G}$ and the path is obtained by performing a shortest path query on the graph.

RRT is a well-established sampling-based motion planner. The RRT is a tree $\mathcal{T}$ rooted at the microparticle's initial state $\mathbf{x}_{start}$ (Sect. II). At each iteration, the algorithm samples a state $\mathbf{x}_{sample} \in \mathcal{F}$, finds its nearest neighbors $\mathbf{x}_{nn}$ in the tree, and computes a feasible control $\mathbf{f}$ that grows the tree toward the sampled state [21]. The tree is grown toward the sampled

state by applying the control input $\mathbf{f}$ to the dynamic model of the micro-sized agent (2). The output of the RRT is a motion plan $[(\mathbf{x}_{start}, \mathbf{f}_{start}), \ldots, (\mathbf{x}_S, \mathbf{f}_S)]$, where S is the number of steps. The most important advantages of RRTs is that they can deal with real-valued spaces of extremely high dimension and they can handle dynamics.

V. EXPERIMENTAL VALIDATION

We conduct experimental studies in which we apply the path planners to a microparticle moving in a real 2D environment. The environment contains virtual dynamic and static obstacles and a goal area (see Fig. 4). The objective is to move the microparticle from its initial position to the goal region while avoiding the obstacles. We use virtual dynamic and static obstacle in order to evaluate the planners in similar environments and thus provide a fair comparison among them; *the video of the experiments is available at: https://youtu.be/RuAQG8hFrGI.*

The main components of the experimental setup are a fluid reservoir for the paramagnetic particles and six orthogonally oriented electromagnets (Fig. 1). Each electromagnet is powered by an Elmo Whistle 1/60 servo controller (Elmo Motion Control, Petach-Tikva, Israel). A Blackfly 1.4 MP Color GigE PoE (Point Grey Research Inc., Richmond, Canada) camera is mounted on a Mitutoyo FS70 microscope unit (Mitutoyo, Kawasaki, Japan) using a Mitutoyo M Plan Apo 2 / 0.055 Objective. We use paramagnetic microparticles, consisting of iron-oxide in a poly(lactic acid) matrix (PLA Particles-M-redF-plain from Micromod Partikeltechnologie GmbH, Rostock-Warnemuende, Germany). These particles have radius $r_p = 50$ μm and mass $m = 7.33 \times 10^{-10}$ Kg. Concerning the magnetic system, $\chi_m = 0.075$, $\mu_0 = 4\pi \times 10^{-7}$ T.m/A, $\eta = 1$ mPa.s, $\rho_f = 998.2$ kg/m^3, $\rho_b = 1.4 \times 103$ kg/m^3, $\max \mathbf{I}_c = 1$ A, and $\max |\widetilde{\mathbf{B}}(\mathbf{p})| = 15$ mT. Four scenarios are considered (see Table I). We evaluate the path planners using three metrics:

- Computation time (it is the time to partition $\mathcal{C}$ plus the time to compute the path).
- Length of the trajectory traveled by the microparticle.
- Time to reach the goal area.

Fig. 4 depicts the initial setup of the experimental scenarios. Dynamic obstacles are represented as circles; it is a common choice to represent obstacles by their bounding area or volume. Static obstacles are represented as line segments. In order to simulate tracking errors, we add zero-mean white Gaussian noise with standard deviation $\sigma = 50$ μm to the $x-$ and $y-$ coordinates of the dynamic obstacles. Concerning the path planners, the quadrants of the uniform grid have a side of 300 μm. For the quadtree, we use a minimal cell side of 300 μm

TABLE I
THIS STUDY CONSIDERS FOUR EXPERIMENTAL SCENARIOS. IN SCENARIOS II AND IV, WE SIMULATE LIMITATIONS IN THE SENSING RANGE OF THE SYSTEM ASSUMING A SENSING RANGE OF 1 MM AROUND THE MICROPARTICLE.

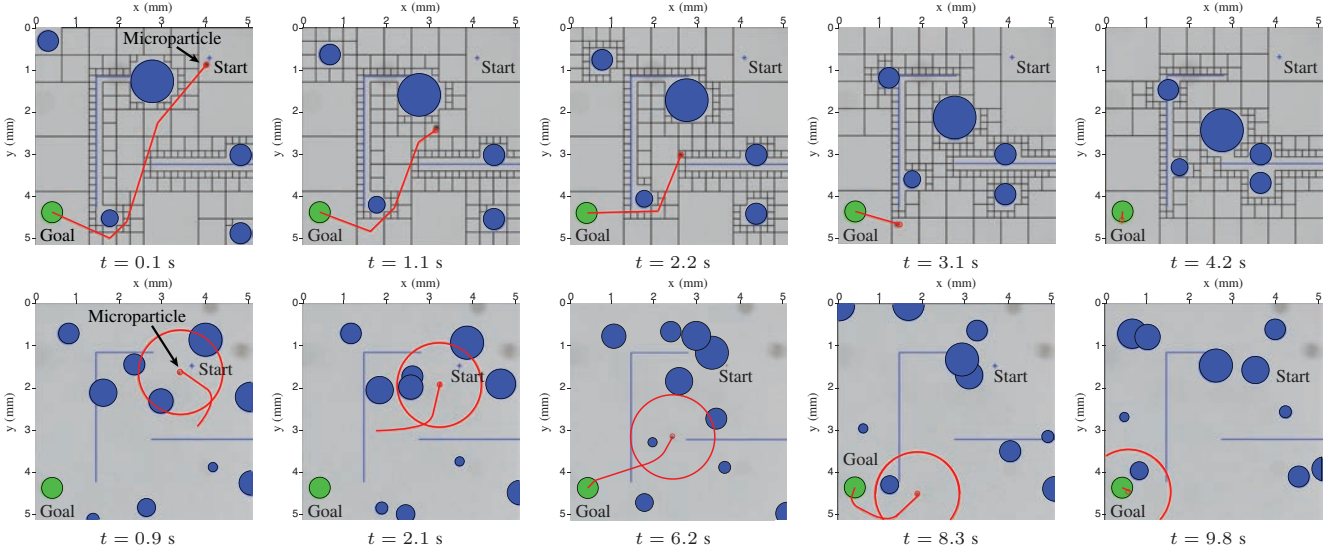| Scenario	Static obstacles	Dynamic obstacles	Sensing range (mm)
I	3	5	∞
II	3	5	1
III	3	10	∞
IV	3	10	1

Fig. 5. Representative experimental results of the A* with quadtree (top sequence) and the Rapidly-exploring Random Tree (bottom sequence) applied to the motion planning of a paramagnetic microparticle. The top sequence shows the results of Scenario I while the bottom sequence shows the results of Scenario IV. The objective is to move the microparticle from its actual position to the goal region (green) while avoiding static and dynamic obstacles (blue). The planned trajectory is shown in red. In the bottom sequence, the red circle represents the sensing range of the system (the radius is about 1 mm). In this case, only the obstacles which are within the sensing range are considered by the planner. *The video of the experiments is available at: https://youtu.be/RuAQG8hFrGI.*

and at each time frame we perform a full recalculation of the quadtree rather than updating the moving entities in the tree since we have more mobile entities than static. For the PRM algorithm we sample $\mathcal{F}$ with 200 samples and we connect each new sample with a maximum of 10 neighbors. Concerning the RRT planner, we use a standard RRT-GoalZoom policy [21] where the new random state is generated based on a biased coin toss that chooses a random sample from either a region around the goal or the whole space. Instead of extending an RRT by an incremental step, we iterate until the random state or an obstacle is reached. Moreover, we use a fixed planning time interval of $T = 0.1$ s. At the beginning the RRT computes the motion plan. Then, for each period of duration T, the system executes the previously computed trajectory and simultaneously the motion planner updates the plan that will be ready for the next time interval. For each path planner except for the RRT, we perform a path smoothing based on the line-of-sight using a voxel traversal algorithm [22]. We test our C++ implementations on a PC with an Intel Xeon CPU 3.2 GHz processor. For each scenario and for each path planners, we perform 7 trials. Fig. 5 shows the motion of a paramagnetic microparticle along an obstacle-free path generated by an A* with quadtree and an RRT planner. The experimental results are provided in Fig. 6.

In order to compare the six planners in the different considered conditions, we run a three-way repeated-measures ANOVA. The type of planner, number of obstacles, and sensing range are considered within-subject factors. Ratings are subjected to the arcsine square root transformation to stabilize variance. All the transformed data pass the Shapiro-Wilk normality test and Mauchly's Test of Sphericity. Sphericity is assumed for variables with only two levels of repeated measures. The test reveals a statistically significant change in the completion time due to the sensing range ($F_{(1, 5)} = 56.305$, $p = 0.001$) and the type of planner ($F_{(5, 25)} = 6.082$,

$p = 0.001$). For the type of planner, post hoc analysis with Bonferroni adjustments reveals a significant difference in the performance of A* with quadtrees vs. APF ($p = 0.006$). A second ANOVA test shows a statistically significant change in the trajectory due to the type of planner ($F_{(5, 25)} = 6.082$, $p = 0.001$). Post hoc analysis with Bonferroni adjustments reveals a significant difference in the performance of A* with quadtrees vs. APF ($p = 0.012$). We perform an additional analysis on the conditions found non statistically different in the previous ANOVA analysis. In order to understand if the difference among them can be considered statistically negligible, we use the two one-sided t-test approach (TOST) [23]. The null hypothesis of the TOST states that the mean values of two groups are different by at least a certain amount ε. In this work we evaluate ε as suggested in [23]. In order to test for equivalence, the 90% confidence intervals for the difference between the two groups are evaluated. The tests reveal statistical equivalence between all the considered planners except D* Lite vs. APF and A* with quadtree vs. APF, for both trajectory length and completion time. Statistical analysis reveals that, with the exception of APF, there are no statistical differences between deterministic and probabilistic methods, mainly due to the fact that the space analyzed does not have a huge size, the configuration space is low-dimensional, and the simple dynamic model of the microparticle. For the computation time, in all the scenarios A* with quadtree and APF have the lowest computation time. For the APF, the motivation is that some computation can be performed offline and stored. For the A* with quadtree the motivation is that less states had to be visited.

VI. Conclusions and Future Work

Micro-sized agents can be accurately controlled to perform minimally invasive procedures, such as targeted therapy and drug delivery. These procedures require obstacle-free paths
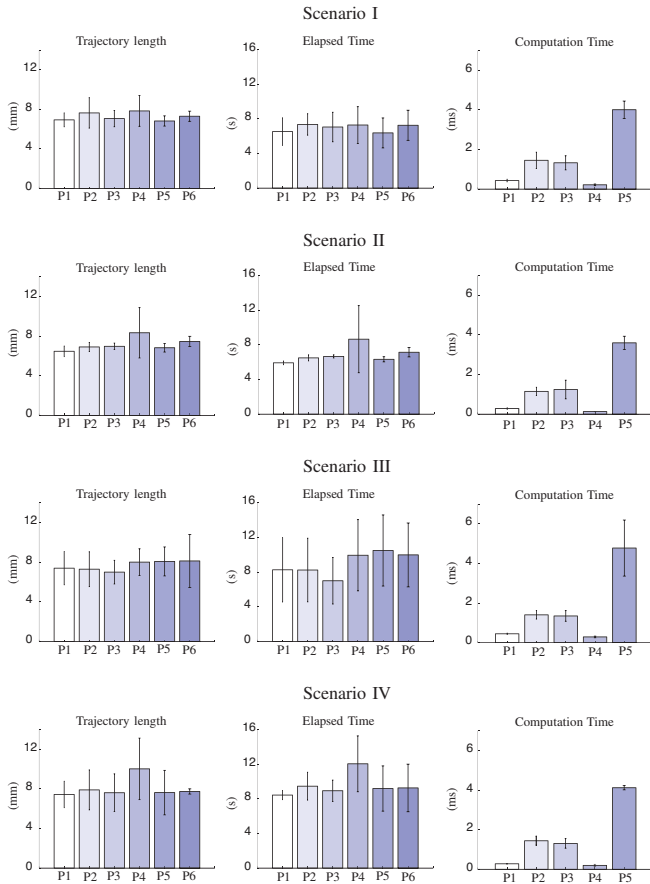
Fig. 6. For each path planner we evaluate the length of the trajectory performed by the microparticle, the time to reach the goal, and the computation time (mean and standard deviation are computed over seven trials). In each graph, each bar represents a different planner, from left to right: A* with quadtree (P1), A* with uniform grid (P2), D* Lite (P3), Artificial Potential Field (P4), Probabilistic Roadmap (P5) and Rapidly-exploring Random Tree (P6). In the graphs of the computation time we do not report the Rapidly-exploring Random Tree (P6) since it has a fixed computation time of 100 ms.

to autonomously move these agents toward a target/goal area while avoiding collisions. Obstacle-free paths can be provided by path planning algorithms. In order to help future researchers to choose the most suitable planning approach, in this paper we perform an experimental comparison of six planning algorithms applied to the motion control of paramagnetic microparticles. The algorithms are chosen among the families of deterministic (A* with quadtree, A* with uniform grid, D* Lite, APF) and probabilistic (PRM, RRT) planners. Statistical analysis reveals equivalence between almost all the considered planners in terms of trajectory length and completion time. Statistical differences are found between D* Lite and APF and between A* with quadtree and APF. Statistical analysis of our experimental data reveals that there are almost no statistical differences between deterministic and probabilistic planners. The main reason can be found in the fact that the configuration space is low-dimensional and the simple dynamic model of the microparticle. From our experiments, A* with quadtree and APF show the lowest computation time.

In future work, we plan to extend the proposed evaluation by considering moving targets and more complex environments. We will extend our results to a 3D setup. Moreover, we will consider clinically-relevant scenarios: the path planners will be evaluated in scenarios in combination with clinical imaging modalities which provide them with the position of the micro-sized agents in situations where visual feedback cannot be provided via a microscope system.

REFERENCES

[1] I. S. M. Khalil, V. Magdanz, S. Sanchez, O. G. Schmidt, and S. Misra. The control of self-propelled microjets inside a microchannel with time-varying flow rates. *IEEE Transaction on Robotics*, 30(1):49–58, 2014.

[2] I. S. M. Khalil, H. C. Dijkslag, L. Abelmann, and S. Misra. MagnetoSperm: A microrobot that navigates using weak magnetic fields. *Applied Physics Letters*, 104(22):223701, 2014.

[3] I. S. M. Khalil, M. P. Pichel, L. Abelmann, and S. Misra. Closed-loop control of magnetotactic bacteria. *International Journal of Robotic Research*, 32(6):637–649, 2013.

[4] I. S. M. Khalil, J. D. Keuning, L. Abelmann, and S. Misra. Wireless magnetic-based control of paramagnetic microparticles. In *Proceedings IEEE International Conference on Biomedical Robotics and Biomechatronics*, pages 460–466, 2012.

[5] I. S. M. Khalil, V. Magdanz, S. Sanchez, O. G. Schmidt, and S. Misra. Precise localization and control of catalytic Janus micromotors using weak magnetic fields. *International Journal of Advanced Robotic Systems*, 12:2, 2015.

[6] I. S. M. Khalil, V. Magdanz, S. Sanchez, O. G. Schmidt, and S. Misra. Biocompatible, accurate, and fully autonomous: a sperm-driven micro-bio-robot. *Journal of Micro-Bio Robotics*, 9(3-4):79–86, 2014.

[7] M. Sitti. Microscale and nanoscale robotics systems: characteristics, state of the art, and grand challenges. *IEEE Robotic & Automation Magazine*, 14(1):53–60, 2007.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transaction on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[9] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, 1985.

[10] W. Sun, I. S. M. Khalil, S. Misra, and R. Alterovitz. Motion planning for paramagnetic microparticles under motion and sensing uncertainty. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 5811–5817, 2014.

[11] S. Koenig and M. Likhachev. D* Lite. In *Proceedings National Conference on Artificial Intelligence*, pages 476–483, 2002.

[12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transaction on Robotics*, 12(4):566–580, 1996.

[13] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 1, pages 113–120 vol.1, 1996.

[14] S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept., Iowa State University, 1998.

[15] S. Fusco, M. S. Sakar, S. Kennedy, C. Peters, R. Bottani, F. Starsich, A. Mao, G. A. Sotiriou, S. Pané, S. E. Pratsinis, D. Mooney, and B. J. Nelson. An integrated microrobotic platform for on-demand, targeted therapeutic interventions. *Advanced Materials*, 26(6):952–957, 2014.

[16] David H Gracias. Stimuli responsive self-folding using thin polymer films. *Current Opinion in Chemical Engineering*, 2(1):112–119, 2013.

[17] Evin Gultepe, Jatinder S Randhawa, Sachin Kadam, Sumitaka Yamanaka, Florin M Selaru, Eun J Shin, Anthony N Kalloo, and David H Gracias. Biopsy with thermally-responsive untethered microtools. *Advanced Materials*, 25(4):514–519, 2013.

[18] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

[19] S. Koenig, M. Likhachev, and D. Furcy. Lifelong Planning A*. *Artificial Intelligence*, 155(1-2):93–146, 2004.

[20] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3310–3317, 1994.

[21] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.

[22] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, pages 3–10, 1987.

[23] G. B. Limentani, M. C. Ringo, F. Ye, M. L. Bergquist, and E. O. McSorley. Beyond the t-test: statistical equivalence testing. *Analytical Chemistry*, 77(11):221–226, 2005.