<table>
<thead>
<tr>
<th>Title</th>
<th>CVT-based 2D motion planning with maximal clearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Zheng, L; Choi, YK; Liu, X; Wang, WP</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2011</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/140007">http://hdl.handle.net/10722/140007</a></td>
</tr>
<tr>
<td>Rights</td>
<td>IEEE International Conference on Robotics and Automation Proceedings. Copyright © IEEE, Computer Society.; This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
Abstract—Maximal clearance is an important property that is highly desirable in multi-agent motion planning. However, it is also inherently difficult to attain. We propose a novel approach to achieve maximal clearance by exploiting the ability of evenly distributing a set of points by a centroidal Voronoi tessellation (CVT). We adapt the CVT framework to multi-agent motion planning by adding an extra time dimension and optimize the trajectories of the agents in the augmented domain. As an optimization framework, our method can work naturally on complex regions. We demonstrate the effectiveness of our algorithm in achieving maximal clearance in motion planning with some examples.

I. INTRODUCTION

Motion planning of an agent is to determine a route parameterized in time between the given starting and ending positions. Motion planning of multiple agents, also called multi-agent motion planning, is more challenging since it is required that no two agents collide with each other while moving along their routes. Multi-agent motion planning finds a spectrum of applications in robotics, artificial intelligence, control theory, computer simulation and computer animation. We will propose in this paper a novel approach to handle this problem.

In a typical multi-agent motion planning scenario, the starting and ending positions of some agents (e.g., robots) to be deployed in a region are given and a route for each agent from its starting to its ending position is to be found. Due to the uncertainty of agents [1], it is often required that their routes to be of maximal clearance, that is, the agents are required to be as far as possible from each other during motion. Other major factors under consideration include the smoothness of routes, the speeds as well as the deployment time of the moving agents.

A popular approach in the literature utilizes the generalized Voronoi diagram to find a maximal clearance route for an agent by treating all other agents as static ones [2]. However, this method only provides the best local movement at a certain time instant, while ignoring the coherence between two consecutive time instants and therefore the global trajectory of an agent. This kind of approach may generally lead to zig-zag and disorganized motions.

Centroidal Voronoi tessellation (CVT) is a special Voronoi tessellation in which each site of an Voronoi cell coincides with the centroid of the cell. For a domain of uniform density, the sites of a CVT are evenly distributed. In this paper, we exploit this favourable property of a uniform point distribution given by a CVT and devise an algorithm for multi-agent motion planning. We formulate the motion planning problem as a CVT optimization problem and propose an energy function which takes into account route smoothness, motion speeds and the deployment time. By adding a time dimension to the original path planning domain, our CVT framework naturally leads to a globally planned maximal clearance trajectories.

II. RELATED WORKS

There is extensive literature on the topic of motion planning. Most prior work on multi-agent motion planning can be classified into centralized and decentralized planning. Centralized planners treat all agents as a single multi-body robot, which generally provide a complete solution and allow global optimization, but however suffer from inefficiency due to the high dimensionality of the configuration spaces. Decentralized planners, on the other hand, utilize distributed methodology, which first plan for an initial path for each robot independently which is collision-free with respect to the obstacles only and then consider the spontaneous interactions among the robots by varying the velocities along their paths to avoid collisions. They are in general more efficient but completeness is not assured [3]. Our proposed method is a kind of centralized planning as we consider all agents at the same time to achieve an optimal planning.

We refer to the texts by LaValle [1] and by Choset et al. [4] and the comprehensive surveys from López et al. [5], Kavraki and LaValle [6], and Lindemann and LaValle [7] for details. We focus here only on trajectory planning, maximal clearance and CVT based navigation, which are related to our work.

A. Trajectory planning

In robotics, trajectory planning refers to the problem of determining both a route and a velocity function for a robot, and is different from path planning as it is parametrized by time [8]. Decoupled trajectory planning works in a similar way as decoupled planning, while direct trajectory planning generates a path directly in the state space, which is usually hard to solve analytically and therefore nonlinear optimization or grid-based search methods are generally involved [4].
Roque and Doering [9] performed trajectory planning using Voronoi roadmaps to generate the shortest feasible path with maximal clearance. Kant [10] proposed to use path-time space for decoupled trajectory planning, where time was added as an extra dimension as to the original domain. We method also works in the path-time space, but we adopt a direct approach with nonlinear optimization.

B. Maximal clearance

Uncertainty of moving agents will increase the probability of collision if their paths are too close to each other. o´Dúnlaing et al. first used generalized Voronoi diagram (GVD) to compute a maximal clearance path for a disc and a ladder using planar and 3-dimensional Voronoi diagrams, respectively [2]. Geraerts et al. enlarged the pre-planned path clearance by retracting it to the medial axis as a post processing step [11], and further applied the same idea for robots in high-dimensional configuration spaces [12]. Geraerts presented a data structure called “Explicit Corridor Map” based on GVD from which a shortest path is generated with adjustable amount of clearance in real-time [13]. Champagne and Tang used 2D GVD on environment maps for planning maximal-clearance paths of crowds comprising groups of agents [14]. Pettré et al. obtained maximal clearance paths in a multilayered and uneven domain by means of a navigation graph constructed from the Voronoi diagram [15]. Bhattacharya and Gavriloiva proposed an algorithm to generate a path that can keep a specific value of minimum clearance from obstacles [16], [17]. Sud et al. presented a maximal clearance method for global path planning for multiple agents, using the multi-agent navigation graph (MaNG) which combines the first- and second-order Voronoi graphs [18]. In general, GVD-based methods can well handle scenarios with static obstacles. For situations with dynamic obstacles or agents, these methods are not guaranteed to converge and may result in oscillating motions.

C. CVT-based navigation

CVTs have been used in many fields such as computational geometry, numerical PDEs, mesh-free computation, image segmentation, surface discretization, etc. Readers may refer to [19], [20] for more details about the theory and applications of the CVT.

Recently, there are some literature extending the application of CVT to robot navigation. Cortés et al. proposed to use continuous- and discrete-time Lloyd descent methods for coverage control and coordination of autonomous robots for distributed sensing [21]. Pimenta et al. [22] further extended [21] to cover also non-convex domains with a team of heterogeneous mobile sensors of finite sizes. Chen et al. [23] used CVT to generate crowd behavior of mobile robots for coverage control and multiple-target tracking. CVT-based approach has also been used in other sensing applications, e.g., diffusion control, for distributing networked mobile robots [24].

III. PROBLEM DEFINITION

We consider the maximal clearance trajectory planning for multiple agents in $\mathbb{R}^2$. Suppose that we are given $k$ agents and their starting position $p_i^0 \in \mathbb{R}^2$ and ending position $p_i^s \in \mathbb{R}^2$, $i \in \{1, \ldots, k\}$. We assume that each agent is a point agent which takes zero area. A maximal clearance trajectory planning in this setting is to assign a path as a function of time to each agent $i$ from $p_i^0$ to $p_i^s$, so that the agents are as far from each other as possible at each time instant. In this section, we present a solution to the problem in which the resulting path of each agent $i$ is given as a piecewise linear curve defined by an ordered set of $n+2$ points $\{p_j^i\}_{j=0}^{n+1}$, which we called the checkpoints, with $p_0^i = p_i^0$ and $p_{n+1}^i = p_i^s$. We suppose that all agents start moving at $t = 0$. Moreover, we require that each agent $i$ takes equal time increment $\Delta t_i$ to move from each checkpoint $p_j^i$ to $p_{j+1}^i$, for $j = 0, \ldots, n$ and $i \in \{1, \ldots, k\}$, that is, the checkpoints $\{p_j^i\}_{j=0}^{n+1}$ are equally sampled with respect to time. Each checkpoint $p_j^i$ is therefore associated with a time $t_j = j\Delta t_i$, which is the time at which agent $i$ arrives at $p_j^i$. The position of agent $i$ in the interval $[t_j, t_{j+1}]$ is then determined by linearly interpolating the checkpoints $p_j^i$ and $p_{j+1}^i$.

We note here that our solution to maximal clearance trajectory planning for multiple agents aims at achieving maximal clearance at the checkpoints $\{p_j^i\}_{j=0}^{n+1}$ only. It means that the two agents can be in close proximity or even in collision when one of them is not at the checkpoints. Nevertheless, the time span of this “error window” can be reduced by using more checkpoints (i.e., with a larger $n$) to sample a path.

IV. KEY IDEA

In this section, we shall present the basic idea of how we solve the problem of maximal clearance trajectory planning for multiple agents. Let us consider a path-time domain $D = \mathbb{R}^2 \times T$, where $T = [t_0, t_1] \subset \mathbb{R}$ (Fig. 1). Each point in $D$ is a pair $(p, t)$ representing a position $p \in \mathbb{R}^2$ associating with a time $t$. A trajectory $g(t) : T \rightarrow \mathbb{R}^2$ is then given by a space curve (or what we call a 3D path) $C : (g(t), t)$ in $D$. A trajectory planning problem in $\mathbb{R}^2$ can be transformed to a 3D path planning problem in $D$ such that:

- two trajectories in $\mathbb{R}^2$ are collision-free if and only if their corresponding 3D paths in $D$ do not intersect; also, maximal clearance trajectories can be achieved by designing 3D paths in $D$ which are as far away from each other as possible, in the following two senses. Firstly, maximal geometric clearance is realized so that two agents will be as far apart as possible at a particular time $t$. Secondly, maximal temporal clearance can also be attained so that the arrival time difference between two agents at a particular location is as large as possible.

We represent a 3D path $C$ in $D$ as a piecewise linear curve connecting a set of equally sampled points $\{(p_j, t_j)\}_{j=0}^{n+1}$ against the time axis. We also call these sample points the checkpoints where the context is clear. Given $k$ paths $\{C_i\}_{i=1}^k$ in $D$, our central idea is to distribute the checkpoints
Fig. 1. The path-time domain \( \mathbb{D} = \mathbb{R}^2 \times T \).

\[
\{(p^i_j, t^i_j)\}_{j=0}^{n+1}, i = 1, \ldots, k, \text{ in } \mathbb{D} \text{ under some pre-specified constraints}
\]

The smoothness of a 3D path is the smaller \( 1 - \cos \theta_j \) ensures that the function \( S_{\text{path}} \) is nonnegative and nondecreasing.

\[
\sum_{i=1}^{n} \int_{V_{i}} \|x - p_i\|^2 \, d\sigma
\]

where \( d\sigma \) is the differential area element of \( \Omega \). A local minimizer of \( F(X) \) therefore yields a CVT of \( \Omega \).

CVT is known to evenly distribute a set of points (i.e., the sites) in a domain. We therefore use CVT to distribute the checkpoints of the paths evenly in \( \mathbb{D} \) which in turn achieves maximal clearance as discussed in Section IV. Given the checkpoints \( z^i_j = (p^i_j, t^i_j) \) in \( \mathbb{D}, i = 1, \ldots, k, j = 1, \ldots, n \), the CVT energy of all checkpoints on all the paths in \( \mathbb{D} \) is defined as:

\[
F_{\text{cvt}} = \sum_{i=1}^{k} \sum_{j=1}^{n} \int_{V_{z^i_j}} \|z - z^i_j\|^2 \, d\sigma,
\]

where \( V_{z^i_j} \) is the Voronoi region of \( z^i_j \) and \( d\sigma \) is the differential volume element of \( \mathbb{D} \).

B. Trajectory Smoothing

CVT tends to well distribute the checkpoints in the path-time domain to give us maximal clearance, but it does not take into account what the trajectories should behave like. A reasonable requirement is that an agent should take a path involving fewer turns to the destination, which means that zig-zag paths are not desirable. Also, abrupt changes in the speed of motion of an agent should also be avoided so as to maintain a natural movement. We therefore introduce two smoothing terms: path smoothness and speed smoothness. The two factors are considered separately so that we may adjust their individual contribution to produce desired result.

1) Path smoothness: The smoothness of a 3D path is defined by the smoothness of its projection onto the 2D \( x \)-\( y \) plane, which is measured by an approximation of the sum of its turning angles squared. Specifically, at any checkpoint \( p_j, j = 1, \ldots, n \), the turning angle \( \theta_j \) of a path is the smaller angle between the vectors \( p_{j-1}p_j \) and \( p_jp_{j+1} \). Then the path smoothness term is defined as:

\[
S_{\text{path}} = \sum_{i=1}^{k} \sum_{j=1}^{n} (1 - \cos \theta_j),
\]

where

\[
\cos \theta_j^i = \frac{p^i_{j-1}p^i_j \cdot p^i_jp^i_{j+1}}{\|p^i_{j-1}p^i_j\|\|p^i_jp^i_{j+1}\|}.
\]

The quantity \( 1 - \cos \theta_j \) ensures that the function \( S_{\text{path}} \) is nonnegative and nondecreasing.
2) Speed smoothness: The speed variation along a trajectory can be observed by projecting a path to the xt-planes in \( D \). In the same spirit as in how path smoothness is determined above, speed smoothness is measured approximately by the sum of the turning angles squared. Let the coordinates of \( p_j \) be \((x_j^*, y_j^*)\) \( \in \mathbb{R}^2 \). Then we define

\[
S^x_{\text{speed}} = \sum_{i=1}^{k} \sum_{j=1}^{n} (1 - \cos \theta^i_j),
\]

\[
S^y_{\text{speed}} = \sum_{i=1}^{k} \sum_{j=1}^{n} (1 - \cos \phi^i_j),
\]

where

\[
\cos \theta^i_j = \frac{v_{i,j-1} \cdot v_{i,j} \cdot v_{i,j+1}}{\|v_{i,j-1}\| \|v_{i,j}\| \|v_{i,j+1}\|},
\]

\[
\cos \phi^i_j = \frac{w_{i,j-1} \cdot w_{i,j} \cdot w_{i,j+1}}{\|w_{i,j-1}\| \|w_{i,j}\| \|w_{i,j+1}\|},
\]

\[
v^i_j = (x^i_j, j \Delta t_i),\]

\[
w^i_j = (y^i_j, j \Delta t_i).
\]

C. Total time taken for motion

It is often useful to be able to control the amount of time required for an agent to complete an entire path. In some applications, one may require that the agents be able to arrive at the destinations as soon as possible. Hence, we introduce an energy term for such a control which is defined as:

\[
T_{\text{path}} = \sum_{i=1}^{k} \Delta t_i.
\]

By minimizing \( T_{\text{path}} \), we aim to navigate the agents to move as fast as possible.

D. Speed constraint

It is natural to require that all agents be not moving with a velocity that is less than zero or exceeds a user-pre-specified limit, denoted as \( \text{MaxSpeed} \). The constraint is imposed by having

\[
C_{\text{speed}} : 0 \leq \frac{\|p_{i-1}^j - p_i^j\|}{\Delta t_i} \leq \text{MaxSpeed},
\]

\( i = 1, \ldots, k; \ j = 1, \ldots, n \).

VI. ALGORITHMIC DETAILS

In this section, we present a solution to formulation (1) and also discuss how we shall handle the end points of the paths in the path-time domain \( D \) which determines the journey time of the agents. The detailed steps of our algorithm are given in Algorithm 1.

A. Solution to nonlinear optimization with inequality constraints

We use the Powell-Hestenes-Rockafellar (PHR) method [25], a well-known augmented Lagrangian algorithm, to solve the nonlinear optimization problem with inequality constraints given in (1). The constraint \( C_{\text{speed}} \) is then incorporated as a penalty function as follows:

Minimize \( M(X, \lambda_{C_{\text{speed}}}, \sigma) = f(X) + c(X, \lambda_{C_{\text{speed}}}, \sigma) \) (2)

where

\[
X = (x_i), \ x_i = (p_i^1, \ldots, p_i^n, \Delta t_i),
\]

\[
c(X, \lambda_{C_{\text{speed}}}, \sigma) = \frac{1}{2\sigma} \sum_{i=1}^{k} \sum_{j=1}^{n} (\max(0, \lambda_{C_{\text{speed}}}) + \sigma C^i_j(X))^2
\]

where \( \lambda_{C_{\text{speed}}} \) is the Lagrangian multiplier vector for \( C_{\text{speed}} \), \( \sigma \) is the penalty parameter and \( C^i_j(X) = \frac{\|p^i_{j-1} - p^i_j\|}{\Delta t_i} - \text{MaxSpeed} \). Details on updating of parameters \( \lambda_{C_{\text{speed}}}, \sigma \) and the terminal condition can be found in [25].

Now that we have converted the constrained optimization problem to an unconstrained one, we adopt the L-BFGS (limited-memory BFGS) algorithm [26], a quasi-Newton method proposed by Liu et al. [27] for CVT computation, to minimize \( M(X, \lambda_{C_{\text{speed}}}, \sigma) \).

B. Determining the total journey time

Minimizing the CVT energy function tends to evenly distribute the checkpoints in the domain \( D = \mathbb{R}^2 \times T \), which means that given a fixed time interval \( T \), there is most likely an agent who takes the maximum possible time to reach his destination (without considering the effect of the energy term \( T_{\text{path}} \)). The problem therefore lies in how a proper time interval for the domain \( D \) is selected. Choosing a too small interval may result in nonexistence of solution, while a too large interval may render a journey completion time much longer than necessary. Our solution is as follows. The initial time interval is taken as the maximum of the shortest journey time of all agents, assuming that they head straight towards their destination with the maximum speed. Recall that in the optimization framework, the speed constraint is only guaranteed for the journey from \( p^i_0 \) up to \( p^i_n \), not including the destination for each agent, which is \( p^i_{n+1} \). Hence, after each iteration of energy minimization, we check if each agent \( i \) is able to move from the last checkpoint \( p^i_n \) to the destination \( p^i_{n+1} \) within the speed limit in a time interval \( \Delta t_i \). A valid solution is found only if there is no agent violating the speed limit in this regard; otherwise the time interval \( T \) needs to be incremented so that the “speeding agents” should all observe the speed limit. In the latter case, the optimization will then be carried out again in order to seek for a better solution.

C. The algorithm

Algorithm 1 details the procedure of our method. We first determine in step 1 the initial path (which are the shortest paths towards the destination), as well as the time increment \( \Delta t_i \) for each agent. In step 2 we construct the domain \( D \), and perform the optimization process in steps 3 and 4. The last step serves to update the time interval of \( D \) to allow a valid solution.
Algorithm 1: CVT-based maximal clearance trajectory planning in $\mathbb{R}^2$

Input:
Number of agents $k$;
Number of checkpoints on each path $n$;
Starting points $\{p_0^i\}$ and ending points $\{p_{n+1}^i\}$, $i=1,\ldots,k$.

STEP 1 For $i=1,\ldots,k$, set $\Delta t_i = \frac{||p_i^i p_{i+1}^i||}{(n+1)\text{Max speed}}$.
Also, set $p_j^i = p_0^i + \sum_{j=1}^{i-1} p_j^i p_{j+1}^i$, $j=1,\ldots,n$.

STEP 2 Let $D = \mathbb{R}^2 \times T$, where $T = [0,\max_{i}((n+1)\Delta t_i)]$.

STEP 3 Minimizing Eq. (2) with L-BFGS method.

STEP 4 If terminal condition is satisfied, go to step 5; otherwise, update the penalty parameters and go to step 3.

STEP 5 Check if, for each $i=1,\ldots,k$, $\frac{p_j^i p_{j+1}^i}{\Delta t_j} \leq \text{Max speed}$.
If not, set $T = [0,\max_{i}((n+1)\Delta t_i)]$, goto step 2.
Otherwise, a valid solution is found and return all checkpoints.

VII. EXPERIMENTAL RESULTS

In this section, we shall give the implementation details and show some experimental results to demonstrate the effectiveness of our algorithm.

A. Implementation details

We use CGAL version 3.5.1 [28] and exact predicates inexact constructions kernel [29] to compute the Voronoi diagrams. We run all experiments on a workstation with an Intel Xeon 3.33 GHz CPU and 12GB RAM.

The four terms $F_{\text{cvt}}$, $S_{\text{path}}$, $S_{\text{speed}}$ and $T_{\text{path}}$ in the objective function (1) are generally of different magnitudes and are therefore first normalized. The three weights $\alpha_1$, $\alpha_2$ and $\alpha_3$ are then applied to the normalized terms.

B. Parameter tuning

The following two examples demonstrates the importance of the terms $S_{\text{path}}$ and $S_{\text{speed}}$, and how their weights can be tuned to achieve a balance between maximal clearance and the quality of paths. Fig. 2(a) (resp. (b)) is a result without path (resp. speed) smoothness control by setting $\alpha_1$ (resp. $\alpha_2$) to zero. Without path smoothness control, sharp turns appear in both paths of the two agents, as is shown in (a). It is also easy to see that the unsmoothed velocity leads to an undesirable zig-zag path in (b). Fig. 3 illustrates the effect of using different values of $\alpha_1$ and $\alpha_2$. An unacceptable result with $\alpha_1 = 0$ and $\alpha_2 = 0$ is shown in (a). Setting $\alpha_1$ and $\alpha_2$ with larger values keeps improving the path quality in terms of smoothness as shown in (b-d), but however sacrificing maximum clearance of the paths. We found that $\alpha_1 = 0.16$ and $\alpha_2 = 0.08$ is a suitable choice in a typical setting. Our testing shows that together with $\alpha_3 = 0.4$, the resulting trajectories also allow the agents to move to their ending positions as quickly as possible. Hence, these values are used in all experiments in this paper.

C. Results

Fig. 4 shows some examples with two agents. It can be seen that the two agents always keep clear of each other in the examples. The agents also keep the largest distance at midway of their journey. Fig. 5 shows examples with four agents and we can see that not only geometric but also temporal maximal clearance is achieved.

More complex scenarios are shown in Fig. 6 with 16 agents and Fig. 7 with 40 agents. We show the projection of the trajectories of the agents on $\mathbb{R}^2$, as three dimensional views cannot provide useful visualization in this case. The snapshots of the moving agents with their trails at six time stamps are given. From these snapshots, we can see that the agents are distributed evenly and their trajectories are also smooth. In Fig. 7(d), about 10 agents appear to be crowded in the central region of the square. The reason is that most agents choose to take a shorter path passing through the central region in order to more quickly reach the ending positions.
D. Other domains

Since the CVT energy function is defined on general two-dimensional domains, our algorithm can handle also complex domains naturally. An example with two agents in a circular domain is shown in Fig. 8. Another example with four agents in a non-simple domain is shown in Fig. 9. Such non-simple domains may also model scenarios with moving obstacles. For each moving obstacle, a cylinder is constructed by sweeping the obstacle along its trajectory in the path-time domain. The path-time domain $D$ for the optimization then becomes the augmented domain minus the cylinders of all the obstacles. Obstacles can naturally be avoided by placing initial checkpoints outside of the cylinders.

VIII. CONCLUSION

We propose a CVT-based framework for motion planning to achieve maximal clearance in a two-dimensional domain. Efficient optimization schemes are employed in our algorithm. Experiments show that our approach handles general domains and is robust. Similar to many other existing methods, we treat robots simply as points without area; although a reasonable approximation, this may lead to some errors in practice. In future, we will adapt our approach by computing the CVT energy function of robots with complex shapes directly to ensure accurate solutions. Our discrete sampling scheme for the trajectories does not assure non-collision of agents in between the checkpoints. One possible solution is to directly define the CVT energy function of the trajectory paths in three dimensions which will be further investigated.

REFERENCES

Fig. 7. An example with 40 agents.

(a) A domain with an obstacle.  (b) The trajectories.

Fig. 8. Two agents in a circular domain.

Fig. 9. Four agents in a non-simple domain.


