Pacific Graphics (2021)
M. Okabe, S. Lee, B. Wuensche, and S. Zollmann (Editors)

Short Paper

Maximum-Clearance Planar Motion Planning Based on Recent Developments in Computing Minkowski Sums and Voronoi Diagrams

Mingyu Jung and Myung-Soo Kim†

Dept. of Computer Science and Eng., Seoul National University, South Korea

Abstract

We present a maximum-clearance motion planning algorithm for planar geometric models with three degrees of freedom (translation and rotation). This work is based on recent developments in real-time algorithms for computing the Minkowski sums and Voronoi diagrams of planar geometric models bounded by G^1-continuous sequences of circular arcs. Compared with their counterparts using polygons with no G^1-continuity at vertices, the circle-based approach greatly simplifies the Voronoi structure of the collision-free space for the motion planning in a plane with three degrees of freedom. We demonstrate the effectiveness of the proposed approach by test sets of maximum-clearance motion planning through narrow passages in a plane.

CCS Concepts

• ***Computing methodologies*** → *Motion planning; planar geometric models; circle-based algorithm; maximum-clearance; Minkowski sum; Voronoi diagram; medial axis;*

1. Introduction

Motion planning has a long history of research and development, the main results of which are very well documented in several textbooks [Brady et al. (1983), Canny (1988), Choset et al. (2005), Latombe (2012), LaValle (2006), Lynch and Park (2017), Sacks and Joskowicz (2010), Scwhartz et al. (1987), Thurn et al. (2006)]. The concept of Configuration-space (C-space) is widely accepted as an important conceptual tool for the design and analysis of motion planning algorithms [Lozano-Pérez (1981), Lozano-Pérez (1983), Lozano-Pérez and Wesley (1979)], not only in robotics [Sacks and Joskowicz (2010)] but also in other motion-related application areas such as in computer graphics, animation, games, and virtual/augmented/mixed realities [Akenine-Möller et al. (2018)]. Nevertheless, an explicit construction of C-space obstacles in high-dimensional space is computationally prohibitive for motion planning with more than three degrees of freedom. Sampling-based methods (constructing probabilistic roadmaps in high-dimensional C-spaces) are thus dominating the design of motion planning algorithms in robotics [Kavraki et al. (1996), Thurn et al. (2006)].

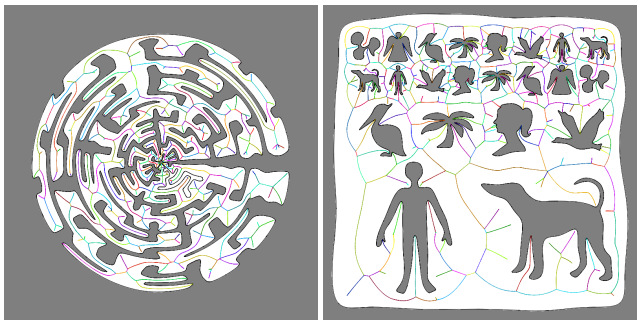
For some special motion planning problems in low-dimensional cases, however, a precise (and often partial) construction of C-space obstacles produces effective solutions to the given non-trivial geometric problems such as continuous contact (deformable) motions [Kim et al. (2014), Kim et al. (2016)]. In a hybrid motion planner, [Foskey et al. (2001)] demonstrated that the number of random samples can be reduced substantially by using a discrete approximation of the generalized Voronoi diagram for a 3D workspace.

In this paper, we revisit the rigid-body motion planning problem in a plane (with three degrees of freedom) with a few highly-efficient computational tools recently developed for the processing of planar geometric models bounded by G^1-continuous circular arcs (called arc-splines) [Meek and Walton (1995), Meek and Walton (1999), Sir et al. (2006)]. The computational advantages of using these circle-based tools are rather too many to articulate in a few words [Aichholzer et al. (2007)]. Here, we will mainly consider the efficiency and robustness issues in the construction of Voronoi diagrams and medial axes for planar objects with smooth boundaries [Aichholzer et al. (2009), Aichholzer et al. (2010), Lee et al. (2016)]. (Voronoi diagrams and medial axes are closely related to each other, as explained using some concrete examples at the end of Section 3; thus we will use these two terms almost interchangeably.) In the planning of maximum-clearance motions, the construction of Voronoi diagrams is known to be the most crucial step [Ó Dúnlaing and Yap (1985), Takahashi and Schiling (1989)].

Figure 1 shows two examples of medial axis construction for planar regions bounded by (a) 2994 and (b) 3906, respectively, cubic curvature-monotone curves (subdivided at x and y-extreme

† Corresponding author: mskim@snu.ac.kr

DOI: 10.2312/pg.20211382



(a) (b)

Figure 1: *Medial axis construction for complex planar regions.*

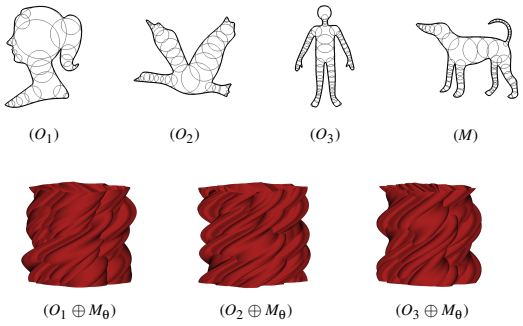
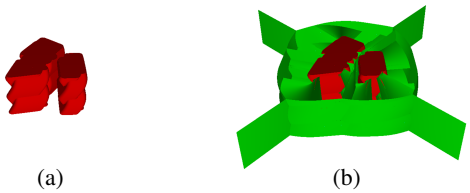


Figure 2: *The Minkowski sum computation for $O_i \oplus M_\theta$.*



(a) (b)

Figure 3: *Medial axis construction for planar slices of the C-space: (a) the C-space obstacle in the $xy\theta$-space, and (b) the medial surface (in green) constructed by stacking up the medial axes along the θ-axis direction.*

points). [Lee et al. (2016)] computed these medial axes highly efficiently, in (a) 90 ms and (b) 124 ms, respectively, which can also guarantee the correct topology of the branching structures. Using these medial axes, we can immediately extend the classical result of [Takahashi and Schiling (1989)] that had been developed for maximum-clearance motion planning based on a generalized Voronoi diagram for polygonal obstacles in a plane.

[Takahashi and Schiling (1989)] simplified the motion planning problem by covering the robot with a small number of circular disks. This approximation would errorneously block certain narrow passages due to relatively large error in the disk-coverage. We can resolve this limitation using a recent circle-based acceleration technique [Han et al. (2019)] for the construction of three-dimensional C-space obstacles. The medial surface will then be generated for these volumetric obstacles in the $xy\theta$-space to make the maximum-clearance motion planning more accurate and efficient than before.

Given a set of obstacles O_i and a moving object M, $(i = 1, \cdots, n)$, the C-space is generated by computing the following set [Lozano-Pérez and Wesley (1979)]:

$$\cup O_i \oplus (-M) = \cup O_i \ominus M = \cup\{\mathbf{p} - \mathbf{q} \mid \mathbf{p} \in O_i, \mathbf{q} \in M\},$$

where the Minkowski sum and difference are defined as follows

$$A \oplus B = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\},\ A \ominus B = \{\mathbf{a} - \mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\}.$$

When the moving object M is rotated to M_θ by angle θ, the corresponding C-space is generated by $\cup O_i \ominus M_\theta$. (Since $-M_\theta = M_{\theta+\pi}$, from now on, we will use $\oplus$ and $\ominus$ interchangeably, adding π to θ if necessary.) By stacking up these planar sets along the θ-axis, we can generate the volumetric C-space obstacles in the $xy\theta$-space (as shown in Figures 2 and 3).

For general non-convex objects, the construction of such a volumetric C-space had long been considered to be an extremely time-consuming task. Using the medial axes of O_i and M, in a recent work, [Han et al. (2019)] accelerated the construction of $\cup O_i \oplus M_\theta$. Figure 2 shows three examples of stacking up 3600 slices of the planar Minkowski sums $\cup O_i \oplus M_\theta$. The construction of all three volumetric C-space obstacles took less than one minute. For typical application scenarios of maximum-clearance motion planning, we use a much smaller number of θ-slices and the moving robot M is further approximated by a rotational sweep $\overline{M} = \cup M_\theta$ [Kim and Moon (1990)], where the union is taken on a small interval $\theta \in [0, \Delta\theta]$. As shown in Figure 3(a), we compute N $(= 2\pi/\Delta\theta)$ different slices of $CO_i = O_i \oplus \overline{M}_{j\Delta\theta}$, $(j = 0, \cdots, N-1)$, for each obstacle O_i. On each of these N slices, the medial axis for the exterior of $\cup CO_i$ is computed [Lee et al. (2016)]. Figure 3(b) shows a green surface that interpolates these medial axes, which is then used as a reference medial surface for the maximum-clearance motion planning.

The start and goal configurations of a moving robot are usually not located on the medial surface. The medial surface is dynamically expanded by adding certain local visibility maps around the start and goal configurations. The shortest path is then computed on the roadmap built around the medial surface. In Section 3, we discuss more technical details on smoothing and optimizing the path so as to maximize the clearance through narrow passages.

2. Previous Work

[Lozano-Pérez and Wesley (1979)] and [Lozano-Pérez (1981), Lozano-Pérez (1983)] converted the collision-free path planning problem for a moving robot among obstacles to a simplified point navigation problem in the C-space of transformed obstacles generated as the result of Minkowski sum operations. For a rigid-body motion in a plane, the C-space obstacles are bounded by non-convex ruled surfaces in the $xy\theta$-space, even when the moving robot and obstacles are simple (non-convex) polygons. [Avnaim et al. (1988)] developed the first algorithm for computing exact C-space obstacles for planar polygons. [Milenkovic et al. (2013a), Milenkovic et al. (2013b)] presented a robust C-space construction algorithm for planar objects bounded by circular arcs.

As shown in the examples of Figure 2, the C-space obstacles have many self-intersections, which correspond to double or multiple contacts between the robot and obstacles. By connecting these self-intersection curves in a correct topology, [Kim et al. (2014)] generated a roadmap for a continuous double (or multiple) contact motion in an environment where objects are bounded by C^1-continuous B-spline plane curves. [Kim et al. (2016)] further enhanced the roadmap for a deformable robot by adding extra routes in a partially extended C-space corresponding to the deformational degrees of freedom. The roadmap can be represented in a compact network of curve segments. However, the construction is quite time-consuming. [Han et al. (2019)] accelerated the computation of Minkowski sums by approximating planar objects by G^1-continuous circular arcs and filling the interior of each object with maximum touching circles sampled along the medial axis (see the interior circles of the four objects O_1, O_2, O_3, and M shown in Figure 2). The Minkowski sums are also bounded by circular arcs, but not necessarily with G^1-continuity, due to the self-intersections in the boundary curve.

[Kim and Moon (1990)] simplified the swept volume computation for a rotating object M_θ (about the origin in the xy-plane) by converting the problem to a translational motion along the θ-axis in the $r\theta$-plane. The boundary of the rotational sweep $\overline{M} = \cup_{0\le\theta\le\Delta\theta} M_\theta$ consists of some boundary curve segments of M and $M_{\Delta\theta}$, and additionally some circular arcs generated by rotating certain critical points on the boundary of M. Starting with a moving object M bounded by G^1-circular arcs, the rotational sweep $\overline{M}$ will be bounded by circular arcs, possibly with some sharp corners as the result of intersections among the circular arcs from the boundaries of M and $M_{\Delta\theta}$ and also those additionally generated from the critical points of M. The sharp corners can be treated as a circular arc of radius 0. Thus the C-space obstacles $\cup O_i \ominus \overline{M}$ can be generated by [Han et al. (2019)], again bounded by circular arcs possibly with some sharp corners.

[Aichholzer et al. (2009), Aichholzer et al. (2010)] presented an efficient and robust algorithm for computing the Voronoi diagram and medial axis of planar regions bounded by circular arcs. Based on the theory developed there, [Lee et al. (2016)] further accelerated the construction by considering curvature-monotone curve segments. Connected sequences of G^1-circular arcs with monotonically increasing/decreasing radii can also be treated as such curve segments. These results can be applied to the medial axis construction for the collision-free space in the exterior of the C-space obstacles $\cup O_i \ominus \overline{M}$.

3. Motion Planning Algorithm

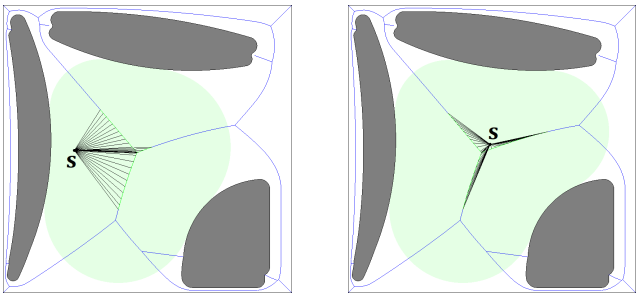
The construction of exact C-space obstacles (in the $xy\theta$-space) is quite time-consuming and numerically unstable even for the planar objects bounded by circular arcs. Following the slicing technique of [Lozano-Pérez (1981), Lozano-Pérez (1983)], in each restricted $xy\theta$-subspace in the θ-range: $j\Delta\theta \le \theta \le (j+1)\Delta\theta$, we slightly expand the C-space obstacles $O_i \ominus M_\theta$ by cylindrical volumes with cross-sections $O_i \ominus \overline{M}_{j\Delta\theta}$, where the rotational swept volume $\overline{M} = \cup_{0\le\theta\le\Delta\theta} M_\theta$. (Thus $\overline{M}_{j\Delta\theta}$ is the rotation of $\overline{M}$ by angle $j\Delta\theta$.) This approximation slightly shrinks the free-space; however, the approximation error introduced in this simplification process is not as serious as in the case of continuous contact motion planning such as those discussed in [Kim et al. (2014), Kim et al. (2016)]. As a result, we essentially reduce the construction of volumetric C-space obstacles to a simple problem of computing $N = 2\pi/\Delta\theta$ slices of planar C-space obstacles, for $j = 0, \cdots, N-1$. On each slice, we constuct the medial axis of the planar free-space.

3.1. Roadmap construction

The construction of maximum-clearance motion path starts with three basic components. The first and last components are projection line segments $\mathbf{l}_s$ and $\mathbf{l}_g$ from the start and goal configurations (denoted as $\mathbf{s} = (x_s, y_s, \theta_s)$ and $\mathbf{g} = (x_g, y_g, \theta_g)$) to their projections $\mathbf{m}_s$ and $\mathbf{m}_g$ on the medial axes of their respective θ-slices. The main component γ_m is the shortest path connecting the two projection points $\mathbf{m}_s$ and $\mathbf{m}_g$ on the medial surface (shown as a green surface in Figure 3(b)). Our algorithm makes local changes to the main part γ_m so that it becomes smoother. After that, we bend the other two components ($\mathbf{l}_s$ and $\mathbf{l}_g$) so that they are connected a bit more smoothly to the main part γ_m. In fact, this is essentially the same as the conventional retraction method for maximum-clearance motion planning [Ó Dúnlaing and Yap (1985), Takahashi and Schiling (1989)]. However, there are some technical problems – the start and goal configurations are often located in relatively open areas. The conventional retraction methods usually start with the projections of these configurations to the medial surface, thus increasing their distances from obstacles. However, this may construct an unnecessarily long paths for the moving robot. Thus we discuss how to add local visibility maps (around the start and goal configurations) to the global roadmap constructed on the medial surface. Note that these local maps are removed once the start and goal configurations are updated for the next stage of motion planning. In the rest of this paper, we employ the shortest path $\gamma(t)$, $(0 \le t \le 1)$, constructed between the start and goal configurations ($\mathbf{s}$ and $\mathbf{g}$), on the expanded roadmap built around the medial surface.

At the start configuration $\mathbf{s} = (x_s, y_s, \theta_s)$, we consider a vertical line segment (passing through $\mathbf{s}$) along the θ-axis: $\mathbf{l}(\theta) = (x_s, y_s, \theta)$, $(\underline{\theta}_s < \theta < \overline{\theta}_s)$, where the rotating robot M_θ with its center fixed at the start location (x_s, y_s) makes no collision with the obstacles. For each θ_j-slice ($\theta_j = j\Delta\theta \in [\underline{\theta}_s, \overline{\theta}_s]$), we project (x_s, y_s, θ_j) to $(x_{m_j}, y_{m_j}, \theta_j)$ on the medial axis of the θ_j-slice. In the neighborhood of the projected point $(x_{m_j}, y_{m_j}, \theta_j)$, we sample vertices on the medial axis and connect each of them to the configuration (x_s, y_s, θ_j) when certain local visibility conditions are met, namely, when the rotated robot M_{θ_j} can translate from (x_s, y_s, θ_j) to the corresponding vertex without colliding with obstacles.

The local visibility test can be done very efficiently using the medial axis, where each vertex (x_m, y_m, θ_j) has at least two different closest points on the C-space obstacles on the θ_j-slice. Taking (x_m, y_m, θ_j) as the circle center and the distance to the closest points as the radius r_m, we can construct a maximum touching circle whose interior contains none of the C-space obstacles on the θ_j-slice. When the distance between (x_m, y_m, θ_j) and (x_s, y_s, θ_j) is less than the radius $r_m > 0$, the linear translation of M_{θ_j} from (x_s, y_s, θ_j) to (x_m, y_m, θ_j) is guaranteed to be collision-free. Thus the local visibility can be tested in a straightforward manner. In Figure 4, the green area is the union of all maximum touching circles that con-

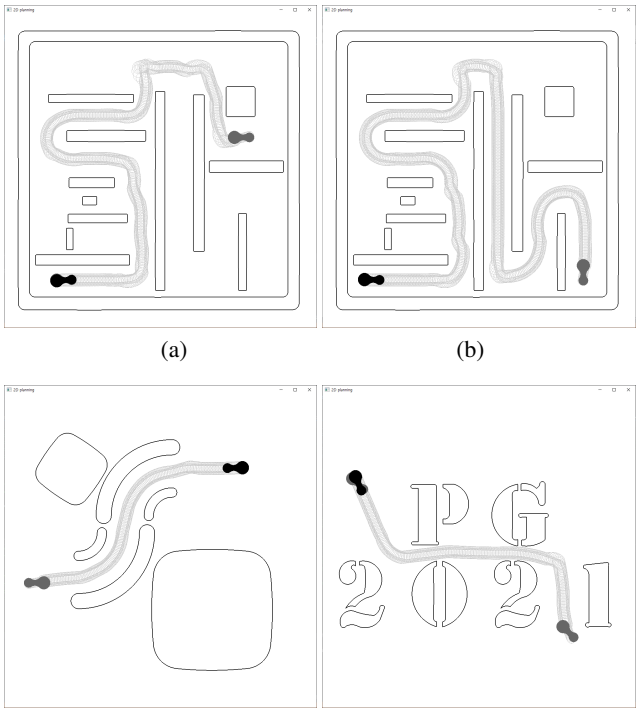


(a) (b)

Figure 4: *Adding local visibilty from the start configuration* **s**.

tain the point (x_s, y_s, θ_j) in their interiors. The local visibility map is generated by connecting the point (x_s, y_s, θ_j) to the center of each of the circles (corresponding to the vertices sampled).

We assume the medial surface is tesselated by sampled vertices along the medial axis on each θ_j-slice. The connection between adjacent vertices by edges may introduce interference with the C-space obstacles. Moreover, we also connect nearby vertices between adjacent θ_j-slices, which may introduce some interferences as well. The interference can also be tested very efficiently using the maximum touching circles associated with the sampled vertices on the medial axes.



(a) (b) (c) (d)

Figure 5: *Maximum-clearance paths through narrow passages.*

3.2. Path smoothing and optimization

Let $C(t)$ denote the projection of $\gamma(t)$ to the xy-plane. As shown in the examples of Figure 5 for the motion paths through narrow passages, it is important to make the swept volume of the moving robot smaller so that the robot can maintain maximum clearance from nearby obstacles in both sides. The major axis of the moving robot is almost lined up with the tangent direction $C'(t)$ in narrow passages. Thus we mainly focus on the smoothing of $C(t)$ in narrow passages, while making the robot orientation more or less dependent on $C'(t)$ and thus on the trajectory $C(t)$ itself. In open spaces with relatively large clearance, the orientation control is less important and the robot can even flip its orientation there so that it can end up with the orientation given at the goal configuration. Note that the orientation flipping is the result of a connected path construction on the roadmap.

The shortest path γ_m is initially constructed on the discretized roadmap. The projected curve $C(t)$ is thus a polygonal path in the xy-plane. Using the distribution of the discrete curvatures of $C(t)$, we can segment the curve $C(t)$ into several pieces, each of which is then approximated by a G^1-continuous connected sequence of circular arcs. An important advantage of using arc-splines is in the simplicity of measuring the maximum deviation (Hausdorff distance) error between $C(t)$ and its approximation with arc-splines, which is quite important in applications dealing with maximum clearance [Kim et al. (2010)].

Let $C_a(s)$ denote the arc-spline approximation of $C(t)$, where s is the arc-length parameterization of C_a. (Note that the arc-length reparameterization is non-trivial for general curves [Farouki and Sakkalis (1991)]; but, it is easy for arc-splines, simply by adding the lengths of their constituent circular arcs.) When the moving robot M has its major axis direction making a constant angle with the tangent direction $C_a'(s)$ of the arc-spline path, the swept volume of M along $C_a(s)$ can be computed using the rotational sweep volume technique of [Kim and Moon (1990)]. In particular, we can also compute the clearances from both sides of the narrow passage quite precisely, since the rotational sweep volume of M is also bounded by circular arcs. (Note that the rotation of M is now about the center of a circular arc of $C_a(s)$, but not around the center of M.) We can maximize the local clearance by adjusting the center and radius of each circular arc of $C_a(s)$ in the neighborhoods of close proximity locations. The swept volume of M is often minimized when the major axis direction of M is aligned to the tangent direction $C_a'(s)$. Thus we make the motion of M completely decided by the arc-spline $C_a(s)$ through narrow passages.

[Kim et al. (2010)] presented a simple technique for constructing the distance field from a set of circular arcs, which is based on the GPU hardware rendering of truncated circular cones to the graphics depth buffer. This technique was originally developed for an efficient computation of Hausdorff distance for measuring the maximum deviation between two plane curves. The distance field can also be used for accelerating the computation of minimum distances in close proximity locations. In particular, we can render the distance fields from both sides of each narrow passage into the GPU depth buffer and read the changes in minimum distance, while walking along the swept volume boundary of M, for the detection of clearance value and the corresponding location.

Figure 6 shows three examples of computing the distance field

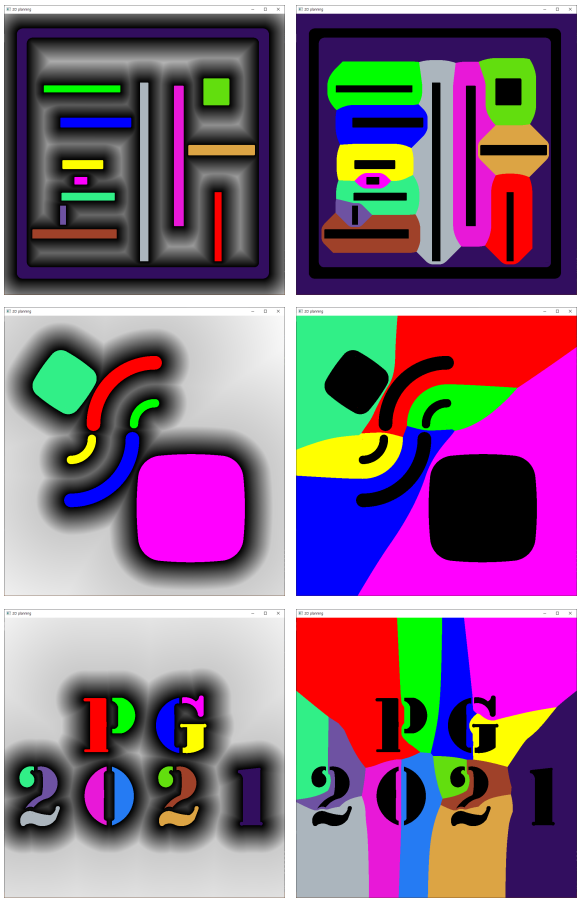


Figure 6: *Distance fields and Voronoi cell decompositions.*

(shown in the left column) and the Voronoi diagram (in the right column) for obstalces in the plane. In the Voronoi diagrams, the Voronoi cells for obstacles are shown in different colors. The common boundary curves of adjacent Voronoi cells are the collection of equi-distance points to the corresponding obstacles. These boundary curves belong to the medial axis for the exterior region of all obstacles. The medial axis is the set of exterior points that have at least two different closest points on the obstacle boundaries. The two different closest points may be on the boundary of the same obstacle; in this case, note that the obstacle must be non-convex. Thus the medial axis may contain some points in the interior of Voronoi cells for non-convex obstacles. These points are on the bisector of the obstacle boundary on the concave side. By adding (or removing) the self-bisector curves to the Voronoi diagram (or from the medial axis), we can construct the medial axis (or the Voronoi diagram). Thus the two concepts and their constructions are closely related to each other.

4. Experimental Results

We have implemented our maximum-clearance motion planning algorithm in C++ on an Intel Core i7-6700K 4.0 GHz PC with a 32 GB main memory. To demonstrate the effectiveness of the proposed approach, we have tested the algorithm to the three examples shown in Figure 6, some motion planning results of which are also shown in Figure 5.

Table 1: *The construction time (in ms) for the C-space, the roadmap, the polygonal path $\gamma(t)$, and the arc-spline path $C_a(s)$.*

Examples	#Arcs	C-space	RMap	$\gamma(t)$	$C_a(s)$
Top(a)	143	1159	5001	285	114
Top(b)				297	96
Middle	32	222	2111	443	25
Bottom	187	1303	4987	910	52

We compare the result of our G^1-circular motion against the polygonal motion generated by the shortest path $\gamma(t)$ on the roadmap built around the medial surface of the C-space obstacles. Due to the self-intersection of the C-space obstacle boundary surfaces and the complex topology of the medial surface, the polygonal motion $\gamma(t)$ has many sharp corners. Moreover, the strict constraint of keeping an equal distance to nearby obstacles also makes the resulting orientation of a moving robot somewhat awkward. By smoothing the result using G^1-circular arcs, we have improved the motion planning results while relaxing the maximal-clearance constraint and thus leaving slightly away from the medial surface of C-space obstacles.

The peanut-shaped moving robot M has 4 circular arcs on the boundary. Table 1 shows the experimental results for the four path constructions of Figure 5 on the three examples given in Figure 6. For the first example, there are two path constructions shown in Figure 5(a)-(b), the results of which are reported in the two rows of Top(a) and Top(b), respectively. In Table 1, the second column is about the total number of circular arcs used in the representation of obstacle boundaries in each example. The C-space construction means the Minkowski sum computation for 360 different slices along the θ-direction, whereas the roadmap construction includes the medial axes on 360 slices and the local visibility maps around the start and goal configurations. The shortest path $\gamma(t)$ is computed using Dijkstra's algorithm, and the arc-spline path is generated quite efficiently compared with other steps. The overall motion planning can be handled in several seconds for these examples.

5. Conclusions

In this work, we have presented circle-based geometric acceleration techniques for maximum-clearance rigid-body motion planning in a plane among obstacles bounded by G^1-circular arcs. By constructing the roadmap that contains the medial axes for collision-free spaces, we have generated an initial polygonal path that is near optimal in the sense of maximum clearance through narrow passages. Using G^1-circular arcs and circular motions around the circle centers, we have generated G^1-continuous robot motions while maintaining the maximum-clearance within a reasonable bound. In future work, we plan to apply the circle-based optimization techniques to other geometric problems.

Acknowledgments

We would like to thank anonymous reviewers for their invaluable comments. This research was supported in part by the European Union Horizon 2020 research and innovation programme, under grant agreement No 862025, and in part by the National Research Foundation of Korea (No. NRF-2019R1A2C1003490, NRF-2019K1A3A1A78112596).

References

[Aichholzer et al. (2007)] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Jüttler, M. Oberneder, and Z. Sir: Computational and structural advantages of circular boundary representation. In: F. Dehne, J.R. Sack , N. Zeh, editors. *Algorithms and Data Structures*. Springer LNCS, 2007. p. 374–385. 1

[Aichholzer et al. (2009)] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, M. Oberneder, and B. Jüttler: Medial axis computation for planar free-form shapes. *Computer-Aided Design*, 41(5):339–349, 2009. 1, 3

[Aichholzer et al. (2010)] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl: Divide-and-conquer for Voronoi diagrams revisited. *Computational Geometry: Theory and Applications*, 43(8):688–699, 2010. 1, 3

[Akenine-Möller et al. (2018)] T. Akenine-Möller, E. Hains, N. Hoffman, A. Pesce, M. Iwanicki, S. Hillaire: *Real-Time Rendering*. CRC Press, Taylor & Francis, Boca Raton, 4th Ed., 2018. 1

[Avnaim et al. (1988)] F. Avnaim, J.-D. Boissonnat, B. Faverjon: A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. *Proc. of 1988 IEEE Conf. on Robotics and Automation*, pp. 1656–1661, 1988. 2

[Brady et al. (1983)] *Robot Motion*. The MIT Press, Cambridge, MA, 1983. 1

[Canny (1988)] J.F. Canny: *The Complexity of Robot Motion Planning*. The MIT Press, Cambridge, MA, 1988. 1

[Choset et al. (2005)] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun: *Principles of Robot Motion: Theory, Algorithms, and Implementation*. The MIT Press, Cambridge, MA, 2005. 1

[Farouki and Sakkalis (1991)] R.T. Farouki and T. Sakkalis: Real rational curves are not 'unit speed'. *Computer Aided Geometric Design*, **8**(2):151-157, 1991. 4

[Foskey et al. (2001)] M. Foskey, M. Garber, M.C. Lin, D. Manocha: A Voronoi-based hybrid motion planner. *Proc. of 2001 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2001. 1

[Han et al. (2019)] S. Han, S.-H. Yoon, M.-S. Kim, G. Elber: Minkowski sum computation for planar freeform geometric models using G^1-biarc approximation and interior disk culling. *The Visual Computer*, **35**:921-933, 2019. 2, 3

[Kavraki et al. (1996)] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans on Robotics and Automation*. **12**(4):566-580, 1996. 1

[Kim and Moon (1990)] M.-S. Kim and S.-R. Moon: Rotational sweep volumes for objects bounded by algebraic curves. *Proc. of 1990 IEEE Conf. on Robotics and Automation*, pp. 311-316, 1990. 2, 3, 4

[Kim et al. (2010)] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, G. Elber: Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer. *The Visual Computer*, **26**(6-8):1007-1016, 2010. 4

[Kim et al. (2014)] Y.-J. Kim, G. Elber, M.-S. Kim: Precise continuous contact motion for planar freeform geometric curves. *Graphical Models*. **76**(5):580-592, 2014. 1, 3

[Kim et al. (2016)] Y.-J. Kim, G. Elber, M.-S. Kim: Precise contact motion planning for deformable planar curved shapes. *Computer-Aided Design*, **70**(1):126-133, 2016. 1, 3

[Kurnosenko (2013)] A.I. Kurnosenko: Biarcs and bilens, *Computer Aided Geometric Design*, **30**3:310-330, 2013.

[Latombe (2012)] J.-C. Latombe: *Robot Motion Planning*, volume 124. Springer Science & Business Media, 2012. 1

[LaValle (2006)] S.M. LaValle: *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. 1

[Lee et al. (2015)] J. Lee, Y.-J. Kim, M.-S. Kim, G. Elber: Comparison of three bounding regions with cubic convergence to planar freeform curves. *The Visual Computer*, **31**(6–8): 809-818, 2015

[Lee et al. (2016)] J. Lee, Y.-J. Kim, M.-S. Kim, G. Elber: Efficient Voronoi diagram construction for planar freeform spiral curves. *Computer Aided Geometric Design*, **43**:131-142, 2016. 1, 2, 3

[Lozano-Pérez (1981)] T. Lozano-Pérez: Automatic planning of manipulator transfer movements. *IEEE Trans. on System, Man, and Cybernetics*, **11**(10):681-698, 1981. 1, 2, 3

[Lozano-Pérez (1983)] T. Lozano-Pérez: Spatial planning: A configuration space approach. *IEEE Trans. on Computers*, **32**(2):108-120, 1983. 1, 2, 3

[Lozano-Pérez and Wesley (1979)] T. Lozano-Pérez and M.A. Wesley: An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, **22**(10):560-570, 1979. 1, 2

[Lynch and Park (2017)] K.M. Lynch and F.C. Park: *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017. 1

[Meek and Walton (1995)] D. Meek and D. Walton. Approximating smooth planar curves by arc splines. *J. of Computational and Applied Mathematics*, **59**(2):221-231, 1995. 1

[Meek and Walton (1999)] D. Meek and D. Walton. Spiral arc spline approximation to a planar spiral. *J. of Computational and Applied Mathematics*, **107**:21-30, 1999. 1

[Milenkovic et al. (2013a)] V. Milenkovic, E. Sacks, S. Trac: Robust free space computation for curved planar bodies. *IEEE Trans. on Automation Science and Engineering*, **10**(4):875-883, 2013. 2

[Milenkovic et al. (2013b)] V. Milenkovic, E. Sacks, S. Trac: Robust complete path planning in the plane. *Proc. of the Tenth Workshop on the Algorithmic Foundations of Robotics*, pp. 37–52, 2013. 2

[Ó Dúnlaing and Yap (1985)] C. Ó Dúnlaing and C.K. Yap. A retraction method for planning the motion of a disc. *Journal of Algorithms*, **6**:104-111. 1985. 1, 3

[Sacks and Joskowicz (2010)] E. Sacks and L. Joskowicz. *The Configuration Space Method for Kinematic Design of Mechanisms*. The MIT Press, Cambridge, MA, 2010. 1

[Scwhartz et al. (1987)] J.T. Scwhartz, M. Sharir and J. Hopcroft. *Planning, Geometry, and Complexity of Robot Motion*, Ablex Publishing Corporation, 1987. 1

[Sir et al. (2006)] Z. Sir, R. Feichtinger, and B. Jüttler. Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Computer-Aided Design*, **38**(6):608-618, 2006. 1

[Takahashi and Schiling (1989)] O. Takahashi and R.J. Schiling: Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans. Robot. Automat*. 6(2):143-150, 1989. 1, 2, 3

[Thurn et al. (2006)] S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2006. 1