Computing Shortest Paths amid Pseudodisks [*]

Danny Z. Chen[†] Haitao Wang[†‡]

Abstract

Multiple objects in the plane are called *pseudodisks* if they are convex and the boundaries of any two of them intersect transversely at most twice. Given a set of n (possibly intersecting) pseudodisks of $O(1)$ complexity each and two points s and t in the plane, we develop an $O(n^2)$ time algorithm for computing a shortest s-to-t path avoiding the pseudodisks. In over two decades, the previously best algorithms for this problem take $O(n^2 \log n)$ time, even when all pseudodisks are pairwise-disjoint disks. Our technique is also applicable to a motion planning problem of finding a shortest path to translate a convex object in the plane from one location to another avoiding a given set of polygonal obstacles, improving the previously best known solution. Our algorithm actually solves a more general version of the motion planning problem. Further, as a by-product of our approach, we present an $O(n^2)$ time algorithm for computing the visibility graph of a set of n (possibly intersecting) pseudodisks in the plane. The previously best known time bound of this visibility problem is $O(n^2 \log n)$. In addition, for n pairwise disjoint (non-polygonal) convex objects of $O(1)$ complexity each in the plane, we compute a shortest s-to-t path avoiding all objects in $O(n \log n + k)$ time, where k is the size of the visibility graph of the objects.

1 Introduction

We say that two curves in the plane *intersect transversely* if they intersect but are not tangent to each other at any intersection point (see Fig. 1). Multiple objects in the plane are called *pseudodisks* if they are *convex* and the boundaries of any two of them intersect transversely at most twice. Given a set $\mathcal{D}$ of n (possibly intersecting) pseudodisks of $O(1)$ complexity each and two points s and t in the plane, we consider the *shortest path problem* (SPP) of finding a path from s to t that avoids the pseudodisks in $\mathcal{D}$ and has the minimum Euclidean length, if such a path exists. In this paper, we present an $O(n^2)$ time algorithm for SPP. If the given n (non-polygonal) convex objects of $O(1)$ complexity each are pairwise disjoint, then our algorithm takes $O(k+n \log n)$ time, where k is the size of the visibility graph of the objects.

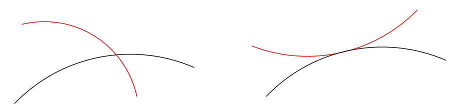

Figure 1: The left two curves intersect transversely while the right two do not.

For a special case in which the n pseudodisks are all disks, the previously best algorithms compute a shortest path in $O(n^2 \log n)$ time [3, 4], which is also the best result when all disks are pairwise disjoint; further, when all disks are of the same size, the algorithm in [16] can be applied to find a shortest path in $O(n^2)$ time.

The problem of computing shortest paths avoiding polygonal obstacles has been studied extensively [7, 8, 10, 11, 12, 14, 16]. Given a set of polygonal obstacles with totally n vertices, after building their visibility graph [5], a shortest path can be obtained in $O(k + n \log n)$ time, where k is the size of the visibility graph. Using the continuous Dijkstra paradigm, Mitchell [11] solved the problem in $O(n^{3/2+\epsilon})$ time (for any constant $\epsilon > 0$). Also based on the continuous Dijkstra approach as well as a conforming planar subdivision, Hershberger and Suri [7] gave an optimal $O(n \log n)$ time solution. However, it is not clear how to generalize these efficient techniques for polygonal obstacles to the curved obstacle case. For example, Mitchell's algorithm [11] uses a data structure for wavelet dragging queries; by modeling as high-dimensional radical-free semialgebraic range queries, some of the key operations can be handled efficiently. In our problem, however, such queries would involve not only radical numbers but also inverse trigonometric operations (e.g., arcsine), and hence similar techniques do not seem to apply. Hershberger and Suri's algorithm [7] relies heavily on a conforming subdivision defined on the vertices of the polygonal obstacles. In our problem, however, it is elusive how to determine a set of $O(n)$ vertices that can help build such a subdivision.

The SPP problem has many applications. For

[*]This research was supported in part by NSF under Grants CCF-0515203 and CCF-0916606.

[†]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: {dchen, hwang6}@nd.edu.

[‡]Corresponding author.

example, a motion planning problem seeks a "good" path to move an object from one location to another without colliding with any obstacles. Hershberger and Guibas [6] studied the problem of finding a shortest path to translate a convex object A between two locations (without rotation) amid polygonal obstacles in the plane, and gave an $O(n^2 + \min\{n, l/w\} n \log n)$ time algorithm, where l is the diameter and w is the width of A. Theoretically, when $l/w = \Omega(n/\log n)$, the running time of their algorithm is up to $O(n^2 \log n)$. An open question was posed in [6] to find an $O(n^2)$ time solution that does not depend on the ratio l/w. In [6], the problem was reduced to finding a shortest path between two points amid a set of convex objects, each of which is defined by a translational copy of A and an obstacle; these convex objects form a set of pseudodisks. Therefore, by using our SPP algorithm, the problem in [6] can be solved in $O(n^2)$ time, thus settling the open question. Further, our SPP algorithm can be applied to a generalized case of this motion planning problem in which each polygonal obstacle B has a specified distance δ_B such that the distance between A and B is at least δ_B during the movement of A.

A by-product of our results is an $O(n^2)$ time algorithm for computing the visibility graph of n (possibly intersecting) pseudodisks of $O(1)$ complexity each in the plane. Constructing visibility graphs is a fundamental research topic and has many applications (e.g., see [2, 5, 13, 15, 17]). For a set of pairwise disjoint polygons with totally n vertices in the plane, $O(n^2)$ time algorithms [2, 17] and an $O(k + n \log n)$ time output-sensitive algorithm [5] for computing the visibility graph were known, where k is the size of the visibility graph. For the curved object case, Pocchiola and Vegter [13] gave an $O(k + n \log n)$ time visibility graph construction algorithm for n pairwise disjoint convex objects in the plane, where k is the size of the visibility graph. If the convex objects are allowed to intersect, then for a special case in which each object is a translational copy of a given convex object, the visibility graph can be computed in $O(n^2)$ time [6]. However, the algorithm in [6], which is based on the feature that all objects are of the same size and the same shape, cannot be generalized to the case of intersecting convex (pseudodisks) objects. Although it is not difficult to build the visibility graph of n (possibly intersecting) pseudodisks in $O(n^2 \log n)$ time, we are not aware of any better previous solution.

Our Approaches Clearly, a shortest s-to-t path avoiding all pseudodisks in $\mathcal{D}$ follows the visibility graph of $\mathcal{D} \cup \{s, t\}$. Our algorithm thus has two main steps: (I) Compute the visibility graph $\mathcal{G}_{\mathcal{V}}$ of $\mathcal{D} \cup \{s, t\}$ (this is the easier step), and (II) find a shortest s-to-t path in $\mathcal{G}_{\mathcal{V}}$.

To construct $\mathcal{G}_{\mathcal{V}}$, we use a global slope or angular sweeping algorithm, in a similar spirit as Welzl's algorithm [17]. Note that Welzl's algorithm does not handle the intersection case (if line segments intersect each other in their interior). Our algorithm, however, is able to deal with the intersection case. Our sweeping algorithm is based on a partial order on all $O(n^2)$ events, each of which is determined by a line that is either a common tangent line of two arcs of $\partial\mathcal{D}$ or tangent to one arc of $\partial\mathcal{D}$ and passing through an intersection point on $\partial\mathcal{D}$. We obtain the partial order of all events by using duality and the curve segment arrangement algorithm [1]. With all these efforts, we build $\mathcal{G}_{\mathcal{V}}$ in $O(n^2)$ time.

Unlike the polygonal obstacle case, the visibility graph $\mathcal{G}_{\mathcal{V}}$ may have $\Omega(n^2)$ vertices. Consequently, running Dijkstra's shortest path algorithm on $\mathcal{G}_{\mathcal{V}}$ would take $O(n^2 \log n)$ time, which is a bottleneck. Our strategy is to transform $\mathcal{G}_{\mathcal{V}}$ to a *coalesced graph* $\mathcal{G}_{\mathcal{V}}^c$ such that: (1) $\mathcal{G}_{\mathcal{V}}^c$ has $O(n)$ vertices and $O(n^2)$ edges; (2) a shortest path in $\mathcal{G}_{\mathcal{V}}^c$ corresponds to a shortest path in $\mathcal{G}_{\mathcal{V}}$. Consequently, by running Dijkstra's algorithm on $\mathcal{G}_{\mathcal{V}}^c$, a shortest path can be computed in $O(n^2)$ time. To build $\mathcal{G}_{\mathcal{V}}^c$, we determine a set of *distinguished points*. Note that the approach in [6] also builds a coalesced graph, which is also based on a set of distinguished points. However, our definitions of distinguished points are different from those in [6]. More importantly, the number of distinguished points computed in [6] is $O(n\frac{l}{w})$ and consequently the coalesced graph in [6] has $O(n\frac{l}{w})$ vertices. We take a completely different approach to compute our distinguished points whose number is bounded by $O(n)$, which is independent of the ratio l/w. Consequently, the number of vertices in our coalesced graph is also bounded by $O(n)$.

Our approach for computing distinguished points is based on the Voronoi diagram $VD(\mathcal{D})$ of the pseudodisks of $\mathcal{D}$. Using $VD(\mathcal{D})$, for each arc e of $\partial\mathcal{D}$, we create distinguished points on e, whose number is proportional to the number of e's neighboring cells in $VD(\mathcal{D})$. Since the Voronoi diagram has $O(n)$ cells, the total number of such distinguished points is $O(n)$. With another set of $O(n)$ distinguished points, we prove that they are sufficient for building the coalesced graph $\mathcal{G}_{\mathcal{V}}^c$, which takes $O(n^2)$ time to build. It turns out that our approach for finding the $O(n)$ distinguished points is very simple but to prove these distinguished points are sufficient for building $\mathcal{G}_{\mathcal{V}}^c$ is quite challenging. To this end, we have to explore the structures of the Voronoi diagram $VD(\mathcal{D})$ and make use of many interesting geometric observations. These observations and techniques may be useful for other problems as well.

In summary, our shortest path algorithm has the following steps: (1) Compute the visibility graph $\mathcal{G}_{\mathcal{V}}$;

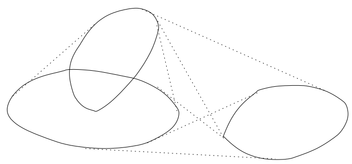

Figure 2: The free common tangents (the dotted line segments) of three pseudodisks.

(2) compute the Voronoi diagram of $\mathcal{D}$; (3) find all distinguished points; (4) compute the coalesced graph $\mathcal{G}^c_\mathcal{V}$; (5) using Dijkstra's algorithm to find a shortest s-to-t path in $\mathcal{G}^c_\mathcal{V}$, from which a shortest s-to-t path in $\mathcal{G}_\mathcal{V}$ can be obtained.

For n pairwise disjoint (non-polygonal) convex objects of $O(1)$ complexity each, we first use the algorithm in [13] to compute the visibility graph in $O(k + n \log n)$ time. Then, by following the same steps as above, a shortest s-to-t path avoiding the objects can be found in $O(k + n \log n)$ time.

The SPP problem (and even some special cases) studied in this paper has been open for a long time and has many applications. Thus, our improvements on the previous work (even by a logarithmic factor) are of significance. Further, our approach based on Voronoi diagrams is very new, and our techniques are likely to be useful in solving other related problems.

The remaining paper is organized as follows. Section 2 describes the visibility graph $\mathcal{G}_\mathcal{V}$, whose construction algorithm is given in Section 6. In Section 3, we discuss the coalesced graph $\mathcal{G}^c_\mathcal{V}$. Section 4 presents the algorithm for determining the distinguished points (for constructing $\mathcal{G}^c_\mathcal{V}$). In Section 5, we briefly discuss the special case when all pseudodisks are pairwise disjoint.

2 The Visibility Graph

In this section, we give the definition of the visibility graph $\mathcal{G}_\mathcal{V}$ of $\mathcal{D} \cup \{s, t\}$, which is needed for finding a shortest s-to-t path in Section 3. Its construction algorithm is given in Section 6.

We treat s and t as two simple pseudodisks. For convenience, we still use $\mathcal{D}$ to denote $\mathcal{D} \cup \{s, t\}$. We assume that the pseudodisks in $\mathcal{D}$ are smooth (i.e, there is a well-defined tangent line through each boundary point), strictly convex (i.e., the open line segment connecting two points of a pseudodisk lies in its interior), and in general position (i.e., no three pseudodisks share a common tangent line). These assumptions are for ease of exposition only and the general case can be treated by standard techniques. We define a *common tangent* of two pseudodisks as the closed line segment connecting the two tangent points on a line that is tangent to both pseudodisks. A common tangent is *free* if it does not in-

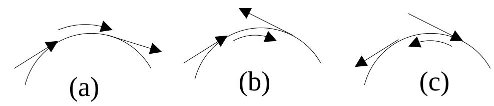

Figure 3: A subpath as in (a) is possible on a shortest path while those in (b) and (c) are not.

tersect the interior of any pseudodisk in $\mathcal{D}$ (see Fig. 2). Since the complexity of each pseudodisk in $\mathcal{D}$ is $O(1)$, a common tangent of any two different pseudodisks in $\mathcal{D}$ can be computed in $O(1)$ time. We define the visibility graph $\mathcal{G}_\mathcal{V}$ as follows. The set of tangent points of all free common tangents of $\mathcal{D}$ forms the vertex set of $\mathcal{G}_\mathcal{V}$. Each free common tangent defines an edge in $\mathcal{G}_\mathcal{V}$ whose weight is the Euclidean length of that common tangent. For each pseudodisk $D \in \mathcal{D}$, if there are multiple vertices of $\mathcal{G}_\mathcal{V}$ on D, then the minimal arcs on D that connect these vertices and do not intersect the interior of any pseudodisk also define edges in $\mathcal{G}_\mathcal{V}$ (no two different such edges overlap except possibly at their endpoints) and the length of each such arc is the weight of the corresponding edge.

Clearly, $\mathcal{G}_\mathcal{V}$ has $O(n^2)$ vertices and $O(n^2)$ edges. Our $O(n^2)$ time algorithm for constructing $\mathcal{G}_\mathcal{V}$ is given in Section 6. We only present the following result here.

THEOREM 2.1. *The visibility graph $\mathcal{G}_\mathcal{V}$ of n pseudodisks can be constructed in $O(n^2)$ time.*

3 The Coalesced Graph

Note that the visibility graph $\mathcal{G}_\mathcal{V}$ is for $\mathcal{D} \cup \{s, t\}$. A shortest s-to-t path avoiding all pseudodisks must be a shortest s-to-t path in $\mathcal{G}_\mathcal{V}$. Without loss of generality, we assume that neither s nor t is in the interior of the union region of $\mathcal{D}$ bounded by $\partial\mathcal{D}$. Since $\mathcal{G}_\mathcal{V}$ has $O(n^2)$ vertices and edges, running Dijkstra's shortest path algorithm on it takes $O(n^2 \log n)$ time. In this section, we describe a *coalesced graph* $\mathcal{G}^c_\mathcal{V}$, which has $O(n^2)$ edges but only $O(n)$ vertices. We show that after $\mathcal{G}^c_\mathcal{V}$ is constructed from $\mathcal{G}_\mathcal{V}$, a shortest s-to-t path in $\mathcal{G}_\mathcal{V}$ corresponds to a shortest s-to-t path in $\mathcal{G}^c_\mathcal{V}$. Since $\mathcal{G}^c_\mathcal{V}$ has $O(n)$ vertices and $O(n^2)$ edges, a shortest s-to-t path in $\mathcal{G}^c_\mathcal{V}$ can be computed in $O(n^2)$ time.

Hershberger and Guibas [6] also built a coalesced graph for a set of objects each of which is a translational copy of a given convex object A. Their graph has $O(n\frac{l}{w})$ vertices, where l (resp., w) is the diameter (resp., width) of A. However, the approach for constructing the kind of coalesced graph in [6] is not applicable to our problem since our pseudodisks can have different sizes and shapes. We present a new approach for building a different coalesced graph $\mathcal{G}^c_\mathcal{V}$ whose number of vertices is strictly bounded by $O(n)$.

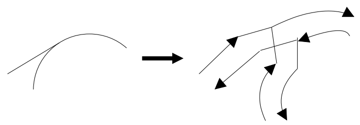

Figure 4: Transforming a joint between an arc edge and a tangent edge from $\mathcal{G}_{\mathcal{V}}$ to $\mathcal{G}^d_{\mathcal{V}}$.

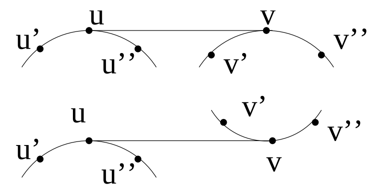

Figure 5: In both figures, $\overline{uv}$ is a tangent edge; u' and u'', v' and v'' are consecutive distinguished points. Property (iii) requires $|uv| \geq |uu''|$ and $|uv| \geq |vv'|$.

We refer to the edges of $\mathcal{G}_{\mathcal{V}}$ defined by free common tangents (resp., arcs on $\partial\mathcal{D}$) as *tangent edges* (resp., *arc edges*). Similar to an observation in [6], for any subpath $a \to b \to c$ of a shortest s-to-t path in $\mathcal{G}_{\mathcal{V}}$, if a and c are tangent edges and b consists of one or more arc edges on the same pseudodisk, then this subpath forms a convex chain. For example, the pattern in Fig. 3(a) is possible on a shortest path but those in Fig. 3(b)-(c) are not. We call any subpath in $\mathcal{G}_{\mathcal{V}}$ of this pattern a *forward-going* path. Note that any edge of $\mathcal{G}_{\mathcal{V}}$ can be traversed in either direction. As in [6], we first transform $\mathcal{G}_{\mathcal{V}}$ to a directed graph $\mathcal{G}^d_{\mathcal{V}}$ by replacing each undirected edge with two directed edges with opposite directions. The new joints in $\mathcal{G}^d_{\mathcal{V}}$ between an arc edge and a tangent edge are shown in Fig. 4. In this way, any path in $\mathcal{G}^d_{\mathcal{V}}$ must be a forward-going path.

We use $\overline{uv}$ to denote the line segment connecting two points u and v and $|uv|$ to denote the length of $\overline{uv}$; if u and v are on the same arc, we use $\widehat{uv}$ to denote the sub-arc between them and $|uv|$ to denote the length of $\widehat{uv}$. In $\mathcal{G}^c_{\mathcal{V}}$ and $\mathcal{G}^d_{\mathcal{V}}$, we use $e(u,v)$ to denote a (directed) edge connecting vertices u and v (directed from u to v in the corresponding graph).

To define our coalesced graph $\mathcal{G}^c_{\mathcal{V}}$, as in [6], we first introduce the *distinguished points* on $\partial\mathcal{D}$, which are different from those in [6] and have the following properties: (i) Each vertex (i.e., an intersection point) of $\partial\mathcal{D}$ is a distinguished point; (ii) the tangent lines to any two consecutive distinguished points on each arc of $\partial\mathcal{D}$ differ in direction by at most $\pi/2$; (iii) for any arc $a \in \partial\mathcal{D}$, if there is a tangent edge $e(u,v)$ with the endpoint u on a and u is not a distinguished point, assuming that u is between two consecutive distinguished points u' and u'' on a such that the sub-arc $\widehat{u''u}$ and the tangent edge $e(u,v)$ do not form a convex chain, then $|uv| \geq |uu''|$ (see Fig. 5).

The main difference between our distinguished points and those in [6] is that property (iii) above is different from (and weaker than) property (3) in [6], which requires $|uv| \geq |u'u''|$. Interestingly, it turns out that our "weaker" version of distinguished points is sufficient for building a coalesced graph for finding a shortest s-to-t path; more importantly, it leads to an efficient way to compute distinguished points such that their total number is bounded by $O(n)$ (that number in [6] is $O(n\frac{l}{w})$). Consequently, the number of vertices in our coalesced graph $\mathcal{G}^c_{\mathcal{V}}$ is also bounded by $O(n)$. Note that both s and t are distinguished points.

Let Δ be the number of distinguished points needed. Obviously, $n = O(\Delta)$ and $\Delta = O(n^2)$. As in [6], we prove the following two lemmas. With the two lemmas below, our remaining major task is to compute $\Delta = O(n)$ distinguished points, which will be discussed in Section 4.

Lemma 3.1. *The coalesced graph $\mathcal{G}^c_{\mathcal{V}}$ has $O(\Delta)$ vertices and $O(n^2)$ edges. Once all distinguished points are given, $\mathcal{G}^c_{\mathcal{V}}$ can be constructed in $O(\Delta \log \Delta + n^2)$ time.*

Proof. Suppose a set of Δ distinguished points for $\mathcal{G}^c_{\mathcal{V}}$ is given. We prove this lemma by defining our coalesced graph $\mathcal{G}^c_{\mathcal{V}}$ precisely, in a similar fashion as in [6].

We first define the vertices of $\mathcal{G}^c_{\mathcal{V}}$. The distinguished points divide the arcs of $\partial\mathcal{D}$ into arc-intervals. We also view the distinguished points themselves as closed intervals distinct from the open intervals they delimit. Each interval has two directions in which a path may traverse. We call an interval (either closed or open) associated with a direction a *directed interval.* Thus, each interval corresponds to two directed intervals. The set of all such directed intervals forms the vertex set of $\mathcal{G}^c_{\mathcal{V}}$. It is easy to see that the number of vertices in $\mathcal{G}^c_{\mathcal{V}}$ is $O(\Delta)$. Below, we use (u', u'') to denote the directed u'-to-u'' interval (open or closed) bounded by two points u' and u'' on an arc of $\partial\mathcal{D}$; if it is closed, then $u' = u''$.

Based on the graph $\mathcal{G}^d_{\mathcal{V}}$, we define the edges of $\mathcal{G}^c_{\mathcal{V}}$, as follows.

For each (directed) tangent edge $e(u,v)$ of $\mathcal{G}^d_{\mathcal{V}}$, with two tangent points u and v on arcs e_u and e_v respectively, there must be a unique directed interval (u', u'') (resp., (v', v'')) containing u (resp., v) on e_u (resp., e_v) such that the path $u' \to u \to v \to v''$ is forward-going (see Fig. 6). We refer to the directed interval (u', u'') (resp., (v', v'')) as the *host interval* of u (resp., v). We then create a *tangent edge* in $\mathcal{G}^c_{\mathcal{V}}$ to link the vertex defined by the interval (u', u'') to the vertex defined by (v', v''), and set its weight to $|uv| - |uu''| + |vv''|$. By property (iii) of our distinguished

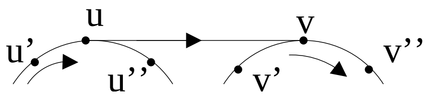

Figure 6: The weight of the edge in $\mathcal{G}^c_{\mathcal{V}}$ corresponding to the directed edge from u to v is $|uv| - |uu''| + |vv''|$.

points, we have $|uv| \geq |uu''|$, and thus the edge weight is non-negative. It should be noted that for any two directed intervals (u', u'') and (v', v''), there is at most one tangent edge $e(u, v)$ in $\mathcal{G}^d_{\mathcal{V}}$ such that (u', u'') contains u and (v', v'') contains v, and the path $u' \to u \to v \to v''$ is forward-going. Each tangent edge of $\mathcal{G}^d_{\mathcal{V}}$ corresponds to a tangent edge of $\mathcal{G}^c_{\mathcal{V}}$ connecting two vertices defined by two intervals on different arcs.

Next, we consider the *arc edges* of $\mathcal{G}^c_{\mathcal{V}}$. For an arc e of $\partial\mathcal{D}$, suppose (u, u), (u, u'), and (u', u') are three consecutive directed intervals on e with the same direction, where (u, u') is an open interval and the other two are closed intervals. Then, we put an arc edge in $\mathcal{G}^c_{\mathcal{V}}$ from the vertex defined by (u, u) to the vertex defined by (u, u') with a weight $|uu'|$, and put an arc edge in $\mathcal{G}^c_{\mathcal{V}}$ from the vertex defined by (u, u') to the vertex defined by (u', u') with a weight 0. Note that the weight of each edge joining adjacent intervals on the same arc is equal to the length of the sub-arc between the forward points of the two intervals. This completes the definition of our coalesced graph $\mathcal{G}^c_{\mathcal{V}}$.

Clearly, the number of edges in $\mathcal{G}^c_{\mathcal{V}}$ is $O(n^2)$.

Once the Δ distinguished points are given, to construct $\mathcal{G}^c_{\mathcal{V}}$, we first sort the distinguished points on each arc of $\partial\mathcal{D}$, in overall $O(\Delta \log \Delta)$ time. Then, the $O(\Delta)$ vertices of $\mathcal{G}^c_{\mathcal{V}}$ can be determined in $O(\Delta)$ time. For each tangent edge $e(u, v)$ of $\mathcal{G}^d_{\mathcal{V}}$, we associate u and v with their corresponding host intervals. This step takes $O(n^2)$ time. Subsequently, the tangent edges of $\mathcal{G}^c_{\mathcal{V}}$ can be created in $O(n^2)$ time. Finally, constructing all arc edges of $\mathcal{G}^c_{\mathcal{V}}$ can be done in $O(\Delta)$ time. In summary, we can construct $\mathcal{G}^c_{\mathcal{V}}$ in $O(\Delta \log \Delta + n^2)$ time.

The proof of the following lemma is in Appendix.

Lemma 3.2. *A shortest s-to-t path in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a shortest s-to-t path in $\mathcal{G}^d_{\mathcal{V}}$ with the same length.*

4 Determining the Distinguished Points

In this section, we compute the distinguished points for constructing $\mathcal{G}^c_{\mathcal{V}}$. Hershberger and Guibas [6] created their distinguished points as follows. First, distinguished points are put on each arc of $\partial\mathcal{D}$ so that no interval on any arc is longer than l and the tangent directions at any two consecutive distinguished points on the arc differ by at most $\pi/2$. There are only $O(n)$ such distinguished points. To satisfy property (3) of the distinguished points in [6], the endpoints of all tangent edges whose lengths are shorter than l are also added as distinguished points, and it was shown in [6] that the number of distinguished points of this type is bounded by $O(n\frac{l}{w})$.

We take a completely different approach to determining our version of distinguished points, and show that their total number is bounded by $O(n)$. First, we construct the Voronoi diagram $VD(\mathcal{D})$ of all pseudodisks in $\mathcal{D}$, which can be done easily in $O(n^2)$ time. We refer to the vertices and edges of $VD(\mathcal{D})$ as *Voronoi vertices* and *Voronoi edges*, respectively. Clearly, $VD(\mathcal{D})$ has $O(n)$ Voronoi edges and $O(n)$ Voronoi vertices. In $VD(\mathcal{D})$, every arc e of $\partial\mathcal{D}$ induces a cell $c(e)$ that contains e. Two cells c and c' in $VD(\mathcal{D})$ are *neighboring* if they share a Voronoi edge (sharing only a Voronoi vertex is not sufficient). We say that an arc e_1 of $\partial\mathcal{D}$ is a *neighbor* of another arc e_2 if $c(e_1)$ and $c(e_2)$ are neighboring in $VD(\mathcal{D})$. For any arc e of $\partial\mathcal{D}$, let $D(e)$ denote the pseudodisk of $\mathcal{D}$ on which e lies. For every arc e of $\partial\mathcal{D}$, with respect to each neighboring arc e' of e in $VD(\mathcal{D})$, we add the following four types of distinguished points on e.

Type (1): Note that there are at most four common tangents (not necessarily free) between $D(e)$ and $D(e')$; for each tangent point p on $D(e)$ of these common tangents, if p lies on e, then take p as a distinguished point on e.

Type (2): Let p' be the closest point on $D(e)$ to $D(e')$; if $p' \in e$, then take p' as a distinguished point on e.

Type (3): For each Voronoi vertex v of $c(e)$, let v' be the closest point on $D(e)$ to v; if $v' \in e$, then take v' as a distinguished point on e.

Type (4): For each Voronoi vertex v of $c(e)$, there are at most two tangent lines of $D(e)$ passing through v; for each such tangent line, let v' be its tangent point on $D(e)$; if $v' \in e$, then take v' as a distinguished point on e.

Obviously, the number of distinguished points thus created on e is proportional to the number of neighboring cells of $c(e)$ in $VD(\mathcal{D})$. We perform this process for each arc of $\partial\mathcal{D}$. Since the size of $VD(\mathcal{D})$ is $O(n)$, the number of all distinguished points thus created is $O(n)$. Finally, we add to each arc of $\partial\mathcal{D}$ its endpoints as distinguished points (for property (i)), and add more distinguished points (if needed) so that the tangent directions at any two consecutive distinguished points along the arc differ by at most $\pi/2$ (for property (ii)). There are $O(n)$ such distinguished points.

In summary, the number of all distinguished points that we create above is bounded by $O(n)$. After $VD(\mathcal{D})$ is built, all distinguished points can be easily generated in $O(n)$ time. The coalesced graph $\mathcal{G}^c_{\mathcal{V}}$ can then be constructed in $O(n^2)$ time by Lemma 3.1. By running Dijkstra's algorithm on $\mathcal{G}^c_{\mathcal{V}}$, a shortest s-to-t path can be obtained in $O(n^2)$ time.

It remains to prove that property (iii) of our version of distinguished points is satisfied, as stated by the next lemma.

LEMMA 4.1. *Property (iii) of our version of distinguished points is satisfied, that is, for any arc e of $\partial\mathcal{D}$, if there is a tangent edge $\overline{uv}$ of $\mathcal{G}_{\mathcal{V}}$ with the endpoint u on e and u is not a distinguished point, then it must be $|uv| \geq |uu''|$, where u'' is the closest distinguished point on e along the direction (called the* reversing direction*) such that the tangent edge $e(v,u)$ and the sub-arc $\widehat{uu''}$ along e do not form a convex chain (e.g., see Fig. 5).*

Before giving the main proof for Lemma 4.1 in Section 4.2, we first show in Section 4.1 some geometric observations about the Voronoi diagram $VD(\mathcal{D})$, which are useful to the main proof. We will continue to use the notation of Lemma 4.1 in the ensuing discussion and proof. Without loss of generality, we assume that no arc of $\partial\mathcal{D}$ is tangent to any interior point of the tangent segment $\overline{uv}$. If this is not the case, i.e., an arc e' of $\partial\mathcal{D}$ is tangent to an interior point v' of $\overline{uv}$, then we simply replace $\overline{uv}$ by $\overline{uv'}$ and rename v' as v (note that the arc e' cannot be a neighbor of e; otherwise, u would be a Type (1) distinguished point of e).

4.1 Notations and Geometric Observations

Without loss of generality, we assume that in $\mathcal{D}$, no pseudodisk is contained in another one. In the following, when we say a point p is *on* a pseudodisk D, we mean that p is on the boundary of D; when we say a pseudodisk *intersects* another pseudodisk or a line, we mean that they intersect but are not tangent to each other.

For two different pseudodisks D_1 and D_2, denote by $B(D_1, D_2)$ their *bisector curve*, which consists of all points each of which has equal distance to D_1 and to D_2. Note that $B(D_1, D_2)$ does not contain any point in the interior of D_1 or D_2. So if $D_1 \cap D_2 \neq \emptyset$, then $B(D_1, D_2)$ can be viewed as having two disconnected portions separated by $D_1 \cap D_2$. Thus, if two cells $c(e_1)$ and $c(e_2)$ share a Voronoi edge h of $VD(\mathcal{D})$, then the edge h is part of $B(D(e_1), D(e_2))$. For a pseudodisk D and a point $p \notin D$, let $\psi_p(D)$ denote the closest point of D to p. For two different points p_1 and p_2 in the plane, we use $l(p_1, p_2)$ to denote the line containing p_1 and p_2. For a pseudodisk D and a point p on D, let $l_p(D)$ denote the tangent line of D touching p, and $\rho_p(D)$ denote the ray perpendicular to $l_p(D)$ and shooting away from D starting at p. For two objects O_1 and O_2 in the plane, use $d(O_1, O_2)$ to denote the shortest Euclidean distance between O_1 and O_2. Note that for any pseudodisk D and any point $p \notin D$, the line $l(p, p')$ is perpendicular to the line $l_{p'}(D)$, where $p' = \psi_p(D)$.

Let e_v be the arc of $\partial\mathcal{D}$ on which the tangent point v of $\overline{uv}$ lies. Since e_v and e are in different cells of $VD(\mathcal{D})$, $\overline{uv}$ must intersect a Voronoi edge h of the cell $c(e)$ (if $\overline{uv}$ intersects a Voronoi vertex of $c(e)$, then we let h be any edge of $c(e)$ incident to that vertex). Let e' be the neighbor of e sharing the Voronoi edge h with e. Note that e' is not e_v since otherwise u would be a (Type (1)) distinguished point. When $D(e) \cap D(e') = \emptyset$, let b be a point of $D(e)$ closest to $D(e')$ and $b' = \psi_b(D(e'))$. When $D(e) \cap D(e') \neq \emptyset$, let a_1 and a_2 be the two intersection points of their boundaries ($a_1 = a_2$ if $D(e)$ and $D(e')$ are tangent to each other). For simplicity of discussion, we view $D(e) \cap D(e') \neq \emptyset$ as one "thick" point and let both b and b' denote this thick point. We define a (thick) line $tl(b, b')$ passing through both b and b' as follows: If $D(e) \cap D(e') = \emptyset$, then $tl(b, b')$ is the (normal) line containing b and b'; if $D(e) \cap D(e') \neq \emptyset$ and $a_1 \neq a_2$, then $tl(b, b')$ is the "thick" line consisting of all lines each of which intersects a point on $\overline{a_1a_2}$ and is perpendicular to $\overline{a_1a_2}$; if $D(e) \cap D(e') \neq \emptyset$ and $a_1 = a_2$ (i.e., $D(e)$ is tangent to $D(e')$), then $b = b' = a_1 = a_2$ and $tl(b, b')$ is the (normal) line containing the point b and perpendicular to the line $l_b(D(e))$. Let b'' denote the intersection of $tl(b, b')$ and $B(D(e), D(e'))$ (if $D(e) \cap D(e') \neq \emptyset$, then $b'' = b = b'$ is the thick point $D(e) \cap D(e')$). For simplicity, let B^* denote $B(D(e), D(e'))$.

The above notations and concepts will be consistently used in the forthcoming discussion.

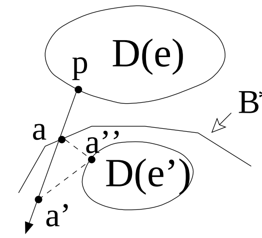

Figure 7: Illustrating Lemma 4.2.

LEMMA 4.2. *For any point p on $D(e) \setminus D(e')$ (resp., $D(e') \setminus D(e)$), the ray $\rho_p(D(e))$ (resp., $\rho_p(D(e'))$) can intersect the bisector B^* at most once and can never be tangent to B^* (e.g., see Fig. 7).*

Proof. We only prove the case when p is on $D(e) \setminus D(e')$.

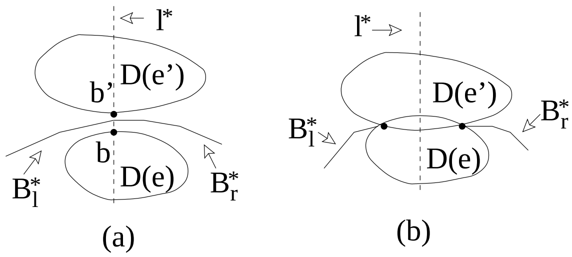

Figure 8: Illustrating the definitions of B_l^* and B_r^*. (a) $D(e) \cap D(e') = \emptyset$; (b) $D(e) \cap D(e') \neq \emptyset$.

The other case (i.e., when p is on $D(e') \setminus D(e)$) can be proved similarly. For simplicity, let ρ denote $\rho_p(D(e))$.

Note that ρ is perpendicular to $l_p(D(e))$. Thus, for any point p' on ρ, $d(p', D(e)) = |pp'|$. Let a be the first intersection point of B^* and ρ when moving from p along ρ (see Fig. 7). Let ρ' be the (open) ray from a along ρ but not including a. Below we show that for any point a' on ρ', $d(a', D(e')) < d(a', D(e))$.

Let $a'' = \psi_a(D(e'))$. For any point a' on ρ', we have $d(a', D(e)) = |a'p| = |a'a| + |ap| = |a'a| + |aa''|$. Since $p \neq a''$, by the triangle inequality, we have $|a'a''| < |a'a| + |aa''|$. Note that $d(a', D(e')) \leq |a'a''|$. Therefore, $d(a', D(e')) < d(a', D(e))$.

Since $d(a', D(e')) < d(a', D(e))$ for any point a' on ρ', ρ' cannot intersect B^* and ρ cannot be tangent to B^* at a. The lemma thus follows.

When $D(e) \cap D(e') \neq \emptyset$, B^* can be viewed as having two disconnected portions separated by $D(e) \cap D(e')$. If $D(e) \cap D(e') = \emptyset$, we also view B^* as having two portions separated by the point b''. In either case, these two portions of B^* are on different sides of the (thick) line $tl(b, b')$. Let B_r^* and B_l^* denote these two portions of B^* respectively, defined as follows. The line $tl(b, b')$ divides the plane into two half-planes $\mathcal{P}_r$ and $\mathcal{P}_l$. We let $\mathcal{P}_r$ be the half-plane that we enter first when moving from b clockwise on $D(e)$, and $\mathcal{P}_l$ be the other one. Let B_r^* (resp., B_l^*) be the portion of B^* contained in $\mathcal{P}_r$ (resp., $\mathcal{P}_l$), as in shown Fig. 8.

The next two simple observations follow from the convexity of the pseudodisks.

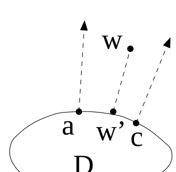

Figure 9: Illustrating Observation 1: $w' = \psi_w(D)$.

OBSERVATION 1. *For any two points a and c on a pseudodisk D, suppose a point p is moved from a to c along D clockwise and the ray $\rho_p(D)$ moves accordingly. Denote by e'' the arc of D swept by p and denote by R the region of the plane swept by $\rho_p(D)$ due to the movement of p. Then for any point w in R, the point $\psi_w(D)$ is on the arc e'' (e.g., see Fig. 9).*

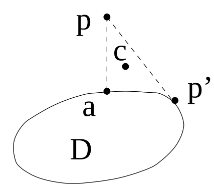

Figure 10: Illustrating Observation 2: $a = \psi_p(D(e))$ and $\overline{pp'}$ is a tangent to D at p'. The point $\psi_c(D(e))$ must be on the arc $\widehat{ap'}$.

OBSERVATION 2. *For any pseudodisk D and a point $p \notin D$, suppose $a = \psi_p(D)$ and $\overline{pp'}$ is a tangent to D at p'. Let $\widehat{ap'}$ be the arc of D contained in the triangle $\triangle$ formed by p, a, and p'. Then for any point $c \in (\triangle - D)$, the point $\psi_c(D)$ is on the arc $\widehat{ap'}$ (e.g., see Fig. 10).*

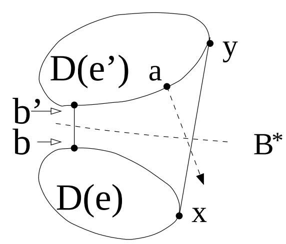

Figure 11: Illustrating Lemma 4.3.

$D(e)$ and $D(e')$ have two outer common tangents (not necessarily free), one intersecting B_r^* and the other intersecting B_l^*. Suppose $\overline{xy}$ is the outer common tangent intersecting B_r^* with x on $D(e)$ and y on $D(e')$ (see Fig. 11). Let $A_r(e')$ be the arc of $D(e')$ from b' counterclockwise to y but excluding the point y, and $A_r(e)$ be the arc of $D(e)$ from b clockwise to x but excluding x. Based on the outer comment tangent intersecting B_l^*, we also define $A_l(e')$ and $A_l(e)$ in a similar way. Let $A(e) = A_l(e) \cup A_r(e)$ and $A(e') = A_l(e') \cup A_r(e')$. We have the following lemma.

LEMMA 4.3. *For any point a on $D(e')$, the ray $\rho_a(D(e'))$ intersects B^* if and only if a is on the arc $A(e')$; further, if a is on $A_r(e')$ (resp., $A_l(e')$), then $\rho_a(D(e'))$ intersects B_r^* (resp., B_l^*) (e.g., see Fig. 11). Similar results also hold for the case when a is on $D(e)$.*

Proof. We only prove the case when a is on $D(e')$; the other case when a is on $D(e)$ can be analyzed in a similar way.

Let a be on $A_r(e')$; then $d(a, D(e')) = 0 < d(a, D(e))$. To prove $\rho_a(D(e'))$ intersecting B^*, it suffices to show that there exists a point p on $\rho_a(D(e'))$ such that $d(p, D(e')) \geq d(p, D(e))$.

Note that $D(e')$ and $\rho_a(D(e')) - \{a\}$ are on different sides of the line $l_a(D(e'))$. Let R be the open half-plane bounded by (but not including) $l_a(D(e'))$ in which $\rho_a(D(e')) - \{a\}$ lies. By the definition of $A_r(e')$ and since a is on $A_r(e')$, there must exist a point c on $D(e)$ that is in R. Let l be the perpendicular bisector of the line segment $\overline{ac}$. Since c is in R (not on the line $l_a(D(e'))$) and $\rho_a(D(e'))$ is perpendicular to $l_a(D(e'))$, the line l must intersect $\rho_a(D(e'))$ inside R. Let p be the intersection point of l and $\rho_a(D(e'))$. Then, we have $d(p, c) = d(p, a)$. Since $d(p, D(e')) = d(p, a)$ and $d(p, D(e)) \leq d(p, c)$, we obtain $d(p, D(e')) \geq d(p, D(e))$.

Additionally, by the convexity of $D(e')$ and the definition of $A_r(e')$, $\rho_a(D(e'))$ must intersect B_r^*. Based on similar analysis, if a is on $A_l(e')$, $\rho_a(D(e'))$ must intersect B_l^*.

On the other hand, if a is on $D(e')$ but not on $A(e')$, then $D(e)$ and $D(e')$ are on the same side of the line $l_a(D(e'))$. For any point p on the ray $\rho_a(D(e'))$, we have $d(p, D(e')) = d(p, a)$. Note that $d(p, l_a(D(e'))) < d(p, D(e))$ and $d(p, l_a(D(e'))) = d(p, a)$. Thus, we have $d(p, D(e')) < d(p, D(e))$. Therefore, the ray $\rho_a(D(e'))$ cannot intersect B^*.

Recall that $\overline{uv}$ intersects the Voronoi edge h. Let i be the intersection point of $\overline{uv}$ and h such that $\overline{ui}$ is contained in the cell $c(e)$. In the following discussion, for any point p on $D(e)$, $\widehat{up}$ always refers to the arc of $D(e)$ from u to p along the *reversing direction* (i.e., the segment $\overline{uv}$ and the arc $\widehat{up}$ do not form a convex chain).

LEMMA 4.4. *Let a be the endpoint of the arc $A_r(e)$ other than b. Suppose the reversing direction is counterclockwise on $D(e)$ and i is on B_r^*. Let $\rho(u, v)$ be the ray starting at u and containing $\overline{uv}$. If $\rho(u, v)$ does not intersect $D(e') \cup \overline{bb'}$, then a must be on the arc $\widehat{uj}$, where $j = \psi_i(D(e))$ (e.g., see Fig. 12).*

Proof. If $a = u$, then a is on $\widehat{uj}$. Below, we assume a is not u.

We first claim that u cannot be on $A_r(e)$. Assume to the contrary that u is on $A_r(e)$. Then u is in the region enclosed by $A_r(e)$, $A_r(e')$, $\overline{bb'}$, and the outer common tangent segment of $D(e)$ and $D(e')$ intersecting B_r^* (i.e., the one touching a in Fig. 12). Since the reversing direction is counterclockwise and by the definition of $A_r(e)$, $\rho(u, v)$ must intersect $A_r(e') \cup \overline{bb'}$,

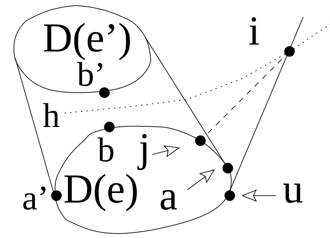

Figure 12: The point i is the intersection of $\overline{uv}$ and h, and $j = \psi_i(D(e))$; a must be on the arc $\widehat{uj}$.

a contradiction. Hence, u cannot be on $A_r(e)$. Since i is on B_r^*, by Lemma 4.3, j is on $A_r(e)$.

For any two points p and p' on $D(e)$, let $\widehat{pp'}$ refer to the arc of $D(e)$ when moving from p to p' along the reversing direction.

Assume to the contrary that a is not on $\widehat{uj}$. Then either j is on $\widehat{ua}$ or u is on $\widehat{aj}$.

Since u is not on $A_r(e)$ and j is on $A_r(e)$, j cannot be on $\widehat{ua}$.

If u is on $\widehat{aj}$, then since j is on $A_r(e)$, u must be on $A_r(e)$, a contradiction.

Thus, a must be on $\widehat{uj}$, and the lemma follows.

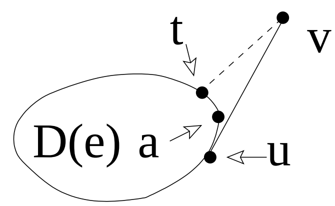

Figure 13: Illustrating Lemma 4.5: $t = \psi_v(D(e))$ and a is a distinguished point of e on $\widehat{ut}$.

In the following discussion, we let $t = \psi_v(D(e))$.

LEMMA 4.5. *If the arc e has a distinguished point, say a, on the arc $\widehat{ut}$, then $|uu''| \leq |uv|$ (see Fig. 13).*

Proof. We first prove $|uv| \geq |ut|$. Let p be the intersection point of the tangent line $l_t(D(e))$ and $\overline{uv}$. Then $|tp| + |pu| \geq |ut|$. Since $l_t(D(e))$ is perpendicular to $l(v, t)$, we have $|pt| \leq |pv|$. Therefore, $|uv| = |up| + |pv| \geq |up| + |pt| \geq |ut|$.

Since a is on $\widehat{ut}$, $|uv| \geq |ut| \geq |ua|$. Because a is a distinguished point on e and u'' is the closest distinguished point on e to u along the reversing direction, it must be $|uu''| \leq |ua|$. Thus we obtain $|uu''| \leq |uv|$.

Lemma 4.5 above is very critical. The main proof in Section 4.2 for Lemma 4.1 consists of several cases. In each case, our goal is always to find a distinguished point of e on the arc $\widehat{ut}$, and then by Lemma 4.5, $|uu''| \leq |uv|$ holds.

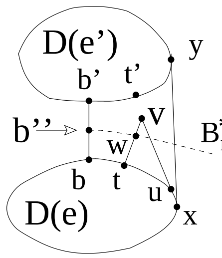

Figure 14: Illustrating Lemma 4.6: $t = \psi_v(D(e))$, $t' = \psi_v(D(e'))$, and w is the intersection of $\overline{vt}$ and B_r^*.

Let $\overline{xy}$ be the outer common tangent intersecting B_r^* with x on $D(e)$ and y on $D(e')$ (see Fig. 14). Suppose i is on B_r^* and v is in the area bounded by $A_r(e)$, $\overline{xy}$, $A_r(e')$, and $\overline{bb'}$, as in Fig. 14. Let $t = \psi_v(D(e))$ and $t' = \psi_v(D(e'))$. Let w be the intersection point of $\overline{vt}$ and B_r^* and z be the intersection point of the line $l(v, t')$ and B_r^*. Let $B^*(b'', w)$ be the continuous portion of B_r^* between b'' and w excluding the point w. The next lemma is due to the convexity of $D(e)$ and $D(e')$.

Lemma 4.6. *The point z cannot be on $B^*(b'', w)$.*

Proof. Without loss of generality, assume $D(e) \cap D(e') = \emptyset$ (the case of $D(e) \cap D(e') \neq \emptyset$ can be proved in a similar fashion). Consider Fig. 14 as an example. By the definition of b and b', $D(e')$ is above the line $l_{b'}(D(e'))$ and $D(e)$ is below the line $l_b(D(e))$. Further, $l_{b'}(D(e'))$ and $l_b(D(e))$ are parallel to each other. Let a (resp., a') be the intersection of $l_b(D(e))$ (resp., $l_{b'}(D(e))$) and the line $l(v, t)$. Due to the convexity of $D(e)$, the angle $\angle bav$ is at least $\pi/2$. Since $l_{b'}(D(e'))$ and $l_b(D(e))$ are parallel to each other and both are perpendicular to $tl(b, b')$, the angle $\angle b'a't$ is at most $\pi/2$. Due to the convexity of $D(e')$, t' cannot be to the right of the line $l(v, t)$, implying that z cannot be on $B^*(b'', w)$.

4.2 The Proof of Lemma 4.1

For simplicity of discussion, we assume that the line $l(u, v)$ is vertical and u is below v. In the following, we analyze only the cases when $D(e)$ is on the left side of $l(u, v)$ (thus the reversing direction is counterclockwise on $D(e)$); the other cases (when $D(e)$ is on the right side of $l(u, v)$) can be analyzed similarly. The proof consists of several cases. In each case, our goal is to find a distinguished point of e on the arc $\widehat{ut}$ with $t = \psi_v(D(e))$; then by Lemma 4.5, $|uu''| \leq |uv|$ follows.

First of all, depending on whether i is on B_l^* or B_r^* (recall that i is the intersection point of $\overline{uv}$ and h such that $\overline{ui}$ is contained in $c(e)$), there are two main cases (L and R).

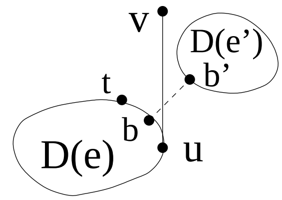

Figure 15: When $\overline{uv}$ intersects $\overline{bb'}$, b must be on $\widehat{ut}$, where $t = \psi_v(D(e))$.

4.2.1 Case L

If i is on B_l^*, then depending on whether $\overline{uv}$ intersects $\overline{bb'}$, there are two subcases (L1 and L2). Note that as a free common tangent, $\overline{uv}$ does not intersect any pseudodisk in $\mathcal{D}$.

Case L1 If $\overline{uv}$ intersects $\overline{bb'}$, then b must be on the arc $\widehat{ut}$ with $t = \psi_v(D(e))$. Fig. 15 shows an example. Note that in this case, $D(e)$ does not intersect $D(e')$. Since u is on e, if b is also on e, then b is a (Type (2)) distinguished point of e; otherwise, there is a distinguished point that is an end point of e on $\widehat{ub}$. In either case, there is a distinguished point of e on $\widehat{ut}$. By Lemma 4.5, we have $|uu''| \leq |uv|$.

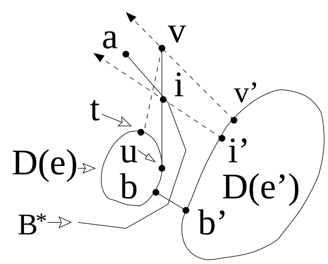

Figure 16: Illustrating Case L2, i.e., when i is on B_l^* and $\overline{uv}$ does not intersect $\overline{bb'}$.

Case L2 If $\overline{uv}$ does not intersect $\overline{bb'}$, then either $D(e) \cap D(e') = \emptyset$ or not can happen. Our analysis below is applicable to both these subcases. Fig. 16 shows an example when $D(e) \cap D(e') = \emptyset$. In the discussion of this case, all arcs we refer to are on $A_l(e)$ or $A_l(e')$. Thus, for any two points p and p' on $A_l(e)$ (or $A_l(e')$), the arc $\widehat{pp'}$ is well defined.

Starting from i, suppose we move along the boundary of the cell $c(e)$ counterclockwise, and let a be the first Voronoi vertex encountered that is incident to h (see Fig. 16). There are two tangent lines of $D(e)$ passing through a. Let a' be the first tangent point of these two tangent lines we meet when moving from u on $D(e)$ along the reversing direction. We will prove that a' must be on the arc $\widehat{ut}$. Since u is on e, if a' is also on e, then a' is a (Type

(4)) distinguished point of e; otherwise, there is a distinguished point that is an end point of e on $\widehat{ua'}$. In either case, by Lemma 4.5, we have $|uu''| \le |uv|$. Below, we prove that a' must be on $\widehat{ut}$.

Let $i' = \psi_i(D(e'))$. Let $\rho'(i)$ be the ray emanating from i with the same direction as $\rho_{i'}(D(e'))$ (i.e., the ray emanating from i' and going through i). Let $v' = \psi_v(D(e'))$. Let $\rho'(v)$ be the ray emanating from v with the same direction as $\rho_{v'}(D(e'))$. Let R' be region enclosed by $\rho'(i)$, $\overline{iv}$, and $\rho'(v)$.

Let $h(a, i)$ denote the part of h between a and i. We first show that for any point p on $h(a, i)$, $\overline{pp'}$ must intersect $\overline{vi}$, where $p' = \psi_p(D(e'))$. To this end, we prove below that $h(a, i)$ lies in the region R'.

We claim that $h(a, i)$ cannot intersect $\rho'(v)$. Assume to the contrary that this is not true and let x be their intersection point. Then $|xv| < |xv'| \le d(x, e')$, implying that x is closer to v (and to the pseudodisk on which v lies) than to e', contradicting with the fact that x is on h, which is the Voronoi edge shared by e and e'.

The line $l(i, i')$ intersects $h(a, i)$ at i. The line $l(i, i')$ divides the plane into two open half-planes. Denote by R the union of $l(i, i')$ and the open half-plane containing v. Since $\rho_{i'}(D(e'))$ intersects B^* at i, by Lemma 4.2, $h(i, a)$ must be contained in R. Since $h(i, a)$ cannot intersect $\rho'(v)$, $h(a, i)$ must lie in the region R'.

Based on Observation 1, for any point p on $h(i, a)$, $p' = \psi_p(D(e'))$ must be on the arc $\widehat{i'v'}$, and thus $\overline{pp'}$ must intersect $\overline{vi}$.

Since $\overline{uv}$ is a free common tangent, $\overline{vi}$ does not intersect $D(e')$. We have assumed that the interior of $\overline{uv}$ is not tangent to any arc of $\partial\mathcal{D}$. Therefore, for any p on $h(i, a)$, $d(p, \overline{vi}) < d(p, D(e'))$.

Assume to the contrary that the tangent point a' is not on $\widehat{ut}$. Then t is on the arc $\widehat{ua'}$. For any point p on $h(a, i)$, there are two tangent lines of $D(e)$ passing through p. Let $f(p)$ denote the first tangent point encountered when moving from u on $D(e)$ along the reversing direction. Since $f(i) = u$, $f(a) = a'$, and t is on the arc $\widehat{ua'}$, there must exist a point p on $h(a, i)$ such that $f(p) = t$. Then, the tangent line $l(p, t)$ is perpendicular to the line $l(v, t)$. Thus, $d(p, l(v, t)) = |pt|$. Note that p and $\overline{vi}$ are on different sides of the line $l(v, t)$. Hence $d(p, l(v, t)) \le d(p, \overline{vi})$. Also, note that $d(p, D(e)) \le |pt|$. Recall that $d(p, \overline{vi}) < d(p, D(e'))$. Therefore, we obtain $d(p, D(e)) < d(p, D(e'))$, contradicting with the fact that p lies on B^*. Therefore, a' is on the arc $\widehat{ut}$.

We conclude that in Case L2, $|uu''| \le |uv|$ holds.

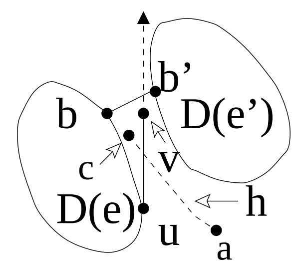

Figure 17: An example of Case R2: $\rho(v)$ first hits $\overline{bb'}$. The dashed vertical upward ray is $\rho(v)$.

4.2.2 Case R

We then analyze the case when i is on B^*_r. Let $\rho(v)$ be the ray starting at v and along the direction from u to v (i.e., u is not on $\rho(v)$). Note that $\overline{uv}$ does not intersect $D(e')$. There are three subcases: (R1) When $\overline{uv}$ intersects $\overline{bb'} \cup D(e')$ (similar to Fig. 15); (R2) when $\rho(v)$ intersects $\overline{bb'} \cup D(e')$ (see Fig. 17 and Fig. 18); (R3) when $\rho(v) \cup \overline{uv}$ does not intersect $\overline{bb'} \cup D(e')$ (see Fig. 20).

Case R1 If $\overline{uv}$ intersects $\overline{bb'} \cup D(e')$, then $\overline{uv}$ must intersect $\overline{bb'}$ since $\overline{uv}$ cannot intersect any pseudodisk in $\mathcal{D}$. (Note that $D(e)$ does not intersect $D(e')$ in this case.) The analysis for this case is the same as for Case L1. Namely, b is on the arc $\widehat{ut}$ (see Fig. 15).

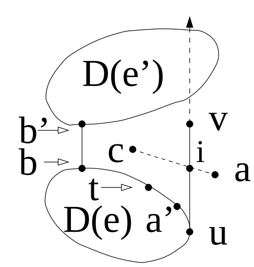

Figure 18: An example of Case R2: a and c are Voronoi vertices incident to h; either $\psi_a(D(e))$ or $\psi_c(D(e))$ must be on the arc $\widehat{ut}$.

Case R2 If $\rho(v)$ intersects $\overline{bb'} \cup D(e')$, then $\rho(v)$ first hits either $\overline{bb'}$ (e.g., see Fig. 17) or $D(e')$ (e.g., see Fig. 18). ($D(e)$ and $D(e')$ may or may not intersect in this case.) Let a be the Voronoi vertex incident to h as going from i on h clockwise around the cell $c(e)$, and c be the other Voronoi vertex incident to h. Let $a' = \psi_a(D(e))$ and $c' = \psi_c(D(e))$. Below, we show that a' or c' must be on the arc $\widehat{ut}$. Both

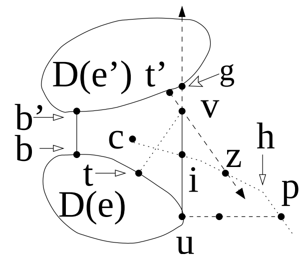

Figure 19: Illustrating an example of Case R2 when z is to the right of $l(u,v)$: The point a' must be on $\widehat{ut}$.

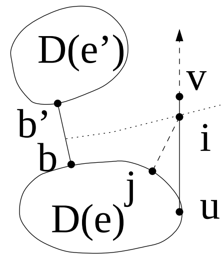

Figure 20: Illustrating an example of Case R3.

a' and c' are (Type (3)) distinguished points if they are on e; thus, if either one is on e, then as before, by Lemma 4.5, $|uu''| \le |uv|$ holds.

In the entire discussion for this case, all arcs we refer to are on $A_r(e)$ or $A_r(e')$. Thus, for any two points p and p' on $A_r(e)$ (or $A_r(e')$), the arc $\widehat{pp'}$ is well defined.

Let $t' = \psi_v(D(e'))$. Let $\overline{xy}$ be the common tangent of $D(e)$ and $D(e')$ intersecting B_r^* with x on $D(e)$ and y on $D(e')$. Since $\rho(v)$ intersects $\overline{bb'} \cup D(e')$ and $\overline{uv}$ is a free common tangent segment, v must be in the region enclosed by $\overline{xy}, \widehat{yb'}, \overline{b'b}$, and $\widehat{bx}$. Thus, t' must be on $\widehat{b'y}$. By Lemma 4.3, $\rho_{t'}(D(e'))$ must intersect B_r^*, and we let z be this intersection point. Recall that b'' is the intersection of the (thick) line $tl(b,b')$ and B^*.

If z is between b'' and i on B_r^*, i.e., z is to the left of $l(u,v)$, then let w be the intersection point of $\overline{vt}$ and B_r^* (e.g., see Fig. 14). Below, we prove that $c' = \psi_c(D(e))$ must be on $\widehat{ut}$. First of all, by Lemma 4.6, z must be between w and i. We claim that the Voronoi vertex c must be between z and i. If this is not the case, then then z must be between c and i, implying that z is on h. Note that $d(z,D(e')) = |zt'|$ and v is on $\overline{zt'}$. Thus, $d(z,v) < d(z,D(e')) \le d(z,e')$, which contradicts with the fact that h is a Voronoi edge shared by $c(e)$ and $c(e')$. Therefore, c is between z and i and thus between w and i. Since $t = \psi_v(D(e))$ and $\overline{uv}$ is tangent to $D(e)$, by Observation 2, c' must be on $\widehat{ut}$.

If z is not between b'' and i (e.g., see Fig. 19), then z is to the right of $l(u,v)$. Below, we prove that $a' = \psi_a(D(e))$ must be on $\widehat{ut}$. Since z is to the right of $l(u,v)$, t' must be to the left of $l(u,v)$. Since t' is on $A_r(e')$, the ray $\rho(v)$ cannot first hit $\overline{bb'}$ and thus $\rho(v)$ must first hit $D(e')$. Let g be the first point on $D(e')$ hit by $\rho(v)$.

Assume to the contrary that a' is not on $\widehat{ut}$. Then u must be on the arc $\widehat{a't}$. We prove in the following that z then lies on h and $d(z,v) < d(z,e')$, which contradicts with the fact that h is a Voronoi edge shared by $c(e)$ and $c(e')$.

Let $h(i,a)$ denote the portion of h between i and a. Since the point $\psi_i(D(e))$ is on $\widehat{ut}$ (by Observation 2) and $a' = \psi_a(D(e))$, there must exist a point p on $h(i,a)$ such that $u = \psi_p(D(e))$. We prove that z must be on h by showing that z is between i and p.

Since $\rho(v)$ first hits $D(e')$ at g, g is on the arc $\widehat{b'y}$. Note that $g \ne y$ since otherwise u would be a (Type (1)) distinguished point. By Lemma 4.3, $\rho_g(D(e'))$ intersects B_r^*, say at p'. Since t' is on $\widehat{b'g}$, by Observation 1, z is between i and p'. Thus, to prove z is between i and p, it suffices to prove that p' is between i and p. Let $u' = \psi_p(D(e'))$. Since p is on B_r^*, we have $|pu| = |pu'|$ and $\angle pu'u = \angle puu' \le \pi/2$. Due to $\angle pug = \pi/2$, it is easy to see that $g = \psi_{p'}(D(e'))$ is on the arc $\widehat{b'u'}$. By Observation 1, p' must be between b'' and p. Since p' is not between b'' and i, we obtain that p' is between i and p. Consequently, z is between i and p. Since both i and p are on h, z is on h.

But, we then have $d(z,v) < d(z,t') = d(z,D(e')) \le d(z,e')$, which contradicts with the fact that h is a Voronoi edge shared by $c(e)$ and $c(e')$. Hence, our assumption cannot be true, and a' must be on $\widehat{ut}$.

We conclude that in Case R2, we have $|uu''| \le |uv|$.

Case R3 If $\rho(v) \cup \overline{uv}$ does not intersect $\overline{bb'} \cup D(e')$ (e.g., see Fig. 20), then let $j = \psi_i(D(e))$. ($D(e)$ and $D(e')$ may or may not intersect in this case.) Suppose a is the endpoint of $A_r(e)$ other than b; then by Lemma 4.4, a is on $\widehat{uj}$. Note that j must be on $\widehat{ut}$. Thus a is on $\widehat{ut}$. If a is on e, then a is a (Type (1)) distinguished point of e; otherwise, there is a distinguished point which is an end point of e on $\widehat{ua}$. In either case, by Lemma 4.5, we have $|uu''| \le |uv|$.

The above analysis leads to Lemma 4.1 as well as the following results.

THEOREM 4.1. *A shortest s-to-t path amid n pseudodisks can be computed in $O(n^2)$ time.*

5 The Pairwise-disjoint Case

When $\mathcal{D}$ consists of n pairwise disjoint (non-polygonal) convex objects of $O(1)$ complexity each in the plane, our algorithm takes the following steps. (1) Use the algorithm in [13] to compute the visibility graph $\mathcal{G}_\mathcal{V}$ in $O(k + n \log n)$ time, where k is the size of $\mathcal{G}_\mathcal{V}$; (2) compute the Voronoi diagram of $\mathcal{D}$, which can be done in $O(n \log n)$ time; (3) find all distinguished points in $O(n)$ time; (4) compute the coalesced graph $\mathcal{G}^c_\mathcal{V}$, which takes $O(k+n \log n)$ time; (5) apply Dijkstra's algorithm with the Fibonacci heap to find a shortest s-to-t path in $\mathcal{G}^c_\mathcal{V}$, in $O(k + n \log n)$ time.

THEOREM 5.1. *A shortest s-to-t path amid n pairwise disjoint (non-polygonal) convex objects can be computed in $O(k+n \log n)$ time, where k is the size of the visibility graph of the objects.*

6 Constructing the Visibility Graph

In this section, we present an $O(n^2)$ time algorithm for constructing $\mathcal{G}_\mathcal{V}$, the visibility graph of $\mathcal{D}$. Note that since the complexity of each pseudodisk in $\mathcal{D}$ is $O(1)$, a common tangent of any two different pseudodisks in $\mathcal{D}$ can be computed in $O(1)$ time.

To construct $\mathcal{G}_\mathcal{V}$, we first compute $\partial\mathcal{D}$, the union boundary of all pseudodisks in $\mathcal{D}$. According to the results in [9], $\partial\mathcal{D}$ has $O(n)$ vertices formed by the intersections of the pseudodisks (each such vertex is a common endpoint of two arcs) and can be constructed in $O(n \log^2 n)$ time. It is easy to see that $\partial\mathcal{D}$ also has $O(n)$ arcs connecting pairs of consecutive vertices. For convenience, for each arc e of $\partial\mathcal{D}$, if there is a vertical line tangent to e, then we break e into two arcs by adding an artificial vertex to $\partial\mathcal{D}$ at the tangent point. Note that the new $\partial\mathcal{D}$ still has $O(n)$ vertices and arcs. Denote by L the set of lines each of which is either tangent to two arcs of $\partial\mathcal{D}$ (e.g., l_2 in Fig. 21) or tangent to one arc and passing through a vertex of $\partial\mathcal{D}$ (e.g., l_1 in Fig. 21). Then L has $O(n^2)$ lines. We should point out that the subset L_f of the lines in L which are tangent to two arcs of $\partial\mathcal{D}$ contains all the sought free common tangents for the visibility graph $\mathcal{G}_\mathcal{V}$; while the lines in the subset $L - L_f$ (which are tangent to one arc and passing through a vertex of $\partial\mathcal{D}$) do not contain free common tangents, they are needed by our algorithm for computing free common tangents.

For simplicity of discussion, we assume that no line in L is vertical. This assumption can be easily removed

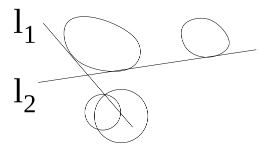

Figure 21: Two types of lines in L: l_1 is tangent to an arc and passes through a vertex of $\partial\mathcal{D}$, l_2 is tangent to two arcs, and $l_1 \prec l_2$.

without deteriorating the performance of our algorithm. In the following, we first create a partial order $\prec$ on the lines in L such that if two lines l_1 and l_2 are tangent to the same arc of $\partial\mathcal{D}$ or passing through the same vertex of $\partial\mathcal{D}$, then $l_1 \prec l_2$ if and only if the slope of l_1 is less than that of l_2 (see Fig. 21). Later, we will use this partial order to guide a slope sweeping algorithm to compute all free common tangents of $\partial\mathcal{D}$.

For each open arc of $\partial\mathcal{D}$, it is well known that its duality in the dual plane is a curve segment of constant complexity such that each point of the curve corresponds in the primal plane to a tangent line of the arc; further, the dual curve segment is x-monotone, i.e., any vertical line intersects it at most once. Denote by $\mathcal{A}$ the arrangement of all dual curve segments. By the duality, each intersection in $\mathcal{A}$ corresponds to a common tangent of two arcs of $\partial\mathcal{D}$ in the primal plane. Further, the dual of a vertex of $\partial\mathcal{D}$ is a line, and an intersection between the dual line and a curve in $\mathcal{A}$ corresponds in the primal plane to a line tangent to an arc of $\partial\mathcal{D}$ and passing through the vertex. Let $\mathcal{A}'$ be the arrangement obtained by adding the dual lines of all vertices of $\partial\mathcal{D}$ to $\mathcal{A}$. Denote by I the set of all intersections of $\mathcal{A}'$ except those formed by two intersecting dual lines of the vertices of $\partial\mathcal{D}$. We call the intersections in I the *interesting intersections*. Note that I corresponds in the primal plane to the line set L. Below, we create a partial order on I, which induces our sought partial order on L.

We first construct the arrangement $\mathcal{A}'$. The deterministic algorithm in [1] can compute the arrangement of $O(n)$ curve segments in $O(k+n \log n)$ time, where k is the number of intersections in the arrangement. Thus, $\mathcal{A}'$ can be built in $O(n^2)$ time. By sweeping $\mathcal{A}'$, we create a directed acyclic graph (DAG) G as follows. Each interesting intersection defines a vertex of G. On each curve segment or line in $\mathcal{A}'$, two interesting intersections v_1 and v_2 are *adjacent* if no other interesting intersection is in between them on that curve segment or line. For each pair of adjacent interesting intersections v_1 and v_2 on the same curve segment or line, without loss of generality, suppose v_1 is to the left of v_2; we add a directed edge from v_1 to v_2 in G. Thus, G has $O(n^2)$ vertices and $O(n^2)$ directed edges, and can be built in $O(n^2)$

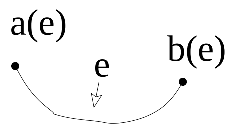

Figure 22: An arc e and its two endpoints $a(e)$ and $b(e)$.

time by sweeping $\mathcal{A}'$. For convenience, we add two additional vertices v' and v''; for each vertex v in G, add a directed edge from v' to v and a directed edge from v to v''. This completes our construction of G, which takes $O(n^2)$ time. By a topological sort of G from v' to v'', we obtain a list of all interesting intersections in I. Note that for any pair of interesting intersections v_1 and v_2 on the same curve segment or line, v_1 precedes v_2 in this list if and only if v_1 is to the left of v_2, that is, the line dual to v_1 in the primal plane has a smaller slope than the one dual to v_2. Therefore, this list on I corresponds exactly in the primal plane to a partial order $\prec$ on the line set L. Clearly, the topological sort of G takes $O(n^2)$ time.

LEMMA 6.1. *A partial order $\prec$ on the lines of L can be produced in $O(n^2)$ time.*

Next, by using the partial order on L obtained in Lemma 6.1, we compute all free common tangents of $\mathcal{D}$ and subsequently build the visibility graph $\mathcal{G}_{\mathcal{V}}$. Our idea is that for each arc, we shoot two rays from a point p on the arc along opposite directions such that the line formed by these two rays is tangent to the arc at p; then we move p along the arc and keep updating the corresponding two rays. Throughout the algorithm, the arc in $\partial\mathcal{D}$ hit first by every ray is maintained and updated accordingly. The algorithm stops when each arc has been fully swept by the moving point on it. Note that while our algorithm can be viewed as a generalization of Welzl's algorithm [17], it is capable of handling the intersecting case.

We first introduce some terminology. Each arc e of $\partial\mathcal{D}$ has two endpoints (which are two vertices of $\partial\mathcal{D}$); denote by $a(e)$ the endpoint such that when moving along e counterclockwise from $a(e)$, one stays on e until the other endpoint, denoted by $b(e)$, is met (see Fig. 22). Let $F(e)$ be the set of all free common tangents between e and other arcs of $\partial\mathcal{D}$. For any point p on e, denote by $l(p)$ the line tangent at p to the pseudodisk on which the arc e lies. We split $l(p)$ into two rays at p such that the two rays have the same origin p but opposite directions; let $\rho_1(p)$ be any one of these two rays and $\rho_2(p)$ be the other ray. Further, for a ray ρ, let $\sigma(\rho)$ denote the arc of $\partial\mathcal{D}$ that is hit first by ρ if it exists and $\emptyset$ otherwise.

For each arc e, if we move a point p along e from $a(e)$ to $b(e)$, then the values of $\sigma(\rho_1(p))$ and $\sigma(\rho_2(p))$ possibly change accordingly, and at the moment when either value changes, $l(p)$ is either tangent to another arc of $\partial\mathcal{D}$ or passing through a vertex of $\partial\mathcal{D}$. Given a slope-sorted list of all lines each of which is either a common tangent between e and another arc, or tangent to e and passing through a vertex of $\partial\mathcal{D}$, the set $F(e)$ can be computed easily in $O(n \log n)$ time as for a visibility problem (with $\partial\mathcal{D}$ as the opaque objects) by sweeping a point p (and thus the line $l(p)$) from $a(e)$ to $b(e)$. Actually, we can obtain this sorted list in $O(n)$ time from the arrangement $\mathcal{A}'$. But, doing this one by one for each arc would yield only an $O(n^2 \log n)$ time solution. For a faster algorithm, we use the following slope or angular sweep scheme which sweeps all arcs of $\partial\mathcal{D}$ simultaneously to compute $\cup_{e\in\partial\mathcal{D}} F(e)$ in $O(n^2)$ time. Note that $\cup_{e\in\partial\mathcal{D}} F(e)$ is the set of all free common tangents of $\mathcal{D}$.

For each arc $e \in \partial\mathcal{D}$, denote by α_e and β_e the slopes of the two lines $l(a(e))$ and $l(b(e))$, respectively; if $l(a(e))$ (resp., $l(b(e))$) is vertical, then let α_e (resp., β_e) be $-\infty$ (resp., $+\infty$). Even if there is an artificial vertex on $e \in \partial\mathcal{D}$, as a point p moves from $a(e)$ to $b(e)$ along e, the slope of $l(p)$ increases continuously from α_e to β_e. Thus, for any slope γ with $\alpha_e \le \gamma \le \beta_e$, there exists a point p on e such that the slope of $l(p)$ is γ; we denote this line $l(p)$ by $l(e,\gamma)$, and denote $\rho_1(p)$, $\rho_2(p)$, $\sigma(\rho_1(p))$, and $\sigma(\rho_2(p))$ by $\rho_1(e,\gamma)$, $\rho_2(e,\gamma)$, $\sigma_1(e,\gamma)$, and $\sigma_2(e,\gamma)$, respectively. Consider the slope sweep process with a sweep slope γ. If $\alpha_e \le \gamma \le \beta_e$, then we say the arc e is *active*. Initially, we compute, for each arc $e \in \partial\mathcal{D}$, the values of $\sigma(\rho_1(a(e)))$ and $\sigma(\rho_2(a(e)))$. This can be easily done in totally $O(n^2)$ time. During the sweeping, at any moment with a sweep slope γ, for each active arc e, we maintain the values $\sigma_1(e,\gamma)$ and $\sigma_2(e,\gamma)$. As γ increases, the values of $\sigma_1(e,\gamma)$ and $\sigma_2(e,\gamma)$ of all arcs $e \in \partial\mathcal{D}$ do not change until one of the following events occurs.

1. If an arc e starts to be active, i.e., $\gamma = \alpha_e$, then since we know the values $\sigma(\rho_1(a(e)))$ and $\sigma(\rho_2(a(e)))$, we simply set $\sigma_1(e,\gamma) = \sigma(\rho_1(a(e)))$ and $\sigma_2(e,\gamma) = \sigma(\rho_2(a(e)))$.

2. If an arc e is about to be deactivated, i.e., $\gamma = \beta_e$, we set $\sigma_1(e,\gamma) = \emptyset$ and $\sigma_2(e,\gamma) = \emptyset$.

3. There is an active arc e whose line $l(e,\gamma)$ becomes tangent to another arc e'. Thus, one of the two rays $\rho_1(e,\gamma)$ and $\rho_2(e,\gamma)$ must be tangent to e' (say, it is $\rho_1(e,\gamma)$). Note that we already know the value of $\sigma_1(e,\gamma)$. If $\rho_1(e,\gamma)$ first hits the arc $\sigma_1(e,\gamma)$ strictly before e', then we do nothing; otherwise, there are two cases to consider.

   Since $l(e,\gamma)$ is tangent to e', $l(e,\gamma)$ must be over-

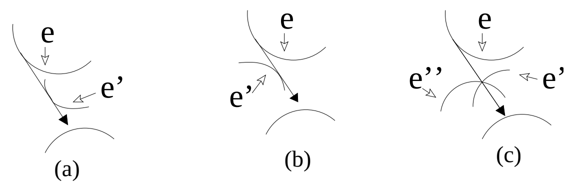

Figure 23: Illustrating some events of the sweep algorithm: (a) The ray $\rho_1(e,\gamma)$ is "entering" the arc e'; (b) $\rho_1(e,\gamma)$ is "leaving" e'; (c) $\rho_1(e,\gamma)$ is passing through the intersection of e' and e''.

lapping with $l(e',\gamma)$ right at this moment. Without loss of generality, assume $\rho_1(e',\gamma)$ is of the same direction as $\rho_1(e,\gamma)$. For an infinitesimal $\epsilon > 0$, if the ray $\rho_1(e,\gamma+\epsilon)$ still intersects e' (i.e., the ray is "entering" e'; see Fig. 23(a)), then we set the value of $\sigma_1(e,\gamma)$ to e'; otherwise (i.e., the ray is "leaving" e'; see Fig. 23(b)), we set $\sigma_1(e,\gamma)$ to the value of $\sigma_1(e',\gamma)$.

4. The line $l(e,\gamma)$ of an active arc e intersects a vertex of $\partial\mathcal{D}$, which is a common endpoint of two arcs, say e' and e'' (see Fig. 23(c)). Without loss of generality, assume that the ray $\rho_1(e,\gamma)$ intersects the vertex. Note that we already know the value of $\sigma_1(e,\gamma)$. If $\rho_1(e,\gamma)$ first hits the arc $\sigma_1(e,\gamma)$ strictly before this vertex, then we do nothing; otherwise, the ray $\rho_1(e,\gamma+\epsilon)$ must intersect one of e' and e'' for an infinitesimal $\epsilon > 0$ (say, $\rho_1(e,\gamma+\epsilon)$ intersects e'; see Fig. 23(c)), and we set $\sigma_1(e,\gamma)$ to e'.

Note that each of the above events can be handled in $O(1)$ time. The algorithm stops when every arc has been deactivated. The number of the total events is bounded by $O(n^2)$.

It remains to find a method to guide the above slope sweep algorithm. That is, we need an appropriately ordered list of all events. Note that except those events for activating or deactivating an arc (i.e., the first two types of events above), each event is determined by a line in L that is either tangent to two arcs or tangent to one arc and passing through a vertex of $\partial\mathcal{D}$, which we call a *relevant line*. Every event for activating an arc e is determined by an *activating line* that is tangent to e at $a(e)$. Actually, handling those events for deactivating arcs is trivial (and thus is ignored). Denote by L' the set of all activating lines. Clearly, $|L| = O(n^2)$ and $|L'| = O(n)$. A sorted list of the lines in $L \cup L'$ by their slopes will certainly do the job for our sweep algorithm. But, it takes $O(n^2 \log n)$ time to compute such a sorted list. As in [17], it turns out that a partial order of $L \cup L'$ is sufficient, as shown below.

For an arc e, denote by $l'(e,e')$ the relevant line tangent to e that is also tangent to e' if e' is an arc or passes through e' if e' is a vertex of $\partial\mathcal{D}$; we say that e and e' *support* $l'(e,e')$. Consider two relevant lines $l'(e_1,e'_1)$ and $l'(e_2,e'_2)$. A critical observation is: If $\{e_1,e'_1\} \cap \{e_2,e'_2\} = \emptyset$, then no matter which one of the two events associated with $l'(e_1,e'_1)$ and $l'(e_2,e'_2)$ is processed first, the sweep algorithm always works correctly. In other words, the algorithm works as long as the order of the lines in $L \cup L'$ has the following properties: (1) For every arc e, its activating line is in front of any relevant line supported by e; (2) for any pair of relevant lines l_1 and l_2 supported by the same arc or vertex, l_1 is in front of l_2 if and only if the slope of l_1 is smaller than that of l_2.

Note that property (2) above on the order of the relevant lines in L is captured by the partial order $\prec$ of L from Lemma 6.1. On the other hand, in $O(n)$ time, we can compute the set L'. To satisfy property (1) above, a simple way is to process all activating lines (in an arbitrary order) at the beginning of the sweeping process. Hence, a needed (partial) order of $L \cup L'$ can be generated in $O(n^2)$ time. Consequently, all free common tangents of $\mathcal{D}$ are computed in $O(n^2)$ time.

Note that after the sweeping, for each arc e, we also obtain the ordered list of all tangent points of the free common tangents in $F(e)$ along e from $a(e)$ to $b(e)$. Therefore, by scanning each arc of $\partial\mathcal{D}$, the visibility graph $\mathcal{G}_\mathcal{V}$ is built in $O(n^2)$ time. Thus, we obtain Theorem 2.1.

7 Conclusion

When using visibility graph to find a shortest path, Dijkstra's algorithm normally gives an $O(n^2 \log n)$ time solution. The visibility graph usually has $O(n^2)$ vertices and $O(n^2)$ edges, which has been a bottleneck for further improving the Dijkstra's shortest path algorithm. In this paper, we derive a new approach to transform the visibility graph to a coalesced graph which has $O(n)$ vertices and $O(n^2)$ edges such that a shortest path in the visibility graph can be obtained from a corresponding shortest path in the coalesced graph. Running Dijkstra's algorithm on the coalesced graph needs only $O(n^2)$ time, which improves the previous result. Our techniques are likely to be extended for solving other related problems.

References

[1] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing the arrangement of curve segments: Divide-and-conquer algorithms via sampling. In *Proc. of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 705–706, 2000.

[2] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1(1):49–63, 1986.

[3] E. Chang, S. Choi, D. Kwon, H. Park, and C. Yap. Shortest path amidst disc obstacles is computable. In *Proc. of the 21st Annual Symposium on Computational Geometry*, pages 116–125, 2005.

[4] L.P. Chew. Planning the shortest path for a disc in $O(n^2 \log n)$ time. In *Proc. of the first Annual Symposium on Computational Geometry*, pages 214–220, 1985.

[5] S.K. Ghosh and D.M. Mount. An output-sensitive algorithm for computing visibility. *SIAM Journal on Computing*, 20(5):888–910, 1991.

[6] J. Hershberger and L.J. Guibas. An $O(n^2)$ shortest path algorithm for a non-rotating convex body. *Journal of Algorithms*, 9(1):18–46, 1988.

[7] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.

[8] S. Kapoor and S.N. Maheshwari. Efficient algorithms for Euclidean shortest path and visibility problems with polygonal obstacles. In *Proc. of the 4th Annual Symposium on Computational Geometry*, pages 172–182, 1988.

[9] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete and Computational Geometry*, 1(1):59–71, 1986.

[10] J.S.B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Annals of Mathematics and Artificial Intelligence*, 3(1):83–105, 1991.

[11] J.S.B. Mitchell. Shortest paths among obstacles in the plane. *International Journal of Computational Geometry and Applications*, 6(3):309–332, 1996.

[12] M.H. Overmars and E. Welzl. New methods for computing visibility graphs. In *Proc. of the 4th Annual Symposium on Computational Geometry*, pages 164–171, 1988.

[13] M. Pocchiola and G. Vegter. Computing the visibility graph via pseudo-triangulations. In *Proc. of the 11th Annual Symposium on Computational Geometry*, pages 248–257, 1995.

[14] H. Rohnert. Shortest paths in the plane with convex polygonal obstacles. *Information Processing Letters*, 23(2):71–76, 1986.

[15] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15(1):193–215, 1986.

[16] J.A. Storer and J.H. Reif. Shortest paths in the plane with polygonal obstacles. *Journal of the ACM (JACM)*, 41(5):982–1012, 1994.

[17] E. Welzl. Constructing the visibility graph for n line segments in $O(n^2)$ time. *Information Processing Letters*, 20:167–171, 1985.

Appendix

Lemma 3.2 *A shortest s-to-t path in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a shortest s-to-t path in $\mathcal{G}^d_{\mathcal{V}}$ with the same length.*

Proof. For discrimination, we refer to the length of a path in $\mathcal{G}^c_{\mathcal{V}}$ as its *weight* and let s' and t' denote s and t in $\mathcal{G}^c_{\mathcal{V}}$, respectively. Our goal is to show that a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a shortest s-to-t path in $\mathcal{G}^d_{\mathcal{V}}$. To this end, we first show that any shortest s-to-t path P in $\mathcal{G}^d_{\mathcal{V}}$ corresponds to an s'-to-t' path P' in $\mathcal{G}^c_{\mathcal{V}}$ such that the length of P is equal to the weight of P'; we then show that any s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ that does not correspond to a (forward-going) path in $\mathcal{G}^d_{\mathcal{V}}$ cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$.

For any path P_x in $\mathcal{G}^d_{\mathcal{V}}$, we use $L(P_x)$ to denote the length of P_x. For any path P_y in $\mathcal{G}^c_{\mathcal{V}}$, we use $W(P_y)$ to denote the *weight* of P_y, i.e., the sum of the weights of all edges on P_y. We use $e(y_i, y_j)$ to denote an edge in $\mathcal{G}^c_{\mathcal{V}}$ connecting two vertices y_i and y_j with direction from y_i to y_j. For any vertex $x \in \mathcal{G}^d_{\mathcal{V}}$, let $H(x)$ be its host interval on an arc of $\partial\mathcal{D}$. For any vertex $y \in \mathcal{G}^c_{\mathcal{V}}$, let $I(y)$ be the directed interval defining y.

We first show that any shortest s-to-t path P_x in $\mathcal{G}^d_{\mathcal{V}}$ corresponds to an s'-to-t' path P_y in $\mathcal{G}^c_{\mathcal{V}}$ with $L(P_x) = W(P_y)$. Let $P_x = (s = x_1, x_2, \ldots, x_{k-1}, x_k = t)$, where each x_i $(1 \le i \le k)$ is a vertex in $\mathcal{G}^d_{\mathcal{V}}$. Recall that any path in $\mathcal{G}^d_{\mathcal{V}}$, particularly P_x, is forward-going. Based on P_x, an s'-to-t' path P_y in $\mathcal{G}^c_{\mathcal{V}}$ can be constructed in a natural way. We give the details below.

Let $P_y = (s' = y_1, y_2, \ldots, y_{m-1}, y_m = t')$, where each y_i $(1 \le i \le m)$ is a vertex in $\mathcal{G}^c_{\mathcal{V}}$. We construct P_y using P_x. First, $y_1 = s'$. Suppose we already constructed the subpath of P_y from y_1 to y_i based on the subpath of P_x from x_1 to x_j (initially, $i = 1$ and $j = 1$). The next vertex y_{i+1} of P_y is obtained as follows. Consider the next edge $e(x_j, x_{j+1})$ of P_x. If $H(x_j)$ and $H(x_{j+1})$ are two different intervals, we let y_{i+1} be the vertex defined by $H(x_{j+1})$. Note that in this case, regardless of whether $e(x_j, x_{j+1})$ is a tangent edge or an arc edge in $\mathcal{G}^d_{\mathcal{V}}$, there must be a directed edge $e(y_i, y_{i+1})$ in $\mathcal{G}^c_{\mathcal{V}}$. If $H(x_j)$ and $H(x_{j+1})$ are the same interval, we do nothing (i.e., y_{i+1} is not produced) and simply continue on considering the next edge $e(x_{j+1}, x_{j+2})$ of P_x. P_y is obtained after the last edge $e(x_{k-1}, t)$ of P_x is considered.

We next prove $L(P_x) = W(P_y)$. We first define a set of *artificial vertices* in both $\mathcal{G}^c_{\mathcal{V}}$ and $\mathcal{G}^d_{\mathcal{V}}$ based on the tangent edges in $\mathcal{G}^d_{\mathcal{V}}$. For any tangent edge $e(u, v)$ in $\mathcal{G}^d_{\mathcal{V}}$, suppose $H(u)$ is (u', u'') and $H(v)$ is (v', v''), and $u' \to u \to v \to v''$ forms a forward-going path (see Fig. 6). Let y_u (resp., y_v) be the vertex in $\mathcal{G}^c_{\mathcal{V}}$ defined by (u', u'') (resp., (v', v'')). Then there is an edge $e(y_u, y_v)$ in $\mathcal{G}^c_\mathcal{V}$. We add in $\mathcal{G}^c_\mathcal{V}$ an *artificial vertex* y_{uv} and replace the edge $e(y_u, y_v)$ by two *artificial edges* $e(y_u, y_{uv})$ and $e(y_{uv}, y_v)$. Set $W(e(y_u, y_{uv})) = |uv| - |uu''|$ and $W(e(y_{uv}, y_v)) = |vv''|$. Thus, the weight of either artificial edge is nonnegative, and the sum of their weights is equal to the weight of the original edge $e(y_u, y_v)$. Note that the above operations are used only for analyzing this lemma. We call the original edge $e(y_u, y_v)$ in $\mathcal{G}^c_\mathcal{V}$ the *broken edge* of y_{uv} and the tangent edge $e(u, v)$ in $\mathcal{G}^d_\mathcal{V}$ the *generator edge* of y_{uv}. We also call the vertex v in $\mathcal{G}^d_\mathcal{V}$ the *corresponding artificial vertex* of y_{uv}, denoted by $f(y_{uv})$, i.e., $f(y_{uv}) = v$. Note that t' in $\mathcal{G}^c_\mathcal{V}$ is an artificial vertex and $f(t') = t$. In summary, each tangent edge in $\mathcal{G}^d_\mathcal{V}$ defines an artificial vertex in both $\mathcal{G}^c_\mathcal{V}$ and $\mathcal{G}^d_\mathcal{V}$. For convenience, s' in $\mathcal{G}^c_\mathcal{V}$ is also considered as an artificial vertex, and $f(s') = s$ in $\mathcal{G}^d_\mathcal{V}$. Below, we prove that for each artificial vertex y_{uv} on P_y, the weight of the subpath from s' to y_{uv} along P_y is equal to the length of the subpath from s to $f(y_{uv})$ along P_x.

Let $(s = x'_1, x'_2, \ldots, x'_l = t)$ be the sequence of artificial vertices along P_x from s to t, and $(s' = y'_1, y'_2, \ldots, y'_{l'} = t')$ be the sequence of artificial vertices along P_y from s' to t'. By the definitions of artificial vertices, we have $l = l'$ and for each $1 \leq i \leq l$, $f(y'_i) = x'_i$. For any $x'_i \in P_x$, let $L(s, x'_i)$ denote the length of the s-to-x'_i subpath of P_x, and for any $y'_i \in P_y$, let $W(s', y'_i)$ denote the weight of the s'-to-y'_i subpath of P_y. Below we prove by induction $L(s, x'_i) = W(s', y'_i)$ for each $1 \leq i \leq l$. Initially, $L(s, x'_1) = W(s', y'_1) = 0$ holds. Assume $L(s, x'_{i-1}) = W(s', y'_{i-1})$ for $1 < i \leq l$. To show $L(s, x'_i) = W(s', y'_i)$, let $e(y_j, y_{j+1})$ in $\mathcal{G}^c_\mathcal{V}$ be the broken edge of the artificial vertex y'_{i-1}, and $e(y_h, y_{h+1})$ in $\mathcal{G}^c_\mathcal{V}$ be the broken edge of y'_i; let $e(u', u)$ in $\mathcal{G}^d_\mathcal{V}$ be the generator edge of y'_{i-1}, and $e(v, v')$ be the generator edge of y'_i. (Note that $x'_{i-1} = u$ and $x'_i = v'$.) Since y'_{i-1} and y'_i are two consecutive artificial vertices on P_y, they are defined by two tangent edges on P_x (i.e., $e(u', u)$ and $e(v, v')$) and no other tangent edge is between them on P_x. Hence, the two tangent points u and v must be on the same arc of $\partial\mathcal{D}$. Thus, $L(s, x'_i) = L(s, x'_{i-1}) + |uv| + |vv'|$ (with $u = x'_{i-1}$ and $x'_i = v'$). To compute $W(s', y'_i)$, note that y_{j+1} is defined by the host interval of u and y_h is defined by the host interval of v. Depending on whether u and v have the same host interval, there are two cases.

1. If u and v have the same host interval (i.e., $y_{j+1} = y_h$), say (u'', v'') (e.g., see Fig. 24), then we have $W(s', y'_i) = W(s', y'_{i-1}) + W(e(y'_{i-1}, y_h)) + W(e(y_h, y'_i))$. Note that $W(e(y'_{i-1}, y_h)) = |uv''|$ and $W(e(y_h, y'_i)) = |vv'| - |vv''|$. Thus, we have $W(s', y'_i) = W(s', y'_{i-1}) + |uv''| + |vv'| - |vv''| = W(s', y'_{i-1}) + |uv| + |vv'|$. Consequently, due to $L(s, x'_{i-1}) = W(s', y'_{i-1})$, we have $L(s, x'_i) = W(s', y'_i)$.

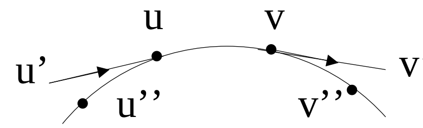

Figure 24: The artificial vertex x'_{i-1} is u and x'_i is v'. The subpath from x'_{i-1} to x'_i in $\mathcal{G}^d_\mathcal{V}$ has a length $|uv| + |vv'|$, which is equal to the weight of the subpath from y'_{i-1} to y'_i in $\mathcal{G}^c_\mathcal{V}$, i.e., $|uv''| + (|vv'| - |vv''|)$.

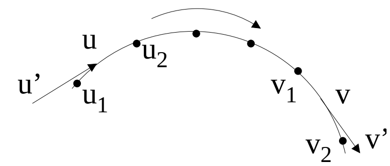

Figure 25: All black points are distinguished points; u is in the interval (u_1, u_2), and v is in (v_1, v_2).

2. If u and v have different host intervals, (i.e., y_{j+1} and y_h are two different vertices in $\mathcal{G}^c_\mathcal{V}$), then let the vertices on P_y between y_j and y_{h+1} be $y_j, y'_{i-1}, y_{j+1}, z_1, z_2, \ldots, z_k, y_h, y'_i, y_{h+1}$. Since the two intervals defining y_{j+1} and y_h are on the same arc, the intervals defining the vertices $y_{j+1}, z_1, z_2, \ldots, z_k, y_h$ are all on the same arc. It is possible that no other vertex is between y_{j+1} and y_h on P_y. Fig. 25 shows an example.

   Let W' be the sum of the weights of all edges on the subpath of P_y from y_{j+1} to y_h. We then have $W(s', y'_i) = W(s', y'_{i-1}) + W(e(y'_{i-1}, y_{j+1})) + W' + W(e(y_h, y'_i))$. Suppose the interval defining y_{j+1} is (u_1, u_2) and the interval defining y_h is (v_1, v_2) (thus the tangent point u is contained in (u_1, u_2) and the tangent point v is contained in (v_1, v_2); see Fig. 25). Based on the definitions of the edge weights in $\mathcal{G}^c_\mathcal{V}$, $W(e(y'_{i-1}, y_{j+1})) = |uu_2|$ and $W(e(y_h, y'_i)) = |vv'| - |vv_2|$. Further, it is easy to verify that $W' = |u_2v_2|$. Thus, we obtain $W(s', y'_i) = W(s', y'_{i-1}) + |uu_2| + |u_2v_2| + |vv'| - |vv_2| = W(s', y'_{i-1}) + |uv| + |vv'|$. Due to $L(s, x'_{i-1}) = W(s', y'_{i-1})$, we have $L(s, x'_i) = W(s', y'_i)$.

Consequently, we obtain $L(P_x) = W(P_y)$.

A similar analysis as above can also give the following result: For any artificial vertex $y \in \mathcal{G}^c_\mathcal{V}$, a shortest path P from s to the artificial vertex $f(y)$ in $\mathcal{G}^d_\mathcal{V}$ corresponds to a path P' from s' to y in $\mathcal{G}^c_\mathcal{V}$ with $L(P) = W(P')$. We omit the details of this proof here. Since for any vertex x in $\mathcal{G}^d_\mathcal{V}$, a shortest path from s to x in $\mathcal{G}^d_\mathcal{V}$ is also a shortest path from s to the point x

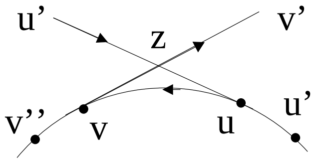

Figure 26: The subpath $u' \to u \to v \to v'$ has a weight $|u'u|+|vv'|-|vu|$ in $\mathcal{G}^c_{\mathcal{V}}$, which is actually smaller than the length of the corresponding geometric path in $\mathcal{G}_{\mathcal{V}}$, i.e., $|u'u|+|vv'|+|vu|$, but is still bigger than the length of the geometric path $u' \to z \to v'$.

in the plane, we have the following observation (which will be useful later): For any artificial vertex $y \in \mathcal{G}^c_{\mathcal{V}}$, suppose P is a shortest path from s to the point $f(y)$ in the plane and P' is a minimum weight path from s' ($= s$) to y in $\mathcal{G}^c_{\mathcal{V}}$, then it must be $W(P') \le L(P)$.

We then show that any s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ that does not correspond to a path in $\mathcal{G}^d_{\mathcal{V}}$ cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$. Note that any path in $\mathcal{G}^d_{\mathcal{V}}$ is a forward-going path in $\mathcal{G}_{\mathcal{V}}$ and any forward-going path in $\mathcal{G}_{\mathcal{V}}$ is a path in $\mathcal{G}^d_{\mathcal{V}}$. Thus, it suffices to show that any s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ that does not correspond to a forward-going path in $\mathcal{G}_{\mathcal{V}}$ cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$.

Given an s'-to-t' path P_y in $\mathcal{G}^c_{\mathcal{V}}$, let $P_y = (s' = y_1, y_2, \ldots, y_m = t')$, where each y_i is a vertex in $\mathcal{G}^c_{\mathcal{V}}$. Based on P_y, we obtain an s-to-t path $P_x = (s = x_1, x_2, \ldots, x_{k-1}, x_k = t)$ in $\mathcal{G}_{\mathcal{V}}$, as follows. Initially, $x_1 = s$. Assume we already obtained the subpath $(x_1, \ldots, x_j)$ based on the subpath $(y_1, \ldots, y_i)$. We obtain x_{j+1} and beyond on P_x in the following way. Consider the next vertex y_{i+1} of P_y. For any $1 \le g \le m$, let e_g be the arc of $\partial\mathcal{D}$ on which the interval $I(y_g)$ lies (recall that $I(y_g)$ is the directed interval defining y_g). Depending on whether e_i and e_{i+1} are the same arc, there are two cases.

1. If e_i and e_{i+1} are two different arcs, then the edge $e(y_i, y_{i+1})$ in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a tangent edge $e(u', u)$ in $\mathcal{G}_{\mathcal{V}}$ that is a free common tangent between e_i and e_{i+1}. Suppose u' is on e_i and u is on e_{i+1}. If x_j is u', we let $x_{j+1} = u$; otherwise, we let $x_{j+1} = u'$ and $x_{j+2} = u$. We then continue considering the next vertex y_{i+2} of P_y.

2. If e_i and e_{i+1} are the same arc, which means that the two intervals $I(y_i)$ and $I(y_{i+1})$ are adjacent on e_i, then we do nothing (i.e., x_{j+1} is not determined) and continue considering the next vertex y_{i+2} of P_y.

The path P_x in $\mathcal{G}_{\mathcal{V}}$ is obtained after $y_m = t'$ in P_y is considered.

Suppose P_x thus produced from P_y is not a forward-going path. Then we show below that P_y cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$. To this end, we first examine what can cause P_x to be not forward-going. Suppose we have just constructed $x_{j+1} = u'$ and $x_{j+2} = u$ as in Case 1 above. Next, we discuss one more step to construct P_x, i.e., constructing x_{j+3} and beyond. We consider the next vertex y_{i+2} of P_y.

1. If e_{i+2} is the same as e_{i+1}, implying that the two intervals $I(y_{i+1})$ and $I(y_{i+2})$ are adjacent and have the same direction, then we do nothing (i.e., x_{j+3} is not determined) and continue considering the next vertex y_{i+3} of P_y. Actually, x_{j+3} is not determined until we find the first (i.e., the smallest) l with $l \ge 3$ such that e_{i+l} is not e_{i+1}. Then, the edge $e(y_{i+l-1}, y_{i+l})$ in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a tangent edge $e(v, v')$ in $\mathcal{G}_{\mathcal{V}}$ that is a free common tangent between e_{i+1} and e_{i+l}. Suppose v is on e_{i+1} and v' is on e_{i+l}. Fig. 25 shows an example. Call the tangent edge $e(u', u)$ the *incoming tangent* of the arc e_{i+1} and $e(v, v')$ the *outgoing tangent* of e_{i+1}.

   Let $I(y_{i+1})$ be (u_1, u_2) and $I(y_{i+l-1})$ be (v_1, v_2) (see Fig. 25). Thus u is in (u_1, u_2) and v is in (v_1, v_2). Also, $\overline{u'u}$ and $\widehat{uu_2}$ form a convex chain, and $\widehat{v_1v}$ and $\overline{vv'}$ form a convex chain. Since $I(y_{i+1})$ and $I(y_{i+2})$ are not the same interval, intervals $I(y_{i+1})$ and $I(y_{i+l-1})$ are also not the same, implying that the direction from u' to u is consistent with the direction from u_2 to v_1 along the arc. In other words, $\overline{u'u}$, $\widehat{uu_2}$, and $\widehat{u_2v_1}$ form a convex chain; similarly, $\widehat{u_2v_1}$ $\widehat{v_1v}$, and $\overline{vv'}$ form a convex chain. (Note that $u_2 = v_1$ is possible; in that case, (u_2, v_1) is a closed directed interval.) Thus, $\overline{u'u}$, $\widehat{uv}$, and $\overline{vv'}$ must form a convex chain, i.e., the subpath $u' \to u \to v \to v'$ is forward-going.

   Hence, if $e_{i+2} = e_{i+1}$, then the subpath of P_x formed by the incoming tangent of e_{i+1}, the sub-arc of e_{i+1} on P_x, and the outgoing tangent of e_{i+1} is always forward-going.

2. If $e_{i+2} \neq e_{i+1}$, then the edge $e(y_{i+1}, y_{i+2})$ in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a tangent edge $e(v, v')$ in $\mathcal{G}_{\mathcal{V}}$ that is a free common tangent between e_{i+1} and e_{i+2}. Suppose v is on e_{i+1} and v' is on e_{i+2}. Then the two tangent points, u of $e(u', u)$ and v of $e(v, v')$, are both in the interval $I(y_{i+1})$. But, it is possible that the subpath $u' \to u \to v \to v'$ enters the interval $I(y_{i+1})$, backs up, and then leaves (see Fig. 26), making it not forward-going.

When P_x is not a forward-going path, based on the analysis above, P_x must contain a subpath $u' \to u \to v \to v'$ that enters an interval, backs up, and then leaves, as shown in Fig. 26. We denote this subpath by P^x_{sub}. Without loss of generality, we assume that P^x_{sub} is the first such subpath on P_x from s. Denote by (v'', u'') the directed interval containing both u and v. Let y_{i+1} be the vertex on P_y defined by (v'', u''). Since y_{i+1} cannot be s' or t' (otherwise, P^x_{sub} is forward-going), y_{i+1} has both a predecessor y_i and a successor y_{i+2} on P_y. We add a new vertex y' to P_y, and replace the edge $e(y_i, y_{i+1})$ by two new edges $e(y_i, y')$ and $e(y', y_{i+1})$. Set $W(e(y', y_{i+1})) = |u'u| + |uu''|$ and $W(e(y_i, y')) = W(e(y_i, y_{i+1})) - W(e(y', y_{i+1}))$. Thus, the sum of the weights of the two new edges is the same as the weight of the original edge. For any vertex x on P_x, let $L(s, x)$ be the length of the subpath of P_x from s to x. For any vertex y on P_y, let $W(s', y)$ be the weight of the subpath of P_y from s' to y. Since P^x_{sub} is the first such non-forward-going subpath on P_x, it is easy to verify that $L(s, u') = W(s', y')$. Note that on the edge $e(y_{i+1}, y_{i+2})$ of P_y, there is an artificial vertex y'' in $\mathcal{G}^c_{\mathcal{V}}$ with two artificial edges $e(y_{i+1}, y'')$ and $e(y'', y_{i+2})$. Recall that $W(e(y_{i+1}, y'')) = |vv'| - |vu''|$ and $W(e(y'', y_{i+2})) = W(e(y_{i+1}, y_{i+2})) - W(e(y_{i+1}, y''))$. The corresponding artificial vertex of y'' in $\mathcal{G}^d_{\mathcal{V}}$ is v', i.e., $v' = f(y'')$. Denote by P^y_{sub} the subpath $y' \to y_{i+1} \to y''$ of P_y. To compute $L(s, v')$ and $W(s', y'')$, we have $L(s, v') = L(s, u') + L(P^x_{sub})$ and $W(s', y'') = W(s', y') + W(P^y_{sub})$.

It is easy to see that $L(P^x_{sub}) = |u'u| + |uv| + |vv'|$ and $W(P^y_{sub}) = |u'u| + |uu''| + |vv'| - |vu''| = |u'u| + |vv'| - |uv|$. Thus, $W(P^y_{sub}) \leq L(P^x_{sub})$, and $W(s', y'') \leq L(s, v')$. Yet, we show below that $W(s', y'')$ is still bigger than the length of a shortest path from $s' = s$ to v' in $\mathcal{G}_{\mathcal{V}}$.

First of all, we claim that $\overline{u'u}$ and $\overline{vv'}$ must intersect (say, at a point z). Indeed, by property (iii) of our distinguished points, $|u'u| \geq |uv''|$ and $|vv'| \geq |vu''|$ hold. Let $|\overline{uv}|$ be the length of the line segment $\overline{uv}$. Due to the convexity of the pseudodisk, on which u and v lie, $|uv| \geq |\overline{uv}|$, and thus $|u'u| \geq |\overline{uv}|$ and $|vv'| \geq |\overline{uv}|$. Let z be the intersection of the ray emanating from u and going through u' and the ray emanating from v and going through v'. By property (ii) of our distinguished points, we have $\angle vzu \geq \pi/2$, and thus $|\overline{uv}| > |zu|$ and $|\overline{uv}| > |zv|$. Hence, $|uz| < |uu'|$ and $|vz| < |vv'|$, implying that $\overline{u'u}$ and $\overline{vv'}$ must intersect. Since u and v are not the same point (otherwise, P^x_{sub} is forward-going), it must be $|zv| + |zu| > |uv|$ due to the convexity of the pseudodisk on which u and v lie. Thus, $W(P^y_{sub})$ $(= |u'u| + |vv'| - |vu|)$ is bigger than $|u'z| + |zv'|$, which is the length of another subpath P'_{sub} connecting u' and v' in the plane that consists of $\overline{u'z}$ and $\overline{zv'}$. Let $P^x(s, v')$ be the path in the plane from s to v' by joining the subpath of P_x from s to u' with the subpath P'_{sub}. We then have $L(P^x(s, v')) = L(s, u') + L(P'_{sub})$. Due to $L(P'_{sub}) < W(P^y_{sub})$, we have $L(P^x(s, v')) < W(s', y'')$. Therefore, $W(s', y'')$ is larger than the length of a shortest path in the plane from s' $(= s)$ to v' $(= f(y''))$.

Assume to the contrary that P_y is nevertheless a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$. Then, for any vertex on P_y, and in particular the artificial vertex y'', the subpath of P_y from s' to y'' is a minimum weight s'-to-y'' path in $\mathcal{G}^c_{\mathcal{V}}$. Based on our earlier observation in this proof, since y'' is an artificial vertex, for any shortest path P' from s to $v' = f(y'')$ in the plane, it must be $W(s', y'') \leq L(P')$; but this contradicts with the fact that $W(s', y'')$ is larger than the length of a shortest path in the plane from s to v'. Therefore, P_y cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$.

In summary, any s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ that does not correspond to a path in $\mathcal{G}^d_{\mathcal{V}}$ cannot be a minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$. Hence, any minimum weight s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ corresponds to a path in $\mathcal{G}^d_{\mathcal{V}}$. Since any shortest s-to-t path in $\mathcal{G}^d_{\mathcal{V}}$ corresponds to an s'-to-t' path in $\mathcal{G}^c_{\mathcal{V}}$ of the same length, the lemma follows.