Deformable Free Space Tilings for Kinetic Collision Detection*

Pankaj K. Agarwal† Julien Basch‡ Leonidas J. Guibas‡
John Hershberger§ Li Zhang‡¶

Submission to *International Journal of Robotics Research*

Abstract

We present kinetic data structures for detecting collisions between a set of polygons that are not only moving continuously but whose shapes can also vary continuously with time. Unlike classical collision detection methods that rely on bounding volume hierarchies, our method is based on deformable tilings of the free space surrounding the polygons. The basic shape of our tiles is that of a *pseudo-triangle*, a shape sufficiently flexible to allow extensive deformation, yet structured enough to make detection of self-collisions easy. We show different schemes for maintaining pseudo-triangulations as a kinetic data structure, and we analyze their performance. Specifically, we first describe an algorithm for maintaining a pseudo-triangulation of a point set, and show that the pseudo-triangulation changes only quadratically many times if points move along algebraic arcs of constant degree. We then describe an algorithm for maintaining a pseudo-triangulation of a set of convex polygons. Finally, we extend our algorithm to the general case of maintaining a pseudo-triangulation of a set of moving or deforming simple polygons.

*A preliminary version of this paper appeared in the *4th International Workshop on Algorithmic Foundation of Robotics*, 2000.

†Center for Geometric Computing, Department of Computer Science, Box 90129, Duke University, Durham, NC 27708. Supported by Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by NSF grants EIA–9870724, EIA–997287, ITR–333–1050, and CCR–9732787, and by a grant from the US-Israeli Binational Science Foundation.

‡Computer Science Department, Stanford University, Stanford, CA 94305. Supported in part by National Science Foundation grant CCR–9623851 and by US Army MURI grant 5–23542–A.

§Mentor Graphics Corp., 8005 SW Boeckman Road, Wilsonville, OR 97070.

¶Current address: Compaq Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301.

1 Introduction

Collision detection between moving objects is a fundamental problem in computational simulations of the physical world. Because of its universality, it has been studied by several different communities, including robotics, computer graphics, computer-aided design, and computational geometry. Several methods have been developed for the case of rigid bodies moving freely in two and three dimensions. Though a physical simulation involves several other computational tasks, such as motion dynamics integration, graphics rendering, and collision response, collision detection remains one of the bottlenecks in such a system. A commonly used approach to expedite the collision detection between complex shapes is based on hierarchies of simple bounding volumes surrounding each of the objects. For a given placement of two non-intersecting objects, their respective hierarchies are refined only to the coarsest level at which the primitive shapes in the two hierarchies can be shown to be pairwise disjoint.

Motion in the physical world is in general continuous over time, and many systems attempt to speed up collision checking by exploiting this temporal coherence, instead of repeating a full collision check *ab initio* at each time step [20]. Swept volumes in space or space-time have been used towards this goal [8, 15]. Though fixed time-sampling is customary for motion integration, collisions tend to be rather irregularly spaced over time. If we know precisely the motion laws of the objects, then it makes sense to try to predict exactly when collisions will happen, instead of hoping to locate them with time sampling. There have been a few theoretical papers in computational geometry along these lines [9, 12, 21], but their results are not so useful in practice because they use complex data structures and are only applicable for limited types of motion.

Recently Basch *et al.* [5] and Erickson *et al.* [10] presented algorithms for detecting collision between two polygons using the *kinetic data structure* framework, which was originally introduced by Basch *et al.* [6, 11]. Their algorithms avoid many of the problems that arise in the fixed time-sampling method. A kinetic data structure, or KDS for short, is built on the idea of maintaining a discrete attribute of objects in motion by animating a proof of its correctness through time. The proof consists of a set of elementary conditions, called *certificates*, based on the kinds of tests performed by ordinary geometric algorithms (CCW tests in our case). Those certificates that can fail as a result of the motion of the polygons are placed in an event queue, ordered according to their earliest failure time. When a certificate fails, the proof needs to be updated. Unless a collision has occurred, we perform this update and continue the simulation. In contrast to fixed time step methods, for which the fastest moving object determines the time step for the entire system, a kinetic method is based on *events* (the certificate failures) that have a natural significance in terms of the problem being addressed (collision detection in this case).

Unlike the previous hierarchy-based algorithms, the algorithm by Basch *et al.* [5] maintains a decomposition of the common exterior of the two moving polygons. The cells of this decomposition deform continuously as the objects move. As long as all the cells in the

decomposition remain disjoint, the decomposition itself acts as a KDS proof of non-collision between the objects. At certain times, cells become invalid because they self-intersect and the decomposition has to be modified. Unfortunately, extending their approach to collision detection between many polygons is very expensive—we have to construct such a decomposition for every pair of polygons.

In this paper we present an algorithm for detecting collision between members of arbitrary sets of simple polygons as they move and/or deform in the plane. As in [5], we maintain a decomposition of the common exterior of polygons, the free space, into deformable tiles. Ideally, we would like to maintain a decomposition so that the self-intersection of a cell of the decomposition is easy to detect, the decomposition is simple to update when a self-intersection occurs, and the decomposition conforms to the motion of polygons so that self-intersections do not happen too many times. An obvious choice for the decomposition is a triangulation of the free space. Although a triangulation satisfies the first two criteria, it contains too many cells and therefore has to be updated frequently. We will therefore use a *pseudo-triangulation* as the decomposition; a pseudo-triangulation has considerably fewer cells than a triangulation. Pseudo-triangles can flex as the objects move, and therefore the combinatorial structure of the tilings needs fewer updates. At the same time, the cells in a pseudo-triangulation have sufficiently simple shapes that their self-intersections are easy to detect and the triangulation is easy to update. Pseudo-triangulations have been used in the past, but primarily for various visibility problems [19]. An additional benefit of our structure is that it can gracefully adapt to object shapes that are themselves flexible. As our polygons move they can also deform and change shape. This additional flexibility impacts our data structure primarily on the number of events it has to process—but its basic nature and certification remain unchanged.

In Section 2, we describe the kinetic data structures framework and our model for motion, define pseudo-triangulations, and describe the certificates needed to maintain a pseudo-triangulation. In Section 3, we first describe how to maintain a pseudo-triangulation for a set $\mathcal{P}$ of n moving points in the plane, and show that it can be maintained in an output-sensitive manner. For low-degree algebraic motion, the pseudo-triangulation changes about n^2 times. We can further refine the pseudo-triangulation to maintain a triangulation of $\mathcal{P}$ that changes $O(n^{7/3})$ times if each point in $\mathcal{P}$ is moving with fixed velocity. To our knowledge, this is the first triangulation (without Steiner points) that provably changes a sub-cubic number of times.

In Section 4, we describe a scheme for maintaining the pseudo-triangulation of a set $\mathcal{P}$ of k disjoint convex polygons with a total of n vertices. We show that the greedy vertical pseudo-triangulation proposed by Pocchiola and Vegter [19] can be maintained efficiently. A nice feature of this (or any minimum) pseudo-triangulation is that the number of cells is only $O(k)$, which is typically much smaller than n, the total complexity of the polygons. On the other hand, the size of any triangulation has to be at least $\Omega(n)$. We will show that we need a kinetic data structure of size $O(k)$ to maintain this pseudo-triangulation. However, we do not have sharp bounds on the number of events, as we do for the case of points.

Finally, in Section 5, we combine our algorithm for the convex case with the one in [5] to construct a pseudo-triangulation for a set of pairwise-disjoint simple polygons moving in the plane. It can easily be updated when a certificate fails. The number of certificates needed to maintain the correctness of the pseudo-triangulation is proportional to the size of a *minimum-link subdivision* separating the polygons [24], and is thus optimal in some sense. Our separation proof automatically adapts to the complexity of the relative placement of the polygons, and its size will vary between $O(k)$ (when the polygons are far from each other) and $O(n)$ (when they are closely intertwined). A compact separation proof is important when objects are allowed to change their *motion plan* unpredictably.

Traditionally collision detection has been divided into two phases: the *broad phase*, in which one uses a simple-minded algorithm (typically a bounding box check) to determine which pairs of objects might collide and thus need further testing, and the *narrow phase*, in which these candidate pairs get a more detailed collision test using a sophisticated algorithm. Note that our approach based on deformable free space tilings completely obviates this distinction. The tiling effectively "hides" features of objects that are far away and treats the objects as equivalent to their convex hulls. As the objects get closer and more intertwined, their features are progressively revealed and participate in collision checks. Furthermore our framework can be extended in a straightforward way to deformable objects, a setting that no previous collision detection method had successfully addressed. Though this paper describes how to implement the deformable tiling idea only in 2D, we are hopeful that 3D extensions will also be possible.

Related work. Independent of our work, Kirkpatrick *et al.* [17] recently discovered another extension of the structure in [5] to multiple objects. In their structure, the number of certificates used is within a constant ratio of the size of the minimum-link subdivision. However, the structure they proposed is non-canonical, i.e., it depends on the history of the motion, and thus more difficult to analyze. In a more recent paper [18], they further extend their structures by incorporating the approach of [10], thereby obtaining separation sensitive kinetic collision detection for multiple moving polygons. There is also some recent work related to pseudo-triangulations of points. Kettner *et al.* show how to construct bounded degree pseudo-triangulations for points [16]. Streinu uses pseudo-triangulations of points to plan non-colliding motions for polygonal frameworks [23].

2 Kinetic Data Structures, Models of Motion, and Pseudo-Triangulations

In this section we describe the kinetic data structure framework, discuss our models of motion, define pseudo-triangulations, and discuss how we maintain them as objects move continuously.

2.1 Kinetic data structures

Kinetic data structures are based on assertions about the world called *certificates*. In our examples these certificates are always simple geometric relations among a few features of our moving objects—an example might be that a vertex of one object is above the plane supporting a facet of another object. Mathematically, a certificate can be written as the assertion $g(x_1, x_2, \cdots, x_k) > 0$ where k is a small constant, g is a polynomial with constant degree, and x_i's are the description of the position of the geometric objects. For moving objects, x_i's are functions in the time t.

A KDS maintains a set of such assertions tailored to facilitating the computation of the attribute of interest—for example, determining the collision in a set of moving objects. In the plainest form of a KDS, the certificates being maintained prove (in the strict mathematical sense) the correctness of an easy computation for the attribute of interest. As long as the certificates stay valid, this computation can be used to quickly get the value of the relevant system attributes, for example, the time before a collision can occur. Obviously failures of these certificates correspond to the time t when $g(x_1(t), \cdots, x_k(t)) = 0$ and are events of interest to the KDS. At each certificate failure, the certificate set being maintained must be repaired, and in the process the attribute computation possibly updated as well. A good KDS will choose a certificate set that can be repaired locally and efficiently when one of its members fails. This is possible because a KDS exploits continuity or coherence of the motion to obtain "continuity" of the proof. Fundamentally a KDS consists of a *mathematical proof animated through time.*

For this framework to work, the certificate failure times must be either detected or predicted. Prediction is possible if the objects follow known motion laws. The predicted failure times of all kinetic certificates can then be placed in an event queue and the system clock can be advanced until either one of the objects changes its motion plan (at which time all certificates involving it need to have their failure time recomputed), or a certificate fails. The process defines the "eternal KDS loop" shown in Figure 1. It is worth pointing out that motion plan updates will sometimes occur as a result of certificate failures; in the example KDSs we will present, a collision will be detected through a certificate failure and (usually) this will then result in motion plan updates for the colliding objects.

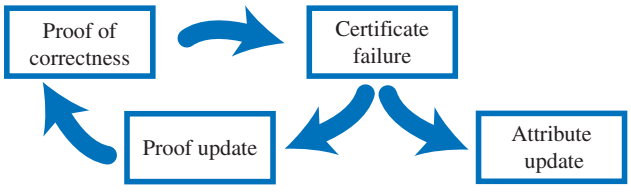

Figure 1. The eternal KDS loop.

The essence of designing a good kinetic structure is in choosing the certificate set to be maintained. Here the designer faces a dilemma. On the one hand, the more we know about

the world, the easier it will be to repair the kinetic proof after a certificate failure. On the other hand, the more assertions about the world we maintain, the more certificate failures we will have to process. In the end, just like in classical data structures where we need to balance the cost of query and update operations, in a KDS design we need to balance the cost of maintaining the information we track against its utility to our task.

Classical complexity measures for algorithms need to be adapted to the on-line KDS setting. We will use the following terminology: events at which the computation of the attribute of interest needs to change are called *external*; the rest of the events are *internal*. The following four measures are used to describe the quality of a KDS [6]:

responsiveness: a measure of how quickly the KDS is able to repair the kinetic proof after a single certificate failure;

efficiency: a measure of the ratio of internal vs. external events that need to be processed by the structure (external events generally reflect a change in the attribute of interest in the outside world; internal events are there solely for the convenience of the KDS);

compactness: a measure of the number of certificates needed by the KDS; and

locality: a measure of the largest number of certificates that an object can participate in; this number reflects the cost of updating the kinetic event queue when an object changes its motion plan.

KDSs that are good in terms of most of these measures have been developed for a variety of problems, including various extent [3], proximity [7], connectivity [1], visibility [4, 2], and collision detection [5, 10] problems.

It is instructive to compare a KDS-based simulation to a a more traditional fixed time-step simulation. The latter approach is the standard method for implementing physical simulations when the evolution law of the system is fixed. Typically the system evolution is obtained by numerically integrating an ordinary or partial differential equation. However, when discrete events can happen that change the evolution law, and especially when the distribution of these events is hard to predict far in advance (as in the case of collisions), the choice of the step size becomes problematic. If it is chosen too small, the simulation will slow down with no perceptible difference in its quality; if it is too large, important events may be missed and the wrong evolution may result. In a certain sense the proof being maintained by a KDS provides intelligent advice to the simulation as to when it is important to examine the system: the certificate failure times are the only times when something of interest to the attribute being maintained can occur. Thus a KDS samples the system just at moments of potential significance, as determined by the certificate structure chosen.

2.2 Models of motion

Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a set of k simple polygons with a total of n vertices. We allow single-point degenerate polygons, but we do not allow polygonal chains, which are instead treated as thin polygons by doubling the chain. We assume that all polygons are bounded. Let $\overline{\mathcal{P}}$ and $\mathcal{P}^\circ$ denote the convex hull and interior of $\mathcal{P}$, respectively. We define the *free space* $\mathcal{F}(\mathcal{P})$ of $\mathcal{P}$ to be $\overline{\mathcal{P}} \setminus \mathcal{P}^\circ$. The boundary of $\mathcal{F}(\mathcal{P})$, denoted by $\partial\mathcal{F}(\mathcal{P})$, consists of polygon boundaries and convex hull edges.

We call a function $x(t)$ algebraic in t if there is a bi-variate polynomial f so that $f(x(t), t) = 0$. The degree of $x(t)$ is defined to be the minimum degree of all such f's so that $f(x, t) = 0$. In this paper, we will consider the motions that can be described by algebraic functions with bounded degree. If the polygons in $\mathcal{P}$ are rigid, then the motion of each P_i can be defined by a moving orthogonal reference frame: a point $o_i(t)$ and two orthogonal unit vectors $x_i(t), y_i(t)$, whose coordinates are low-degree algebraic functions of t. Each vertex v of P_i is assumed to have constant coordinates (v_x, v_y). The position of v at time t, denoted by $v(t)$, is $o_i(t) + v_x x_i(t) + v_y y_i(t)$. If P_i is a point, then o_i is the same as P_i and the vectors x_i, y_i are not needed. We will use $P_i(t)$ to denote the polygon P_i at time t, and $\mathcal{P}(t)$ to denote the set $\mathcal{P}$ at time t. The *degree of motion* of $\mathcal{P}$ is the maximum degree of a polynomial defining the coordinates of o_i, x_i, y_i, for $1 \leq i \leq k$. We call a motion *translational* if the coordinates of vectors x_i, y_i, for all $1 \leq i \leq k$, are constant. For a translational motion, if the o_i's $(1 \leq i \leq k)$ are linear polynomials, we say that the motion is *linear*.

While algebraic motions do not include all the motions, they do represent a broad family of motions and are flexible enough to approximate any motion to any order and accuracy, for a limited time. Note however that an algebraic rotation is necessarily of non-uniform angular velocity and can cover only constant number of full turns. As noted in the kinetic model, each certificate is in the form of $g(x_1(t), x_2(t), \cdots, x_k(t)) > 0$ where k is a small constant ($k \leq 3$ in all the certificates involved in this paper) and x_i's are algebraic function with constant degree. By Bezout's Theorem, the equation $g(x_1(t), x_2(t), \cdots, x_k(t)) = 0$ has constant number of solutions in non-degenerate case. Or in other words, any certificate can only fail constant number of times for constant degree algebraic motions. Since both g and x_i's are of constant degree, we also assume that computing the roots of $g(x_1, \cdots, x_k)$ can be done with $O(1)$ cost.

In summary, the two key properties that we need for our methods to work are that

- for each certificate or condition in the kinetic proof, we must be able to predict when it will fail next; we assume that our knowledge of the polygon motions allows us to compute the time of such an event at $O(1)$ cost.
- for our analysis we need the property that the polygon motions are such that any particular certificate can fail at most a constant number of times; motions satisfying this constraint are called *pseudo-algebraic*.

2.3 Pseudo-triangulation

A *pseudo-triangle* in the plane is a simple polygon whose boundary consists of three concave chains, called *side chains*, that join at their endpoints. The three endpoints of a pseudo-triangle are called *corners*. The edges incident upon the corners are called *corner edges* (Figure 2 (i)). For a vertex p on the boundary of a pseudo-triangle, we denote p_L, p_R be the predecessor and the successor of p along $\partial\Delta$ in the counterclockwise direction.

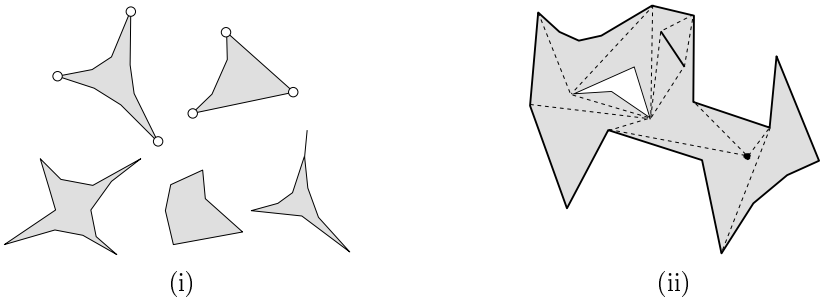

Figure 2. (i) The examples of some pseudo-triangles and non-pseudo-triangles. The first two figures are pseudo-triangles, but the others are not. (ii) An example of pseudo-triangulation of a polygon with holes.

Let S be a polygon with holes; some of the holes may be degenerate, i.e., they may be points or segments. A *pseudo-triangulation* $\mathcal{T}(S)$ of S is a planar subdivision of the closure of S, so that each face of $\mathcal{T}(S)$ is a pseudo-triangle and so that the interior of each face lies in the interior of S; see Figure 2 (ii) for an example. In other words, $\mathcal{T}(S)$ is a collection of pseudo-triangles with pairwise-disjoint interiors, each lying inside S, that cover S. The vertices of $\mathcal{T}(S)$ are the same as the vertices of S, and each edge of $\mathcal{T}(S)$ is either an edge of ∂S or a segment whose interior lies inside S; edges of the latter type are called *diagonals*.

We are interested in maintaining the pseudo-triangulation of $\mathcal{F}(\mathcal{P})$, which, for brevity, we denote by $\mathcal{T}(\mathcal{P})$.

Since a polygon (even with degenerate holes) can always be triangulated and a triangulated planar subdivision is obviously a pseudo-triangulation, a pseudo-triangulation of S always exists. A pseudo-triangulation $\mathcal{T}(\mathcal{P})$ is called *minimum* if $\mathcal{T}$ has the minimum number of pseudo-triangles, among all the pseudo-triangulations of $\mathcal{P}$. For a given set $\mathcal{P}$ of simple polygons, although minimum pseudo-triangulations are not unique, the size of any minimum pseudo-triangulation of $\mathcal{F}(\mathcal{P})$ can easily be determined by the following lemma.

Lemma 2.1. *Let $\mathcal{P}$ be a set of k simple pairwise-disjoint polygons as defined above. Suppose m of these polygons are not points, and suppose there are a total of r reflex vertices in $\mathcal{P}$. Then there are at least $k + m + r - 2$ pseudo-triangles in any minimum pseudo-triangulation of $\mathcal{F}(\mathcal{P})$. A pseudo-triangulation has the minimum number of pseudo-triangles if and only if any non-reflex vertex or point that is not on the convex hull of $\mathcal{P}$ is a non-corner vertex of some pseudo-triangle in the triangulation.*

Proof: Denote by s the total number of vertices on the polygons (which are not points), and by t the number of vertices on the convex hull of $\mathcal{P}$. There are a total of $s+k-m$ vertices in $\mathcal{P}$ (there are $k-m$ points) among which $s+k-m-t-r$ are non-reflex vertices that do not appear on the boundary of convex hull. Suppose also that a pseudo-triangulation $\mathcal{T}(\mathcal{P})$ contains n pseudo-triangles, and that the number of *non-corner* vertices on those pseudo-triangles are $a_1, a_2, \ldots, a_n$.

Now, we consider the sum of the interior angles, denoted by A, of the pseudo-triangles in $\mathcal{T}(\mathcal{P})$. Since the total interior angles of a pseudo-triangle with x non-corner vertices is $(x+1)\pi$, $A = \sum_{i=1}^{n}(a_i+1)\pi = (\sum_{i=1}^{n} a_i + n)\pi$. Any point can be a non-corner vertex of at most one pseudo-triangle. Moreover, a non-corner vertex of a pseudo-triangle cannot be a reflex vertex of a polygon or a vertex of the convex hull of $\mathcal{P}$. Therefore

$$\sum_{i=1}^{n} a_i \leq s+k-m-t-r \quad \text{and} \quad A \leq (s+k-m-t-r+n)\pi.$$

On the other hand, let B be the total interior angles of all the polygons. Then $A = (s+k-m-t)2\pi + (t-2)\pi - B$, where the first term counts the angles around each interior point and the second term counts those on the convex hull boundary. Clearly $B = (s-2m)\pi$. Thus, we have that $A = (s+2k-t-2)\pi$. Combining with the previous inequality, we have that $n \geq k+m+r-2$.

This proof also shows that the equality holds if and only if every non-reflex, non-convex hull node is a non-corner vertex of some pseudo-triangle. □

The above lemma gives a lower bound on the number of pseudo-triangles in a pseudo-triangulation. This bound is actually tight, as we show later. Indeed, the major goal in the rest of the paper is to show how to construct and maintain minimum pseudo-triangulations when objects are moving. Notice that the number of pseudo-triangles in the bound stated above depends only on the number of objects and the number of reflex vertices, which is usually much smaller than the total number of vertices.

2.4 Maintaining a pseudo-triangulation

We are interested in maintaining $\mathcal{T}(\mathcal{P})$ as $\mathcal{F}(\mathcal{P})$ deforms continuously. Specifically, we want to maintain $\mathcal{T}(\mathcal{P})$ as a kinetic data structure. As mentioned in the introduction, this is accomplished by maintaining a proof of the correctness of $\mathcal{T}(\mathcal{P})$, which consists of a small set of elementary conditions called *certificates*. As long as the certificates remain valid, the current combinatorial structure of $\mathcal{T}(\mathcal{P})$ is valid. $\mathcal{T}(\mathcal{P})$ is updated only when some certificate fails. A certificate failure is called an *event*. All the events are placed in an event queue according to their failure time. The structure is then maintained by processing those events one by one. The efficiency of such a data structure depends on the size of the proof, the number of events, and the number of certificates that need to be updated at each event.

In our set-up, we certify that each face in $\mathcal{T}(\mathcal{P})$ is a pseudo-triangle and that the faces cover $\mathcal{F}(\mathcal{P})$. In view of the definition of $\mathcal{T}(\mathcal{P})$, we need to certify the following two conditions for each pseudo-triangle Δ in $\mathcal{T}(\mathcal{P})$ (Figure 3):

1. Each side chain C is concave, i.e., for each interior vertex p of C, the angle $\angle p_L p p_R > \pi$ (or the signed area of $\triangle p_L p p_R$ is negative). We call these certificates *reflex* certificates.
2. The side chains of a pseudo-triangle Δ join only at their endpoints, i.e., for each corner vertex q of Δ, the angle $\angle q_L q q_R < \pi$ (or the signed area of $\triangle q_L q q_R$ is positive). We call such certificates *corner* certificates.

When the reflex certificate of p fails, we connect the vertices adjacent to p, say p_1, p_2, to maintain the pseudo-triangularity of Δ. To maintain the pseudo-triangulation, we may also have to update other pseudo-triangles, depending on the edges adjacent to p. The only other pseudo-triangles affected are those that share the edge pp_1 or pp_2 with Δ. Let us consider Δ_1, which contains the edge pp_1 (Figure 3 (ii)). Since p cannot be a non-corner vertex on two pseudo-triangles, p must be a corner of Δ_1. Thus, it is legitimate to merge triangle pp_1p_2 into Δ_1, or equivalently to delete the edge pp_1, to form a new pseudo-triangle. In the new pseudo-triangle, p becomes a non-corner vertex. However, we may need to do it carefully because either pp_1 might be a polygon edge, which we cannot delete, or both pp_1 and pp_2 are deletable, where we have to make a choice. If both pp_1 and pp_2 are polygon edges, then we add a triangle pp_1p_2 to $\mathcal{T}(\mathcal{P})$. If only one edge is a diagonal edge, then we simply delete it and merge the triangle pp_1p_2 into the corresponding pseudo-triangle. If both pp_1 and pp_2 are diagonal edges, we then delete one of pp_1 or pp_2. Which of the two edges is deleted depends on the specific pseudo-triangulation we are maintaining; we will explain the choice for each pseudo-triangulation later in the paper.

When the corner certificate of q fails, we collapse the corner. Suppose that the vertices adjacent to q are q_1 and q_2, and when the concave certificate corresponding to q fails, q_2 lies on the edge qq_1. We then delete the edge qq_1 and add the edge q_1q_2 to maintain the pseudo-triangularity of Δ. This process also effectively adds the point q_2 to the side chain of the other pseudo-triangle incident to the edge qq_1. Since there is only one way to update the pseudo-triangulation when a corner certificate fails, we will not describe updates for this case in the following sections.

In addition, we also have to maintain the boundary $\partial\mathcal{F}(\mathcal{P})$. $\partial\mathcal{F}(\mathcal{P}) \setminus \partial\overline{\mathcal{P}}$ is automatically maintained by the algorithm. If we regard $\partial\overline{\mathcal{P}}$ as a concave chain with respect to the exterior of $\overline{\mathcal{P}}$, we can also maintain it using reflex certificates for each vertex on the hull. We omit the details.

Since the pseudo-triangulation is not a canonical structure (there may be many pseudo-triangulations of $\mathcal{F}(\mathcal{P})$), it is not clear which pseudo-triangulation we are maintaining. To avoid this ambiguity, we first describe a static algorithm that constructs a "specific" pseudo-triangulation of $\mathcal{F}(\mathcal{P})$. We then assume that, at any time t, the kinetic data structure maintains the pseudo-triangulation that the static algorithm would have constructed on

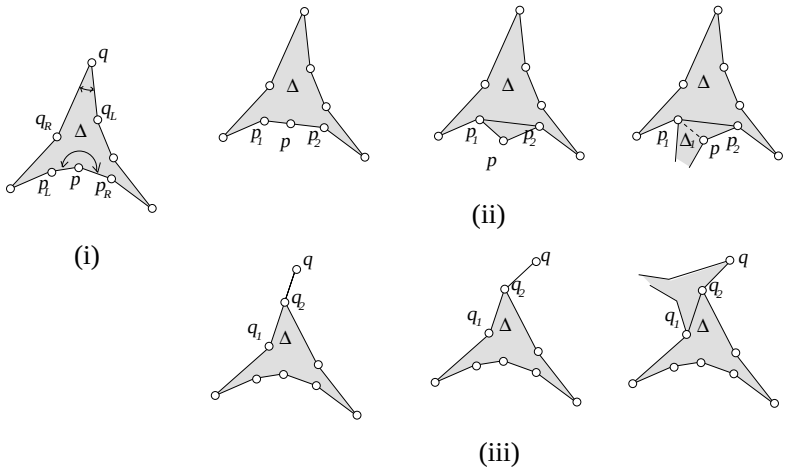

Figure 3. The certificates to prove pseudo-triangularity.

$\mathcal{P}(t)$. To maintain this invariant, we need additional certificates. When these certificates fail, we usually update the structure by the *flipping* operation: For any diagonal edge e, consider the two pseudo-triangles Δ_1, Δ_2 that contain e. The union of Δ_1, Δ_2 forms a pseudo-quadrangle $\diamond$—a cell whose boundary consists of four concave chains. Besides e, there is exactly one other diagonal edge inside $\diamond$, which is called the *shadow edge* of e. The quadrangle $\diamond$ can be decomposed into two pseudo-triangles in two ways, one by adding e and the other by adding e's shadow edge. The flipping operation replaces e by e'.

3 Pseudo-Triangulation for Points

In this section, we assume $\mathcal{P}$ to be a set of n points. We define the *incremental pseudo-triangulation* for $\mathcal{P}$, and show how to maintain it efficiently. We also show it can be refined to maintain a triangulation of $\mathcal{P}$.

3.1 Incremental pseudo-triangulation

The pseudo-triangulation for points is built in an incremental manner as follows. We first sort the points in increasing order of their x-coordinates. Suppose that $p_1, p_2, \ldots, p_n$ is the sorted sequence, and $\mathcal{P}_k = \{p_1, p_2, \ldots, p_k\}$ denotes the leftmost k points of the sequence. We place a vertical line at p_1 and sweep it from left to right to build the pseudo-triangulation incrementally. When the sweep line passes the point p_k, we draw two tangent segments from p_k to the convex hull of $\mathcal{P}_{k-1}$. Denote by $u(p_k)$ and $d(p_k)$ the left endpoints of the upper and lower tangent segments to $\mathcal{P}_{k-1}$ from p_k, respectively (Figure 4). The concave chain on $\overline{\mathcal{P}}_{k-1}$ between $u(p_k)$ and $d(p_k)$ and the line segments $p_k u(p_k)$, $p_k d(p_k)$ form a pseudo-triangle $\Delta(p_k)$—its boundary consists of a concave chain and two single edges. Let $C(p_k)$

denote the concave chain on $\Delta(p_k)$. The three corners of $\Delta(p_k)$ are the points p_k, $d(p_k)$, and $u(p_k)$. After processing every point in $\mathcal{P}$, we obtain a pseudo-triangulation, which is called the *incremental pseudo-triangulation (IPT)* of $\mathcal{P}$. Clearly, except for the first two vertices, we add exactly one pseudo-triangle when adding a point. Therefore, IPT has $n-2$ pseudo-triangles. According to Lemma 2.1, IPT is a minimum pseudo-triangulation.

Figure 4. The incremental pseudo-triangulation of a point set. In (i), the sweep line has just passed p; (ii) shows the final pseudo-triangulation.

3.2 Maintaining the incremental pseudo-triangulation

We now describe how we maintain $IPT(\mathcal{P})$, the incremental pseudo-triangulation of $\mathcal{P}$, which is a "canonical" structure in the sense that it is uniquely defined as long as no two points in $\mathcal{P}$ have the same x-coordinate. As the points move, $IPT(\mathcal{P})$ changes continuously, but its combinatorial structure changes only at discrete times. Since we wish to maintain $IPT(\mathcal{P})$, we need to update the triangulation at certain times even though none of reflex or corner certificates have been violated. For example, the change in x-ordering of two points causes a change in the structure because $IPT(\mathcal{P})$ depends on the x-ordering of the points. The following lemma, whose proof is straightforward, suggests that besides the reflex- and corner-certificate failures, changes in the x-ordering are the only additional events.

Lemma 3.1. *As long as the x-ordering of the points in $\mathcal{P}$ does not change, the reflex and corner certificates certify the incremental pseudo-triangulations.*

We first describe how to handle corner and reflex events. If a corner event fails, then the structure can be updated as described in Section 2.4. If a reflex certificate fails, say p_1, p, p_2 become collinear, then we also proceed as described in Section 2.4: we add the edge p_1p_2. If both pp_1 and pp_2 are diagonal edges, then we have to delete one of them. Suppose $x(p_1) > x(p_2)$. Since the x-ordering has not changed at this event, we have $x(p_1) > x(p) > x(p_2)$. Before the event, p is either $d(p_2)$ or $u(p_2)$, say, $d(p_2)$. Then after the event, p_1 becomes $d(p_2)$, so we delete the edge pp_2; see Figure 3.2.

Next, we consider the case in which the x-ordering of two points changes. Although such an event does not affect the validity of the pseudo-triangulation, it does cause a change to $IPT(\mathcal{P})$ since the incremental ordering of the points changes. Suppose that p passes q from

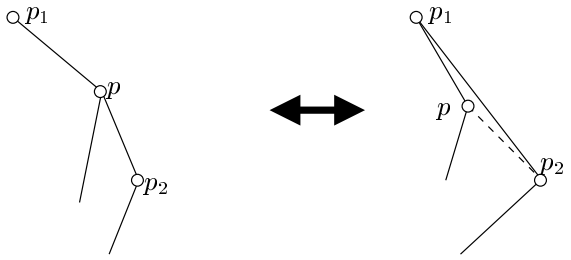

Figure 5. A reflex event at a point p: edge p_1p_2 is added and edge pp_2 is deleted.

the left at time t and p is above q (Figure 6). Let r be $d(p)$ just before time t, and let r' be $u(q)$ immediately after time t. Then the structure is updated by replacing the edge rp with the edge $r'q$; a flipping operation as defined in Section 2.4. The point r' can be found by computing a tangent segment from q to the concave chain between $u(p)$ and $d(q)$, which can be done in $O(\log n)$ time.

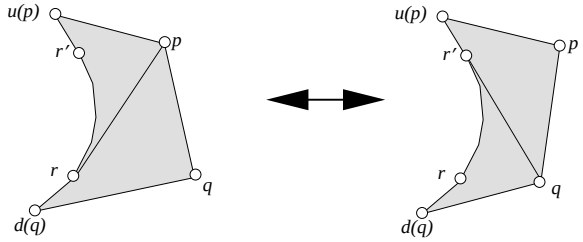

Figure 6. Two points exchange in x-order.

Note that $IPT(\mathcal{P})$ changes at each event, so the method is output-sensitive. We have shown the following.

Theorem 3.2. *The incremental pseudo-triangulation can be maintained in an output-sensitive manner. Each update of the structure takes $O(\log n)$ time.*

In the following, we shall bound the number of changes to $IPT(\mathcal{P})$ for points in constant degree algebraic motion.

3.3 Combinatorial changes to $IPT(\mathcal{P})$

We now obtain an upper bound on the number of combinatorial changes to $IPT(\mathcal{P})$, i.e., the number of events, under the assumption that the degree of motion of $\mathcal{P}$ is fixed. For two distinct points $p, q \in \mathcal{P}$, we define a function $\phi_p^q(t)$ as follows. If q is to the left of p at time t, then $\phi_p^q(t)$ is the slope of the line that passes through p and q; otherwise, $\phi_p^q(t)$

is undefined. Since the motion has constant degree, the x-order of a pair of points can switch only a constant number of times. Thus, each $\phi_p^q(t)$ consists of a constant number of arcs, each of which is a portion of a fixed-degree rational function. Consider the family of functions $\Phi_p = \{\phi_p^q \mid p \neq q \in \mathcal{P}\}$. By our assumptions any two arcs in Φ_p intersect a constant number of times, say, $s-2$. For a point q to be $d(p)$ at time t, $\phi_p^q(t)$ must have the largest value among all the points in $\mathcal{P}$. This is to say, the number of changes to $d(p)$ is the same as the combinatorial complexity of the upper envelope of Φ_p, which is bounded by $\lambda_s(n)$, where $\lambda_s(n)$ is the maximum length of an (n, s) Davenport-Schinzel sequence and is roughly linear [22]. Similarly, we can bound the number of changes to $u(p)$ by $\lambda_s(n)$. Summing over all the vertices in $\mathcal{P}$, we obtain the following.

Theorem 3.3. *If the points of $\mathcal{P}$ move algebraically with constant degree, $IPT(\mathcal{P})$ changes $O(n\lambda_s(n))$ times, where s is a constant that depends on the degree of the motion.*

3.4 Fan triangulation of $\mathcal{P}$

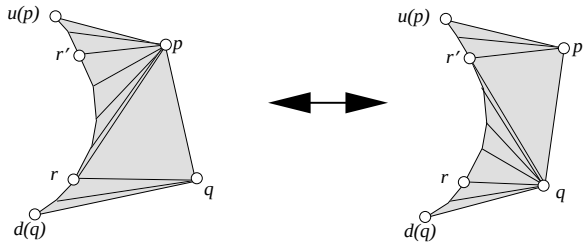

Figure 7. Fan triangulation of $IPT(\mathcal{P})$

It is very easy to obtain a triangulation of $\mathcal{P}$ from $IPT(\mathcal{P})$ by connecting each point p to every interior point on $C(p)$, thereby creating a fan inside each $\Delta(p)$. We call such a triangulation a *fan triangulation* (Figure 7). Although $IPT(\mathcal{P})$ changes only nearly quadratically many times, we are not able to prove a similar bound on the number of changes in the fan triangulation. Although a reflex or corner event still causes only $O(1)$ changes to the fan triangulation, the x-ordering event, which we process by switching a fan of p to a fan of q or vice versa, is expensive (Figure 6 (ii)). Suppose $y(p) \geq y(q)$, and let r be $d(p)$ just before the event and r' be $u(q)$ immediately after the event. The cost of this switching the fan from p to q is proportional to the length of the chain between r and r'. In the worst case, it might be $\Theta(n)$. This would give us a naïve bound of $O(n^3)$ on the number of changes to the fan triangulation.

Using a global argument, we prove that the number of changes to the fan triangulation is roughly $n^{7/3}$ for linear motion. To our knowledge, this is the first triangulation that changes sub-cubically many times for linear motions. In order to prove the improved bound, we need a few lemmas. For each point p, denote by $H_d(p)$ (resp. $H_u(p)$) the half-space below (resp. above) the line defined by p and $d(p)$ (resp. $u(p)$). Now consider a point v on the concave

chain between r and r' at an x-ordering event. Since the chain is concave, the x-coordinate of one of the vertices on the chain adjacent to v is smaller than that of v, and therefore this point must be $u(v)$ or $d(v)$, depending on whether its y-coordinate is larger than that of v. In the following, we assume that $d(v)$ is adjacent to v on the chain. The other case can be handled in the same way.

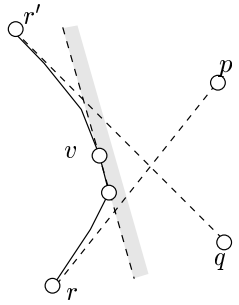

Figure 8. Illustration of ξ_v^p.

Since v is between r and r', the points p and q must be below the line that passes through v and $d(v)$, i.e., p and q are in $H_d(v)$; see Figure 8. Further, among all the points in $H_d(v)$, there is no point to the left of p and q. Indeed, if there were such a point z, then v would lie in the cone formed by the rays $zu(z)$ and $zd(z)$ and therefore would not appear on the concave chain between $u(p)$ and $d(q)$. We define the function $\xi_v^p(t)$ to be the x-coordinate of p if p is in $H_d(v)$ at time t and undefined otherwise. Set $\Xi_v = \{\xi_v^p \mid p \in \mathcal{P} \setminus \{v\}\}$. The point v is on the concave chain between r and r' when p and q exchange their x-ordering if and only if $\xi_v^p(t)$ and $\xi_v^q(t)$ are both defined and they are the smallest among all the ξ_v's at time t, i.e., both of them lie on the lower envelope of Ξ_v. Let $\delta(v)$ denote the complexity of the lower envelope of Ξ_v. The total number of changes to the fan triangulation in this case (in which $d(v)$ is adjacent to v) summed over all points of $\mathcal{P}$ is at most $\sum_v \delta(v)$.

A natural approach to bound the above sum is to obtain a bound on $\delta(v)$. Unfortunately, the graph of function $\xi_v^p(t)$ might consist of $\Omega(n)$ connected arcs. In fact, there are configurations of moving points in which Ξ_v consists of $\Theta(n^2)$ connected arcs. In Figure 9, we show a configuration of n moving points in which this happens; for simplicity assume that n is even. In the figure, the upper $n/2-1$ points are static, and the lower $n/2$ points move horizontally from left to right. A collinearity among v, an upper point, and a lower point corresponds to beginning of an arc in Ξ_v, and it may happen $\Theta(n^2)$ times. We therefore need a more refined analysis to bound $\sum_v \delta(v)$.

When a point q enters or leaves $H_d(v)$, it may or may not appear on the lower envelope of Ξ_v. For a point $p_i \in \mathcal{P}$, let m_i denote the number of times another point appears on the lower envelope at the time when it enters or leaves $H_d(p_i)$. We first bound the number of such events.

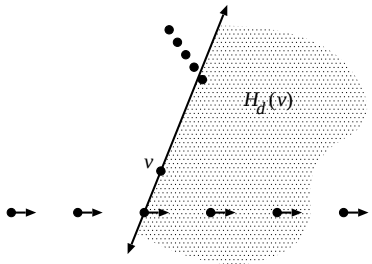

Figure 9. An example where Ξ_v may have $\Theta(n^2)$ arcs.

Lemma 3.4. $\sum_{i=1}^{n} m_i = O(n\lambda_s(n))$.

Proof: Whenever a point leaves of enters $H_d(p_i)$, $IPT(P)$ has to change. Thus the total number of these events is bounded by $O(n\lambda_s(n))$ by Theorem 3.3. □

Consider a point that travels on an arrangement Γ of x-monotone algebraic arcs. The point moves x-monotonically along the arcs in Γ. When it comes to an intersection between two arcs, it either continues to travel on the same arc or makes a right turn, if possible. Call the trajectory of such a point a *pseudo-concave chain*. Clearly, in a line arrangement, this definition gives us exactly a concave chain. Now, we claim the following.

Lemma 3.5. *The lower envelope of the arrangement Ξ_{p_i} consists of m_i disjoint (pseudo-) concave chains.*

Proof: Imagine that a point travels on the lower envelope of Ξ_{p_i}. It has to stop if it encounters an endpoint of an arc. Such an endpoint corresponds to a discontinuity of the function ξ_{p_i}, i.e., corresponds to a point coming into or going out of $H_d(p_i)$. Further, the endpoint is on the lower envelope. By the definition of m_i, this happens m_i times. □

Combining the Lemma 3.4 and 3.5, we are now in position to prove Theorem 3.6.

Theorem 3.6. *For points in linear motion, the number of changes to the fan triangulation is bounded by $O(n^{4/3}\lambda_s(n))$, where s is a constant.*

Proof: If the points move linearly with a constant velocity, then the x-coordinate of each point is a linear function of time. Thus the lower envelope of Ξ_{p_i} consists of m_i disjoint concave chains in an arrangement of n lines. As proved in [13, 14], the complexity of these chains is bounded by:

$$\delta(p_i) = O(\max(nm_i^{1/3}, n^{2/3}m_i^{2/3})).$$

Therefore, the total complexity is bounded by:

$$\sum_i \delta(p_i) = \sum_i O(\max(nm_i^{1/3}, n^{2/3}m_i^{2/3})).$$

By Lemma 3.4, $\sum_i m_i = O(n\lambda_s(n))$. Thus, $\sum_i \delta(p_i)$ is maximized when $m_i = \Theta(\lambda_s(n))$, for all i's. This gives us the bound of $O(n^{4/3}\lambda_s(n))$. □

4 Pseudo-Triangulation for Convex Polygons

Next we consider the case in which $\mathcal{P}$ is a set of k convex polygons with a total of n vertices. We describe a different pseudo-triangulation for $\mathcal{P}$, called the *greedy vertical pseudo-triangulation*, which is based on a result by Pocchiola and Vegter [19]. They introduced *greedy pseudo-triangulation* as a tool to compute the visibility complex of a set of convex polygons. We will modify their algorithm for our application.

4.1 Greedy pseudo-triangulation for convex polygons

There are exactly four bi-tangent segments between any two disjoint convex polygons. A bi-tangent is called *free* if it does not intersect the interior of any object in $\mathcal{P}$. Let s and s' be two free bi-tangents with a common endpoint v, which is a vertex of a polygon $P \in \mathcal{P}$. We say that s and s' cross at v if P lies in the wedge of angle greater than π formed by s and s'; see Figure 10.[1] We say that two free bi-tangents intersect either if they share an interior point or if they have a common endpoint and they cross at that endpoint.

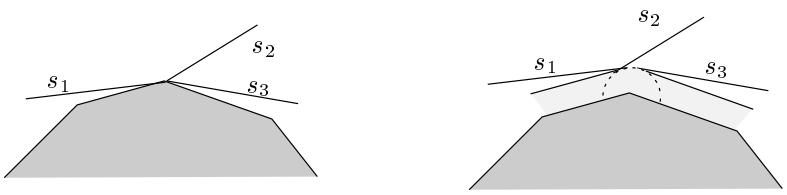

Figure 10. (i)Intersecting and non-intersecting tangents: s_1, s_2 are intersecting, but s_1, s_3 and s_2, s_3 are non-intersecting tangents. (ii) Minkowski sum of P with a ball in the neighborhood of a vertex; the displaced tangents s_1 and s_2 intersect.

For any linear ordering $\prec$ on the free tangents, we can build a corresponding greedy triangulation $\mathcal{T}_\prec(\mathcal{P})$ as follows. We first sort the tangents by $\prec$ ordering. We then scan

[1] Intuitively, if we smooth the polygon P by taking the Minkowski sum of P with a disk of a sufficiently small radius, say δ, and we translate s and s' by at most δ so that they become tangents to the resulting polygons at their endpoints, then s and s' cross at v if and only if the translated copies of s and s' intersect; see Figure 10.

through this list and maintain a set S of tangents that we have added to the pseudo-triangulation so far. At each step, we pick the next tangent segment s in the sorted sequence and check whether it intersects any tangents in S. If s does not intersect any segment of S, we add it to S; otherwise, we discard s. After processing all the segments, we obtain a set of non-intersecting free tangents. It is shown by Pocchiola and Vegter [19] that $|S| = 3k - 3$, and that together with the object boundaries, S forms a pseudo-triangulation of $\mathcal{F}(\mathcal{P})$ consisting of $2k - 2$ pseudo-triangles. By Lemma 2.1, this procedure constructs a minimum pseudo-triangulation.

For any line segment s, define $\theta(s)$ to be the minimum positive angle by which we have to rotate a vertical segment in the clockwise direction so that it becomes parallel to s. We order all the free tangents in increasing order of $\theta(\cdot)$. The greedy pseudo-triangulation created using this ordering is called the *greedy vertical pseudo-triangulation (GVPT)* and is denoted by $\mathcal{G}(\mathcal{P})$ (Figure 11 (i)). It is shown in [19] that $\mathcal{G}(\mathcal{P})$ can be constructed in $O(k \log n)$ time, provided that each polygon is represented as an array storing its vertices in a clockwise (or counterclockwise) order.

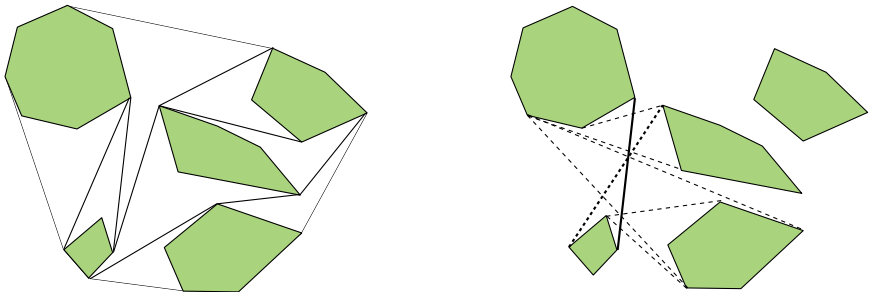

Figure 11. (i) Greedy vertical pseudo-triangulation. (ii) Left-to-right property. The solid segment is an edge in $GVPT$. The dotted ones are the free tangents it crosses. The shadow edge is thickened.

The following local rule determines whether a free bi-tangent is in $\mathcal{G}(\mathcal{P})$.

Lemma 4.1 (Left-to-right property [19]). *A bi-tangent segment s is in $\mathcal{G}(\mathcal{P})$ if and only if $\theta(s) < \theta(s')$ for each free bi-tangent s' intersected by s (Figure 11 (ii)).*

Recall that if we replace an edge in a pseudo-triangulation with its shadow edge, we still obtain a pseudo-triangulation. It turns out that if an edge s has the minimum $\theta(s)$ over all free bi-tangents, then its shadow edge in $\mathcal{G}(\mathcal{P})$ has the minimum $\theta(\cdot)$ value among all free bi-tangents crossing s. This implies the following property, which was proved in [19].

Lemma 4.2 (Local property [19]). *Let $\prec$ be the ordering of tangents determined by $\theta(\cdot)$. Suppose that s is the minimum element of $\prec$, and is therefore an edge in $\mathcal{G}(\mathcal{P}) = \mathcal{T}_{\prec}(\mathcal{P})$. Denote by $\prec'$ the same ordering as $\prec$ except assigning s as the maximum element of $\prec'$. Then $\mathcal{T}_{\prec'}(\mathcal{P})$ can be obtained from $\mathcal{T}_{\prec}(\mathcal{P})$ by replacing s with its shadow edge.*

By this lemma, if an edge or its shadow changes from first to last or vice versa in the $\theta(\cdot)$ ordering, we just perform a local flip operation to fix the greedy pseudo-triangulation.

4.2 Maintaining the greedy pseudo-triangulation

We now describe how to maintain $\mathcal{G}(\mathcal{P})$ as a KDS so that it can be updated efficiently as the polygons in $\mathcal{P}$ move or deform. As described in Section 2.4, we maintain corner and reflex certificates for each pseudo-triangle in $\mathcal{G}(\mathcal{P})$. In addition, for each diagonal edge s of $\mathcal{G}(\mathcal{P})$, we maintain a *diagonal certificate* to certify that $\theta(s) < \theta(s')$, where s' is the shadow edge of s in $\mathcal{G}(\mathcal{P})$. It suffices to detect the time instance at which either $\theta(s)$ or $\theta(s')$ makes a discontinuous transition from zero to π or vice versa because the relative order of $\theta(s)$ and $\theta(s')$ changes only when one of them has a discontinuity. This adds the requirement to maintain the shadow edges even though such edges do not appear in $\mathcal{G}(\mathcal{P})$. Again, the shadow edges can be maintained in the same manner by corner and reflex certificates. We will show that these certificates are sufficient to maintain $\mathcal{G}(\mathcal{P})$.

As in Section 3, we can prove the following analogue to Lemma 3.1.

Lemma 4.3. *If no diagonal certificate fails, then $\mathcal{G}(\mathcal{P})$ changes only when one of the corner or reflex certificates fails.*

Proof: The greedy pseudo-triangulation can change only when a bi-tangent stops or starts being free, the slope ordering of two intersecting tangents changes, or two free tangents start or stop intersecting. Unless some diagonal certificate fails, each of these cases reduces to a collinearity of three objects. (Three objects are *collinear* if there is a line tangent to all of them.) Suppose that the collinear objects are P_1, P_2, and P_3, and that the line ℓ is tangent to them in that order. Consider the line segments s_1,s_2,s_3 on ℓ that connect P_1P_2, P_2P_3, and P_3P_1, respectively. By the left-to-right property, if such a collinearity to change $\mathcal{G}(\mathcal{P})$, then exactly two of these edges are in $\mathcal{G}(\mathcal{P})$ just before the event happens. Furthermore, one of the corner or reflex certificates must fail when such an event happens. □

When a corner or reflex certificate fails, we can update $\mathcal{G}(\mathcal{P})$ as described in Section 2.4, except for one subtle issue in the case of reflex certificates. If an interior vertex p of a side chain $\ldots p_L p p_R \ldots$ ceases to be a reflex vertex, we add the edge p_Lp_R and we delete one of the edges p_Lp and pp_R—we delete p_Lp if and only if $\theta(p_Lp) > \theta(pp_R)$; see Figure 12.

Next, we consider diagonal certificates. If the certificate of a diagonal edge s fails, then either s or its shadow edge is vertical. When such an event happens, the slope ordering of an edge jumps from the minimum to maximum, or vice versa. By the local property of greedy pseudo-triangulations, we can simply perform a flipping operation to s to maintain $\mathcal{G}(\mathcal{P})$.

Since $\mathcal{G}(\mathcal{P})$ has $O(k)$ diagonals and pseudo-triangles, the number of corner and diagonal certificates is $O(k)$. We assume that the polygons remain convex at all times, therefore it

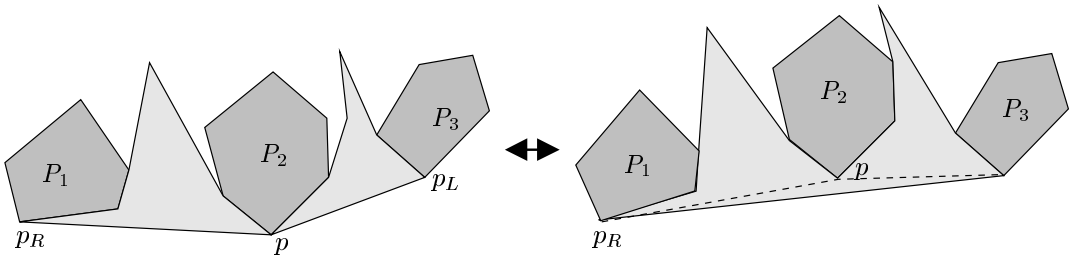

Figure 12. The reflex(from left to right) and corner(from right to left) events and the updates. When a reflex certificate fails, we choose the edge with smaller angle $\theta(\cdot)$, as shown in the right figure.

suffices to maintain reflex certificates for a vertex only if it is adjacent to a diagonal edge, i.e., a free bi-tangent. The number of such vertices is also $O(k)$. Hence, we obtain the following:

Theorem 4.4. *$\mathcal{G}(\mathcal{P})$ can be maintained by using a structure with $O(k)$ certificates. Each event can be processed in time $O(\log n)$.*

This data structure works even if the polygons deform continuously over time, as long as they remain convex at all times. What matters is the number of events. The rigid motion can give us better bounds as stated in Theorem 4.6.

4.3 Combinatorial changes to $\mathcal{G}(\mathcal{P})$

Next, we bound the number of changes to $\mathcal{G}(\mathcal{P})$ if the degree of motion of $\mathcal{P}$ is fixed. We call three polygons *collinear* if they have a line is tangent to all three of them. It can be shown that a corner or reflex certificate fails when three polygons of $\mathcal{P}$ become collinear. Hence, a combinatorial change in $\mathcal{G}(\mathcal{P})$ happens only when three polygons of $\mathcal{P}$ are collinear or when a bi-tangent becomes vertical. The number of such events can be bounded by the following lemma.

Lemma 4.5. *Suppose that P_1, P_2, P_3 are convex polygons with n_1, n_2, n_3 vertices, respectively, and the degree of their motion is constant. They can become collinear $O(n_1+n_2+n_3)$ and $O(n_1n_2+n_1n_3+n_2n_3)$ times for translational and rigid motions, respectively. The bi-tangents between P_1, P_2 can become vertical $O(1)$ and $O(n_1+n_2)$ times for translational and rigid motions, respectively.*

Proof: We first consider two convex polygons P_1 and P_2 with n_1 and n_2 vertices, respectively. As shown in [10], the bi-tangents between them change $O(n_1+n_2)$ and $O(n_1n_2)$ times for constant degree translational and rigid motions, respectively. We bound the number of collinearity events. Since there are four common tangents to two polygons, we have to

consider sixteen possible cases. Without loss of generality, we count the number of collinear events at which all three polygons lie above the tangent line. A similar argument works for the other fifteen cases. Let $\phi_{12}(t)$ denote the slope of the lower outer bi-tangent of the polygons P_1 and P_2 (i.e., P_1 and P_2 lie above the tangent) at time t. By the preceding argument, $\phi_{12}(t)$ consists of $O(n_1 + n_2)$ and $O(n_1 n_2)$ algebraic arcs, for translational and rigid motions, respectively. We can define $\phi_{23}(t)$ similarly for P_2 and P_3. A collinearity event for P_1, P_2, and P_3 corresponds to an intersection between $\phi_{12}(t)$ and $\phi_{23}(t)$. The bounds follow.

As for the second part, a bi-tangent between P_1, P_2 becomes vertical only if the leftmost (or the rightmost) vertices of P_1 and P_2 have the same x coordinates. For translational motions, the leftmost and rightmost vertices of P_1 and P_2 are fixed. Therefore, this can happen only $O(1)$ times. For rigid motions, they can change $O(n_1)$ and $O(n_2)$ times for P_1 and P_2 respectively. Thus, a bi-tangent can become vertical $O(n_1 + n_2)$ times. □

Applying this lemma to all triplets of $\mathcal{P}$, we obtain the following weak upper bounds on the number of changes to $\mathcal{G}(\mathcal{P})$.

Theorem 4.6. *Let $\mathcal{P}$ be a set of k polygons with a total of n vertices. $\mathcal{G}(\mathcal{P})$ changes $O(kn^2)$ times if the degree of motion of $\mathcal{P}$ is fixed. If the motion is translational, the number of changes is only $O(k^2 n)$. If the polygons may deform, the number of changes is bounded by $O(n^3)$.*

5 Pseudo-Triangulation for Simple Polygons

In this section, we consider the case in which $\mathcal{P}$ is a set of k pairwise-disjoint simple polygons with a total of n vertices. We describe a method for maintaining a pseudo-triangulation of $\mathcal{P}$ by combining our algorithm for convex polygons with the algorithm by Basch *et al.* [5] for two simple polygons.

5.1 The mixed pseudo-triangulation

For each $P \in \mathcal{P}$, define the *relative geodesic cycle* $C(P)$ of P to be the shortest cycle in $\mathcal{F}(\mathcal{P})$ with the same homotopy type as ∂P. The region enclosed by $C(P)$, denoted by $\overline{P}$, is called the *relative convex hull* of P. For any two polygons P_1 and P_2 in $\mathcal{P}$, $C(P_1)$ and $C(P_2)$ are interior disjoint because otherwise one of them could not be the shortest cycle. By using relative convex hulls, we decompose $\mathcal{F}(\mathcal{P})$ into two parts $\mathcal{F}_1(\mathcal{P}) = \mathcal{F}(\mathcal{P}) \cap (\bigcup \overline{P})$ and $\mathcal{F}_2(\mathcal{P}) = \mathcal{F}(\mathcal{P}) \setminus \bigcup \overline{P}$. Note that the reflex vertices of polygons do not appear on the boundary of $\mathcal{F}_2(\mathcal{P})$. We compute pseudo-triangulations $\mathcal{T}(\mathcal{F}_1(\mathcal{P}))$ and $\mathcal{T}(\mathcal{F}_2(\mathcal{P}))$ separately. These two triangulations together give the *mixed pseudo-triangulation* of $\mathcal{P}$. First, we consider $\mathcal{F}_1(\mathcal{P})$. Since the interiors of all the relative convex hulls are pairwise disjoint,

we can compute a pseudo-triangulation of each of them separately. We triangulate each $\overline{P}$ as in [5]. Following [5], we define the *pinned relative geodesic cycle* of P, with respect to a pinning point set B (a subset of the vertices of P), to be the shortest cycle in $\mathcal{F}(\mathcal{P})$ that passes through the vertices in B and has the same homotopy type as ∂P. The *pinned relative convex hull* of P with respect to B is the region enclosed by the pinned relative geodesic cycle. If B contains all the vertices of P, then the pinned relative convex hull of P is P itself. Let $w_1, \ldots, w_r$ be a sequence of reflex vertices of P sorted along the boundary of P. We construct a family $C_0, \ldots, C_l$, where $l = \lceil \log r \rceil$, of pinned relative geodesic cycles as follows. The pinning set of C_i is $\{w_{j2^i} \mid 1 \leq j \leq \lceil l/2^i \rceil\}$, i.e., the pinning set of C_i chosen by deleting the every other vertex from the pinning set of C_{i-1}. By overlaying these cycles (i.e., taking the union of the edges of these cycles), we obtain a pseudo-triangulation of $\overline{P} \setminus P$. It can be shown that every nonreflex vertex of P that appears on the boundary of $\overline{P} \setminus P$ is a non-corner vertex of a pseudo-triangle, therefore, by Lemma 2.1, the above pseudo-triangulation of $\overline{P} \setminus P$ is minimum. We refer to this pseudo-triangulation as the *external relative geodesic triangulation*. Refer to [5] for the properties, maintenance, and bounds on the number of changes of this structure.

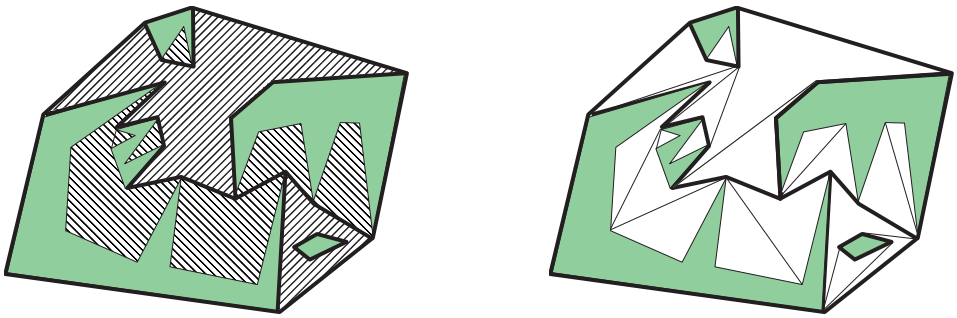

Figure 13. The mixed pseudo-triangulation. (i) $\mathcal{F}_1(\mathcal{P})$ and $\mathcal{F}_2(\mathcal{P})$; $\mathcal{F}_1(\mathcal{P})$ is shaded with the lines of slope -1, and $\mathcal{F}_2(\mathcal{P})$ is shaded with the lines of slope $+1$. (ii) Mixed pseudo-triangulation; thick edges denote the boundary between $\mathcal{F}_1(\mathcal{P})$ and $\mathcal{F}_2(\mathcal{P})$.

Now, we consider $\mathcal{F}_2(\mathcal{P})$. In the following, we generalize the greedy pseudo-triangulation, defined in Section 4, to a connected polygonal subdivision. Consider a connected polygonal region F. A vertex v of F is called a *corner* vertex if the interior angle at v is less than $180°$; otherwise, v is called a *reflex* vertex. Notice that a reflex vertex in the free space is a non-reflex vertex of the object. For a point p on ∂F, a line segment pq in F is called *tangent* to ∂F at p if p is a corner vertex or the line passing through p and q locally supports ∂F at p. (The intuition behind the definition of "tangent" for corner vertices comes from the case in which a corner is formed by two separate convex objects whose boundaries touch. Then a "tangent" segment to the corner is indeed tangent to one of the two convex objects.) For two vertices p, q on ∂F, the line segment pq is called a free bi-tangent if pq lies in F and is tangent to ∂F at both p and q. We now construct the greedy vertical pseudo-triangulation

of F, by sorting the tangents according to their slopes and adding them one by one while maintaining the non-crossing condition, as described in Section 4. For two tangents sharing a corner vertex, we regard them to be non-crossing.

Lemma 5.1. *The greedy pseudo-triangulation of F is a minimal pseudo-triangulation.*

Proof: By Lemma 2.1, it suffices to show that every reflex vertex is a non-corner vertex of one pseudo-triangle in this pseudo-triangulation.[2]. For a reflex vertex p, consider those edges (including the polygon edges) incident upon p. Since these edges do not intersect, in the sense defined in Section 4.1, there must be two adjacent (in counter-clockwise order) edges, say e_1 and e_2, forming an angle greater than 180°. Clearly, e_1 and e_2 belong to the same pseudo-triangle, and p is a non-corner vertex of that pseudo-triangle. Therefore, the greedy pseudo-triangulation is minimum. □

We can then bound the size of such a pseudo-triangulation as follows.

Lemma 5.2. *If ∂F consists of k connected components and m corner vertices, then the number of pseudo-triangles in the greedy pseudo-triangulation of F is $m - 2k$.*

Proof: We just need to prove that for one simple polygon with m corner vertices, the greedy pseudo-triangulation contains $m - 2$ pseudo-triangles. The proof is the same as the one in Lemma 2.1 by assuming the pseudo-triangulation constructed is minimum. □

The relative geodesic triangulation $\mathcal{T}(\mathcal{F}_1(\mathcal{P}))$ and greedy pseudo-triangulation $\mathcal{T}(\mathcal{F}_2(\mathcal{P}))$ together form a pseudo-triangulation of $\mathcal{F}(\mathcal{P})$, which we call the *mixed pseudo-triangulation* (Figure 13). Next, we argue that the mixed pseudo-triangulation is a minimum pseudo-triangulation.

Theorem 5.3. *The mixed pseudo-triangulation is a minimum pseudo-triangulation.*

Proof: We first argue that if a non-reflex vertex p of a polygon $P \in \mathcal{P}$ that is not a vertex of the convex hull of $\mathcal{P}$, then it is a reflex vertex of a connected component of either $\mathcal{F}_1(\mathcal{P})$ or $\mathcal{F}_2(\mathcal{P})$. If p lies in the interior of the relative convex hull $\overline{P}$ or if no diagonal edge is connected to p, then the claim is obvious. So assume that p lies on $\partial\overline{P}$ and that at least one diagonal edge that lies on the boundary of a relative convex hull of a polygon is incident upon p. Let E_p be the set of relative-convex-hull edges, including the edges of P, incident upon p.

Consider the four wedges formed by the lines supporting the edges of P adjacent to p. P lies in one of the wedges, denoted as W, in the neighborhood of p; see Figure 14. None

[2]A reflex vertex here corresponds to a non-reflex vertex in Lemma 2.1 because we are discussing pseudo-triangulation of a polygonal region here, while Lemma 2.1 considers pseudo-triangulation of the free space.

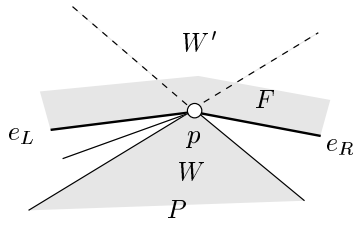

Figure 14. A non-reflex vertex of a polygon that is not a vertex of the convex hull is a reflex vertex of a component of $\mathcal{F}(\mathcal{P})$.

of the edges in E_p can lie in the wedge W' opposite to W because such a segment cannot be an edge of a relative geodesic cycle. Let $e_L \in E_p$ be the rightmost edge lying in the wedge to the left of W, and let $e_R \in E_p$ be the leftmost edge lying in the wedge to the right of W. Since no edge of E_p lies between e_L and e_R, they must lie on the boundary of the some connected component of either $\mathcal{F}_1(\mathcal{P})$ or $\mathcal{F}_2(\mathcal{P})$. By our assumption, at least one of e_L and e_R, say e_L, is not an edge of P. Suppose e_L is an edge of the relative convex hull of polygon Q. The other edge e' of $C(Q)$ incident upon p has to lie in the wedge to the right of W. Since the interiors of relative convex hulls are pairwise disjoint, e' must be e_R. Thus e_L, e_R are two consecutive edges of the relative convex hull $\overline{Q}$ with p as their common endpoint. Consider the connected component F of $\mathcal{F}(\mathcal{P})$ that contains the wedge W' in the neighborhood of p. We claim that p is a reflex vertex of F. Indeed, otherwise, we can shorten the cycle $C(Q)$ by shortcuting it around p, thereby contradicting the assumption that $C(Q)$ is a shortest geodesic cycle. Hence, p is a reflex veretx of a component of either $\mathcal{F}_1(\mathcal{P})$ or $\mathcal{F}_2(\mathcal{P})$.

By the minimality of $\mathcal{T}(\mathcal{F}_1)$ and $\mathcal{T}(\mathcal{F}_2)$, we know that each such vertex is a non-corner vertex of exactly one pseudo-triangle in $\mathcal{T}(\mathcal{F}_1)$ or $\mathcal{T}(\mathcal{F}_2)$. Therefore, by Lemma 2.1, the mixed pseudo-triangulation is minimum. □

While this theorem bounds the size of the pseudo-triangulation, the number of certificates could be significantly smaller since those certificates involving vertices or edges from the same polygon will never fail. We call a diagonal edge of the mixed pseudo-triangulation a *bridge* edge if it connects two different polygons of $\mathcal{P}$. As we will see later, bridge edges play an important role in the maintenance of pseudo-triangulations, as they correspond to the *active* certificates. In the following three lemmas, we show that the number of bridge edges is usually much smaller than the size of the pseudo-triangulation.

Lemma 5.4. *If $\mathcal{P}$ is a set of k polygons, then $\partial\mathcal{F}_2(\mathcal{P})$ consists of $O(k)$ connected components, corner vertices, and bridge edges.*

Proof: The boundary of each connected component of $\mathcal{F}_2(\mathcal{P})$ contains at least one bridge edge, and each bridge edge can lie on the boundary of at most two components of $\mathcal{F}_2(\mathcal{P})$.

The number of connected components is thus at most twice the number of bridge edges. Next, consider a corner vertex p of a connected component $F \in \mathcal{F}_2(\mathcal{P})$. Suppose that q_1, q_2 are the two vertices adjacent to p on ∂F. Then both q_1 and q_2 cannot be on the same polygon as p because this would imply that p is a reflex vertex of that polygon. But as mentioned above, a reflex vertex of a polygon cannot appear on the boundary of $\mathcal{F}_2(\mathcal{P})$. Therefore either pq_1 or pq_2 is a bridge edge. Since each edge has two endpoints, the number of corner vertices is also at most twice the number of bridge edges on $\partial\mathcal{F}$. Hence, it suffices to prove that there are $O(k)$ bridge edges.

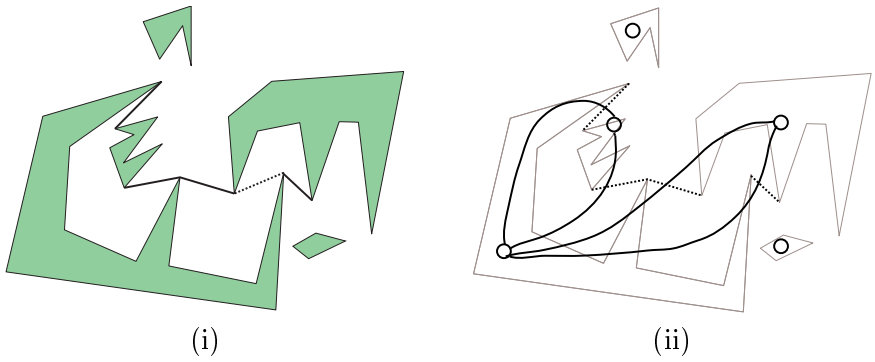

Figure 15. (i) Bridge edges of the polygons in Figure 13; the dashed bridge edge does not appear on $\mathcal{F}_2$. (ii) the underlying graph G.

The number of bridge edges on the convex hull of $\mathcal{P}$ is clearly $O(k)$. For those edges not on the convex hull of $\mathcal{P}$, consider the planar graph G in which a node corresponds to a polygon in $\mathcal{P}$ and an edge between two nodes of G corresponds to a bridge edge on $\partial\mathcal{F}_2(\mathcal{P})$ that connects the corresponding two polygons. The number of bridge edges is bounded by the number of edges in the graph G. The graph G is planar but it is not simple since a pair of nodes may have multiple edges. However, at most two bridge edges between a pair of polygons can lie on the boundary of $\mathcal{F}_2(\mathcal{P})$. Indeed, if two polygons P_1 and P_2 contain more than two bridge edges, then only at most two of them lie on $\mathcal{F}_2(P)$; see Figure 15. By Euler's formula, the number of edges in G is at most $6k - 12$. □

Lemmas 5.2 and 5.4 imply the following.

Corollary 5.5. *The greedy triangulation $\mathcal{T}(\mathcal{F}_2(\mathcal{P}))$ of $\mathcal{F}_2(\mathcal{P})$ has $O(k)$ pseudo-triangles and bridge edges.*

Proof: The number of pseudo-triangles follows immediately from Lemma 5.2. As for the number of bridge edges, consider the construction of the greedy pseudo-triangulation. The number of pseudo-triangles is exactly the same as the number of diagonal edges added. Thus, the total number of bridge edges in the interior of $\mathcal{F}_2(\mathcal{P})$ is bounded by $O(k)$. Including those bridge edges on the boundary of $\mathcal{F}_2(\mathcal{P})$, the total number of bridge edges in $\mathcal{T}(\mathcal{F}_2)$ is bounded by $O(k)$. □

Next, we bound the number of bridge edges in $\mathcal{T}(\mathcal{F}_1(\mathcal{P}))$. Basch *et al.* [5] have shown that any line segment lying inside $\mathcal{F}(\mathcal{P})$ crosses $O(\log n)$ bridge edges, and thus the number of bridge edges is $O(\tau \log n)$, where τ is the size of the minimum-link separator between two simple polygons. Here, we generalize their result to our setup by arguing that any free line segment can cross at most two relative convex hulls.

Lemma 5.6. *Any free line segment crosses $O(\log n)$ bridge edges of $\mathcal{T}(\mathcal{F}_1(\mathcal{P}))$.*

Proof: Consider a free line segment pq. We claim that pq intersects $\partial\overline{P}$, the boundary of the relative convex hull of a polygon P, in at most one point. Indeed, if pq intersects $\partial\overline{P}$ at two points, then we can shortcut $\partial\overline{P}$ between these two points, thereby contradicting the assumption that $\partial\overline{P}$ is a geodesic. This implies that pq can intersect at most two relative convex hulls. As in [5], we can argue that pq crosses $O(\log n)$ bridge edges of each of these two relative convex hulls. □

A *minimum-link subdivision* is a polygonal subdivision of the plane, using as few (line segment) edges as possible, such that each $P \in \mathcal{P}$ is contained in its own face of the subdivision. A *minimum-link separator* for a polygon $P \in \mathcal{P}$ is a simple polygon homotopic to ∂P with as few edges as possible. Consider any Jordan cycle C in $\mathcal{F}(\mathcal{P})$ that is homotopic to ∂P. Clearly, C must intersect all the bridge edges incident upon P because C separates P from other objects in $\mathcal{P}$. Suppose that C consists of τ line segments. Since each line segment on C crosses at most $O(\log n)$ bridge edges connected to P and each bridge edge is crossed at least once, we can conclude that the number of bridge edges incident to P is bounded by $O(\tau \log n)$. Therefore, we have the following result on the number of total bridge edges.

Lemma 5.7. *The number of bridge edges in $\mathcal{T}(\mathcal{F}_1(\mathcal{P}))$ is bounded by $O(\kappa \log n)$, where κ is the number of edges in a minimum-link subdivision of $\mathcal{P}$. For each polygon P in $\mathcal{P}$, the number of bridge edges connected to P is bounded by $O(\tau(P) \log n)$, where $\tau(P)$ is the number of edges in a minimum-link separator of P.*

Thus, we obtain the following bounds on the size of the mixed pseudo-triangulation by combining Theorem 5.3, Corollary 5.5 and Lemma 5.7.

Theorem 5.8. *Let $\mathcal{P}$ be a set of k polygons with a total of n vertices. Suppose there are r reflex vertices, and let κ be the number of edges in a minimum-link subdivision of $\mathcal{P}$. Then the size of the mixed pseudo-triangulation of $\mathcal{F}(\mathcal{P})$ is $O(k + r)$, and only $O(\kappa \log n)$ edges are bridge edges.*

5.2 Maintaining the mixed pseudo-triangulation

We now describe how we maintain the mixed pseudo-triangulation as the polygons in $\mathcal{P}$ move rigidly. We associate with each face Δ (pseudo-triangle) of the mixed pseudo-triangulation a bit $\beta(\Delta)$, which is 0 if $\Delta \in \mathcal{T}(\mathcal{F}_1)$ and 1 if $\Delta \in \mathcal{T}(\mathcal{F}_2)$. For every pseudo-triangle $\Delta \in \mathcal{T}(\mathcal{F}_1)$, we also record the polygon whose relative convex hull contains Δ.

In order to maintain the mixed pseudo-triangulation, we maintain the greedy vertical pseudo-triangulation of $\mathcal{F}_2(\mathcal{P})$, the external relative geodesic triangulation of $\overline{P} \setminus P$ for each polygon in $\mathcal{P}$, and the boundary between $\mathcal{F}_1(\mathcal{P})$ and $\mathcal{F}_2(\mathcal{P})$. As shown in [5], the external relative geodesic triangulation can be maintained by corner and reflex certificates only, and there are local rules to decide which edge to select when a reflex certificate fails. In Section 4, we showed how to maintain the greedy vertical pseudo-triangulation. It thus suffices to describe how to maintain the pseudo-triangulation when the boundary between $\mathcal{F}_1(\mathcal{P})$ and $\mathcal{F}_2(\mathcal{P})$ changes. The boundary changes when a reflex or a corner certificate fails at a vertex lying on the boundary. In the following, we describe the handling of reflex events; handling the corner events is straightforward.

Suppose a reflex certificate p_1pp_2 fails at a vertex p. Let Δ be the pseudo-triangle that contained p as one of its reflex vertices before the event. Let Δ_1 (resp. Δ_2) be the other pseudo-triangle adjacent to the edge pp_1 (resp. pp_2); (Figure 16 (i)). Recall that a reflex event adds the edge p_1p_2 and deletes one of the edges adjacent to p. By examining the bits $\beta(\Delta_1)$ and $\beta(\Delta_2)$, we decide which edge to delete, as follows.

(i) $\beta(\Delta_1) = \beta(\Delta_2) = 1$, i.e., $\Delta_1, \Delta_2 \in \mathcal{T}(\mathcal{F}_2)$: we simply apply the rule for the greedy vertical pseudo-triangulation to determine the edge to be deleted.

(ii) $\beta(\Delta_1) = \beta(\Delta_2) = 0$, i.e., $\Delta_1, \Delta_2 \in \mathcal{F}_1$: they must lie in the relative convex hull of the same polygon, and we can apply the rule for external relative geodesic triangulation.

(iii) $\beta(\Delta_1) = 0$ and $\beta(\Delta_2) = 1$: we keep the edge pp_1 and delete the edge pp_2. (Figure 16(ii))

Finally, the bit $\beta(\cdot)$ can be easily updated as the pseudo-triangulation changes. At each event, we collapse a corner of a pseudo-triangle, attach a triangle to a existing pseudo-triangle, or flip a diagonal edge (in the greedy vertical pseudo-triangulation). In all of these cases, we can simply keep the bit of the affected pseudo-triangle.

In general, the number of certificates for maintaining a pseudo-triangulation is proportional to its size. However, since the objects move rigidly, we can distinguish the certificates to be *passive* or *active*. A passive certificate involves vertices from the same object, and an active certificate involves vertices from different objects. By the rigidity of the motions, those passive certificates will never fail. Clearly, each active certificate has to involve a bridge edge, and each bridge edge correspond to $O(1)$ certificates. By Theorem 5.8, we can bound the number of active certificates for certifying $\mathcal{T}(\mathcal{P})$ as follows.

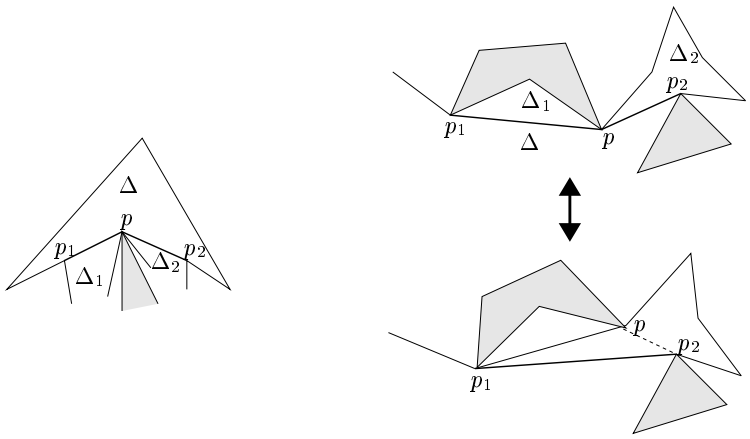

Figure 16. Maintaining mixed pseudo-triangulation.

Theorem 5.9. *The number of active certificates in $\mathcal{T}(\mathcal{P})$ is bounded by $O(\kappa \log n)$, where κ is the number of edges in a minimum-link subdivision of $\mathcal{P}$. For each polygon P in $\mathcal{P}$, the number of active certificates involving P is bounded by $O(\tau(P) \log n + k)$, where $\tau(P)$ is the number of edges in a minimum-link separator of P.*

A trivial bound on the number of changes to the mixed pseudo-triangulation for polygons in rigid motion is $O(n^3)$, which is the number of times when three vertices become collinear.

6 Conclusions

We have shown how to efficiently maintain pseudo-triangulations of the free space for points, convex polygons, and simple polygons moving around in the plane. In addition to collision detection, the pseudo-triangulation structure provides support for walking around the free space and performing operations such as ray shooting and other visibility queries. Unlike any previous collision detection method, our deforming tilings can be adapted to work for deformable objects, thus enabling new applications where simulation of flexible shapes is important. An extension of our ideas to 3D presents the next obvious challenge.

Acknowledgments

The authors would like to thank Bettina Speckmann for pointing out an error in the previous version of Lemma 2.1.

References

[1] P. K. Agarwal, D. Eppstein, L. J. Guibas, and M. H. nger. Parametric and kinetic minimum spanning trees. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 596–605, 1998.

[2] P. K. Agarwal, J. Erickson, and L. J. Guibas. Kinetic BSPs for intersecting segments and disjoint triangles. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 107–116, 1998.

[3] P. K. Agarwal, L. J. Guibas, and J. H. a nd Eric Veach. Maintaining the extent of a moving point set. In *Proc. 5th Workshop Algorithms Data Struct.*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 31–44. Springer-Verlag, 1997.

[4] P. K. Agarwal, L. J. Guibas, T. M. Murali, and J. S. Vitter. Cylindrical static and kinetic binary space partitions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 39–48, 1997.

[5] J. Basch, J. Erickson, L. J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection for two simple polygons. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 102–111, 1999.

[6] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 31:1–28, 1999.

[7] J. Basch, L. J. Guibas, and L. Zhang. Proximity problems on moving points. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 344–351, 1997.

[8] S. Cameron. Collision detection by four-dimensional intersection testing. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 291–302, 1990.

[9] D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 58–67, 1998.

[10] J. Erickson, L. J. Guibas, J. Stolfi, and L. Zhang. Separation-sensitive kinetic collision detection for convex objects. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, 1999.

[11] L. J. Guibas. Kinetic data structures: A state of the art report. In P. K. Agarwal, L. E. Kavraki, and M. Mason, editors, *Proc. 3rd Workshop on Algorithmic Foundations of Robotics*, pages 191–209, Wellesley, MA, 1998. A. K. Peters.

[12] P. Gupta, R. Janardan, and M. Smid. Fast algorithms for collision and proximity problems involving moving geometric objects. *Comput. Geom. Theory Appl.*, 6:371–391, 1996.

[13] D. Halperin and M. Sharir. On disjoint concave chains in arrangements of (pseudo) lines. *Inform. Process. Lett.*, 40(4):189–192, 1991.

[14] D. Halperin and M. Sharir. Corrigendum: On disjoint concave chains in arrangements of (pseudo) lines. *Inform. Process. Lett.*, 51:53–56, 1994.

[15] P. M. Hubbard. Collision detection for interactive graphics applications. *IEEE Trans. Visualization and Computer Graphics*, 1(3):218–230, Sept. 1995.

[16] L. Kettner, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Bounded-degree pseudo-triangulations of points. In *Proc. 17th European Workshop on Computational Geometry*, 2001.

[17] D. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. In *Proc. 16th ACM Symposium on Computational Geometry*, pages 322–330, 2000.

[18] D. Kirkpatrick and B. Speckmann. Separation sensitive kinetic collision detection for simple polygons. In *Proc. Japan Conference on Discrete and Computational Geometry*, 2000.

[19] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comput. Geom.*, 16:419–453, Dec. 1996.

[20] M. K. Ponamgi, D. Manocha, and M. C. Lin. Incremental algorithms for collision detection between general solid models. In *Proc. ACM Siggraph Sympos. Solid Modeling*, pages 293–304, 1995.

[21] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 51–60, 1995.

[22] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[23] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proc. 41st ACM Annual Symposium on Foundations of Computer Science*, pages 443–453, 2000.

[24] S. Suri. *Minimum link paths in polygons and related problems.* Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1987.