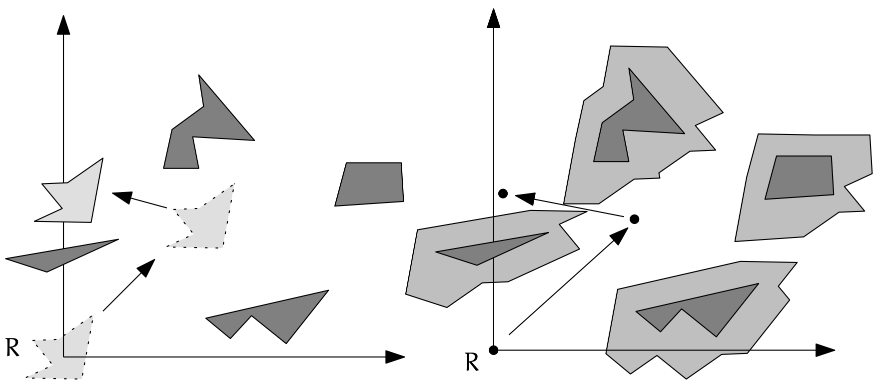
Appendix D

Translational Motion Planning

In a basic instance of motion planning, a robot—modeled as a simple polygon R—moves by translation amidst a set $\mathcal{P}$ of polygonal obstacles. Throughout its motion the robot must not penetrate any of the obstacles but touching them is allowed. In other words, the interior of R must be disjoint from the obstacles at any time. Formulated in this way, we are looking at the motion planning problem in *working space* (Figure D.1a).

However, often it is useful to look at the problem from a different angle, in so-called *configuration space*. Starting point is the observation that the placement of R is fully determined by a vector $\vec{v} = (x, y) \in \mathbb{R}^2$ specifying the translation of R with respect to the origin. Hence, by fixing a *reference point* in R and considering it the origin, the robot becomes a point in configuration space (Figure D.1b). The advantage of this point of view is that it is much easier to think of a point moving around as compared to a simple polygon moving around.



(a) Working Space. **(b)** Configuration Space.

Figure D.1: *Working space and configuration space for a robot* R *and a collection of polygonal obstacles.*

The next question is: How do obstacles look like in configuration space? For an obstacle $P \in \mathcal{P}$ the set $\mathcal{C}(P) = \{\vec{v} \in \mathbb{R}^2 \mid R+\vec{v} \cap P \neq \emptyset\}$ in configuration space corresponds to the obstacle P in the original setting. We write $R+\vec{v}$ for the *Minkowski sum* $\{\vec{r}+\vec{v} \mid \vec{r} \in R\}$. Our interest is focused on the set $\mathcal{F} = \mathbb{R}^2 \setminus \bigcup_{P \in \mathcal{P}} \mathcal{C}(P)$ of *free placements* in which the robot does not intersect any obstacle.

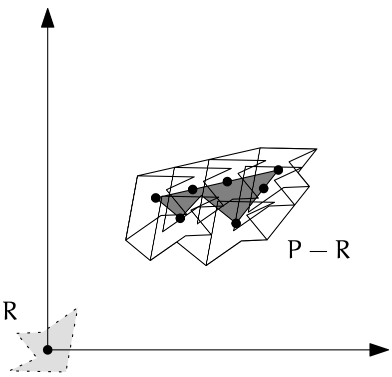


Figure D.2: *The Minkowski sum of an obstacle with an inverted robot.*

Proposition D.1. $\mathcal{C}(P) = P - R$.

Proof. $\vec{v} \in P - R \iff \vec{v} = \vec{p} - \vec{r}$, for some $\vec{p} \in P$ and $\vec{r} \in R$. On the other hand, $R + \vec{v} \cap P \neq \emptyset \iff \vec{r} + \vec{v} = \vec{p}$, for some $\vec{p} \in P$ and $\vec{r} \in R$. □

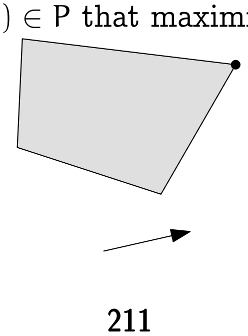
D.1 Complexity of Minkowski sums

Recall that the Minkowski sum of two point sets $P, Q \subseteq \mathbb{R}$ is defined as $P + Q = \{p + q \mid p \in P, q \in Q\}$.

Theorem D.2. *Let P be an m-vertex polygon and Q an n-vertex polygon. Then:*

1. *If both P and Q are convex, then their Minkowski sum $P+Q$ has at most $m+n$ vertices.*
2. *If P or Q is convex, then $P + Q$ has $O(mn)$ vertices.*
3. *In any case, $P + Q$ has $O(m^2n^2)$ vertices.*

Proof. The first claim can be proven using the notion of *extremal points.* If $\overrightarrow{d} = (d_x, d_y)$ is a direction in the plane and $P \subseteq \mathbb{R}^2$ is a set of points, then an *extremal point of* P *in direction* $\overrightarrow{d}$ is a point $p = (p_x, p_y) \in P$ that maximizes $p_x d_y + p_y d_x$:



It is easy to see that, if P and Q are convex polygons and $r \in P + Q$ is an extremal point of $P + Q$ in some direction $\vec{d}$, then r is the sum of some extremal points $p \in P$ and $q \in Q$ in direction $\vec{d}$. If $\vec{d}$ is chosen in "general position", then p, q, and r will be *vertices* of P, Q, and $P + Q$, respectively.

For every $\vec{d}$ in general position, let $p_{\vec{d}}$ and $q_{\vec{d}}$ be the extremal points in P and Q, respectively, in direction $\vec{d}$. Then the vertices of $P + Q$ are the precisely sums $p_{\vec{d}} + q_{\vec{d}}$ over all $\vec{d}$. But as $\vec{d}$ varies continuously and makes a full turn, the pair $(p_{\vec{d}}, q_{\vec{d}})$ can change at most $m + n$ times. This proves the first claim.

For the second claim we need the notion of *pseudodiscs*. A set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of objects in the plane is called a *set of pseudodiscs* if, for every distinct i and j, the boundaries of P_i and P_j intersect in at most two points. That is, the objects "behave" the way discs behave, even though they might not be actual discs.

(Note that it makes no sense to ask whether a single object P_i is a pseudodisc; it only makes sense to ask whether a *set* of objects is a set of pseudodiscs.)

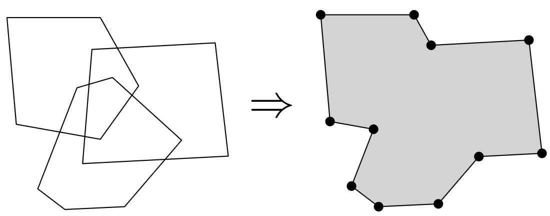
Now we need an auxiliary lemma:

Lemma D.3. *Let $\mathcal{P}$ be a set of convex objects in the plane that are pairwise interior-disjoint, and let Q be a convex object in the plane. Then the set $\mathcal{P} + Q = \{P + Q \mid P \in \mathcal{P}\}$ is a set of pseudodiscs.*

Proof. Suppose for a contradiction that the boundaries of $P_1 + Q$ and $P_2 + Q$ intersect four times, for some $P_1, P_2 \in \mathcal{P}$. This means that there are four directions $\vec{d}_1$, $\vec{d}_2$, $\vec{d}_3$, $\vec{d}_4$, in this circular order, such that P_1 is more extreme than P_2 in directions $\vec{d}_1$ and $\vec{d}_3$, while P_2 is more extreme than P_1 in directions $\vec{d}_2$ and $\vec{d}_4$. But such an alternation cannot happen since P_1 and P_2 are interior-disjoint. □

And we need a second auxiliary lemma:

Lemma D.4. *Let $\mathcal{P}$ be a set of polygonal pseudodiscs with n vertices in total. Then their union $U = \bigcup \mathcal{P}$ hast at most $2n$ vertices.*



Proof. We charge every vertex of the union U to a vertex of $\mathcal{P}$ in such a way that every vertex of $\mathcal{P}$ receives at most two charges.

Let v be a vertex of U. If v is a vertex of $\mathcal{P}$ then we simply charge v to itself. The other case is where v is the intersection point of two edges e_1, e_2, belonging to the boundaries of two distinct polygons $P_1, P_2 \in \mathcal{P}$. In such a case there must be an endpoint w of one edge e_1 or e_2 that lies inside the other polygon P_2 or P_1 (since $\mathcal{P}$ is a set of pseudodiscs). We charge v to w. It is clear that w can receive at most two charges (coming from the two edges adjacent to w). And every vertex of a polygon in $\mathcal{P}$ that is not contained in any other polygon receives at most one charge. □

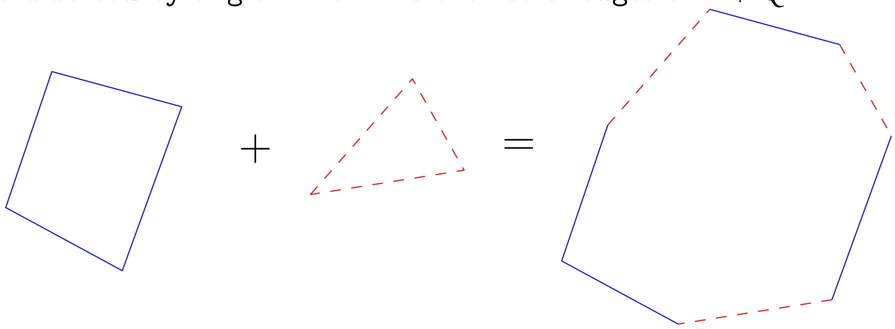
Now we are ready to prove the second claim of Theorem D.2: Suppose P is convex. Triangulate Q into $n-2$ triangles $T_1, T_2, \ldots, T_{n-2}$. Then $P+Q = \bigcup_i (P+T_i)$. By the first claim of our Theorem, each $P+T_i$ is a convex polygon with at most $m+3$ vertices. Therefore, by Lemmas D.3 and D.4, their union has at most $2(m+3)(n-2) = O(mn)$ vertices.

For the third part of our Theorem, let P and Q be arbitrary polygons. Triangulate them into triangles $S_1, \ldots, S_{m-2}$ and $T_1, \ldots, T_{n-2}$, respectively. Then $P+Q = \bigcup_{i,j}(S_i + T_j)$. Arguing as before, for every fixed i, the union $X_i = \bigcup_j (S_i + T_j)$ has at most $12(n-2)$ vertices and as many edges. Each vertex in $P+Q = \bigcup_i X_i$ is either a vertex of some X_i, or the intersection of two edges in two different X_i, X_j. There are at most $\binom{12(m-2)(n-2)}{2} = O(m^2n^2)$ vertices of the latter type. □

We leave as an exercise to show that each case of Theorem D.2 is tight in the worst case.

D.2 Minkowski sum of two convex polygons

Let P and Q be convex polygons, given as circular lists of vertices. Construct the corresponding circular lists of edges E_P and E_Q. Merge E_P and E_Q into a single list of edges E that is sorted by angle. Then E is the list of edges of $P+Q$:



Merging of two sorted lists can be done in linear time.

D.3 Constructing a single face

Theorem D.5. *Given a set S of n line segments and a point $x \in \mathbb{R}^2$, the face of $\mathcal{A}(S)$ that contains x can be constructed in $O(\lambda_3(n) \log n)$ expected time.*

Phrased in terms of translational motion planning this means the following.

Corollary D.6. *Consider a simple polygon R with k edges (robot) and a polygonal environment $\mathcal{P}$ that consists of n edges in total. The free space of all positions of R*

that can be reached by translating it without properly intersecting an obstacle from $\mathcal{P}$ *has complexity* $O(\lambda_3(kn))$ *and it can be constructed in* $O(\lambda_3(kn)\log(kn))$ *expected time.*

Below we sketch[1] a proof of Theorem D.5 using a randomized incremental construction, by constructing the trapezoidal map induced by the given set S of segments. As before, suppose without loss of generality that no two points (intersection points or endpoints) have the same x-coordinate.

In contrast to the algorithm you know, here we want to construct a single cell only, the cell that contains x. Whenever a segment closes a face, splitting it into two, we discard one of the two resulting faces and keep only the one that contains x. To detect whether a face is closed, use a disjoint-set (union-find) data structure on S. Initially, all segments are in separate components. The runtime needed for the disjoint-set data structure is $O(n\alpha(n))$, which is not a dominating factor in the bound we are heading for.

Insert the segments of S in order $s_1, \ldots, s_n$, chosen uniformly at random. Maintain (as a doubly connected edge list) the trapezoidal decomposition of the face f_i, $1 \leqslant i \leqslant n$, of the arrangement $\mathcal{A}_i$ of $\{s_1, \ldots, s_i\}$ that contains x.

As a third data structure, maintain a *history dag* (directed acyclic graph) on all trapezoids that appeared at some point during the construction. For each trapezoid, store the (at most four) segments that define it. The root of this dag corresponds to the entire plane and has no segments associated to it.

Those trapezoids that are part of the current face f_i appear as *active* leaves in the history dag. There are two more categories of vertices: Either the trapezoid was destroyed at some step by a segment crossing it; in this case, it is an interior vertex of the history dag and stores links to the (at most four) new trapezoids that replaced it. Or the trapezoid was cut off at some step by a segment that did not cross it but excluded it from the face containing x; these vertices are called *inactive* leaves and they will remain so for the rest of the construction.

Insertion of a segment s_{r+1} comprises the following steps.

1. Find the cells of the trapezoidal map f_r^* of f_r that s_{r+1} intersects by propagating s_{r+1} down the history dag.

2. Trace s_{r+1} through the active cells found in Step 1. For each split, store the new trapezoids with the old one that is replaced.

   Wherever in a split s_{r+1} connects two segments s_j and s_k, join the components of s_j and s_k in the union find data structure. If they were in the same component already, then f_r is split into two faces. Determine which trapezoids are cut off from f_{r+1} at this point by alternately exploring both components using the DCEL structure. (Start two depth-first searches one each from the two local trapezoids incident to s_{r+1}. Proceed in both searches alternately until one is finished. Mark

[1] For more details refer to the book of Agarwal and Sharir [1].

all trapezoids as discarded that are in the component that does not contain x.) In this way, the time spent for the exploration is proportional to the number of trapezoids discarded and every trapezoid can be discarded at most once.

3. Update the history dag using the information stored during splits. This is done only after all splits have been processed in order to avoid updating trapezoids that are discarded in this step.

The analysis is completely analogous to the case where the whole arrangement is constructed, except for the expected number of trapezoids created during the algorithm. Recall that any potential trapezoid τ is defined by at most four segments from S. Denote by t_r the expected number of trapezoids created by the algorithm after insertion of $s_1, \ldots, s_r$. Then in order for τ to be created at a certain step of the algorithm, one of these defining segments has to be inserted last. Therefore,

$$\Pr[\tau \text{ is created by inserting } s_r] \leqslant \frac{4}{r} \Pr[\tau \text{ appears in } f_r^*]$$

and

$$\begin{aligned}
t_r &= \sum_{\tau} \Pr[\tau \text{ is created in one of the first } r \text{ steps}] \\
&\leqslant \sum_{\tau} \sum_{i=1}^{r} \frac{4}{i} \Pr[\tau \text{ appears in } f_i^*] \\
&= \sum_{i=1}^{r} \frac{4}{i} \sum_{\tau} \Pr[\tau \text{ appears in } f_i^*] \\
&\overset{\text{Theorem 8.34}}{\leqslant} \sum_{i=1}^{r} \frac{4}{i} O(\lambda_3(i)) \\
&\leqslant \sum_{i=1}^{r} \frac{4}{i} ci\alpha(i) = 4c \sum_{i=1}^{r} \alpha(i) \leqslant 4cr\alpha(r) = O(\lambda_3(r)) .
\end{aligned}$$

Using the notation of the configuration space framework, the expected number of conflicts is bounded from above by

$$\begin{aligned}
\sum_{r=1}^{n-1} (k_1 - k_2 + k_3) &\leqslant 16(n-1) + 12n \sum_{r=1}^{n-1} \frac{\lambda_3(r+1)}{r(r+1)} \\
&\leqslant 16(n-1) + 12 \sum_{r=1}^{n-1} \frac{n}{r+1} \lambda_3(r+1) \frac{1}{r} \\
&\leqslant 16(n-1) + 12 \sum_{r=1}^{n-1} \frac{\lambda_3(n)}{r} \\
&= 16(n-1) + 12\lambda_3(n) H_{n-1} \\
&= O(\lambda_3(n) \log n) .
\end{aligned}$$

Exercise D.7. *Show that the Minkowski sum of two convex polygons* P *and* Q *with* m *and* n *vertices, respectively, is a convex polygon with at most* $m+n$ *edges. Give an* $O(m+n)$ *time algorithm to construct it.*

Exercise D.8. *Given an ordered set* $X = (x_1, ..., x_n)$ *and a weight function* $w : X \to \mathbb{R}^+$*, show how to construct in* $O(n)$ *time a binary search tree on* X *in which* x_k *has depth* $O(1 + \log(W/w(x_k)))$*, for* $1 \leqslant k \leqslant n$*, where* $W = \sum_{i=1}^{n} w(x_i)$.

Questions

78. *What is the configuration space model for (translational) motion planning, and what does it have to do with arrangements (of line segments)?* Explain the working space/configuration space duality and how to model obstacles in configuration space.
79. *What is a Minkowski sum?*
80. *What is the maximum complexity of the Minkowski sum of two polygons? What if one of them is convex? If both are convex?*
81. *How can one compute the Minkowski sum of two convex polygons in linear time?*
82. *Can one construct a single face of an arrangement (of line segments) more efficiently compared to constructing the whole arrangement?* Explain the statement of Theorem D.5 and give a rough sketch of the proof.

References

[1] Pankaj K. Agarwal and Micha Sharir, *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press, New York, NY, 1995.