Casting with Skewed Ejection Direction Revisited*

Hee-Kap Ahn Siu-Wing Cheng Otfried Cheong

Department of Computer Science
Hong Kong University of Science and Technology
E-mail: {pastel,scheng,otfried}@cs.ust.hk

1 Introduction

The manufacturing industry has at its disposal a number of processes for constructing objects, including gravity casting, injection molding, stereolithography, NC-machining, deformation, composition, and spray deposition. The survey by Bose and Toussaint [3, 5] gives an overview of geometric problems and algorithms arising in manufacturing processes.

Casting is one of the manufacuring processes in which liquified material is poured into a cast that has a cavity formed by two cast parts. After the material hardens, the movable cast part first retracts from the fixed cast part carrying the object with it. Afterwards, the object is ejected from the retracted cast part. In most existing machinery, the retraction and ejection directions are identical, and previous work on this problem has assumed this restriction on casting. Existing technology for injection molding, however, already has the flexibility to accommodate an ejection direction that is different from the retraction direction of the moving cast part. Exploiting this possibility allows to cast more parts, or to cast parts with simpler moulds, and is the subject of the present paper.

To simulate the retraction, the fixed cast part will first be removed in a direction opposite to the retraction. Then to simulate the ejection, the remaining cast part will be removed in a direction opposite to the ejection. In our model of casting, the two cast parts are to be removed in two given directions and these directions need not be opposite. Note that the ordering of removal is important.

The cast parts should be removed from the object without destroying either cast parts or the object. This ensures that the given object can be mass produced by re-using the same cast parts. The casting process may fail in the removal of the cast parts: if the cast is not designed properly, then one or more of the cast parts may be stuck during the removal phase. The problem we address here concerns this aspect: Given a 3-dimensional object, is there a cast for it whose two parts can be removed after the liquid has solidified? An object for which this is the case is called *castable*.

Some heuristic approaches [6, 7, 8] has been proposed to solve the 3-dimensional castability problem. Bose et al. [4] considered a special model of casting, the *sand casting model*, where the partition of the cast into two parts must be done by a plane. The first complete algorithm to determine the castability of polyhedral parts for opposite directional cast removal was proposed by Ahn et al. [2]. All the results for opposite cast parts removal in [2, 8, 9] rely on the property that an object is castable if its boundary surface is completely visible from the two opposite removal directions. This is not true when the removal directions are non-opposite: there are polyhedra whose whole boundary is visible from the removal directions but which are not castable with respect to those directions [2]. Ahn et al. [1] gave a complete characterization of castability, under the assumption that the cast has to consist of two parts that are to be removed in two not necessarily opposite directions. They presented an $O(n^3 \log n)$ algorithm to determine the castability. In this paper we improve the running time to $O(n^2 \log n)$. We do not assume any special separability of the two cast parts, and allow parts of arbitrary genus.

2 Preliminaries

Throughout this paper, $\mathcal{P}$ denotes a polyhedron, that is, a (not necessarily convex) solid bounded by a piecewise linear surface. The union of vertices, edges, and facets on this surface forms the boundary of $\mathcal{P}$, which we denote by $\mathrm{bd}(\mathcal{P})$. We require $\mathrm{bd}(\mathcal{P})$ to be a connected 2-manifold. Each facet of $\mathcal{P}$ is a connected planar polygon, which is allowed to have polygonal holes. Two facets of $\mathcal{P}$ are called *adjacent* if they share an edge. We assume that adjacent facets are not coplanar—they should be merged into one—but we do allow coplanar non-adjacent facets. We also assume that $\mathcal{P}$ is *simple*, which means that no two non-adjacent facets share a point. Our assumptions imply that $\mathcal{P}$ may contain tunnels,

*Research supported in part by RGC CERG HKUST 6074/97E

but no voids—a polyhedron with a void is not castable anyway.

Our characterization of castability applies to general objects $\mathcal{Q}$. This is important since many industrial parts are not polyhedral. However, our algorithms are only designed for polyhedra.

We call the two removal directions the *red* and the *blue* direction, and we denote them by $\vec{d}_r$ and $\vec{d}_b$ respectively. We assume that the outer shape of the cast equals a box denoted by $\mathcal{B}$, which is large enough so that the object is contained strictly in its interior. This assumption is necessary for producing connected cast parts. Our goal is to decompose the cast into two parts which only overlap along their boundaries. The cast part to be removed first is called the *red part* and is denoted by $\mathcal{C}_r$. The other cast part is called the *blue part* and is denoted by $\mathcal{C}_b$. The removal directions for $\mathcal{C}_r$ and $\mathcal{C}_b$ are $\vec{d}_r$ and $\vec{d}_b$ respectively. Each of $\mathcal{C}_r$ and $\mathcal{C}_b$ is a connected subset of $\mathcal{B}$. The union of $\mathcal{Q}$ and $\mathcal{C}_r \cup \mathcal{C}_b$ equals $\mathcal{B}$. Note that $\mathcal{B} \setminus (\mathcal{C}_r \cup \mathcal{C}_b)$ is an open set and $\mathrm{cl}(\mathcal{B} \setminus (\mathcal{C}_r \cup \mathcal{C}_b))$ is the object to be manufactured.

3 A Characterization of Castability

We call an object $\mathcal{Q}$ *castable* with respect to $(\vec{d}_r, \vec{d}_b)$ if we can translate $\mathcal{C}_r$ to infinity in direction $\vec{d}_r$ without collision with $\mathrm{int}(\mathcal{Q})$ and $\mathrm{int}(\mathcal{C}_b)$, and then translate $\mathcal{C}_b$ to infinity in direction $\vec{d}_b$ without collision with $\mathrm{int}(\mathcal{Q})$. The order of removal is important.

We illuminate $\mathcal{Q}$ with two sources of parallel light. The red light source is at infinity in direction $\vec{d}_r$ and the blue light source is at infinity in direction $\vec{d}_b$. We say that a point p in space is illuminated by red light if a red ray from direction $\vec{d}_r$ can reach p without intersecting $\mathrm{int}(\mathcal{Q})$. The definition for a point p being illuminated by blue light is similar. Note that we assume that a light ray will not stop when it grazes the boundary of $\mathcal{Q}$.

We call the (possibly disconnected) subset of $\mathcal{B} \setminus \mathcal{Q}$ not illuminated by red light the *red shadow volume*. We denote it by $\mathcal{V}_r$. Similarly, We call the subset of $\mathcal{B} \setminus \mathcal{Q}$ not illuminated by blue light the *blue shadow volume*. We denote it by $\mathcal{V}_b$. If we sweep $\mathcal{V}_b$ to infinity in direction $\vec{d}_r$, then we will encounter a set of points in $\mathcal{B}$ and we denote this set of points by $\mathcal{V}_b^*$. Note that $\mathcal{V}_b^*$ includes $\mathcal{V}_b$ itself.

Lemma 1 *If $\mathcal{Q}$ is castable, then $\mathcal{V}_r \subseteq \mathcal{C}_b$ and $\mathcal{V}_b^* \subseteq \mathcal{C}_r$.*

We are now ready to prove the necessary and sufficient condition for an object to be castable. Its proof can be found in the full paper.

Theorem 2 *Given an object $\mathcal{Q}$ and removal directions $(\vec{d}_r, \vec{d}_b)$, $\mathcal{Q}$ is castable if and only if $\mathcal{V}_r$ lies in one connected component of $\mathrm{cl}(\mathcal{B} \setminus (\mathcal{V}_b^* \cup \mathcal{Q}))$.*

The condition in Theorem 2 also implies that $\mathcal{V}_r \cap \mathcal{V}_b$ is empty. However, unlike the case where the two removal directions are opposites of each other, the emptiness of $\mathcal{V}_r \cap \mathcal{V}_b$ does not guarantee castability. Figure 1 shows an object that is not castable, even though $\mathcal{V}_r \cap \mathcal{V}_b$ is empty.

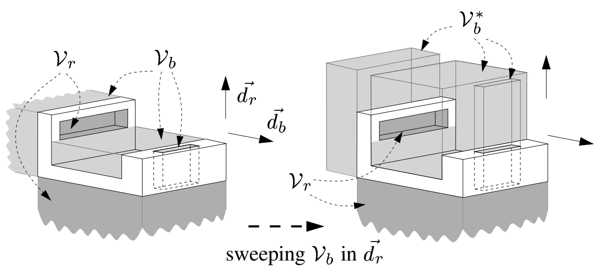

Figure 1: An object $\mathcal{Q}$ and its shadow volume. $\mathcal{V}_r$ intersects two connected components of $\mathcal{B} \setminus \mathrm{int}(\mathcal{V}_b^* \cup \mathcal{Q})$.

4 Feasibility Test for a Polyhedron

Throughout this section, we treat $\vec{d}_r$ as the positive vertical direction and we assume that $\mathcal{P}$ lies above the xy-plane. We use $\ell(p)$ to denote the vertical line through a point p.

The *red shadow*, denoted by $\mathcal{S}_r$, is the complement of the set of points on $\mathrm{bd}(\mathcal{P})$ that can be illuminated by red light from $\vec{d}_r$ without being obscured by $\mathrm{int}(\mathcal{P})$. The *blue shadow*, denoted by $\mathcal{S}_b$, is defined similarly for blue light from $\vec{d}_d$, and the intersection $\mathcal{S}_r \cap \mathcal{S}_b$ is called the *black shadow*.

For each polyhedron edge e, let $h_b(e)$ denote the plane through e and parallel to $\vec{d}_b$. Then e is a *blue silhouette edge* if it satisfies two requirements. The first requirement is that the two facets incident to e lie in a closed halfspace bounded by $h_b(e)$ and the dihedral angle through $\mathrm{int}(\mathcal{P})$ is less than π. The second requirement is that if a facet incident to e is parallel to $\vec{d}_b$, then e should be behind that facet when viewing from direction $\vec{d}_b$. A *lower blue silhouette edge* is a blue silhouette edge e where $\mathcal{P}$ lies *above* $h_b(e)$ locally at e. Similarly, an *upper blue silhouette edge* is a blue silhouette edge e where $\mathcal{P}$ lies *below* $h_b(e)$ locally at e.

For each lower blue silhouette edge e, imagine that e is a neon tube shooting blue rays in direction $-\vec{d}_b$. We trace the "sheet" of blue rays emanating from e until they hit $\mathrm{int}(\mathcal{P})$, or hit an edge or facet parallel to $\vec{d}_b$ and below $\mathrm{int}(\mathcal{P})$ locally, or reach infinity in direction $-\vec{d}_b$. The union of these intercepted or unintercepted blue rays define a subset of the plane $h_b(e)$ called a *lower blue curtain*. Note that a lower blue curtain may pass through a facet of $\mathcal{P}$ parallel to $\vec{d}_b$. Such a facet must then be locally above $\mathcal{P}$. For each upper blue silhouette edge e, we define an *upper blue curtain* similarly.

Given a blue silhouette edge e, we use $\Gamma(e)$ to denote the blue curtain defined by e. If $\Gamma(e)$ is nonempty, then it is bounded by a silhouette edge e called the head, two edges parallel to $\vec{d}_b$ and incident to the endpoints of e called the *side edges*, edges parallel to $\vec{d}_b$ but not incident to the endpoints of e called the *finger edges* and a set $\xi(e)$ of polygonal chains opposite to e called the *tail*. Note that the head and tail of a blue curtain lie on $\mathrm{bd}(\mathcal{P})$

We divide castability testing into three steps. We first verify that the boundary of $\mathcal{P}$ is completely illuminated by red and blue light. That is, $\mathcal{S}_r \cap \mathcal{S}_b$ is empty. Once this test is passed, we then check whether $\mathcal{V}_r \cap \mathcal{V}_b$ is empty. If this test is passed, then we construct the red and blue cast parts and verify that they are connected.

4.1 Testing emptiness of black shadow

We can determine the parts of $\mathrm{bd}(\mathcal{P})$ illuminated by red light in time $O(n^2 \log n)$ by computing the visibility map from $\vec{d_r}$. Similarly, we can determine the parts of $\mathrm{bd}(\mathcal{P})$ illuminated by blue light. We can then decide whether $\mathcal{S}_r \cap \mathcal{S}_b$ is empty by testing the intersection separately on every facet of $\mathrm{bd}(\mathcal{P})$, for instance with a plane sweep algorithm. In total, this test takes time $O(n^2 \log n)$. Details of the testing can be found in the full paper.

4.2 Testing emptiness of black shadow volume

Once we know that $\mathcal{S}_r \cap \mathcal{S}_b$ is empty, we can determine if the black shadow volume is empty by examining the lower envelope, denoted by $\mathcal{L}$, of blue shadow facets and lower blue curtains. This is more efficient than computing $\mathcal{V}_r \cap \mathcal{V}_b$ directly. We show how this is done in the following. Let π be the projection function.

Lemma 3 *Let $\mathcal{L}^*$ be the set of points in $\mathcal{B}$ encountered while we sweep $\mathcal{L}$ to infinity vertically upward. Then $\mathcal{L}^* = \mathcal{V}_b^*$.*

The emptiness of the black shadow volume is now determined by the necessary and sufficient condition stated in the following result. Its proof can be found in the full paper.

Lemma 4 *Suppose that $\mathcal{S}_r \cap \mathcal{S}_b$ is empty. Then $\mathcal{V}_r \cap \mathcal{V}_b$ is non-empty if and only if for two lower blue silhouette edge e and f, $\ell(p) \cap e$ is not below $\ell(p) \cap \Gamma(f)$ for some point p in $\pi(e) \cap \mathrm{int}(\pi(\Gamma(f)))$ or $\pi(e) \cap \pi(\xi(f))$.*

To test the condition in Lemma 4, we identify all lower blue silhouette edges and construct the lower blue curtains. Then we identify all blue shadow facets and construct $\mathcal{L}$, the lower envelope of all lower blue curtains and all blue shadow facets. While we construct $\mathcal{L}$ we check whether $\mathcal{V}_r \cap \mathcal{V}_b$ is empty. This implies that we can construct the swept volume $\mathcal{V}_b^*$ in time $O(n^2 \log n)$ by computing this lower envelope $\mathcal{L}$.

Lemma 5 *Suppose that $\mathcal{S}_r \cap \mathcal{S}_b$ is empty. Then the lower envelope $\mathcal{L}$ has complexity $O(n^2)$.*

Proof. (Sketch) There are three different kinds of edges: shadow facet edges including heads and tails of lower blue curtains, side edges and finger edges. The complexity of $\mathcal{L}$ is determined by the number of these edges and new vertices generated by these edges.

By the absence of black shadow, it can be shown that shadow facet edges do not cross each other in the projection, that is, they do not introduce any new vertex in $\mathcal{L}$.

Now consider a finger edge of a lower blue curtain $\Gamma(e)$. It can be divided into $O(n)$ segments of two types : segments lying on facets of $\mathcal{P}$ which are parallel to $\vec{d_b}$, and segments lying on the other blue curtains. The segments of the former type are shadow facet edges and there are $O(n^2)$ of them. So we only consider the segments of the latter type. A segment of the latter type is the intersection of $\Gamma(e)$ and a blue curtain, say $\Gamma(f)$. If $\Gamma(f)$ is a lower blue curtain, then one of two facets incident to f is a red shadow facet. Since a lower blue curtain $\Gamma(e)$ intersect this red shadow facet, it contains a black point, which contradicts $\mathcal{S}_r \cap \mathcal{S}_b = \emptyset$. So the segement is on an upper blue curtain. Since upper blue curtains do not appear in $\mathcal{L}$ they do not introduce any new vertex in $\mathcal{L}$.

Now we only need to check how many new vertices are generated by side edges and shadow facet edges. Let e be a side edge of a lower blue curtain and h be a vertical plane containing e. Then h intersects a shadow facet f_i in a line segment, denoted by s_i. Since a shadow facet edge lies on either an edge of $\mathcal{P}$ or the projection of an edge of $\mathcal{P}$ on $\mathrm{bd}(\mathcal{P})$ in $-\vec{d_b}$ direction, h intersects linear number of shadow facets in linear number of nonintersecting line segments. The 2D lower envelope of line segments e and $s_1, s_2, \ldots$ on h has linear complexity. Since $\mathcal{P}$ has linear number of side edges, $\mathcal{L}$ has $O(n^2)$ vertices in total. ◻

Based on Lemma 5, we can develop an algorithm to construct $\mathcal{L}$ in $O(n^2 \log n)$ time. In the process, we also verify whether $\mathcal{V}_r \cap \mathcal{V}_b$ is empty. Details can be found in the full paper.

Lemma 6 *Suppose that $\mathcal{S}_r \cap \mathcal{S}_b$ is empty. Then we can test emptiness of $\mathcal{V}_r \cap \mathcal{V}_b$, and if $\mathcal{V}_r \cap \mathcal{V}_b$ is empty we can construct $\mathcal{L}$ in $O(n^2 \log n)$ time.*

4.3 Cast part construction

Points on blue shadow facets and points close to and above lower blue curtains are not illuminated by blue light. So they can only be removed in direction $\vec{d_r}$. Other points encountered while translating these points towards infinity in $\vec{d_r}$ should then belong to $\mathcal{C}_r$ too. This is exactly the subset of $\mathcal{B}$ swept by the lower envelope of lower blue curtains and blue shadow facets in direction $\vec{d_r}$. Denote this subset of $\mathcal{B}$ by X. X may be disconnected. Since $\mathcal{P}$ is strictly contained in $\mathcal{B}$, we can take a layer of material M beneath the top facet of $\mathcal{B}$ and above $\mathcal{P}$ and use M to connect all the components in X. By Lemma 5, we can compute the lower envelope of lower blue curtains and blue shadow facets in $O(n^2 \log n)$ time and so $X \cup M$ can then be computed in the same time. $X \cup M$ is our potential red cast part. All the points in $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ are removable in direction $\vec{d_b}$. So

$\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ is our potential blue cast part. However, $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ may be disconnected. Thus, we will try to attach some components in $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ to $X \cup M$ instead.

Such a process is guided by the condition in Theorem 2. Observe that $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ is a subset of $\mathcal{B} \setminus (\mathcal{V}_b^* \cup \mathcal{P})$. From the above analysis, any blue cast part and hence $\mathcal{V}_r$ lies inside $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$. Thus, we can attach every component of $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ not containing any point in $\mathcal{V}_r$ to $X \cup M$. These components are removable in direction $\vec{d}_r$ as they do not contain points in $\mathcal{V}_r$. In addition, if there are more than one remaining component of $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ containing $\mathcal{V}_r$, then we can abort and report that $\mathcal{P}$ is not castable. Otherwise, we have the cast parts.

It is unnecessary to compute $\mathcal{V}_r$. Every facet bounding $\mathcal{V}_r$ is connected to some red shadow facet. The red shadow facets can be computed in $O(n^2 \log n)$ time using visibility maps. Each red shadow facet lies on a facet bounding $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$. We identify the set of facets bounding $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ that contain the red shadow. Then we test whether this set of facets lie in the same component of $\mathrm{cl}(\mathcal{B} \setminus (X \cup M))$ using a linear-time graph traversal.

Theorem 7 *Let $\mathcal{P}$ be a simple polyhedron with n vertices. Given a pair of directions, we can determine castability and construct cast parts, if castable, of $\mathcal{P}$ in $O(n^2 \log n)$ time and $O(n^2)$ space.*

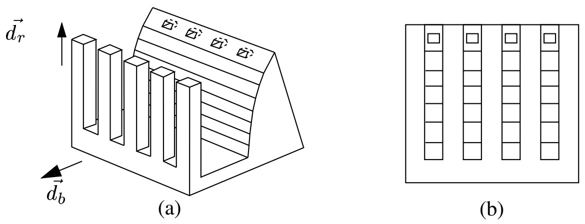

Figure 2: (a)A polyhedron with five vertical legs and four small holes, and (b)The visibility map from $\vec{d}_b$

The analysis of our algorithm is tight as seen from the example in Figure 2 (a). The visibility map from the blue light source shown in Figure 2 (b) has $O(n^2)$ complexity, which shows that $\mathcal{V}_b$ has $O(n^2)$ complexity. It follows that the lower envelope $\mathcal{L}$ of all the lower curtains and blue shadow has complexity $\Omega(n^2)$. In fact, any cast for the polyhedron in Figure 2 (a) must have complexity $\Omega(n^2)$. The proof is omitted in this abstract.

5 Finding a Pair of Directions

In this section we briefly sketch an algorithm to solve the following problem: Given a polyhedron $\mathcal{P}$, decide whether there is a pair of directions $(\vec{d}_r, \vec{d}_b)$ in which $\mathcal{P}$ is castable. In fact, we will solve the more general problem of finding all pairs of directions $(\vec{d}_r, \vec{d}_b)$ for which $\mathcal{P}$ can be cast.

The set of all pairs of directions forms a 4-dimensional parameter space Ψ. We choose an appropriate parameterization that gives rise to algebraic surfaces in Ψ, see for instance Latombe's book [10]. Our goal is to compute that part of Ψ that corresponds to pairs of directions in which $\mathcal{P}$ is castable. As we have proven before, castability depends on a number of simple combinatorial properties: the emptiness of the black shadow, the configuration of the curtain projections, and the connectedness of the blue cast part. We will compute an arrangement of algebraic surfaces in Ψ that includes all pairs of directions where one of these properties could possibly change. The following lemma enumerates all relevant situations. Its proof can be found in the full paper.

Lemma 8 *Let γ_1 and γ_2 be two pairs of directions, such that $\mathcal{P}$ is castable in γ_1 but not in γ_2. Let π be any path in 4-dimensional configuration space Ψ connecting γ_1 and γ_2. Then on π there is a pair of directions $(\vec{d}_r, \vec{d}_b)$ such that one of the following conditions holds:*

- *(i) A facet of $\mathcal{P}$ is parallel to $\vec{d}_r$ or $\vec{d}_b$.*
- *(ii) The projection in direction $\vec{d}_r$ of a vertex v coincides with the projection of an edge e. Here edges and vertices are edges and vertices of $\mathcal{P}$ or of the blue shadow $\mathcal{S}_b$.*
- *(iii) Two polyhedron vertices lie in a plane parallel to the plane determined by $\vec{d}_r$ and $\vec{d}_b$.*

This characterization can be turned into an algorithm that computes the arrangement of these surfaces and tests each cell separately.

Theorem 9 *Given a polyhedral object $\mathcal{P}$ with n vertices and edges, we can in time $O(n^{14} \log n)$ construct a set of all possible pairs of directions in which $\mathcal{P}$ is castable.*

References

[1] H.K. Ahn, S.W. Cheng, and O. Cheong. Casting with skewed ejection direction. In *Proc. 9th Annu. International Symp. on Algorithms and Computation,*, volume 1533 of *Lecture Notes in Computer Science*, pages 139–148. Springer-Verlag, 1998.

[2] H.K. Ahn, M. de Berg, P. Bose, S.W. Cheng, D. Halperin, J. Matoušek, and O. Schwarzkopf. Separating an object from its cast. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 221–230, 1997.

[3] P. Bose. *Geometric and Computational Aspects of Manufacturing Processes*. PhD thesis, McGill University, 1994. Also available as UBC Tech Rep 95-02, Dept. of Comp. Sci, Univ. of British Columbia, 1995.

[4] P. Bose, D. Bremner, and M. van Kreveld. Determining the castability of simple polyhedra. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 123–131, 1994. To appear in Algorithmica.

[5] P. Bose and G. Toussaint. Geometric and computational aspects of manufacturing processes. *Comput. & Graphics*, 18:487–497, 1994.

[6] L.L. Chen, S.Y. Chou, and T.C. Woo. Parting directions for mould and die design. *Computer-Aided Design*, 25:762–768, 1993.

[7] K. Hui. Geometric aspects of mouldability of parts. *Computer Aided Design*, 29(3):197–208, 1997.

[8] K.C. Hui and S.T. Tan. Mould design with sweep operations—a heuristic search approach. *Computer-Aided Design*, 24:81–91, 1992.

[9] K. K. Kwong. *Computer-aided parting line and parting surface generation in mould design*. PhD thesis, The University of Hong Kong, Hong Kong, 1992.

[10] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.