Fourier Collision Detection

Mikola Lysenko
Spatial Automation Lab
University of Wisconsins-Madison

March 7, 2011

Abstract

In this paper, we investigate a new approach to narrowphase collision detection for rigid objects based on the Fourier transform. This new collision test scales with respect to accuracy, and we are able to rigorously establish an upper bound on the error of our test relating the Hausdorff metric to the number of Fourier coefficients used. Because our new form of the collision test is also a smooth inequality, it can be used as a holonomic unilateral constraint in many applications, such as path planning, rigid body dynamics, nesting or tool placement, replacing the need for more ad-hoc normal/contact based constraint solvers. Moreover, we also show how this constraint can be directly differentiated via Fourier multipliers with only a constant factor overhead, which leads to a simple method for constructing a Jacobian for both normal forces and rotational torques.

1 Introduction to Collision Detection

Let us begin by defining the collision detection problem for solids in $\mathbb{R}^d$. As usual, we consider a *solid*, $S \subset \mathbb{R}^d$ to be a compact regular subset of d-dimensional Euclidean space. By regular, we mean that

$$S = \kappa \iota S$$

where κ denotes the closure operator and ι denotes the interior. We say that two solids, $S, T \subset \mathbb{R}^d$ *collide* when the following predicate holds,

$$\text{collide}(S,T) \overset{\text{def}}{\equiv} \iota \left(S \cap T \right) \neq \emptyset. \tag{1}$$

Under this definition, two solids collide if and only if their interiors overlap – surface-surface contact is not considered to be a collision – and for this reason, some authors call this particular test *interference detection*[4]. In literature where the general problem of collision detection for multiple solids is considered, Eqn. 1 is sometimes named *narrowphase collision detection* to distinguish it from *broadphase collision detection*[34]. In other sources which deal with moving objects, a further distinction is made between *dynamic* and *static* collision detection[15]. For the sake of conciseness we do not concern ourselves with these other more complicated variants, and henceforth we shall refer to the "*static narrowphase interference detection*" problem (Eqn. 1) as simply "*collision detection*".

1.1 Background

The problem of numerical collision detection is very old, perhaps originating in industrial engineering, optimization and control research from the 1960s. From our reading of the literature it appears that the very first explicit statement of the problem can be traced to a paper by Comba[6], though some of the basic ideas appear to have been around much longer and are implicit in the variational formulation of rigid body dynamics[43]. Over the last few decades collision detection has been researched extensively, and the field is now in such a state where it would be impossible to give a thorough account in anything less than a full paper. Thus, we refer the reader to an external source, such as one of the following surveys: [26, 10, 53].

Much of the interest in collision detection stems from the fact that it has proven to be a fundamental tool in computer graphics, design and robotics. Some of the many applications include animation[35], rigid body dynamics[7], accessibility for machined parts[4], robot path planning[28], packaging/nesting[19, 52, 9], CSG[55, 45] and virtual reality[57]. Collision detection is also computationally demanding and even today it remains a major bottleneck in many applications[1]. As a result, there is still a great need for ever faster collision detection methods.

One important perspective on collision detection that has been highly influential in the development of our thinking is the concept of a configuration space, which was introduced into robotics by Lozano-Perez in the context of motion planning[28]. Configuration space methods parametrize the predicate in Eqn. 1 by fixing a pair of solids and then looking at the intersections in terms of their relative transformations (or configurations). These configurations form a group[1] called a configuration space which acts on the homogeneous space consisting of all solids. Under this identification the set of all configurations satisfying the collision predicate forms a proper subset of the configuration space which is known as the configuration space obstacle of the solids, and thus the collision predicate becomes a point membership classification relative to this set.

1.1.1 Combinatorial Collision Detection

At a very abstract level there are two fundamental approaches to collision detection: combinatorial and analytic. Combinatorial methods resolve collisions by searching for a feasible point in the interior of the intersection region as a certificate of collision; or if no such point exists they attempt to produce a certificate of non-intersection. Computer graphics has largely favored this approach due to its reliance on triangulated meshes and the relative speed of these calculations. The papers by Maruyama and Boyse first described collision detection procedures for polyhedral shapes[31, 4]. Later on, Gilbert, Johnson and Keerthi proposed one of the first closest-feature detection algorithms for convex polytopes[17]. These basic techniques were further extended to exploit temporal coherence by Lin and Canny[27], leading to substantial performance improvements.

Despite its popularity the combinatorial approach can be quite problematic when used in practice. As an explanation for why these methods appear to suffer from such difficulties, we observe that many applications of collision detection are essentially constrained optimization problems in disguise. In rigid body dynamics, for example, this optimization problem is given by the Lagrangian formulation of the equations of motion, and the requirement that no two solids intersect is equivalent to a holonomic unilateral constraint. Impulse forces and torques associated with an impact then originate from the Lagrange multipliers associated to these constraints by Gauss' principle of least constraint[36, 51, 11]. Similar optimization formulations exists for path planning[28, 16], tool placement[6] and nesting[52, 9].

But just knowing the result of Eqn. 1 is not quite enough information to solve such a constrained optimization problem; rather one needs at least a C^1 constraint function and its gradient in order to apply the Karush-Kuhn-Tucker conditions[22, 24, 36, 20, 30]. In some very special cases it is possible to recover a 'gradient-like' quantity from the certificate point. For example, in the case of ellipsoid or non-degenerate polyhedral collision, one can take the surface normal associated to the instantaneous point of contact as the gradient of the constraint function and work from there. However, for general surface-surface contacts, this is not always possible and so one is forced to introduce increasingly elaborate schemes for handling each type of contact surface.

1.1.2 Analytic Collision Detection

These optimization issues relating to collision response can be addressed in a more efficient and uniform way using the analytical formulation, which is often favored in robotics. In the analytical approach the collision predicate is simply treated as a boolean combination of smooth inequalities over some underlying configuration space of the solids, which are ultimately fed into some generic optimization code. From the configuration space perpsective these inequalities are implicit representations of configuration space obstacles[28]. The basic ground work for a theory of analytic collision detection for convex sets was sketched out in Comba's original paper[6], albeit absent some of the subtleties associated with rotational motion. Similar concepts can be found in Moreau's complementarity formulation of rigid body dynamics[36] and the potential field formulation of configuration space based path planning[16].

[1] Or more technically a torsor.

In this paper we shall focus on the analytic approach, motivated by its greater robustness and simplicity. The key issue in analytic collision detection, like implicit modeling, is the representation of the constraint function, which is ultimately non-unique. In choosing this function, there are a number of trade-offs which can be made between ease of evaluation, local convexity and the size of the representation.

One choice for the constraint function is a signed distance field, or generalized penetration depth measurement. The value of a signed distance field at a point is the distance to the closest point on the boundary of the solid/configuration space obstacle and the sign encodes the point membership value (typically +1 for interior, 0 for boundary, -1 for exterior). The idea of representing solids as signed distance fields is due to Ricci[41], though it appears to have been used in image processing much before that[44]. Many authors have proposed methods to compute translational configuration space obstacles via distance fields[16, 14, 13]. More recently, Zhang and Manocha developed a distance transform for the non-commutative group $SE(3)$, which is useful in robotics and motion planning[58]. From an optimization perspective, the primary benefit of distance fields is that they regularize the process of gradient descent to a stable point[25]. However storing distance fields bears a number of hidden costs, since they require either dense samplings[44], Voronoi decompositions[18] or else must be approximated (which partially negates their regularization benefits)[12].

Our approach departs from the distance field formulation, and instead we view the underlying constraint as a smooth, nonnegative, 'gap' function[20] on the configuration space. Our construction of the gap function is closely related to the Minkowski-sum formulation of the translational configuration space obstacle as first proposed by Lozano-Perez[28]. The interpretation of the Minkowski-sum as a convolution was discovered by Kavraki[23], along with the application of the Fourier analysis. Chirkijian and Kyatkin then extended this basic commutative formulation to general rigid motions using representation theory[5]. The commutative Minkowski sum for translational obstacles was also shown by Roerdink to be a special case of a more general non-commutative Minkowski product[42], which can be interpreted as a non-commutative convolution analogously uniting both perspectives[29].

1.2 Summary of Contributions

With the exception of [37], all of the convolutional methods have so far focused on generating uniformly sampled configuration space obstacles for all configurations simultaneously[23, 5, 8, 29]. This is asymptotically optimal if one absolutely needs a complete and exact description of the configuration space obstacle, but is overkill for collision detection, since at most a very tiny subset of the whole obstacle will ever be queried. And given that in rigid body dynamics or packaging applications, the number of configuration space obstacles grows quadratically with respect to the number of distinct solids, it is not usually possible to explicitly store an obstacle for each pair of objects, and so it is sensible to look for online methods that support the efficient evaluation of just a single query.

Another missing feature in these methods is the ability to compute a gradient. Part of the reason for this failure can be understood to come from the choice of discontinuous indicator functions as the underlying representation of the solids. Though there exist automatic methods which can accomplish this task for symbolically defined functions[40], in applications based on sampled data (such as this), the problem is much more difficult. These complexities are further compounded by the fact that rigid motion gap functions need to be differentiated over the Riemannian manifold of rotations and translations, $SE(d)$, which is an intrinsically curved, non-Euclidean space.

Our new approach, based on interpolated Fourier expansions of smooth bump functions, solves both problems simultaneously. Like many analytic tests, our method supports arbitrary geometry, regardless of connectivity, topology or smoothness. Moreover, our proposed approximation scheme degrades continuously, and so unlike combinatorial tests which must typically fix some approximation up front, our method allows for the collision test to be evaluated to an arbitrary degree of accuracy, which in the limit converges to an exact test. In addition, we also claim the following more specific advantages,

1. *We give an explicit, analytical formula for the $SE(d)$ gradient of the gap function.* Fast and accurate differentiation is crucial for applying collision detection in optimization problems as already discussed. In principle, this result also extends to other Fourier obstacle methods such as [23, 8, 54].

2. *We introduce an early-miss condition in the evaluation of the Fourier integral to reduce the cost of intersection testing.* This property is novel in that unlike other early-hit narrowphase collision

detection tests which terminate early upon detecting a collision, our Fourier test terminates early in the case of non-intersection. From the perspective of most applications, this situation is preferable since collisions are relatively infrequent events compared to non-collisions.

3. *We derive an analytical expression for the accuracy of our collision test relative to the ground truth configuration space obstacle.* This partially addresses a major omission in previous works where the question of their accuracy was either addressed only empirically or else neglected altogether.

The structure of this paper is as follows; in the next section we discuss some of the basic theoretical issues associated with collision detection before launching into our development of collision detection in the Fourier domain and gradient calculations. Following that, we investigate the discrete representation of these shapes and some numerical problems. The theoretical performance our method is then independently verified using experimental data, illustrating the rate of convergence, accuracy and the associated time complexity. Finally, we conclude by discussing some of the open problems which are asked in the body of this work.

2 Bump Functions for Collision Detection

In this section we shall outline the basic framework of our theory. As discussed earlier, rather than dealing with solids bounded by triangulated meshes or parametric surfaces, we shall instead consider implicit representations, or solids represented by regularized sublevels of smooth functions. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a smooth function, $t \in \mathbb{R}$ and define the *t-sublevel* of f, $V_t(f)$ to be

$$V_t(f) \overset{\text{def}}{\equiv} \kappa\{x \in \mathbb{R}^d : f(x) > t\}. \tag{2}$$

Thus for any smooth f we get a family of solids by applying the regularized sublevel operator for different choices of t. We can actually be a bit more precise about how these sublevels are related by saying the following,

Proposition 1. *For any pair of scalars $t_0 \leq t_1$, $V_{t_1}(f) \subseteq V_{t_0}(f)$ for all $f : \mathbb{R}^d \to \mathbb{R}$.*

We shall make heavy use of this property later on, however for the moment let us continue with the basic exposition. The next natural question to ask is what are the class of solids which can be represented by sublevel sets. The answer as it turns out is as good as one could possibly hope for; given any solid S, there always exists a function $f_S \in C^\infty(\mathbb{R}^d)$ such that $V_0(f_S) = S$[46]. This fact is a basic result from analysis, and is at least theoretically reassuring in that it shows that there are no fundamental limitations on the types of solids we can represent.

The treatment of solids as sublevel sets is an important step towards analytic collision detection and was realized early on by Comba[6]. However, it is also true that regularized Boolean set operations on sublevels of real valued functions can be quite complicated. Numerous approaches have been proposed over the years, such as R-functions[47, 49], distance fields[41], or blobby object modeling[2, 3]. While each of these techniques has their relative merits, in this paper, we shall focus on the last approach, blobby object modeling. Our motivation for this choice is that Boolean set operations on blobby objects have a relatively simple analytic interpretation, which leads to a corresponding analytic interpretation of the collision test.

In blobby object modeling solids are described by sublevels of nonnegative real-valued functions. Here we consider a specialization of this technique where $t = 0$ (as in Eqn. 2). We shall use $\mathbb{R}^+ \overset{\text{def}}{\equiv} \{x \in \mathbb{R} : x \geq 0\}$ to denote the collection of all nonnegative reals. Now given a closed regular set, $S \subset \mathbb{R}^d$, we define a *bump function on S*, $\chi_S : \mathbb{R}^d \to \mathbb{R}^+$, to be a smooth, compactly supported nonnegative function such that

$$\chi_S(x) > 0 \Leftrightarrow x \in \iota S. \tag{3}$$

An immediate consequence of this definition is that

$$V_0(\chi_S) = S.$$

Though the restriction to smooth nonnegative functions may seem quite severe, it is actually no stronger than the elementary requirement of a smooth implicit representation. In fact, given a smooth bounded f_S such that $V_t(f_S) = S$, it is quite easy to construct a bump function, χ_S, directly, for example,

$$\chi_S(x) = \begin{cases} \exp(-(f_S(x) - t)^{-2}) & \text{if } f_S(x) > t \\ 0 & \text{otherwise} \end{cases}.$$

And thus we conclude that the set of all blobby objects is equivalent to the set of all closed regular subsets of $\mathbb{R}^d$.

The collection of all bump functions forms an $\mathbb{R}^+$-algebra under addition and pointwise multiplication. In this algebra, the representation of a solid, S, is essentially non-unique. In fact, given a bump function, χ_S, we can always construct a new bump function by multiplying χ_S with any positive function. Therefore, the collection of all bump functions whose 0-sublevel sets are subsets of S forms an ideal (or cone) in the $\mathbb{R}^+$-algebra, which is generated by χ_S. In later sections we shall discuss the considerations associated with choosing a bump function (or in other words a member of this ideal), but for now we continue with the development of our main theory since these issues are mostly orthogonal.

2.1 Boolean Operations on Bump Functions

As we alluded to earlier, the main advantage of bump functions over general real-valued functions is the existence of simpler expressions for regularized Boolean set operations, which we summarize in the following proposition:

Proposition 2. [2] *For all solids $S, T \subset \mathbb{R}^d$*

$$\begin{aligned} S \cup^* T &= V_0(\chi_S + \chi_T) \\ S \cap^* T &= V_0(\chi_S \chi_T) \end{aligned}$$

In addition to modelling basic set-theoretic structures, the $\mathbb{R}^+$-algebra embedding also preserves all direct limits in the lattice structure of regularized sets (under the containment relation). We shall make use of a more basic version of this property in the following lemma, which allows us to check whether a set is empty or not by integrating over its bump function:

Lemma 1 (The Null-Volume Lemma). *For all solids $S \subset \mathbb{R}^d$,*

$$S \neq \emptyset \iff \int\limits_{\mathbb{R}^d} \chi_S(x) dx > 0$$

One major advantage of working with bump functions (aka an $\mathbb{R}^+$-algebra) instead of volumetric/voxelized representations (aka a Boolean algebra) or distance fields (aka a $(\max, +)$-algebra[39]) is that they faithfully embed into the Hilbert space (or more specifically the C^*-algebra) of Lebesgue measurable complex functions on $\mathbb{R}^d$. Recall that a *Hilbert space* is a complex vector space with a complete inner product. Given $f, g \in L^2(\mathbb{R}^d)$, the *inner product* $\langle \cdot, \cdot \rangle$ is defined such that,

$$\langle f, g \rangle = \int\limits_{\mathbb{R}^d} f(x)\overline{g(x)} dx.$$

Now it should be clear that if f and g are bump functions, then the body of this integral is identical to the intersection formula in Prop. 2. If we then apply the Null-Volume Lemma (Lemma 1), we get our first version of an analytic collision test,

Theorem 1. *For all solids $S, T \subset \mathbb{R}^d$,*

$$\text{collide}(S, T) \iff \langle \chi_S, \chi_T \rangle > 0.$$

Theorem 1 is the basis for our approach to analytic collision, since it translates the interference test into a real-valued integral inequality. A weaker version of this theorem is often used implicitly in

[2] Following standard conventions in solid modeling, we use a superscript $*$ to denote the regularization of an operator, for example, $S \cap^* T \overset{\text{def}}{\equiv} \kappa\iota(S \cap T)$.

volumetric collision detection methods. In the volumetric approach, the bump functions are replaced by *indicator functions*, $\mathbf{1}_S : \mathbb{R}^d \to \mathbb{R}$, which are defined such that for a solid S,

$$\mathbf{1}_S(x) = \left\{ \begin{array}{cc} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{array} \right. .$$

The connection to the volumetric formulation then follows from the observation that the volume of the intersection between two solids is equivalent to the inner product of their indicators[29],

$$\langle \mathbf{1}_S, \mathbf{1}_T \rangle = \text{vol } S \cap T.$$

2.2 Early-Hit Tests and Intersections

While theoretically interesting, the direct application of Theorem 1 does not lead to a very practical collision test. As a first optimization, we observe that if two objects intersect in some subset, then they must intersect globally. This simple idea is the basis behind the 'feasible point' strategy which is often applied in combinatorial methods. The underlying reason that this approach works is summarized in the following proposition,

Proposition 3 (Early-Hit Test)**.** *Let $S, T \subset \mathbb{R}^d$ be solids and let $S' \subseteq S$, $T' \subseteq T$. Then,*

$$\text{collide}(S', T') \implies \text{collide}(S, T)$$

This proposition provides a weaker, but sufficient condition for the collision predicate. We call this type of test an *early-hit* test, since it allows one to quickly terminate collision detection once some smaller subset of the solids is found to collide. Of course this still leaves open the issue of which subsets, $S' \subset S$, $T' \subset T$ one should choose. One simple option is to just select some other solid, $U \subset \mathbb{R}^d$ and then pick $S' = S \cap U$ or $T' = T \cap U$. Doing this has the effect of localizing the collision detection test to the region U, unlike the naïve test which requires a global search over all of $\mathbb{R}^d$.

To construct an analytic version of this test, we introduce the idea of a weighted inner product, which gives us the capability to localize the intersection between two solids. Formally it is defined as follows, given a triple of functions, $f, g, h \in L^2(\mathbb{R})$, we define the *$h$-weighted inner product of f, g* to be,

$$\langle f, g \rangle_h \overset{\text{def}}{\equiv} \int\limits_{\mathbb{R}^d} f(x)\overline{g(x)}h(x)dx. \tag{4}$$

When f, g and h are bump functions, it is easy to see why the weighted inner product localizes the collision test. Let $S, T, U \subseteq \mathbb{R}^d$ be solids, then

$$\langle \chi_S, \chi_T \rangle_{\chi_U} = \int\limits_{\mathbb{R}^d} \chi_S(x)\chi_T(x)\chi_U(x)dx.$$

But by an argument analogous to Thm. 1, we can relate this back to a collision predicate on S, T and U,

$$\text{collide}(S, T \cap U) \Leftrightarrow \langle \chi_S, \chi_T \rangle_{\chi_U} > 0.$$

Thus, we conclude that the ultimate effect of reweighting the inner product is to localize the intersection test to some neighborhood U. We summarize this result in the following theorem,

Theorem 2. *For all solids $S, T, U \subset \mathbb{R}^d$,*

$$\langle \chi_S, \chi_T \rangle_{\chi_U} > 0 \implies \text{collide}(S, T)$$

2.3 Early-Miss Tests and Minkowski Sums

Theorem 2 is essentially an analytic reinterpretation of the early-hit test (Prop. 3). However, from an application's perspective the early-out conditions described in Prop. 3 are somewhat backwards. For example, in a typical configuration in rigid body dynamics, most objects are not colliding and thus the underlying inequality is always 0. But in order to rule out a collision (ie demonstrate that the collision inner product is identically 0), we must completely integrate over the support of at least one of the underlying solids which can potentially be very expensive. In sharp contrast, to resolve a much less likely collision event, it is sufficient to locate only a single point in the interior to demonstrate a proper intersection.

This situation is clearly less than optimal; ideally we would like to have some form of collision detection which favors early rejection instead of early acceptance. One possible basis for such a test to take the dual[3] of Proposition 3. This gives a *early-miss* or necessary condition for the collision predicate to hold,

Proposition 4 (Early-Miss Test)**.** *Let $S, T \subset \mathbb{R}^d$ be solids and let $S' \supseteq S$, $T' \supseteq T$. Then,*

$$\text{collide}(S, T) \implies \text{collide}(S', T')$$

In order to test this condition, we need to somehow construct supersets of the underlying solids, S, T. Continuing by analogy to the early-hit test, where the construction of subsets was given canonically by intersecting with some localized region, we would like to find some corresponding dual method for constructing supersets. To do this, we shall take a cue from the theory of mathematical morphology[48], and construct these sets using morpholigcal dilations, or Minkowski sums. The *Minkowski sum* of two solids, $S, T \subset \mathbb{R}^d$ is a binary operator, $\oplus$, which is defined such that,

$$S \oplus T = \{x + y : x \in S, y \in T\}.$$

The Minkowski sum was introduced by Minkowski in 1903, and has become a fundamental tool in computer vision[48], solid modeling[33] and motion planning[28]. Intuitively, the Minkowski sum 'offsets' one solid by another. We can use this property to construct supersets of solids, as needed in Prop. 4, as follows: Let U be a subset of $\mathbb{R}^d$ which contains the origin, or in other words $\{0\} \subseteq U \subseteq \mathbb{R}^d$; then given a solid S, we construct an offset solid $S' \supseteq S$ according to the following rule,

$$S' = S \oplus^* U.$$

In order to make use of the offset in our current programme of analytic collision detection, we need to formulate some interpretation of the Minkowski sum in terms of bump functions. We shall do this using *convolution*, which is a bilinear operator, $* : L^2(\mathbb{R}^d) \times L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$, defined such that for any pair of functions $f, g \in L^2(\mathbb{R}^d)$,

$$(f * g)(x) = \int\limits_{\mathbb{R}^d} f(y) g(x - y) dy.$$

Also, we denote the *reflection* of a function $f \in L^2(\mathbb{R}^d)$ by $\widetilde{f}$ which is defined such that,

$$\widetilde{f}(x) = f(-x).$$

Combining these two definitions naturally leads to our desired reinterpretation of the Minkowski sum[29],

Proposition 5. *Let S, T be solids with bump functions χ_S, χ_T. Then,*

$$S \oplus^* T = V_0 \left(S * \widetilde{T} \right)$$

We are now ready to state our first analytic version of the early-miss test. Proving this becomes a fairly easy exercise, we just substitute Prop. 5 into Prop. 4, and then apply Theorem 1,

Theorem 3. *Let $S, T \subset \mathbb{R}^d$ and let $\{0\} \subseteq U \subseteq \mathbb{R}^d$. Then,*

$$\langle \chi_S, \chi_T * \chi_U \rangle = 0 \implies \neg\text{collide}(S, T).$$

2.4 Fourier Collisions

At first glance, Theorem 3 seems to have actually made the situation worse. Instead of doing a single integral to evaluate the inner product, we are now forced to take a double integral in order to evaluate the convolution term as well. However, we can simplify this expression substantially by moving to the Fourier domain, which allows us to rewrite this convolution as pointwise multiplication while simultaneously

[3] In the categorical sense; or in other words the same predicate but with the order of all relations reversed.

preserving the inner product structure. This lets us reinterpret the theorem as a weighted innner product, analogous to Theorem 2.

The orthonormal Fourier basis for $\mathbb{R}^d$ are functions of the form $e^{2\pi i \langle \omega, x \rangle}$ where $\omega \in \mathbb{R}^d$. Using this basis, we can rewrite a particular function f using orthogonal projection,

$$f(x) = \int\limits_{\mathbb{R}^d} \left\langle f, e^{2\pi i \langle \omega, x \rangle} \right\rangle e^{2\pi i \langle \omega, x \rangle} d\omega.$$

Splitting this up, we define the change of basis to the Fourier domain as the *Fourier transform* of f, written as

$$\mathcal{F}\left(f\right)(\omega) = \left\langle f(x), e^{2\pi i \langle \omega, x \rangle} \right\rangle = \int\limits_{\mathbb{R}^d} f(x) e^{-2\pi i \langle \omega, x \rangle} dx.$$

We also use the common shorthand that $\widehat{f} = \mathcal{F}\left(f\right)$. The inverse Fourier transform is given by

$$\mathcal{F}^{-1}\left(\widehat{f}\right)(x) = \int\limits_{\mathbb{R}^d} \widehat{f}(\omega) e^{2\pi i \langle \omega, x \rangle} d\omega.$$

Our reason for choosing this particular form is that it gives a unitary transformation which behaves nicely under differentiation. This basic result is summarized in Parseval's theorem, which says that for any integrable f, g,

$$\langle f, g \rangle = \left\langle \widehat{f}, \widehat{g} \right\rangle$$

Combined with Theorem 1, this yields a Fourier domain version of the collision test as an immediate corollary,

Corollary 1. *For solids, $S, T \subset \mathbb{R}^d$,*

$$\text{collide}(S, T) \iff \langle \widehat{\chi_S}, \widehat{\chi_T} \rangle > 0$$

However, we still need to translate the offset solid into the Fourier domain. To do this, we shall make use of two additional facts. The first property is that the Fourier transform translates reflection into conjugation,

$$\widehat{\widetilde{f}}(\omega) = \overline{\widehat{f}(\omega)}.$$

And the second property we shall use is the convolution theorem, which states that,

$$\mathcal{F}\left(f * g\right)(\omega) = \widehat{f}(\omega)\widehat{g}(\omega).$$

These properties combine to give us a Fourier domain interpretation of Theorem 3,

Corollary 2. *For all pairs of solids, $S, T \subset \mathbb{R}^d$, and $\{0\} \subseteq U \subseteq \mathbb{R}^d$,*

$$\langle \widehat{\chi_S}, \widehat{\chi_T} \rangle_{\widehat{\chi_U}} = 0 \implies \neg\text{collide}(S, T)$$

3 Configuration Space Obstacles in $SE(d)$

As we discussed in the introduction, typically collision queries are not made with respect arbitrary fixed solids, but are instead selected from transformed instances of some fixed collection of base shapes. These transformations, or *configurations*, are in turn selected from a group called a *configuration space* which acts on the solids. In this section, we consider a special instance of this problem where the configuration space is the group of rigid motions on $\mathbb{R}^d$; or in other words the the transformations of $\mathbb{R}^d$ that fix the Euclidean distance metric and orientation.

3.1 The Special Euclidean Group

The space of rigid motions forms a Lie group, known as the special Euclidean group or $SE(d)$. $SE(d)$ can be written as the extension of the special orthogonal group, $SO(d)$, over its action on $\mathbb{R}^d$; or more compactly as a semidirect product, $SE(d) \cong SO(d) \ltimes \mathbb{R}^d$. Because of this product structure, elements of $SE(d)$ can be parameterized by a pair consisting of a $d \times d$ rotation matrix and a d-dimensional translation vector. In other words, we can write any such element as a pair $(R, x) \in SE(d)$, where $R \in SO(d)$ and $x \in \mathbb{R}^d$.

Given $(R, x), (R', x') \in SE(d)$, we define their product as follows,

$$(R, x) \left(R', x'\right) = \left(RR', x + Rx'\right).$$

This product is invertible, and has the following inverse operator,

$$(R, x)^{-1} = \left(R^{-1}, -R^{-1}x\right).$$

And finally, the (left) action of $SE(d)$ on $\mathbb{R}^d$ is given by the following rule; for $y \in \mathbb{R}^d$,

$$(R, x)\, y = Ry + x.$$

In our applications, we shall apply this action to solids in order to associate a single configuration (R, x) to a rotated/translated instance of some base solid S, which we shall usually abbreviate as $(R, x)\, S = \{(R, x)\, y : y \in S\}$.

3.2 Gap Functions

Now using these facts, let us investigate the collision predicate for a pair of solids, $S, T \subset \mathbb{R}^d$, undergoing rigid motions $(R_S, x_S), (R_T, x_T)$ respectively. Clearly,

$$\begin{aligned} \text{collide}\left((R_S, x_S)\, S, (R_T, x_T)\, T\right) \quad &\Leftrightarrow \quad (R_S, x_S)\, S \cap^* (R_T, x_T)\, T \neq \emptyset \\ &\Leftrightarrow \quad (R_T, x_T)^{-1} (R_S, x_S)\, S \cap^* T \neq \emptyset. \end{aligned}$$

And so we can reparameterize the collision predicate for a pair of moving solids in terms just one relative transformation, $(R, x) = (R_S, x_S)^{-1} (R_T, x_T)$, leading us to define the *relative collision predicate*,

$$\text{relcollide}\left((R, x), S, T\right) \overset{\text{def}}{\equiv} \text{collide}\left((R, x)^{-1} S, T\right). \tag{5}$$

The locus of all group elements which satisfy Eqn. 5 cuts out a subset of $SE(d)$, which when regularized, is called the *configuration obstacle of* S, T, or just the *obstacle*, which we denote by $O_{S,T} \subset SE(d)$,

$$O_{S,T} = \kappa\iota \left\{(R, x) \in SE(d) : \text{relcollide}\left((R, x), S, T\right)\right\}.$$

Much like a solid, the obstacle can be represented by a bump function on the configuration space, $\chi_{O_{S,T}} : SE(d) \to \mathbb{R}^+$, which we shall use as our analytic extension of the collision predicate, Eqn. 5. We call this new function the *gap function* of the obstacle $O_{S,T}$. The gap function is related to the collision predicate according to the following rule,

$$\text{relcollide}\left((R, x), S, T\right) \Longleftrightarrow \chi_{O_{S,T}}(R, x) > 0. \tag{6}$$

We now turn to the problem of constructing a gap function given bump functions for the underlying solids. A straightforward way to do this is to apply Theorem 1 and the convolution operator. This gives a direct construction for a bump on $O_{S,T}$, given bump functions for S and T,

Theorem 4. *Let $S, T \subset \mathbb{R}^d$ be solids with bump functions χ_S, χ_T respectively. Then there is a gap function, $\chi_{O_{S,T}} : SE(d) \to \mathbb{R}^+$ such that,*

$$\chi_{O_{S,T}}(R, x) = \left(\left(\chi_S \circ R\right) * \widetilde{\chi_T}\right)(x).$$

We can also rewrite Thm. 4 in the Fourier domain. We perform this conversion in anticipation of the sequel, where we shall investigate the problem of differentiating the gap function.

Corollary 3. *Let $S, T \subset \mathbb{R}^d$ be solids with bump functions χ_S, χ_T. Then there is a gap function, $\chi_{O_{S,T}} : SE(d) \to \mathbb{R}^+$ such that,*

$$\chi_{O_{S,T}}(R, x) = \int\limits_{\mathbb{R}^d} \widehat{\chi_S}(R\omega)\overline{\widehat{\chi_T}(\omega)} e^{2\pi i \langle \omega, x \rangle} d\omega.$$

4 Gradients of Gap Functions

Now that we have translated the relative collision predicate into a smooth inequality on the configuration space it can be used as a constraint for the purposes of collision response. As we have discussed previously, the only substantial requirement here is that we are able to differentiate the constraint function – a problem which is fortunately simplified by the Fourier transform in Cor. 3. According to Eqn. 6, two rigid bodies, S, T, collide at the configurations $q_S, q_T \in SE(d)$ if and only if,

$$\chi_{O_{S,T}}\left(q_S^{-1} q_T\right) > 0.$$

Thus, to solve for the collision impulse forces and torques, we need to differentiate this function with respect to q_S, q_T. Unfortunately, this is complicated by the fact that $SE(d)$ (for $d > 2$) can not be covered by a single chart, and as a result we are forced to resort to the techniques of differential geometry in order to correctly differentiate gap functions on $SE(d)$.

4.1 Review of Notation and Conventions

We introduce our conventions by quickly recalling some basic definitions. Given a differential manifold M and point $x \in M$, we write the *tangent space at x in M* as $T_x M$. Now let M, N be differential manifolds and $f : M \to N$ be a diffeomorphism. Then, we denote the *gradient*[4] of f at $x \in M$ by $\frac{d}{dx} f(x) : T_x M \to T_{f(x)} N$. This function is defined for any tangent vector $\eta \in T_x M$ along a smooth curve $\gamma : \mathbb{R} \to M$ with $\gamma(0) = x$, and $\eta = \frac{d}{ds}\gamma(s)\big|_{s=0}$, such that:

$$\left(\frac{d}{dx} f(x)\right)(\eta) = \frac{d}{ds} f(\gamma(s))\bigg|_{s=0}. \tag{7}$$

Given a collection of manifolds, M, N, P and a sequence of diffeomorphisms, $M \xrightarrow{f} N \xrightarrow{g} P$, the *chain rule* states that the gradient of the diffeomorphism $M \xrightarrow{g \circ f} P$ (which is of type $\frac{d}{dx} g(f(x)) : T_x M \to T_{g(f(x))} P$) is given by the following; for all $x \in M, \eta \in T_x M$

$$\left(\frac{d}{dx} g(f(x))\right)(\eta) = \left(\frac{d}{df(x)} g(f(x))\right)\left(\left(\frac{d}{dx} f(x)\right)(\eta)\right). \tag{8}$$

4.2 The Case of $SE(d)$

Returning to $SE(d)$; like all Lie groups, the tangent space at any point in $SE(d)$ is given by a left (or right) transformation of the tangent space at the identity, which is the Lie algebra for the group. For example, in the case of $SO(d)$, observe that for any curve $R(t) : \mathbb{R} \to SO(d)$, we can rewrite a tangent vector in $T_{R(t)} SO(d)$ at any point t along the curve as follows,

$$\frac{dR(t)}{dt} = R(t)\left(R(t)^{-1}\frac{dR(t)}{dt}\right) = R(t)\Omega(t),$$

Where we make the substitution, $\left(R(t)^{-1}\frac{dR(t)}{dt}\right) \mapsto \Omega(t)$. It can be shown that $\Omega(t)$ is skew-symmetric and is also a tangent vector at the identity[5], and is thus contained in the Lie algebra for $SO(d)$, which we write as $\mathfrak{so}(d)$ (or in otherwords the space of all $d \times d$ real skew-symmetric matrices).

For $SE(d)$, its Lie algebra, $\mathfrak{se}(d)$, splits into a direct sum of rotational and translational components, ie

$$\mathfrak{se}(d) \cong \mathfrak{so}(d) \oplus \mathbb{R}^d \cong T_{id} SE(d).$$

Therefore, we can express any tangent vector in the Lie algebra as a tuple, $(\Omega, \tau) \in \mathfrak{se}(d)$, where $\Omega \in \mathfrak{so}(d)$ and $\tau \in \mathbb{R}^d$. Moreover, following an analogous argument to the $SO(d)$ case this extends to tangent vectors at any point in the group $SE(d)$ by left translation; in other words for a tangent vector $\eta \in T_{(R,x)} SE(d)$, there exists a pair $(\Omega, \tau) \in \mathfrak{se}(d)$ such that $\eta = (R\Omega, \tau)$, and $R\Omega \in T_R SO(d)$ and $\tau \in T_x \mathbb{R}^d \cong \mathbb{R}^d$.

[4] Also known as the *(total) derivative*, *pushforward* or *differential* in other sources.

[5] This fact can be verified by differentiating $R(t)R(t)^{-1} = R(t)R(t)^T = id_{SO(3)}$ at $t = 0$ in the case where $R(0) = id_{SO(3)}$.

Using this decomposition and the linearity of the gradient operator, we can find an explicit form for the gradients of a smooth function on $SE(d)$: Let M be a differential manifold, $q = (R, x) \in SE(d)$ and $\eta = (R\Omega, \tau) \in T_q SE(d)$, then for all C^1 functions $f : SE(d) \to M$,

$$\left(\frac{d}{dq} f(q)\right)(\eta) = \left(\frac{d}{dx} f(R, x)\right)(\tau) + \left(\frac{d}{dR} f(R, x)\right)(R\Omega).$$

But observe that we can expand these derivatives using Eqn. 7 as follows, giving us our desired component-wise expansion,

$$\left(\frac{d}{dx} f(R, x)\right)(\tau) = \left.\frac{d}{ds} f(R, x + s\tau)\right|_{s=0}, \tag{9}$$

$$\left(\frac{d}{dR} f(R, x)\right)(R\Omega) = \left.\frac{d}{ds} f(R \exp(s\Omega), x)\right|_{s=0}. \tag{10}$$

4.3 Gradients of Gap Functions

Putting this together, we can now calculate the gradient of the gap function. One great advantage of working in the Fourier domain is that translational differentiation can be replaced by a multiplier. This result is exactly the derivative; no stencils or other approximations are necessary. And because the multiplier commutes with the convolution of the shapes, it can either be precomputed or calculated in parallel to the value of the gap function. These results are summarized in Theorem 5,

Theorem 5. *Let $S, T \subset \mathbb{R}^d$ be solids, $(R, x) \in SE(d)$ and $(\Omega, \tau) \in \mathfrak{se}(d)$. Then,*

$$\left(\frac{d}{dx} \chi_{O_{S,T}}(R, x)\right)(\tau) = \int_{\mathbb{R}^d} \frac{i}{2\pi} \langle \omega, \tau \rangle \widehat{\chi_S}(R\omega) \overline{\widehat{\chi_T}(\omega)} e^{2\pi i \langle \omega, x \rangle} d\omega \tag{11}$$

$$\left(\frac{d}{dR} \chi_{O_{S,T}}(R, x)\right)(R\Omega) = \int_{\mathbb{R}^d} \left(\left(\frac{d}{dR\omega} \widehat{\chi_S}(R\omega)\right)(R\Omega\omega)\right) \overline{\widehat{\chi_T}(\omega)} e^{2\pi i \langle \omega, x \rangle} d\omega. \tag{12}$$

The only slightly complicated term in these equations is the $\left(\frac{d}{dR\omega} \widehat{\chi_S}(R\omega)\right)(R\Omega\omega)$ which appears in the rotational gradient. To understand this subexpression, we need to make use of two key properties of the Fourier transform and gradient. First, we observe that we can apply the dual of the multiplier formula in the spatial domain to get the following expression,

$$\left(\frac{d}{d\omega} \widehat{\chi_S}(\omega)\right)(\tau) = \mathcal{F}\left(-\frac{i}{2\pi} \langle x, \tau \rangle \chi_S(x)\right).$$

And second, since both the derivative and Fourier transform are linear, we can split the above inner product up into components, and thus we need only calculate the above multiplier for each of the d-basis directions.

Theorem 5 gives an explicit form for calculating the gradeint of the gap function with respect to relative transformations of solids. However, for many applications (such as rigid body dynamics), it may be more useful to compute the absolute gradient (or in other words to differentiate the constraint function in terms of q_S, q_T). To do this one, only needs to apply the chain rule for differential manifolds (Eqn. 8). This is a basic, though somewhat tedious calculation. To save the reader some headaches, we just summarize the results in the following corollary,

Corollary 4. *Let $S, T \subset \mathbb{R}^d$ be solids as before, $(R, x) = (R_S, x_S)^{-1} (R_T, x_T) \in SE(d)$ and $(\Omega, \tau) \in \mathfrak{se}(d)$. Then,*

$$\begin{aligned}
\left(\frac{d}{dx_S} \chi_{O_{S,T}}(R, x)\right)(\tau) &= \left(\frac{d}{dx} \chi_{O_{S,T}}(R, x)\right)(-R_S^T \tau) \\
\left(\frac{d}{dR_S} \chi_{O_{S,T}}(R, x)\right)(R_S \Omega) &= \left(\frac{d}{dR} \chi_{O_{S,T}}(R, x)\right)(-\Omega R_S^T R_T) + \left(\frac{d}{dx} \chi_{O_{S,T}}(R, x)\right)(\Omega R_S^T (x_S - x_T)) \\
\left(\frac{d}{dx_T} \chi_{O_{S,T}}(R, x)\right)(\tau) &= \left(\frac{d}{dx} \chi_{O_{S,T}}(R, x)\right)(R_S^T \tau) \\
\left(\frac{d}{dR_T} \chi_{O_{S,T}}(R, x)\right)(R_T \Omega) &= \left(\frac{d}{dR} \chi_{O_{S,T}}(R, x)\right)(R_S^T R_T \Omega)
\end{aligned}$$

The choice of whether to use the form in Thm. 5 or Cor. 4 largely depends on the application. In the case of path planning, for example, one usually considers the environment to be static and so all that matters is the relative configuration of the moving robot, and thus the form in Thm 5 would be preferred. On the other hand, in rigid body dynamics or nesting problems one has many moving objects and so the gradient of the constraints need to be calculated with respect to the coordinates of each object. In this case Cor. 4 is more likely to be useful.

5 Numerical Analysis

To summarize, we have given some formulas for constructing (Thm. 4 and Cor. 3) and differentiating (Thm. 5 and Cor. 4) a collision gap function for a pair of rigidly moving solids starting from their bump functions. But to make these results useful, we need an efficient process for computing a finite approximation of the gap function, and as a secondary issue a finite representation of our bump functions. To simplify the discussion, we will for the moment disregard the effects of rotation and focus exclusively on the translational component of the obstacle. We will return to rotations in the next section once we have settled this more fundamental issue. Because of this restriction we may regard all of our configuration space obstacles as being subsets of $\mathbb{R}^d$ and our gap functions as being elements of $L^2(\mathbb{R}^d)$.

Our general strategy is to write the gap function as a low pass filtered version of some exact gap function. This parallels the process of discretizing a periodic function by low-pass filtering, and can be thought of as a set-theoretic analogue of resampling/anti-aliasing. In order to make this work, we need to show that given two solids it is possible to recover an approximation of their Minkowski sum (up to some error) given their discretizations. In forming this approximation, we identify three variables which determine its characteristics,

1. n - The number of linear samples.
2. ϵ - The sampling period.
3. L - The maximum linear dimension of the solids.

The first of these variables, n, is just the number of sample points along each linear dimension, or alternatively we shall use n^d independent variables to specify a bump function. We shall assume that all of our solids/bump functions are encoded to the same accuracy using the same number of samples. Ultimately, the time and storage complexity of our proposed algorithms will be bounded asymptotically by functions of n, and so our first goal is to find some relation between this parameter and the overall accuracy of the approximation. To do this, we shall assume that the bump functions are effectively sampled at some uniform interval at a rate of $\frac{1}{\epsilon}$.

Given n and ϵ, we define the scalar, $L = n\epsilon$, which is the fundamental length scale of all objects. To understand the meaning of L observe that because any finite collection of solids is bounded, it is sufficient to discretize our underlying geometry within a cube of some fixed size. Therefore, we shall assume without loss of generality that all solids are contained in the box $(-\frac{L}{4}, \frac{L}{4})^d$. The reason for the factor of $\frac{1}{2}$ is that if we want to represent the configuration space obstacle of two solids as a subset of this interval, then the interval needs to be at least twice as large as the largest object in each dimension.

5.1 Hausdorff Distance

Our next goal is to demonstrate that the overall accuracy of our approximation scales with n in some nice way. Showing this is the case appears to be highly non-trivial, and is the primary problem we shall be occupied with for the remainder of this section. Because we are dealing with solids – not functions – we need to be a bit careful with how we measure the accuracy of our approximate configuration space obstacle. One naïve choice for such a metric would be the L^p distance of our discretized gap function from some ground truth gap function. While in the limiting case where ϵ goes to 0 this ensures that the resulting sublevel sets will be exact, in practice it is not a very useful tool for estimating error. Even ignoring the basic fact that a ground truth gap function is non-unique, there is still the bigger problem that this convergence is not uniform in a set-theoretic sense (where we consider the convergence at each point in the obstacle). In particular, one could add a number of tiny oscillations to the gap function which would greatly distort the sublevel set, while only introducing a very small change in the overall L^p distance, as illustrated in Fig. 1. Thus, we are forced to conclude that knowing a bound on the L^p

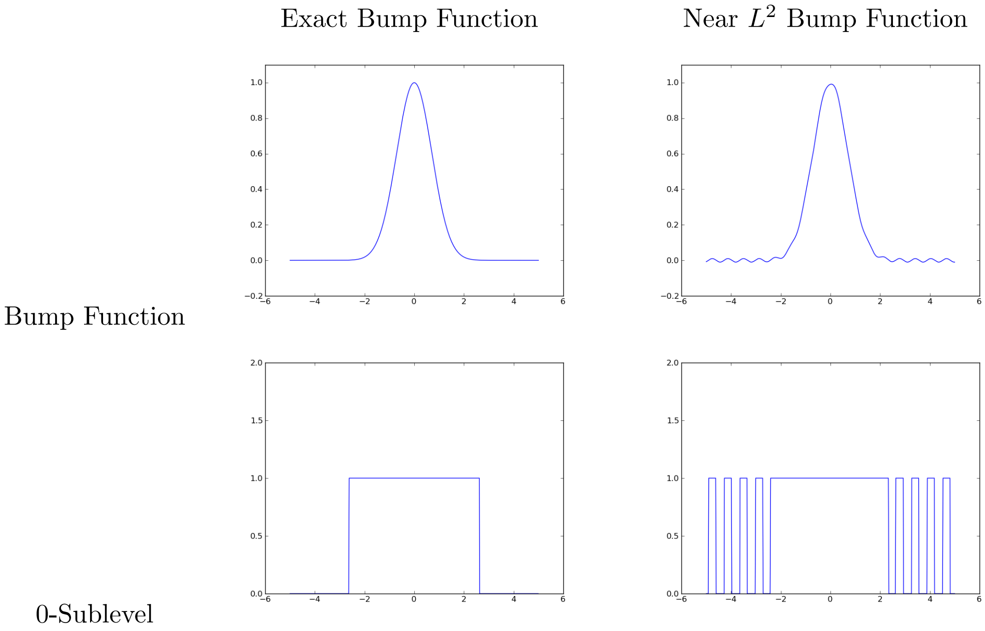


Figure 1: Illustrating the non-uniform convergence of the solid in the L^p norm. Small deviations near the 0-level of the function have little effect on the L^p norm, but can have a huge influence on the structure of the 0-sublevel, and consequently the Hausdorff distance. In the upper left figure, we have some exact 1D bump function, which if we take the 0-sublevel gives the line segment as indicated in the figure below. However, if we add a small oscillatory term to the function, the overall L^p error changes very slightly while the reconstructed 0-sublevel set is completely different.

metric tells very little about the overall convergence of the configuration space obstacle, except in the limiting case where the error is 0.

Because of these issues we are motivated to look for a more geometric notion of distance for solids, and to this end we propose the Hausdorff metric. Recall, given a pair of solids $S, T \subset \mathbb{R}^d$, define the *left L^p Hausdorff distance* from S to T as,

$$d_h^p\left(S,T\right) \overset{\text{def}}{\equiv} \sup_{x\ \ inS} \inf_{y \in T} \left\|x-y\right\|_p ,$$

where $\left\|\cdot\right\|_p$ denotes the L^p norm on $\mathbb{R}^d$. The choice of L^p vs. L^q in the Hausdorff distance is somewhat arbitrary for low dimensions, since for any $p, q > 0$,

$$d_h^q\left(S,T\right) \leq d \cdot d_h^p\left(S,T\right).$$

As a result, we shall simplify the discussion by fixing $p = \infty$. The left Hausdorff distance is not properly a metric since it is not symmetric. However, this is easy to fix by just taking the max over both permutations. Doing this leads to the symmetric *Hausdorff distance*

$$d_H\left(S,T\right) \overset{\text{def}}{\equiv} \max\left(d_h^\infty\left(S,T\right), d_h^\infty\left(T,S\right)\right).$$

We can relate the Hausdorff distance back to the Minkowski sum using offseting. The r-offset of a solid is defined by Minkowski summing with an L^p ball, B_r^p, with radius r,

$$B_r^p \overset{\text{def}}{\equiv} \{x \in \mathbb{R}^d : \left\|x\right\|_p \leq r\}.$$

This relation is summarized in the following theorem,

Proposition 6. *Given $S, T \subset \mathbb{R}^d$ and $\epsilon \geq 0$, then,*

$$\begin{array}{rcl} S & \subseteq & T \oplus B_\epsilon^\infty \\ T & \subseteq & S \oplus B_\epsilon^\infty \end{array}$$

if and only if,

$$d_H\left(S,T\right) \leq \epsilon.$$

It is a well known fact that the Hausdorff distance forms a metric on the space of solids, and thus endows it with a notion of convergence for infinite series. Another consequence is that two solids are equal if and only if their Hausdorff distance is exactly zero. More importantly, unlike metric spaces of functions such as $L^p(\mathbb{R}^d)$, the Hausdorff metric space of solids is endowed with a notion of geometric distance that is directly relevant to comparing the accuracy of our collision test, since it can be used to directly measure the distance between two obstacles.

5.2 Sinc Interpolation

Equipped with a suitable notion of distance for solids, we now turn to the problem of discretization. For the reasons described in the earlier sections, if we wish to compute derivatives or convolutions of bump functions, it is advantageous to work in the frequency domain. As a result, instead of discretizing our bump functions spatially, we shall represent them by a finite sum of uniformly spaced Fourier coefficients. In other words, given a finite d-dimensional sequence $\{Y_k\}_{k \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d}$, we say a function f is the Frequency interpolation of Y if,

$$\widehat{f}(\omega) = \sum_{k \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d} Y_k \delta\left(\omega - \frac{2\pi}{L}k\right). \tag{13}$$

And so,

$$f(x) = \sum_{j \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d} \left(\sum_{k \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d} Y_k e^{\frac{2\pi i}{L}\langle x - \epsilon j, k\rangle} \right). \tag{14}$$

Splitting up the terms on the right, we can then rewrite f as follows,

$$f(x) = \sum_{j \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d} X_j C_j(x). \tag{15}$$

Where the functions C_j are the *(Whittaker) cardinal functions*[6], which we index by the vector $j \in \mathbb{Z}^d \cap \left(-\frac{n}{2}, \frac{n}{2}\right)^d$,

$$C_j(x) = \prod_{\substack{k \in \mathbb{N}^d \\ |k|=1}} \frac{\sin\left(\frac{\pi}{\epsilon}\langle k, x - \epsilon j\rangle\right)}{n \sin\left(\frac{\pi}{n\epsilon}\langle k, x - \epsilon j\rangle\right)}. \tag{16}$$

And the similarly indexed coefficients $\{X_j\}$ are defined in terms of $\{Y_k\}$ as follows,

$$X_j = \sum_{k \in \left(-\frac{n}{2}, \frac{n}{2}\right)^d \cap \mathbb{Z}^d} Y_k e^{\frac{2\pi i}{n}\langle j,k\rangle}. \tag{17}$$

The cardinal functions have a number of nice properties, such as the fact that they are closed under convolution, or in other words for all $j, k \in \mathbb{Z}^d$,

$$C_j * C_k = C_{j+k}, \tag{18}$$

where the addition is cyclic modulo n per component. Another useful fact is that involution inverts the index,

$$\widetilde{C_j} = C_{-j}. \tag{19}$$

Together, these results imply that the convolution of any two sinc interpolated functions is also a sinc interpolant. Moreover, if f not a sinc interpolant, convolving f with C_0 will yield a corresponding 'discretized' version of f. Put another way, the sapn of the cardinal functions forms an ideal in the convolution algebra $\mathbb{C}[\mathbb{R}^d]$ generated by C_0; and so we keep with the tradition of calling sinc interpolation *ideal filtering* (and not because it particularly nice for our application).

5.3 Reconstructing Obstacles

Ideal filtering is an inevitable consequence of discretization in the frequency domain. However, from the perspective of collision detection it also has a number of unfortunate consequences; the most troubling of which is that C_0 is not a bump function (since it takes on negative values), and so it is not necessarily the case that a given bump function will remain nonnegative after ideal filtering.

To rectify this issue, we must show that inspite of the oscillations introduced by sinc interpolation, it is still possible to recover an approximate sublevel set from an ideal filtered bump function (in the Hausdorff sense). While there exist numerous results which study the effect of the L^p error on sinc interpolation as well as various strategies for minimizing it[21, 38], there appear to be very few resources describing techniques for estimating the Hausdorff error of reconstructed sublevel sets.

In this section, we shall outline one possible strategy to obtain a reconstruction with a provably (conservative) bounded Hausdorff error. To do this, we propose the following construction for an approximate bump function. Given a solid S and a scalar $x_0 > 0$, we define the x_0-discretized bump function of S to be the sinc interpolation of its x_0-offset indicator. Or in other words,

$$\Phi_S \overset{\text{def}}{\equiv} \mathbf{1}_{S \oplus^* B^\infty_{x_0}} * C_0. \tag{20}$$

Now given a pair of solids, S, T, by Eqn. 18 the convolution of their discretized bump functions is as follows,

$$\Phi_S * \widetilde{\Phi_T} = \mathbf{1}_{S \oplus^* B^\infty_{x_0}} * \widetilde{\mathbf{1}_{T \oplus^* B^\infty_{x_0}}} * C_0. \tag{21}$$

Our goal will be to recover an approximation of the configuration space obstacle, $O_{S,T}$ from this function such that the Hausdorff distance between this approximation and the exact obstacle decays asymptotically in the number of samples, n. To do this, we shall make use of the following lemma,

[6] C_0 is also known as the *Dirichlet kernel* or *aliased sinc function*.

Lemma 2. *Given a solid $S \subseteq \mathbb{R}^d$, $f \in C^\infty(\mathbb{R}^d)$ and $\upsilon \geq 0$, define*

$$\begin{aligned} t_0 &= \sup_{x \in \neg(S \oplus^* B^\infty_\upsilon)} f(x) \\ t_1 &= \inf_{x \in \iota S} f(x) \end{aligned}$$

Then, for all $t_0 \leq t < t_1$,

$$d_H\left(V_t\left(f\right), S\right) \leq \upsilon.$$

Proof. We shall prove the bound using two sub-parts:

1. $S \subseteq \lim_{\epsilon \to 0^+} V_{t_1 - \epsilon}\left(f\right)$

   Since both sets are regularized, it suffices to show that $\iota S \subseteq \iota V_{t_1}\left(f\right)$. Suppose by contradiction that $\iota S \not\subseteq \iota V_{t_1 - \epsilon}\left(f\right)$ for some $\epsilon > 0$; then there exists some $x \in \iota S$ with $x \notin \iota V_{t_1}\left(f\right)$. However, this implies that $f(x) < t_1 - \epsilon$, which is a contradiction for any $\epsilon > 0$.

2. $V_{t_0}\left(f\right) \subseteq S \oplus^* B^\infty_\upsilon$

   Again, let us suppose that $V_{t_0}\left(f\right) \not\subseteq S \oplus^* B^\infty_\upsilon$. Since the sets are both regular, this then means that there must be some $x \in \neg(S \oplus^* B^\infty_\upsilon)$ such that $f(x) > L$. However, this is a contradiction and so the containment must hold.

Now to combine these results, we apply Prop. 1 to obtain the sequence of containment relations for any t such that $t_0 \leq t < t_1$,

$$S \subseteq V_{t_0}\left(f\right) \subseteq V_t\left(f\right) \subseteq \lim_{\epsilon \to 0^+} V_{t_1 - \epsilon}\left(f\right) \subseteq S \oplus B^\infty_\upsilon.$$

Which then implies that,

$$S \subseteq V_t\left(f\right) \subseteq S \oplus B^\infty_\upsilon.$$

But by Prop. 6, this induces a bound on the Hausdorff distance,

$$d_H\left(V_t\left(f\right), S\right) \leq \upsilon.$$

□

Now it could happen that the interval $[t_0, t_1)$ might be empty, in which case there would be no feasible strategy for recovering an approximate sublevel set for S. Thus, we have reformulated our goal into the following problem: Given an υ which decays in n, find a pair of upper/lower bounds on t_0 / t_1 respectively such that the above interval is non-empty. Unfortunately, using our (admittedly naive) direct techniques, we were unable to obtain such a pair of bounds for Eqn. 21 directly. An intuitive explanation for why our analysis fails is that the extraneous 'ringing' artefacts caused by sinc interpolation (otherwise known as the Gibbs phenomenon) do not nearly fast enough.

To fix this problem, we shall 'smooth out' some of the negative oscillations induced by the sinc kernel by convolving with an *apodization kernel*,[7] $A : \mathbb{R}^d \to \mathbb{C}$. In other words, we shall be investigating sublevel sets of the following function,

$$V_{S,T} = \Phi_S * \widetilde{\Phi_T} * A = \left(\mathbf{1}_{S \oplus B^\infty_{x_0}} * \widetilde{\mathbf{1}_{T \oplus B^\infty_{x_0}}}\right) * \left(C_0 * A\right). \tag{22}$$

We call the terms grouped on the right the *instrument function* for the apodization kernel, which is given by,

$$K = C_0 * A.$$

For convenience, we shall also define the quantity $H_{S,T}$ to be,

$$H_{S,T} = \mathbf{1}_{S \oplus B^\infty_{x_0}} * \widetilde{\mathbf{1}_{T \oplus B^\infty_{x_0}}}.$$

And so,

$$V_{S,T} = H_{S,T} * K.$$

[7] Alternatively, convolving with an apodization kernel is equivalent to multiplying by a *window function* in the Fourier domain.

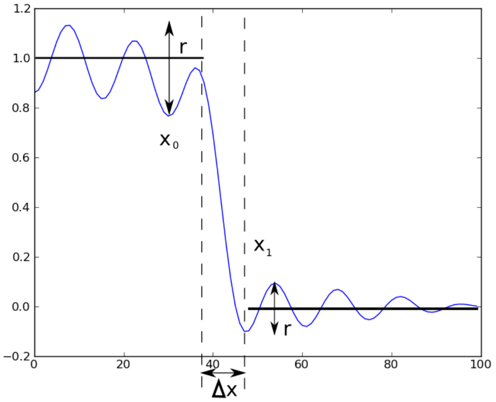


Figure 2: Anatomy of an instrument function.

Ultimately our bound for the Hausdorff error is determined by the shape of the instrument function K. Following the conventions in digital signal processing, we characterize K in terms of three factors: its *pass-band*, x_0; its *stop-band* x_1; and its *ripple level*, r. The basic relation between these parameters and K is illustrated in Fig. 2 and given as follows:

1. For all $x \in B^{\infty}_{x_0}$,
$$|K(x) - 1| \leq r$$
2. For all $x \in [-L/2, L/2]^d \setminus B^{\infty}_{x_1}$,
$$|K(x)| \leq r.$$

The gap between x_0 and x_1 is known as *transition width*, denoted by $\Delta x = x_1 - x_0$. In the design of digital filters, there is a tradeoff between minimizing Δx and r, which we shall investigate later. It is also clear that given a kernel which satisifies this condition, shifting K by r gives a non-negative instrument function $K' = r + K$, such that $K'(x) \geq 1$ for all $x \in B^{\infty}_{x_0}$. We shall use these bounds on the modified K' to derive bounds for t_0 and t_1 which minimize the Hausdorff error according to Lemma 2. Starting with t_1, we get the following result:

Lemma 3. *Suppose K is nonnegative and that for all $x \in B^{\infty}_{x_0}$, $K(x) \geq 1$. Then,*

$$(2x_0)^{2d} \leq \inf_{\iota(S \oplus T)} V_{S,T}$$

Proof. Each point in S and T is contained in a ball of radius x_0 in their offset, and so the value of $H_{S,T}$ at a point in $O_{S,T} \oplus B^{\infty}_{x_0}$ must be at least $\mu(B^{\infty}_{x_0})$. Thus, for all $x \in O_{S,T} \oplus B^{\infty}_{x_0}$, $H_{S,T}(x) \geq (2x_0)^d$. Now if we fix a $x \in O_{S,T}$ and examine $H_{S,T} \star K$; by the nonnegativity K and our lower bound, β_0,

$$V_{S,T}(x) \geq (2x_0)^d \int\limits_{B^{\infty}_{x_0}} K(y) dy \geq (2x_0)^{2d}.$$

□

To get a bound for t_0, we proceed analogously:

Lemma 4. *Suppose that for all $x \in [-L/2, L/2]^d \setminus B^{\infty}_{x_1}$, $K(x) \leq 2r$. Then,*

$$L^{2d} r \geq \sup_{\neg(O_{S,T} \oplus B^{\infty}_{2x_0 + x_1})} V_{S,T}$$

Proof. First, we note that for all $x \in [-L/2, L/2]$; $(H_{S,T})(x) \leq \min(\mu(S \oplus B^{\infty}_{x_0}), \mu(T \oplus B^{\infty}_{x_0})) \leq L^d$. Second, we observe that for all $x \notin S \oplus T \oplus B^{\infty}_{2x_0}$, $H_{S,T} = 0$. Now fix a point $x \notin O_{S,T} \oplus B^{\infty}_{2x_0+x_1}$; then

$$V_{S,T}(x) \leq L^d \int\limits_{[-L/2,L/2]^d \setminus B^{\infty}_{x_1}} K(y)dy \leq 2L^{2d}r$$

□

Now combining these results, if we are given a particular transition width and ripple level, we can compute a pass-band x_0 and t parameter which gives an optimal bound for the Hausdorff error according to Lemma 2. This is summarized in the following theorem,

Theorem 6. *Given a ripple level r and transition width Δx; fix the pass-band $x_0 = \frac{L}{2}(2r)^{\frac{1}{2d}}$. Now let K be a non-negative instrument function such that,*

1. *For all $x \in B^{\infty}_{x_0}$:*
$$1 \leq K(x).$$
2. *For all $x \in [-L/2, L/2]^d \setminus B^{\infty}_{x_0+\Delta x}$:*
$$0 \leq K(x) \leq 2r.$$

Then,

$$d_H\left(V_{L^{2d}r}\left(V_{S,T}\right), O_{S,T}\right) \leq \frac{3L}{2}(2r)^{\frac{1}{2d}} + \Delta x.$$

Proof. Lemmas 2, 3 and 4, if we pick $x_1 = x_0 + \Delta x$, and if

$$2L^{2d}r \leq (2x_0)^{2d}$$

Then,

$$d_H\left(V_{L^{2d}r}\left(V_{S,T}\right), O_{S,T}\right) \leq 3x_0 + \Delta x.$$

However, the Hausdorff distance is linear in x_0, and so the minimal value for x_0 is simply given by solving the equation,

$$2L^{2d}r = (2x_0)^{2d}.$$

Taking $2d^{th}$ roots of both sides gives a minimal value for x_0, and substituting into the Hausdorff bound completes the proof. □

5.4 Optimal Instrument Functions

To the best of our knowledge, no closed form is yet known for the structure of an optimal instrument function. However, given a particular n, it is always possible to find such an arbitrarily close numerical approximation using the Parks-McClellan algorithm[32]. The asymptotic properties of such windows were studied in detail by Shen and Strang[50], where they rigorously showed that as the number of samples n goes to infinity, the minimal transition width, Δx, and minimal the ripple factor, r, are related according to the following rule:

$$n \approx \frac{-L\log(-r\log(r))}{\Delta x}. \tag{23}$$

Using this relation as a constraint, we can solve for the optimal value of r and Δx in Thm 6:

Corollary 5. *For any sufficiently large n and $x_0 \leq \frac{L}{4}$, there exists an instrument function K, such that,*

$$\begin{aligned} r &\in O\left(\frac{1}{n}\right) \\ \Delta x &\in O\left(\frac{L\log(n)}{n}\right) \end{aligned}$$

And so reconstruction by K gives an $O(\epsilon\log(n))$ approximation.

As a result, we have shown that it is possible to reconstruct the Minkowski sumd via Frequency domain interpolation up to a Hausdorff error of at most $O(\epsilon\log(n))$, thus giving our desired asymptotic error bound. It is of course not clear that this bound is tight; the obvious lower bound for the Hausdorff error is $\Omega(\epsilon)$, and based on experimental evidence we conjecture that a sharper analysis could yield such a lower bound.

6 Implementation Issues

6.1 Practical Cutoff Estimation

With the main theoretical question of asymptotic performance now resolved, we turn to the problem of how to actually construct such a kernel in practice. One simple method is to take an existing finite impulse response low-pass filter (for example one of the Butterworth, Chebyshev or elliptic kernels), and then a posteriori calculate a cut-off threshold and minimal error bound. Using this strategy one can then do a parameter sweep or gradient descent on the parameters for the kernel in order to find a good candidate kernel from the space of all possible kernels.

However, to make this work we need to be able to find the cutoff parameter and Hausdorff error associated to an arbitrary instrument function K. To do this, we use an adaptation of our previous results for estimating t_0 and t_1 in terms of x_0 and x_1 respectively. Effectively, we may weaken Lemmas 3, 4 to get the following bounds for t_0, t_1 in terms of x_0, x_1,

$$\begin{aligned} t_0 &\geq \min_{x \in B^\infty_{x_0}} \left(\mathbf{1}_{B^\infty_{x_0}} * \mathbf{1}_{B^\infty_{x_0}} * K \right)(x) \\ t_1 &\leq \max_{x \in B^\infty_{L/2} \setminus B^\infty_{L/2 - x_1}} \left(\mathbf{1}_{B^\infty_{L/4 - x_1}} * \mathbf{1}_{B^\infty_{L/4 - x_1}} * K \right)(x) \end{aligned}$$

This can then be further simplified to give the following bounds,

$$t_0 \geq (2x_0)^d \int\limits_{B^\infty_{x_0}} K(x)dx \tag{24}$$

$$t_1 \leq \left(\frac{L}{2} - x_1 \right)^d \int\limits_{B^\infty_{L/2} \setminus B^\infty_{x_1}} K(x)dx \tag{25}$$

And so we can rephrase the problem of cutoff estimation in terms of the following optimization problem:

$$\begin{array}{rc} \text{Minimize:} & 2x_0 + x_1 \\ \text{Subject to:} & t_0 \geq t_1 \end{array} \tag{26}$$

To solve this problem, we shall take the bounds in Eqns. 24, 25 as an equality and make the additional assumption that K is a symmetric, non-negative, separable function. In other words, assume there exists some $w : \mathbb{R}^+ \to \mathbb{R}^+$ such that

$$K(x) = \prod_{j=1}^{d} w(|x_j|) \tag{27}$$

Where x_j is the j^{th} component of x. We may now rewrite Eqns. 24, 25 as follows,

$$\begin{aligned} t_0 &\geq \left(4x_0 \int\limits_0^{x_0} w(t)dt \right)^d \\ t_1 &\leq 2^d \left(\frac{L}{2} - x_1 \right)^d \left(\left(\int\limits_0^{L/2} w(t)dt \right)^d - \left(\int\limits_0^{x_1} w(t)dt \right)^d \right) \end{aligned}$$

However, we observe that both of these quantities are dependent on the integral of W, and so we define, $W : \mathbb{R}^+ \to \mathbb{R}^+$ to be,

$$W(x) = \int\limits_0^{x_0} w(t)dt$$

Which then gives the following simplified form of the above bounds,

$$\begin{aligned} t_0 &\geq (4x_0 W(x_0))^d \\ t_1 &\leq 2^d \left(\frac{L}{2} - x_1\right)^d \left(W(L/2)^d - W(x_1)^d\right) \end{aligned}$$

Now we observe that since w is nonnegative, W is monotone and so is our bound for t_0 (with respect to x_0). Therefore, if we are given a bound for t_1, then we can always find an optimal choice for x_0 via a binary search. Thus, to find a good approximation to the optimal cutoff parameter, we adopt the following simple strategy: we brute force search over all values of x_1, which yields our bound for t, then binary search to find the optimal x_0 and Hausdorff error.

The accuracy of this method is determined by how well we compute W. To do this efficiently, we propose using a multiplier as well as the FFT to densely sample W on a regular grid. To do this, we will use the fact that $\widehat{w}$ has finite support and construct W as follows,

$$W(x) = x\widehat{w}(0) + \mathcal{F}\left(\frac{2\pi i}{\omega}\left(1 - \delta(\omega)\right)\widehat{w}(\omega)\right).$$

The accuracy of this approach will be determined by the resolution at which the above Fourier transform is evaluated. Now using this, we can write the following algorithm for solving for the minimal Hausdorff error to within some accuracy h. Here is a simple Python implementation using the SciPy/NumPy libraries,

```
'''
SciPy code for calculating optimal cutoff parameter, Hausdorff error and pass-band radii
  what is a 1D numpy array containing the Fourier coefficients of the kernel
  L is the length scale
  d is the dimension
  h is the accuracy to which the error will be estimated
'''
def calculate_cutoff(what, L, d, h):
  #Generate x coordinates
  x = arange(0, L, h)

  #Construct W
  wpad = zeros((2*len(x)+1))
  for k in range(1, len(what)):
    wpad[k]  = what[k] * 2.j * pi / float(k)
    wpad[-k] = conjugate(wpad[k])
  W = ifft(wpad) / float(len(wpad))
  W = W[:len(x)] + x * what[0]

  #Construct t0 and t1 bounds
  t0 = (4. * x * W)**d
  t1 = (L - 2. * x)**d * (W[-1]**d - W**d)

  #Initialize parameters
  best_t, best_err, best_x0, best_x1 = 0, inf, 0, 0

  #Brute force search for x1
  for i, x1 in enumerate(x):
    t   = t1[i]
    x0  = bisect.bisect_left(t0, t) #Binary search for x0
    err = 2 * x0 + x1
    if(err < best_err and x0 <= x1):
      best_t, best_err, best_x0, best_x1 = t, err, x0, x1

  return best_t, best_err, best_x0, best_x1
```

The total time complexity for this approach is dominated by the cost of the Fourier transform, and so it runs in $O(L/h(\log(L) - \log(h))$.

6.2 Rotations

Rotations are much more subtle than translations. The reason for this difficulty is due to the fact that there are only finitely many non-trivial discrete subgroups of the discrete motion, and so there is a limit to the degree to which one can uniformly sample the space of rotations.

For example, in the case of $SE(2)$, there are only 5 possible choices for the type of this discrete subgroup, each of which corresponds to one of the nonreflective tiling groups[56]; taking the finest of these, which is the hexagonal tiling group, gives only a maximal rotational resolution of 60°. As a result, there is a fundamental limitation on the resolution of the rotations that can be represented exactly; and unlike translations it is impossible to drive this down to some arbitrarily fine limit with more samples. Clearly, restricting rotations to increments of either 60° or 90° is undesirable for most applications, and so one inevitably forced to make do with some type of approximation.

One standard technique for dealing with rotations is to simply use interpolation. In practice this has proven very effective in image processing, and can be achieved in any number of ways. Presently, we have adopted this strategy for the sake of expediency and convenience; and empirical evidence suggests that it is a workable solution. However, we do not yet have a precise formulation for the errors associated with this process and so at the moment it can be considered an unproven, but pragmatic, heuristic.

6.3 The Simplified Discretized Formulas

With this analysis in mind, we now turn to some more specific optimizations to improve the performance of evaluating the discretized version of the above summation. The first is based on the observation that since all of our bump functions are real, their Fourier transforms are Hermitian symmetric; which means that for any $\omega \in \mathbb{R}^d$,

$$\widehat{\chi_S}(-\omega) = \overline{\widehat{\chi_S}(\omega)}.$$

Consequently, we only need to store half the number of floating point values in order to represent the Fourier transform of a bump function.

Second, we also observe that if the Fourier transform of each solid is precalculated and stored, we can premultiply each Fourier transform by the square root of the window function to save on the run time cost of evaluating the kernel. The reasoning for this is fairly straightforward, since

$$\mathcal{F}\left(K * f * g\right) = \widehat{K}\widehat{f}\widehat{\widehat{h}} = \left(\sqrt{\widehat{K}}\widehat{f}\right)\left(\sqrt{\widehat{K}}\widehat{g}\right).$$

Using these two facts, we now construct the pre-multiplied Fourier window for a solid according to the following formulas for the discretized collision test and gradient. Let S be a solid and let K be an instrument function. Then we define, the following pair of functions, $Q_S : \mathbb{R}^d \to \mathbb{R}$ and $DQ_S : \mathbb{R}^d \to \mathbb{R}^d$ such that,

$$Q_S(\omega) \quad \stackrel{\text{def}}{\equiv} \quad \sqrt{\widehat{K}\left(\frac{2\pi}{L}\omega\right)}\widehat{\mathbf{1}_{S \oplus B^{\infty}_{x_0}}}\left(\frac{2\pi}{L}\omega\right) \tag{28}$$

$$DQ_{S,k}(\omega) \quad \stackrel{\text{def}}{\equiv} \quad \sqrt{\widehat{K}\left(\frac{2\pi}{L}\omega\right)}\mathcal{F}\left(ix_k \mathbf{1}_{S \oplus B^{\infty}_{x_0}}(x)\right)\left(\frac{2\pi}{L}\omega\right) \tag{29}$$

Where the variable x_k denotes the k^{th} coordinate of the x vector, and similarly for the subscript on $DQ_{S,k}$. Using these conventions, we now formulate a discretized version of the collision test and gradient calculation as given by Cor. 3 and Theorem 5. As a shorthand, we let $J = [0, n/2)^d \cap \mathbb{Z}^d \setminus \{0\}$ denote the indexing set,

$$V_{S,T}(R,x) \quad = \quad Q_S(0)Q_T(0) + 2\sum_{j \in J} \mathfrak{re}\left(Q_S(Rj)\overline{Q_T(j)}e^{\frac{2\pi i}{L}\langle j,x \rangle}\right) \tag{30}$$

$$\left(\frac{d}{dx}V_{S,T}(R,x)\right)(\tau) \quad = \quad \frac{2}{L}\sum_{j \in J} \mathfrak{re}\left(i\left\langle j,\tau\right\rangle Q_S(Rj)\overline{Q_T(j)}e^{\frac{2\pi i}{L}\langle j,x \rangle}\right) \tag{31}$$

$$\left(\frac{d}{dR}V_{S,T}(R,x)\right)(R\Omega) \quad = \quad \frac{2}{L}\sum_{j \in J} \mathfrak{re}\left(\left\langle DQ_S(Rj), R\Omega j\right\rangle \overline{DQ_T(Rj)}e^{\frac{2\pi i}{L}\langle j,x \rangle}\right) \tag{32}$$

Where the quantity $V_{S,T}$ is our discrete analogue of the obstacle bump function, $\chi_{O_{S,T}}$ as defined in Eqn. 22. The quantities W and DW only need to be sampled at the points in $J \cup \{0\}$, and so they require at most $n^d(d+1)$ floating point values. Moreover, for many applications they may be precalculated using the Fourier transform, and so their construction cost is mostly negligible.

6.4 Short Circuit Evaluation

Each time a collision test is performed on a pair of solids S, T, Eqns. 30, 31 and 32 must be evaluated, and the predicate $V_{S,T}(R, x) > t$ must be evaluated. To speed up this process we, first note that all these expressions share a common indexing set as well as a number of common factors. This can be exploited by an implementation to reduce the number of operations required by a constant factor.

Second, we also observe that when evaluating the collision predicate one can perform a 'short circuit' during the calculation if it can be quickly determined that the partial sum of the two solids is vanishingly small. To avoid calculating these extra terms, we use a simple application of Cauchy's theorem to bound the remainder of the collision predicate. Given a subindexing set, $K \subseteq J$, we can obtain the following bounds on the partial sum for $V_{S,T}$:

$$0 \leq 2 \sum_{k \in K} \mathfrak{re}\left(Q_S(Rj)\overline{Q_T(j)} e^{\frac{2\pi i}{L}\langle j,x \rangle} \right) \leq \sqrt{\sum_{k \in K} |Q_S(k)|^2} \sqrt{\sum_{k \in K} |Q_T(k)|^2}$$

Thus, it is always true that,

$$V_{S,T}(R,x) \leq Q_S(0)Q_T(0) + 2\left(\sum_{k \in K} \mathfrak{re}\left(Q_S(Rj)\overline{Q_T(j)} e^{\frac{2\pi i}{L}\langle j,x \rangle} \right) \right) + \sqrt{\sum_{k \in J \setminus K} |Q_S(k)|^2} \sqrt{\sum_{k \in J \setminus K} |Q_T(k)|^2}.$$

And so if at any point the right hand partial summation is less than t, then the two objects can not collide. Using this strategy, one can often early out in the case of non-collision much faster. To speed up the calculation of the residual term, we observe that the partial sums for the square norms of Q_S and Q_T can also be precalculated and incrementally subtracted as the sum for $V_{S,T}$ is evaluated.

In principle, this could be sped up even further by reordering the summation so that the highest energy terms are summed first, thus increasing the rate of convergence. Experimentally, it appears that the highest energy terms in a Fourier expansion are centered near the origin (which makes sense given that our bump functions are smooth, and thus their Fourier expansions decay quickly). As a result, it appears that an efficient strategy for improving the performance of the method is to evaluate the terms in the summation in the order of their distance to the origin. For a simple rectilinear sampling, this can be done using a spiral traversal of the space. Still another possibility would be to convert the objects to polar coordinates, and then sum in outward expanding radial slices. Converting to polar coordinates also has the added benefit of simplifying rotations, though at the slight expense of introducing more errors.

7 Experiments

We now present some experimental results illustrating the performance of the proposed method. In Fig. 3, we show a plot of the Hausdorff error with respect to the number of radial samples for a number of different solids. Overall the convergence is very close to a rate of $O(\frac{1}{n})$. To generate our initial window, we used the Parks-McClellan algorithm with a guess of $\Delta x = \frac{L \log(n)}{2n}$, then applied gradient descent to tune our choice of x_0 and x_1. In Fig. 5, we show the effects increasing the number of samples for a single pair of solids (as shown in Fig. 4). An approximate SE(2) obstacle is shown in Fig. 6.

8 Conclusion

In conclusion, we have demonstrated a novel approach for numerical collision detection. It has a number of interesting features which distinguish it from previous algebraic/geometric approaches. In particular, it can give provably good approximations to the collision query as well as evaluate normal forces for both translational and rotational motion at no additional overhead. These features suggest that it could be

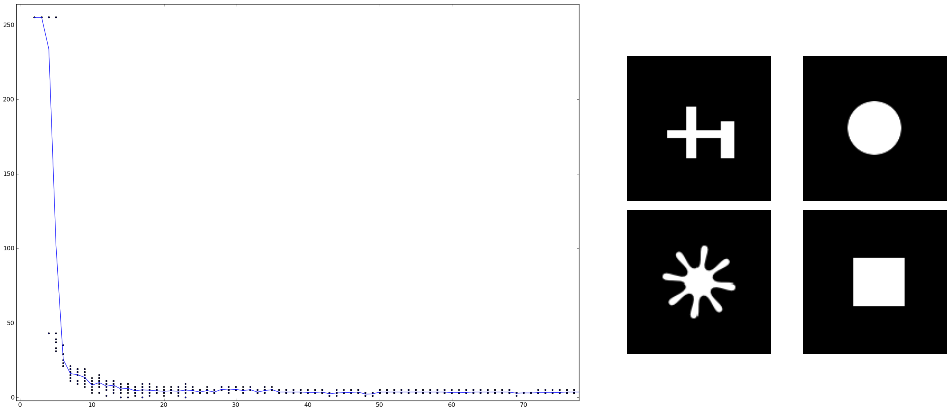
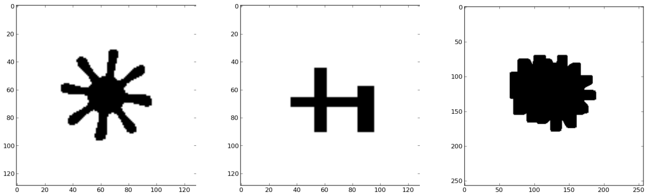


Figure 3: Comparison of Hausdorff accuracy for the proposed method. The horizontal axis indicates the number of samples, while the vertical axis indicates the Hausdorff error. Samples were taken from a collection of pairwise tests consisting of the 4 images on the right. The solid blue curve is the average Hausdorff error.



S T $O_{S,T}$

Figure 4: A pair of test solids, S, T and their exact translational configuration space obstacle, $O_{S,T}$.

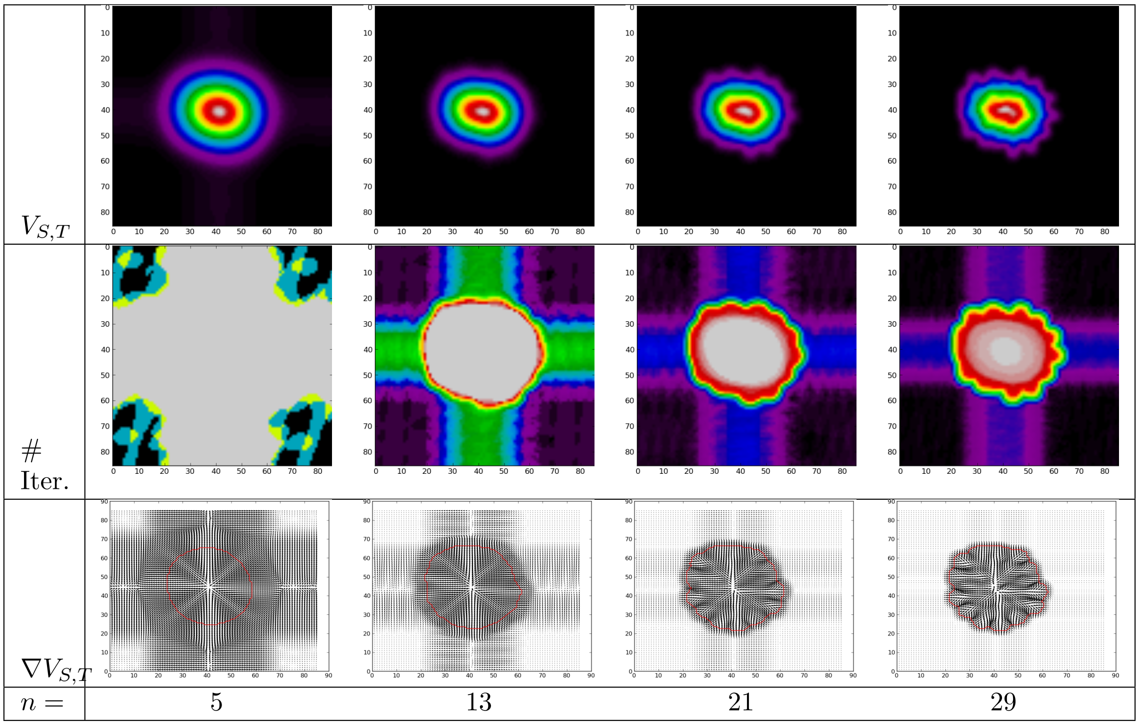


Figure 5: Illustration of rate of convergence for the translational obstacle shown in Fig. 4. On the right are the approximate obstacles constructed with samples of radius n. The top row shows the approximate gap function for the solids; the middle row the number of iterations required to reject a point not contained in the obstacle; and the bottom row shows the gradient as well as the t-level set of the gap function.

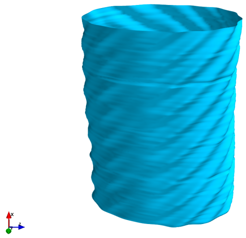


Figure 6: A $SE(2)$ configuration space obstacle generated with $n = 20$ samples for the solids in Fig. 4. The vertical direction indicates the relative rotation of the two solids (in radians), while the horizontal plane indicates relative translation.

appealing for use in situations where fast, approximate and scalable collision detection is desirable (such as in video games).

8.1 Open Conjectures and Future Work

Our basic technique resolves instantaneous collisions between rigidly moving solids. In principle, there are a number of interesting extensions which may be used to improve the overall performance of this method. In particular, it would be useful to extend these results from simple rigid motions to general affine transformations. A secondary issue would also be to incorporate continuous collision detection into this framework as well. Similarly, we have relied primarily on Fourier series expansions since they give a convenient closed form for translation and differentiation operators. However, this approach is somewhat inefficient for rotations. We conjecture that it may be possible to improve performance in this area using some more general class of special functions, such as either the spherical Bessel functions or spherical harmonics. Finally, there remain some open issues about the convergence properties of this method. It is not yet clear that the convergence rate of $O\left(\frac{\log(n)}{n}\right)$ optimal, and we conjecture based on empirical observations that it may be possible to improve this bound to $O\left(\frac{1}{n}\right)$.

9 Acknowledgments

Thanks to Vadim Shapiro, Josh Dancyzk and John Uicker for help with proof reading and many useful comments. This research was supported by NSF grants CMMI-0856778 and CMMI-1029553.

References

[1] Q. Avril, V. Gouranton, and B. Arnaldi. New trends in collision detection performance. In *VRIC, Laval: France*, 2009.

[2] J.F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235 – 256, 1982.

[3] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. In *Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 109 – 116, 1990.

[4] J.W. Boyse. Interference detection among solids and surfaces. *Communications of the ACM*, 22(3):3 – 9, 1979.

[5] G. S. Chirikjian and A. B. Kyatkin. *Engineering Applications of Noncommutative Harmonic Analysis.* CRC Press, October 2000.

[6] P.G. Comba. A procedure for detecting intersections of three-dimensional objects. *Journal of the ACM*, 15(3):354 – 356, 1968.

[7] P.A. Cundall and O.D.L Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47 – 65, 1979.

[8] Belen Curto, Vidal Moreno, and Francisco Blanco. A general method for c-space evaluation and its application to articulated robots. *IEEE Trans. on Robotics and Automation*, 18:24–31, 2002.

[9] F. Eisenbrand, S. Funke, A. Karrenbauer, J. Reicherl, and E. Schömer. Packing a trunk: now with a twist! In *ACM Symposium on Solid and Physical Modeling*, pages 197 – 206, 2005.

[10] C. Ericson. *Realtime Collision Detection.* Morgan-Kaufman, 1 edition, 2005.

[11] R.C. Fetecau, J.E. Marsden, M. Oritz, and M. West. Nonsmooth lagrangian mechanics and variational collision integrators. *SIAM Journal of Applied Dynamical Systems*, 2(3):381 – 416, 2003.

[12] S.F. Frisken, R.N. Perry, A.P. Rockwood, and T.R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of 27th annual conference on computer graphics and interactive techniques*, pages 249 – 254, 2009.

[13] C. Fuenfzig, T. Ullrich, and D.W. Fellner. Hierarchical spherical distance fields for collision detection. *Computer Graphics and Applications*, 26(1):64 – 74, 2006.

[14] A. Fuhrmann, G. Sobottka, and C. Gross. Distance fields for rapid collision detection in physically based modeling. In *Proceedings of GraphiCon 2003*, pages 58 – 65, 2003.

[15] M.A. Ganter and J. Uicker. Dynamic collision detection using swept solids. *ASME Journal of Mechanism, Transmission and Automation in Design*, 108:549 – 555, 1986.

[16] E.G. Gilbert and D.W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, 1(1):21 – 30, 1985.

[17] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 4:193 – 203, 1988.

[18] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisons and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4:74 – 123, 1985.

[19] O. Gurel. Marker layout problem via graph theory: An attempt for optimal layout of irregular patterns. Technical Report 320-2921, IBM Scientific Centre, 1968.

[20] P.T. Harker and J.S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming*, 48:161 – 220, 1990.

[21] F.J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proc. of the IEEE*, 66:51 – 83, 1978.

[22] W. Karush. Minima of Functions of Several variables with Inequalities with Side Constraints. Master's thesis, Univ. of Chicago, Chicago, Illinois, 1939.

[23] L.E. Kavraki. Computation of Configuration Space Obstacles Using the Fast Fourier Transform. *IEEE Conference on Robotics and Automation*, 3:255 – 261, 1995.

[24] H.W. Kuhn and A.W. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, pages 481–492, 1951.

[25] M.B. Lignola. Regularized gap functions for variational problems. *Operations Research Letters*, 36(6):710 – 714, 2008.

[26] M. C. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proc. of IMA Conference on Mathematics of Surfaces*, 1998.

[27] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation. In *IEEE International Conference on Robotics and Automation*, pages 1008 – 1014, 1991.

[28] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE transactions on computers*, 100(32):108–120, 1983.

[29] M. Lysenko, S. Nelaturi, and V. Shapiro. Group morphology with convolution algebras. In *ACM Symposium on Solid and Physical Modeling*, 2010.

[30] M. Mabrouk. A unified varitational model for the dynamics of perfect unilateral constraints. *European Journal of Mechanics*, 17(5):819 – 842, 1998.

[31] K. Maruyama. A procedure to determine intersections between polyhedral objects. *International Journal of Parallel Programming*, 1(3), 1972.

[32] J. McClellan and T. Parks. A computer program for designing optimum fir linear phase digital filters. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 21:506, 1973.

[33] A. Middleditch. The representation and manipulation of convex polygons. *Theoretical Foundations of Computer Graphics and CAD*, 40:211 – 252, 1988.

[34] B. Mitrich. Efficient algorithms for two-phase collision detection. Technical report, MERL, 1997.

[35] M. Moore and J. Wilhems. Collision detection and response for computer animation. In *Proceedings of SIGGRAPH 1988*, pages 289 – 298, 1988.

[36] J.J. Moreau. *Standard inelastic shocks and the dynamics of unilateral constraints*, pages 173 – 221. Springer, 1983.

[37] S. Nelaturi and V. Shapiro. Configuration products in geometric modeling. In *ACM Symposium on Solid and Physical Modeling*, pages 247 – 258, 2009.

[38] A.H. Nuttall. Some windows with very good sidelobe behavior. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29:84 – 91, 1981.

[39] Max Plus. Linear systems in (max, +) algebra. In *Proceedings of the 29th Conference on Decision and Control*, 1990.

[40] L.B. Rall. Automatic differentiation: Techniques and applications. *Lecture Notes in Computer Science*, 120, 1981.

[41] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157 – 160, 1973.

[42] J.B.T.M Roerdink. Group morphology. *Pattern Recognition*, 33(6):877 – 895, 2000.

[43] R. M. Rosenberg. On geometric representations of dynamics. *Acta Mechanica*, 2(2):144 – 170, 1965.

[44] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13:471 – 494, 1966.

[45] J.R. Rossignac and H.B. Voelcker. Active zones in CSG for accelerating boundary evaluation, redundancy elimination, interference detection, and shading algorithms. *ACM Transactions on Graphics*, 8(1):51 – 87, 1988.

[46] W. Rudin. *Real and Complex Analysis.* McGraw-Hill, 1970.

[47] V.L. Rvachev. Geometric applications of logic algebra. *Naukova Dumka*, 1967. In Russian.

[48] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1983.

[49] V. Shapiro. Semi-analytic geometry with R-functions. *Acta Numerica*, 16:239–303, 2007.

[50] J. Shen and G. Strang. The asymptotics of optimal (equiripple) filters. *IEEE Trans. on Signal Processing*, 47:1087, 2002.

[51] D.E. Stewart. Rigid-body dynamics with firction and impact. *SIAM Review*, 42:3 – 40, 2000.

[52] Y.G. Stoyan. Mathematical methods for geometry design. In *Advances in CAD/CAM, Proceedings of PROLAMAT*, pages 67 – 83, 1983.

[53] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, Laks Raghupathi, A. Fuhrmann, Marie-Paule Cani, François Faure, N. Magnetat-Thalmann, and W. Strasser. Collision detection for deformable objects. In *Eurographics State-of-the-Art Report (EG-STAR)*, pages 119–139. Eurographics Association, Eurographics Association, 2004.

[54] Roberto Theron, Vidal Moreno, Belen Curto, and Francisco J Blanco. Collision detection trough deconstruction of articulated objects. In *Lecture Notes in Computer Science. AMDO 2006. Vol. 4069*, pages 165–174. Springer, 2006.

[55] R.B. Tilove. A null-object detection algorithm for constructive solid geometry. *Communications of the ACM*, 27(7):684 – 694, 1984.

[56] H. Weyl. *Symmetry.* Princeton University Press, 1st edition, 1952.

[57] A. Witkin, M. Gleicher, and W. Welch. Interactive dynamics. In *Proceedings of SIGGRAPH 1990*, pages 11 – 21, 1990.

[58] L. Zhang, Y.J. Kim, and D. Manocha. C-DIST: Efficient Distance Computation for Rigid and Articulated Models in Configuration Space. In *ACM Symposium on Solid and Physical Modeling*, 2007.