On New Collision Detection Techniques: Minkowski Sums, Fourier Transforms and Spherical Decomposition

Can Erdogan
CS6491: Computer Graphics
Final Project
2015-12-01

***Abstract*— Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like ≫Huardest gefburn≪. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language. Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like ≫Huardest gefburn≪. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language.**

I. Introduction

Collision detect is one of the fundamental tools for robotics, computer graphics and mechanical design. A number of different variants of collision detection is investigated within these fields with examples ranging from static and dynamic objects in the environment to narrowphase and broadphase (single vs. multiple collisions) detections. In this work, we are interested in the collision of a single pair of static objects. The body of the work is focused on examining the role of three concepts: (1) Minkowski sums, (2) fourier transformations and (3) spherical decompositions.

It is a well-known fact that Minkowski sums can be utilized to generate more complex shapes from simpler primitives. One of the main observations that motivate our research is the utilization of Fourier transforms to increase the efficiency of their computations. A second motivation is to observe the effect of shape representation in the efficiency of Fourier transforms and the exploitation of their underlying properties such as the *convolution theorem* and the *time shifting*. The bulk of the work is focused on the recent work (2016) of Behandish and Ilie [1] who promote the use of spherical decompositions along with non-equispaced Fast Fourier transforms (NFFTs).

Figure 1 represents one of the key ideas that allow the transformation of object representations in cartesian space

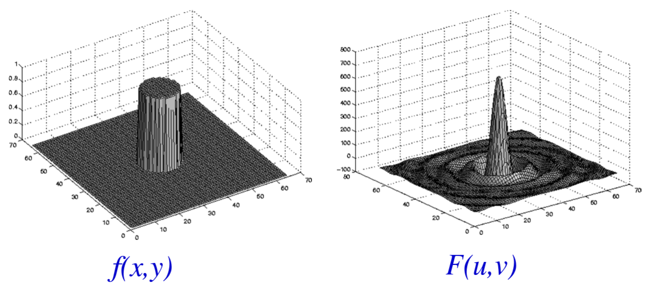

Fig. 1. An obstacle in the cartesian space and its representation in the frequency domain

to the frequency space. The idea is that some function $f(x, y)$ (in $\mathbf{R}^2$ for this example) can be used to represent a cylindrical object (possible an obstacle) and if this so-called *bump function* is chosen properly (e.g. smoothness, differentiability, etc.), its Fourier transform, $F(x, y)$, can be exploited in collision detection computations.

This report is organized as follows. First, we present briefly the related work in Fourier transforms, their use for collision detection in robotics, and the variants of shape descriptors in computer graphics. Then, we present the idea of Minkowski sums and bump functions to formulate collision detection rules. In Section IV, we briefly give an overview of the idea of spherical decomposition and its use with bump functions, followed by in Section V, with the main treatment of Fourier transformations in the context of collision detection. We conclusion with preliminary results that confirm the findings of [1] and point out to a number of future speed-up techniques accumulated from different sources ([1], [2], [3]).

II. Related Work

The Fourier transform, reportedly first used by Carl Friedrich Gauss to compute the period of an asteroid as a series of sinusoids, has been the linchpin of modern signal processing techniques since the introduction of Fast Fourier Transforms (FFT)s in 1965 [4]. There are two key properties that are of interest of FFTs to our work. First, the FFT algorithm computes the Fourier transform of a discrete signal of length n in O(nlogn) time whereas the classical discrete transformation by definition takes O(n^2) computations. Second, the Fourier transform of a function in cartesian space represents the data in terms of a summation of different frequencies. As in [3], we propose the use of this representation by first analyzing the values at the higher frequencies and stop the computation of the transformation if the lack of a collision can be guaranteed.

One of the key ideas of this work is how the shape representation affects the collision detection methods. A number of shape descriptors have been proposed in the computer graphics literature such as the OBB trees [5] and voxel-based representations [6]. The benefit of spherical decompositions is that every sub-part is rotation-invariant and although within the scope of this work, we limit ourselves to spheres of constant radius, the extension to variable radii is trivial [1].

The use of spherical decompositions leads to an interesting area of fourier computations, namely nonequispaced Fast Fourier transforms (FFTs). The traditional DFT algorithm and the FFT algorithm both assume that the time-varying signal is sampled with equispaced time intervals. Although this might be the case with a lot of traditional electrical sensors, a number of domains such as health sector (MRIs) [7], computational physics (heat flow computations) [8], etc. have sampled from non-equispaced time nodes. In our domain, the placement of the sphere centers are clearly non-uniform and [1] proposed the use of NFFT algorithms to compute the Fourier transform of a signal composed of their locations. We have examined a number of prior work in this field [9], [10], [11] and foresee that the use of these algorithms will increase the efficiency of the Fourier approach even more [12]. However, within the scope of this work, we limit ourselves to interpolating between the centers of the spheres and generating a sparse uniform signal.

III. Minkowski Sums in Collision Detection

A. Bump functions

The two main shape representations we discuss in this work is uniform-sampling and spherical-sampling which can be interpreted as a list of voxels with 0-1 booleans for where the voxel intersects the shape and a number of 3D points where spheres of some constant radii are within the shape. In discussing such representations, our papers of interest [1], [3] choose the adopt the descriptor functions $f_S : \mathbf{R}^3 \to \mathbf{R}$ for some solid S such that the shape is the domain of f_S where $f_S(x) > 0$. This representation is convenient because if we formalize the idea above with the notation:

$$U_t(f_S) = \{x \in \text{domain of } f_S | f_S(x) > t\}, \tag{1}$$

then, the union and the intersect of two shapes S_1 and S_2 become additions and dot product of their respective descriptor functions:

$$S_1 \cup S_2 = U_0(f_{S_1} + f_{S_2}) \tag{2}$$
$$S_1 \cap S_2 = U_0(f_{S_1} f_{S_2}) \tag{3}$$

For instance, in classical robotics applications, with voxel-based descriptors, the function $f_S = \mathbf{1}_S$ is an indicator function such that $\mathbf{1}_S(x) = 1$ if and only if $x \in S$. In such a case, two shapes can only intersect if there exists some x such that both of the shapes have some volume, that is both $f_{S_1}(x) = f_{S_2}(x) = 1$ and thus, their dot product is greater than 0. Note that since the analytical expressions of shape descriptors is too complex, when we refer to operations on these functions, we imply to operations on a set of sampled values in their domains.

A number of different variants of descriptor functions are available in addition to indicator functions that have more smoothness properties and in this work, for the spherical decomposition approach, we are interested in the following bump function:

$$\psi_\alpha(x) = \begin{cases} e^{(1-|x|^{-\alpha})^{-1}}, & \text{if } |x| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

which is parameterized by constant α that changes how the function behaves close to the non-zero limits. For a single sphere shape S_0, we then define the descriptor function as $f_{B_0} = \psi_\alpha(|x|/r)$ where r is the radius of the shape. In Figure 2, we demonstrate the effect of the α value in the descriptor function outputs. In our work, we adopted $\alpha = 1$. Note that $\alpha = \infty$ corresponds to the indicator function $\mathbf{1}_S(x)$.

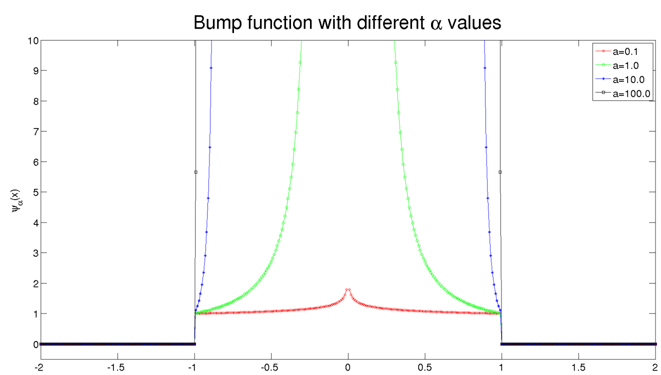

Fig. 2. The effect of the α constants in the descriptor function.

B. Use of Minkowski Sums in Early-Miss Tests

An important observation made in [3] is that in most applications, the frequency of non-collisions is much higher than collisions. Thus, it is important to bias our computations towards detecting lack of collisions early on with so-called "early-miss" tests. One example of such early-miss tests is the common adopted of bounding boxes in computer graphics, specifically ray tracing applications.

Here, we make a brief note on how the Minkowski sum, can be utilized to implement such a early-miss test. First, a brief definition: for two sets A and B, their Minkowski sum $A \oplus B$ is defined as the set:

$$A \oplus B = \{a + b | a \in A, b \in B\} \tag{5}$$

where, in some applications, it is insisted that for one of the sets, say A, the origin $O \in A$.

The idea behind using the Minkowski sum is to swell up one of the objects say, S_2 with a third object M such that, the

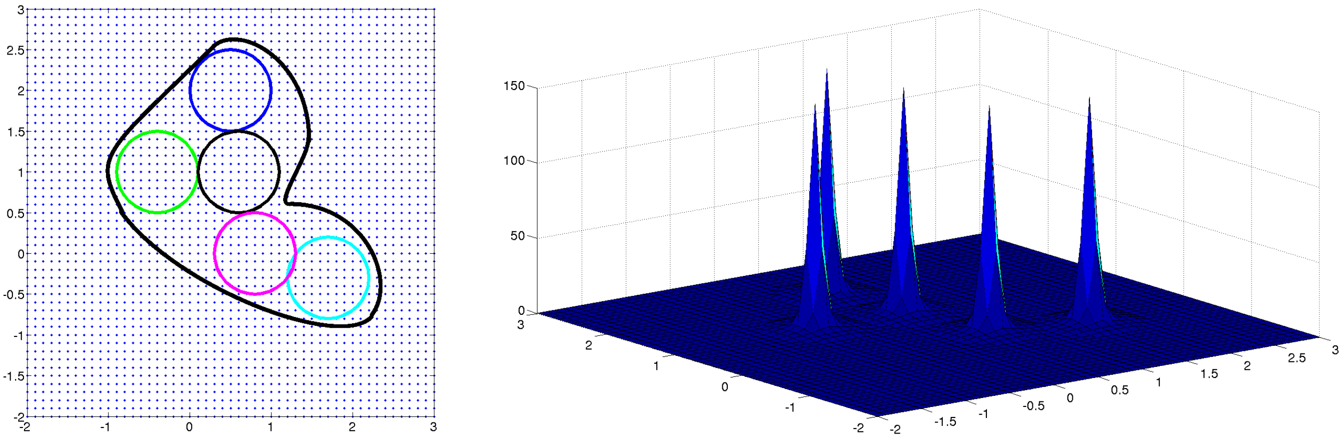

Fig. 3. An example of spherical decomposition for a 2D shape and the visualization of the shape descriptor function composed of the individual bump functions of the spheres.

negative result of the test $S_1 \cap (S_2 \cup M)$ implies that $S_1 \cap S_2 = \emptyset$. Clearly, for some choices of M, such as elliptic objects, this early-miss test is more efficient. Note that the shape descriptor of the Minkowski sum of two objects M and N can be expressed as a convolution of their shape descriptors, a property that will be useful in the frequency space analysis:

$$M \cup N = U_0(f_M * f_N) \tag{6}$$

where the $*$ represents the convolution of the two functions. To learn more about the Fourier interpretation of this operation, see the section "Convolution theorem" in Section V. Note that the early-miss test $S_1 \cap (S_2 \cup M)$ now contains two integrals, one for the convolution and one for the dot product, and in Section V, we will discuss how this situation can be remedied with Fourier transforms.

IV. Spherical Decomposition

Spherical decomposition of three-dimensional objects has been well studied in the computer graphics literature. One of the classical approach is based on the medial axis transform (MAT) where the idea is to represent the shape using spheres with maximal radius such that they are tangential to the inner surface of the object manifold. A challenge with medial axis approaches is that the MAT of an r-set is not necessarily closed, a property that is desirable and achievable with the Minkowski sum formulation we will show in the next section. Instead, [1] proposes a variant of the sphere packing algorithms that exploit the signed distance function field representation of an object, with reportedly less number of output spheres.

In the left image of Figure 3, we demonstrate a spherical decomposition of a two-dimensional object. The idea is to accumulate the bump functions of each sphere, to approximate the overal shape descriptor function as shown on the right. Next, we formulate this approach using Minkowski sums that will help us exploit Fourier representation in Section V.

A. *Minkowski Sum Formulation*

Let P be the set of center points of the balls B_i that are used to describe an object shape $S \approx \hat{S} = \bigcup B_i$. The idea is that the approximate shape $\hat{S}$ can be described as a Minkowski sum of a base ball B_0 and the set of center points P. Here, we explain the proof in [1] to justify this expansion. We begin with the basic description of union of objects as defined in Equation 2:

$$f_{\hat{S}}(x) = \sum_i f_{B_i}(x) = \sum_{p_i \in P} f_{B_0}(x - p_i) \tag{7}$$

where the use of the central ball B_0 is justified by the offset $(x - p_i)$. Then, we expand the definition of the ball description function as:

$$f_{\hat{S}}(x) = \sum_i \int_{\mathbf{R}^3} \delta^3(x' - p_i)[f_{B_0}(x - x')dx' \tag{8}$$

which allows us to reason about the centers of the balls p_i in the continuous samples x'. This is significant because in the next step, by defining a function of spatial impulses,

$$\rho(x) := \sum_i \delta^3(x - p_i) \tag{9}$$

, where δ^3 is a Dirac function, we can now rewrite the Minkowski sum of the approximate shape $\hat{S}$ as a convolution of the primitive ball function B_0 and the points:

$$P \oplus B_0 = U_0(\rho * f_{B_0}) \tag{10}$$

This result is encouraging because first, we have already seen the use of Minkowski sums in early-miss tests and we hope to exploit frequency domain properties in their computations. Second, Lysenko provides theoretical bounds on the Hausdorff distances for Minkowski sum approximations which can be transferred to spherical decomposition approximations. [3]

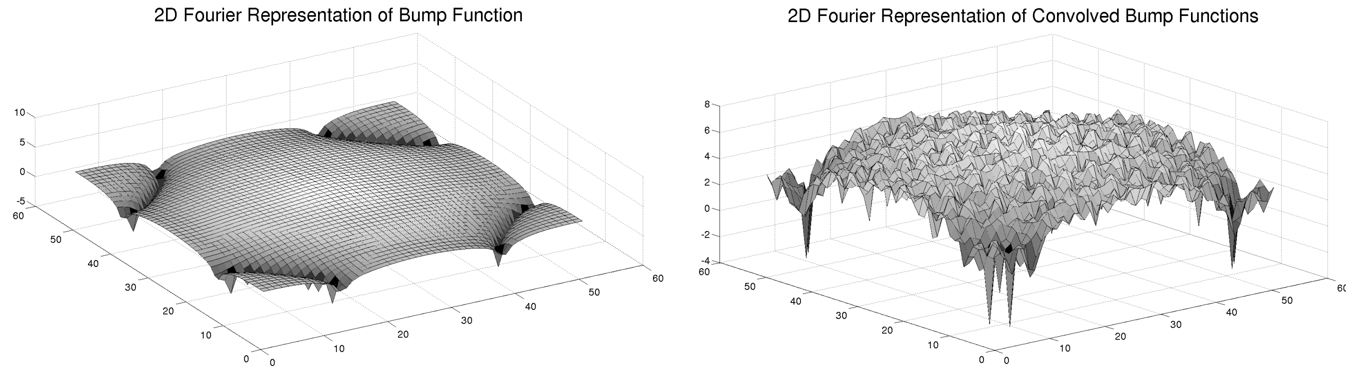

Fig. 4. The frequency domain representations of the B_0 function and the descriptor function for the shape in Figure 3.

V. Fourier transforms

A. Convolution theorem

For two functions f_1 and f_2 and their corresponding Fourier transforms F_1 and F_2, the following two identities hold true under suitable conditions:

$$F(f_1 * f_2) = F_1 \cdot F_2 \tag{11}$$

$$F(f_1 \cdot f_2) = F_1 * F_2 \tag{12}$$

where $\cdot$ represents a point-wise multiplication and the function $F(.)$ takes the Fourier transform of its input. This indicates that the convolution steps for the Minkowski sums of the descriptor functions can be implemented as point-wise multiplications. This is indeed very powerful because the Minkowski sum is an $O(n^2)$ operations whereas the Fourier transform is $O(n)$ and the FFT algorithm is $O(nlog(n))$.

Figure 4 demonstrates, on the left, the Fourier domain representation of the bump function defined in Equation 4 implemented for 2D objects. On the right, we observe the Fourier representation of the entire shape description function for the shape in Figure 3, generated by the convolution of the bump functions in the frequency space.

One advantage of using the spherical decomposition also comes out in the Fourier representation where we can precompute the transform of the function B_0 to any desired resolution and then convolve it with the list of ball centers online. Note that to match the signal lengths, the center list would need to be padded with zeros as FFT is applied.

B. Cartesian Transformations with Dirac functions

Kavraki's initial work on Fourier domain representations of configuration spaces in 1995 [2] dealed only with translations of obstacles and the robot agents. To the most part, most of the literature still focuses mostly on translations, and has heuristic approaches for approximating rotations. Although it should be noted that it is trivial to reflect rotations of multiples of $\frac{pi}{4}$ in the frequency domain. The advantage for the translations is based on the fact that they can be represented as time-shifts where for some signal $f(x)$, the Fourier transform of its shift by Δx, $f(x - \Delta x)$ is:

$$f(x - \Delta x) = e^{-2\pi i \Delta x} F(x) \tag{13}$$

where $F(x)$ is the Fourier transform of the original signal $f(x)$. This basic property allows the recomputation of shape functions without having to first translate the object in the cartesian space, resample and perform FFT on it - an $O(n) < O(nlog(n))$ advantage.

VI. Results

A. Faster Minkowski Summation with Fourier Transform

B. Spherical vs. Uniform Sampling

Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like »Huardest gefburn«. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language. Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like »Huardest gefburn«. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language.

VII. Future Work

A. Possible speed-ups

- Fourier cutoffs - Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like

≫Huardest gefburn≪. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language. Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like ≫Huardest gefburn≪. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language.

VIII. Conclusion

O(n2) - O(nlogn) Spherical ¿ uniform

References

[1] Morad Behandish and Horea T Ilieş. Analytic methods for geometric modeling via spherical decomposition. *Computer-Aided Design*, 70:100–115, 2016.

[2] Lydia E Kavraki. Computation of configuration-space obstacles using the fast fourier transform. *Robotics and Automation, IEEE Transactions on*, 11(3):408–413, 1995.

[3] Mikola Lysenko. Fourier collision detection. *The International Journal of Robotics Research*, 32(4):483–503, 2013.

[4] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[5] Stefan Gottschalk, Ming C Lin, and Dinesh Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180. ACM, 1996.

[6] Mikel Sagardia, Theodoros Stouraitis, and Joao Lopes e Silva. A new fast and robust collision detection and force computation algorithm applied to the physics engine bullet: Method, integration, and evaluation. In *Conference and Exhibition of the European Association of Virtual and Augmented Reality*, 2014.

[7] Tobias Knopp, Stefan Kunis, and Daniel Potts. A note on the iterative mri reconstruction from nonuniform k-space data. *International journal of biomedical imaging*, 2007, 2007.

[8] June-Yub Lee and Leslie Greengard. The type 3 nonuniform fft and its applications. *Journal of Computational Physics*, 206(1):1–5, 2005.

[9] Alok Dutt and Vladimir Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific computing*, 14(6):1368–1393, 1993.

[10] Alok Dutt and Vladimir Rokhlin. Fast fourier transforms for nonequispaced data, ii. *Applied and Computational Harmonic Analysis*, 2(1):85–100, 1995.

[11] Daniel Potts, Gabriele Steidl, and Manfred Tasche. Fast fourier transforms for nonequispaced data: A tutorial. In *Modern sampling theory*, pages 247–270. Springer, 2001.

[12] Jens Keiner, Stefan Kunis, and Daniel Potts. Using nfft 3—a software library for various nonequispaced fast fourier transforms. *ACM Transactions on Mathematical Software (TOMS)*, 36(4):19, 2009.