Shape Representations and Algorithms for 3D Model Retrieval

Michael M. Kazhdan

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science

June 2004

Abstract

With recent improvements in methods for the acquisition and rendering of 3D models, the need for retrieval of models from large repositories of 3D shapes has gained prominence in the graphics and vision communities. A variety of methods have been proposed that enable the efficient querying of model repositories for a desired 3D shape. Many of these methods use a 3D model as a query and attempt to retrieve models from the database that have a similar shape.

In this thesis, we begin by introducing a new shape descriptor that is well suited to the task of 3D model retrieval. The descriptor is designed to enable efficient and meaningful comparison of 3D shapes, thereby satisfying the requirements of efficiency and discriminability that are necessary for an effective, real-time shape retrieval system. We compare our descriptor to other existing descriptors in empirical retrieval experiments, demonstrating that the new shape descriptor provides improved retrieval accuracy and is better suited to the task of shape matching.

One of the specific challenges in matching 3D shapes arises from the fact that in many applications, models should be considered to be the same if they differ by a similarity transformation. Thus in order to match two models, a measure of similarity needs to be computed at the optimal translation, scale and rotation. In this thesis, we review a number of approaches for addressing the alignment challenge and provide new methods for addressing this issue that give rise to better shape matching algorithms.

Additionally, we present two general methods for improving the performance of many extant 3D model matching algorithms by providing a general framework for augmenting existing shape representations with global shape information characterizing salient shape properties. The first approach leverages symmetry information to augment existing representations with information characterizing a model's self-similarity. The second

approach factors the shape matching equation as the disjoint product of anisotropy and geometric comparisons — improving the matching performance of many shape metrics by facilitating the task of shape registration.

Acknowledgments

This thesis represents four years of work in the Princeton University. This work would not have been possible without the boundless assistance of mentors, colleagues, and friends.

It is with the deepest of gratitude that I would like to thank my advisor, Thomas Funkhouser, who would always push me for more, but never for more than I could. I would also like to express my heartfelt thanks to the other professors who have always kept their doors open, ready to discuss and encourage new ideas. Particular thanks go to Szymon Rusinkiewicz who could always find time, despite any other responsibilities or deadlines he may have had.

In my time at Princeton, I have had the opportunity to collaborate with a number of different people without whose help this thesis could not have been completed. I would like to thank Patrick Min, Phil Shilane, Dan Rockmore and Peter Kostelec from Dartmouth, Peter Sarnak, David Dobkin, Bernard Chazelle, Adam Finkelstein, David Jacobs, Ayellet Tal, Joyce Chen, Alex Halderman, and William Kiefer for their time and their efforts. While not a collaborator as such, I would also like to thank Melissa Lawson for her regular guidance and tireless assistance. Additionally I would like to acknowledge the institutions that have provided generous support for my research: this work was funded by a National Science Foundation grant and an Alfred P. Sloan fellowship.

Finally, I would like to express my sincere thanks to friends and family whose unwavering support has been indispensable to me over the last years. Thank you to my family: Cilia, Moussia, Mom, Dad, Eli, Dina and Dani. Thank you to the friends that I came to graduate school with: Yoni, Yoav, Jeremy, and Alexei and Molly. And thank you to the new friends that I have made along the way: Allison, Jason, Paul and Kimberley, and Ann.

For Sanya and Mark.

For Moussia and Cilia.

For those who have left us.

For those that stayed behind.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last few decades, the proliferation of the World Wide Web has resulted in a consolidated repository of information. As a result of this information revolution we have reached a point where much of the data that we seek is available through our web browsers. We can now go online and find the weather, the news, dictionary definitions, biographies, journal articles, restaurant reviews, etc. with a few keystrokes. In this context, one of the challenges of data retrieval has shifted to the design of effective methods for finding desired information: "Given the large amount of information out there, how do I actually find what I am looking for?" To address this challenge, a number of different search engines (e.g. Google [18], Yahoo [65], etc.) have been established that allow a user to specify a simple, textual query and return documents with matching content.

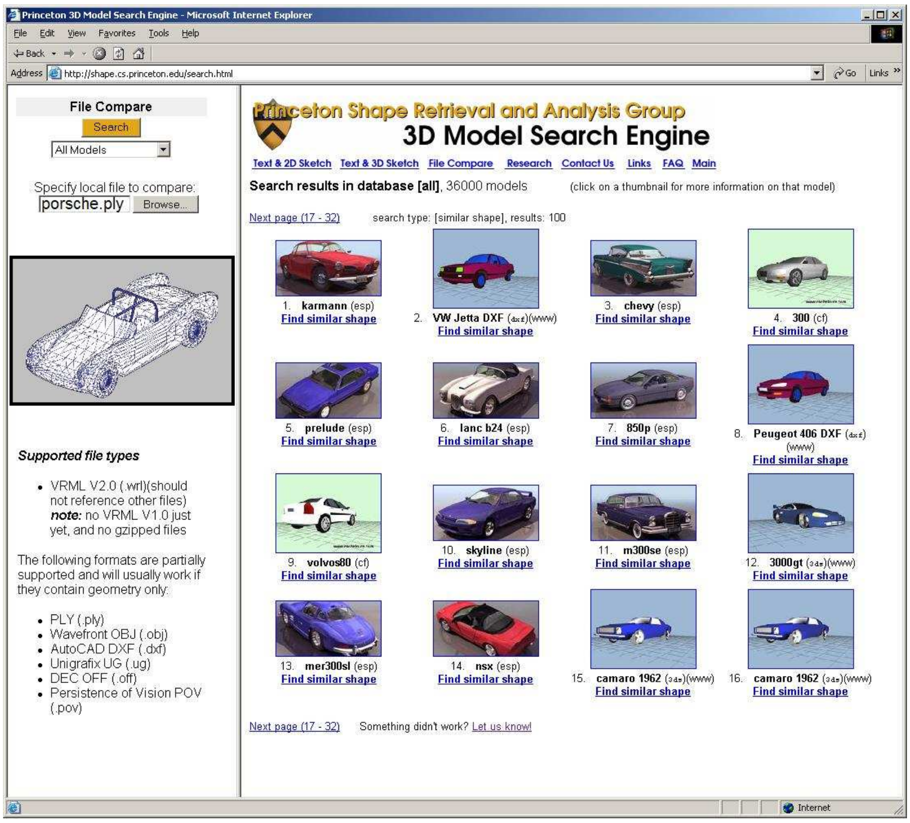
More recently, tools for acquiring and visualizing 3D models have become integral components of data processing in a number of disciplines, including medicine, chemistry, architecture and entertainment. With the increased availability of these tools, we have witnessed an explosion in the number of available 3D models, resulting in the creation of a large number of online repositories of 3D shapes [1, 2, 11, 12, 42, 47, 49, 52, 61].

The ability to retrieve existing models from these repositories facilitates the tasks of professionals in fields ranging from entertainment to scientific research, by allowing them to obtain desired models quickly without requiring the expenditure of large amounts of time modeling a 3D shape. Thus, as we had initially witnessed with text, the proliferation of 3D content in the World Wide Web has changed the focus from the challenge of generating new models — a time consuming endeavor that may take hours or days — to the challenge of retrieving existing ones.

To satisfy this need, a variety of retrieval methods have been proposed that enable the efficient querying of model repositories for a desired 3D shape, many of which use a 3D model as a query and attempt to retrieve models with matching shape from the database. An example of such an application is shown in Figure 1.1. The user specifies a car as a query model (top left). The system then compares the query to every model in the database, returning to the models that are most similar (right).

As in many database retrieval applications, the algorithms for matching 3D shapes are motivated by two principal concerns. First, the algorithms needs to be discriminating — effectively distinguishing between different classes of shapes and returning those models in the database that most closely approximate the ones that a user would want. Second, the algorithms need to be efficient in both space and time. In particular, since many of the existent repositories index thousands, or even tens of thousands of models, the stored representation of a 3D model needs to be compact and the retrieval time needs to be fast enough to return results in real time.

In practice, addressing the run-time efficiency requirement is done with the assistance of a *shape descriptor*. The shape descriptor is an abstraction of the 3D model, capturing salient shape information in a structure that is well-suited for comparison. In many shape matching applications, the shape descriptor represents a 3D model by a fixed-dimensional


Princeton 3D Model Search Engine - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://shape.cs.princeton.edu/search.html
Go
Links
File Compare
Search
All Models
Browse...
Princeton Shape Retrieval and Analysis Group
3D Model Search Engine
Text & 2D Sketch
Text & 3D Sketch
File Compare
Research
Contact Us
Links
FAQ
Main
Internet

Specify local file to compare: porsche.ply

Supported file types

- VRML V2.0 (.wrl)(should not reference other files) ***note:*** no VRML V1.0 just yet, and no gzipped files

The following formats are partially supported and will usually work if they contain geometry only:

- PLY (.ply)
- Wavefront OBJ (.obj)
- AutoCAD DXF (.dxf)
- Unigrafix UG (.ug)
- DEC OFF (.off)
- Persistence of Vision POV (.pov)

Search results in database [all], 36000 models (click on a thumbnail for more information on that model)

Next page (17 - 32) search type: [similar shape], results: 100

1. **karmann** (esp) Find similar shape
2. **VW Jetta DXF** (dxf)(www) Find similar shape
3. **chevy** (esp) Find similar shape
4. **300** (cf) Find similar shape
5. **prelude** (esp) Find similar shape
6. **lanc b24** (esp) Find similar shape
7. **850p** (esp) Find similar shape
8. **Peugeot 406 DXF** (dxf)(www) Find similar shape
9. **volvos80** (cf) Find similar shape
10. **skyline** (esp) Find similar shape
11. **m300se** (esp) Find similar shape
12. **3000gt** (3ds)(www) Find similar shape
13. **mer300sl** (esp) Find similar shape
14. **nsx** (esp) Find similar shape
15. **camaro 1962** (3ds)(www) Find similar shape
16. **camaro 1962** (3ds)(www) Find similar shape

Next page (17 - 32) Something didn't work? Let us know!

Figure 1.1: An example of 3D model retrieval. The user specifies a 3D query to the search engine (top left). The search engine then compares the query to every model in the database and returns snapshots of the models that are most similar to the query (right).

vector, and comparing two models amounts to the computation of the distance between two points in Euclidean space. Since the distance between two points is easy to compute, the underlying matching is efficient and the real-time requirements of a retrieval system can be satisfied. In practice, the shape descriptor is incorporated into the search engine as shown in Figure 1.2. In a pre-processing step the shape descriptor of each model in the database is computed (step 1). Then, at run time, a query is presented to the system, its shape descriptor is computed (step 2), the descriptor is compared against the descriptors of the models in the database (step 3), and the database models that are most similar to the query are returned as matches.

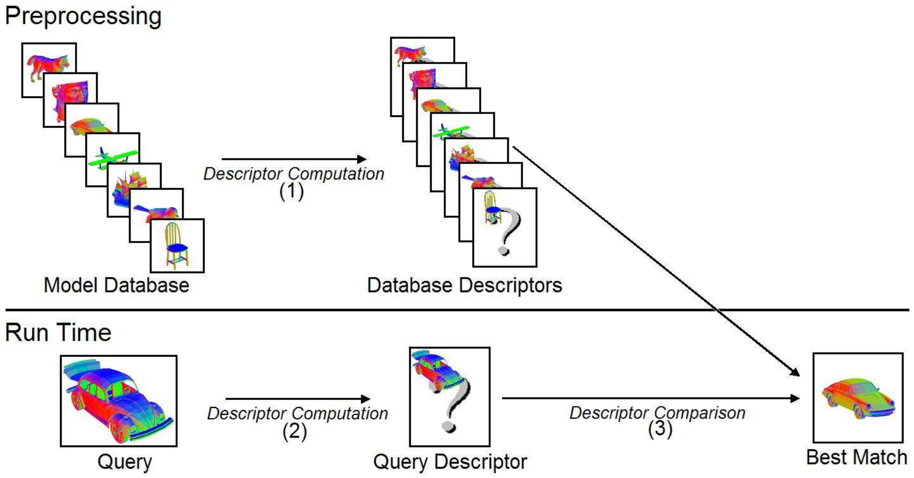


Figure 1.2: A shape descriptor is incorporated into the retrieval algorithm in the following manner: In the preprocessing stage the shape descriptor of every model in the database is computed (step 1). Then, when a query is presented to the database, its shape descriptor is computed (step 2). The query descriptor is compared against the database descriptors (step 3) and the closest matches are returned.

In this thesis, we provide contributions to the challenge of shape matching by presenting novel methods for representing and comparing 3D models. Specifically, the contributions of this thesis to the area of shape matching are four-fold. First, we present a new shape descriptor that is more discriminating than previous descriptors (Chapter 2). Second, we review a number of approaches for addressing the alignment challenge, providing new methods for matching 3D shapes across different transformations (Chapter 3). Third, we provide a general method for computing the symmetries of a 3D model and show how the symmetries can be used to augment shape information, providing a more discriminating representation of shape that is better suited for retrieval tasks (Chapter 4). Fourth, we show that the shape matching equation can be factored as the disjoint product of anisotropy and geometric comparisons — improving the matching performance of many shape metrics by facilitating the task of shape registration (Chapter 5).

Chapter 2

Shape Descriptors

2.1 Related Work

Traditional methods for retrieval of models from large repositories focus on designing a method for defining a measure of similarity between a query model and every target model in the database. The models in the database are then sorted by this measure of similarity, and the nearest models are returned as matches.

In the context of matching 3D shapes, the most common approach is to establish correspondences between the query model and the target model, and then to define the measure of similarity in terms of the distances between corresponding points. Two general classes of methods have been proposed that compute a measure of shape similarity by explicitly establishing such correspondences. The first approach is a local one, seeking to establish correspondences between pairs of points on the two models, and then defining the measure of shape similarity as the sum of the squared distances between pairs of points in correspondence [6, 7, 9, 16, 26, 35, 41, 69]. The second method is more general, decomposing a model into constituent parts, and then representing the model as a graph characterizing the relationship between the different segments [21, 45, 53, 54, 55]. Correspondences between two models can then be established using graph matching techniques, which simultaneously define the correspondences between the nodes of the two graph representations, and give the quality of the correspondences.

For both of these approaches, the establishing of correspondences is a difficult and time consuming task that needs to be performed on a per-pair-of-models basis. Thus, much of the necessary computation needs to be performed at run time, once a query is specified. This makes these methods impractical for the retrieval of models from large databases, where efficient comparison is essential.

The computational complexity of establishing correspondences between models has motivated a large body of research in the area of *shape descriptors*. The general approach of these methods is to define a mapping from the space of models into a fixed-dimensional vector space, and then to define the measure of similarity between two models as the distance between their corresponding descriptors [4, 8, 13, 17, 22, 27, 44, 63]. This approach has the advantage of addressing the correspondence problem on a per-model basis, allowing for the computation of descriptors in an offline process. Thus, correspondences are established not between two models, but between a single model and the coefficients of a fixed dimensional vector. Then, at run time, the descriptor of the query is computed and compared against the (pre-computed) descriptors of all the models in the database, giving rise to methods that can satisfy the efficiency requirements of interactive search. We give a detailed description of a number of these types of shape descriptors below, and a more general survey of shape descriptors can be found in [3, 37, 46, 60].

Extended Gaussian Image (EGI)

The extended Gaussian image is a shape descriptor that represents a 3D model by a spherical function. The EGI was initially proposed by Horn in [22] and is obtained by having each triangle vote on the bin corresponding to its normal direction, with a weight equal to the area of the triangle.

The extended Gaussian image has several important properties that make it useful for shape analysis and matching. First, the EGI is invariant to translation. Second, the EGI scales and rotates with the model. Third, in the case that the initial 3D shape is convex, the EGI is an invertible representation.

Complex Extended Gaussian Image (CEGI)

The complex extended Gaussian image is a generalization of the EGI, proposed by Kang *et al.* in [27]. Rather than just voting with a real value equal to the area of the triangle, this method votes with a complex number whose amplitude is equal to the area of the triangle and whose complex phase is equal to the normal distance of the triangle from the origin. This approach results in a representation of a 3D model that rotates and scales with the 3D model, and which exhibits a simple phase-shift when acted on by translation. Thus, it is particularly well suited for applications in which one would like to register two similar models that are in different poses, as the challenge of solving for the optimal rotation and translation can be decoupled by first solving for the optimal rotation using the complex norm of the CEGI (which is invariant to translation), and then subsequently solving for the optimal translation — effectively decomposing a 6D optimization problem into two independent 3D optimization problems.

Shape Histograms

Motivated by the challenge of using shape matching techniques to address the challenge of protein matching, Ankerst *et al.* [4] developed three different methods for representing 3D models in terms of the distribution of surface points as a function of distance from the center of mass and spherical angle. When only the distance from the surface is used the

Shells descriptor is obtained, when only the spherical angle is used the *Sectors* descriptor is obtained, and when both are used the *Shells and Sectors* descriptor is obtained.

Shells: The Shells descriptor represents a 3D model by a one-dimensional histogram, giving the distribution of distances of surface points from the center of mass. This representation is invariant to rotation since the distance of a point from the center of mass does not change when the model is rotated about its center. While scale transformations act non-trivially on the representation, scaling the domain of the representation rather than the histogram values, a scale-normalized representation can be obtained using well-established methods that we will describe in Chapter 3. In Chapter 3 we will also show that the shells descriptor is a specific instance of an approach for obtaining rotation-invariant representations of 3D models, and can be generalized to obtain a two-dimensional rotation-invariant representation with improved retrieval performance.

Sectors: The Sectors descriptor represents a 3D model by a spherical histogram, giving the distribution of surface points as a function of spherical angle. This representation scales and rotates with the model and exhibits no information loss when the initial model is star-shaped with respect to the origin.

Shells and Sectors: Combining the Shells and Sectors representation, Ankerst *et al.* provide a shape descriptor that represents a 3D model by a collection of spherical functions. Each spherical function is obtained by intersecting the model with a thin spherical shell centered at the origin and then computing the Sectors representation of the intersection of the model with the shell. The resultant descriptor gives rise to a three-dimensional representation that rotates with the model.

Shape Distribution (D2)

In [44], Osada *et al.* present a generalization of the Shells method by generating a histogram of distances between pairs of points on the surface of a model. Similar to the Shells representation, the D2 descriptor is a one-dimensional, rotation-invariant representation of 3D shapes. Furthermore, the binning of distances between pairs of points on the surface of the model results in a shape representation that is also invariant to translation. In Chapter 3 we will show that the D2 descriptor can be generalized to obtain a two-dimensional representation with improved retrieval performance.

Spherical Extent Function

Initially described in [10] though first used in database retrieval applications by Vranic et al. in [63], this descriptor represents a 3D model by a spherical function giving the maximal distance from the center of mass as a function of spherical angle. Similar to the Sectors descriptor, this function scales and rotates with the model and is invertible for star-shaped models.

When comparing the spherical extent functions of two different models, the measure of similarity is defined as the sum of distances, along rays through the origin, that points on the surface of one model need to be moved in order to lie on the surface of the second model. Thus, comparing two models with this descriptor gives a measure of the amount of "work" that needs to be performed in order deform one 3D model into another.

Radial Spherical Extent Function

Using the methodology of Ankerst *et al.*, Vranic proposes a method [62] for obtaining a higher-dimensional shape representation by combining the spherical extent function with the Shells representation. The resultant descriptor, obtained by computing the spherical extent function of the restriction of the model to concentric shells about the origin, gives rise to a representation of a 3D model as a collection of spherical functions that rotates with the shape.

In order to be effective as a representation of a 3D model, the shape descriptor needs to be discriminating, effectively differentiating between similar and dis-similar models. To this end, it is desirable that the shape descriptor be not only invertible but also an isometric representation. That is, the mapping from the space of shapes to the space of shape descriptors should have the property that the distance between the descriptors of two models gives a meaningful measure of the similarity of the underlying shapes.

In our research, we have found that while many shape descriptors are invertible, they do not approximate isometries. As a result the shape metric defined by the Euclidean distance between these shape descriptors does not provide sufficiently effective discrimination between models. In this chapter, we present a new shape descriptor having the property that the distance between the shape descriptors of two models corresponds to the amount of work needed to transform one model into the other. We begin by reviewing the sum of squared distances metric which characterizes the amount of work needed to transform one model into another. We then provide a novel method for analytically computing this metric and introduce a shape descriptor that approximates this metric. We conclude by providing experimental results that demonstrate the empirical limitations of the previous shape descriptors and show that the matching performance obtained using our new shape descriptor is comparable, in efficacy, to the matching performance obtained with the sum of squared distances shape metric.

2.2 Shape Similarity

The notion of shape similarity that we would like to use is based on the sum of squared distances (SSD) for models aligned in the same coordinate system. Specifically, we define the distance between two models as the sum of squared distances from every point on one surface to the closest point on the other, and vice-versa. This definition of shape similarity gives a measure of the extent to which each model is a subset of the other and is commonly used in shape registration techniques such as the Iterative Closest Point algorithm presented by Besl and McKay[9].

While a direct approach for computing the minimum SSD would require a complex integration over the surfaces of the models, we present a new method for computing this distance that is easy to implement. For each model M in the database, we represent the model by two voxel grids, R_M and E_M. The first voxel grid, R_M, is the rasterization of the boundary, with value 1 at a voxel if the voxel intersects the boundary, and value 0 if it does not. The second voxel grid, E_M is the (square) Euclidean Distance Transform of the boundary, with the value at a voxel equal to the square of the distance to the nearest point on the boundary. In order to compare two models M and N we simply set the distance between the two of them to be equal to:

$$d(M,N) = \langle R_M, E_N \rangle + \langle E_M, R_N \rangle,$$

the dot product of the rasterization of the first model with the squared distance transform of the second, plus the dot product of the rasterization of the second model with the squared-distance transform of the first. The dot product $\langle R_M, E_N \rangle$ is equal to the integral over the surface of M of the square distance transform of N. Thus, it is precisely equal to the minimum sum of squared distances that points on the surface of M need to be moved in to order to lie on the surface of N. Figure 2.1 demonstrates this process for computing the similarity between two models.

While this method provides a direct means for computing the minimum SSD, it cannot be used directly to design a shape descriptor, as the method for comparison involves

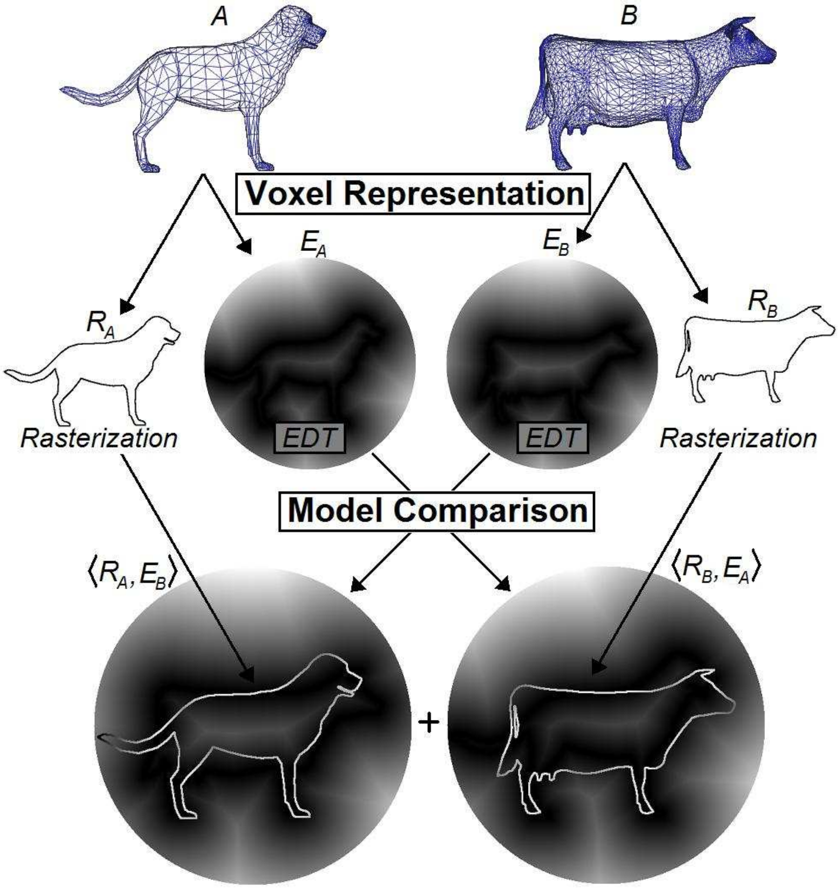


Figure 2.1: Two models are compared by computing the voxel rasterization and squared distance transform of each one, and then defining the distance measure of model similarity as the dot product of the rasterization of the first with the distance transform of the second, plus the dot product of the distance transform of the first with the rasterization of the second. The resultant value is equal to the minimum sum of squared distances that points on each model need to be moved in order to lie on the other model.

summing the asymmetric dot products of two vectors rather than computing Euclidean distances. Consequently, algorithms for nearest neighbor search (e.g. [25]) cannot be used to provide efficient retrieval, limiting the practical efficacy of this method. In the next section we show that this dot-product representation can be approximated with a shape descriptor, so that the minimum sum of squared distances between two surfaces can be approximated by the Euclidean distance between their corresponding shape descriptors.

Thus, we provide a matching method that implicitly establishes point correspondences while maintaining the simplicity of a vector-based representation.

2.3 Gaussian Euclidean Distance Transform

Since we compare two models by computing the distance between their shape descriptors, we would like to use a 3D voxel representation that describes not only where the points on the model are, but also how far an arbitrary point is from the surface of a model. Furthermore, the values of the voxel grid should fall off to zero for voxels further from the model, allowing us to treat the voxel grid as a sampling of a compactly supported function and to restrict the domain over which we integrate. To address these issues we define the voxel grid as a sampling of an exponentially decaying Euclidean Distance Transform. In particular, given a model S we define its *Gaussian Euclidean Distance Transform* to be the function $GEDT_S$ with:

$$GEDT_S(x) = \exp\left(-EDT_S^2(x)\right)$$

where $EDT_S(x)$ is the Euclidean Distance Transform, giving the distance from x to the nearest point on the model S.

The advantage of this shape descriptor are three-fold. First, iso-surfacing methods such as marching cubes [38] can be used to get back the surface of the model, so that the shape descriptor is invertible. Second, the L_2 difference between two Gaussian Euclidean Distance Transforms gives a measure of the proximity of the two underlying surfaces to each other, providing a mapping from shape-space to descriptor-space that approximates an isometry. Finally, this descriptor allows us to compare across a wide class of models, including models that are not topologically consistent, models that have cracks, and models with flipped triangles.

In order to maintain the context of spherical shape descriptors, the Gaussian Euclidean Distance Transform is represented by a collection of spherical functions obtained by restricting the 3D function to concentric spheres about the origin. That is, given a shape descriptor F defined in the Cartesian coordinate system, we reparameterize the function in terms of radius and spherical angle to obtain:

$$\begin{array}{rcl} F:\mathbb{R}^3 \to \mathbb{R} & \longrightarrow & \left\{F_r : S^2 \to \mathbb{R}\right\}, r \in [0,1] \\ F_r(\vec{v}) & = & F(r\cdot\vec{v})\sqrt{4\pi r^2} \end{array}$$

where the factor of $\sqrt{4\pi r^2}$ accounts for the change of variable, so that given two descriptors F and G, we get:

$$\|F-G\|^2 = \int_0^1 \|F_r - G_r\|^2 \, dr.$$

2.4 Retrieval Performance

One of the motivations for the design of the Gaussian Euclidean Distance Transform is the limitation of earlier techniques in defining shape descriptors whose difference corresponds to the distance between the underlying shapes. The Gaussian Euclidean Distance Transform, by contrast, was designed to provide a measure of similarity that represents the amount of work needed to transform one model into another. In order to evaluate how well the Gaussian Euclidean Distance Transform approximates the underlying minimum SSD metric, we compared the retrieval performance of our shape descriptor, with the retrieval performance obtained when model similarity was computed by explicitly

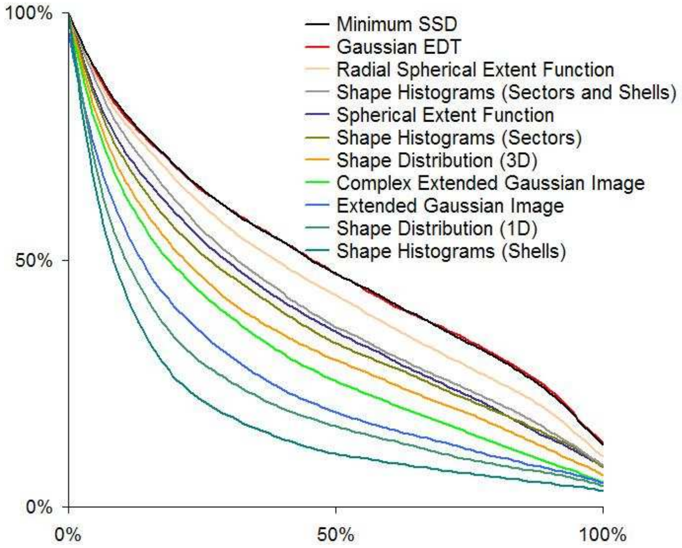


Figure 2.2: Comparison of the precision of different shape descriptors in classification experiments run with the test dataset of the Princeton Shape Benchmark. Note that the minimum sum of squared distances metric gives the best matching results with performance that is very closely approximated by the Gaussian Euclidean Distance Transform descriptor.

evaluating the sum of squared distances shape metric.

In order to evaluate the retrieval performance of a given shape descriptor, we measure how well it classifies models within a test database. The database was provided by the Princeton Shape Benchmark [48], and consists of 1814 models decomposed into two groups of roughly 900 models, corresponding to training and test datasets. Each group is provided with a classification, associating each of the models to one of roughly 90 distinct classes. Classification performance was measured using precision versus recall plots, which give the percentage of retrieved information that is relevant as a function of the percentage of relevant information retrieved. That is, for each target model in class C and any number K of top matches, "recall" represents the ratio of models in class C returned within the top K matches, while "precision" indicates the ratio of the top K matches that are in class C. Thus, plots that appear shifted up correspond to methods with superior matching performance.

We computed the precision versus recall results for the test dataset using the minimum SSD metric, the Gaussian Euclidean Distance Transform descriptor, and the descriptors reviewed in Chapter 2.1. The plots for these retrieval experiments are shown in Figure 2.2. In order to address the alignment problem in a uniform manner, we normalized all the models for translation and scale using center of mass and mean variance, as described in Chapter 3. For all but the Shells and D2 descriptors, we solved for rotation by computing the difference between descriptors at all rotations, and used the measure of similarity obtained at the optimal rotation, as described in Appendix A. (The Shells and D2 descriptors are rotation-invariant by design and hence we do not need to solve for the optimal rotation. However, at the end of Chapter 3, we show that both the Shells and D2 descriptors can be obtained from more descriptive, rotation-varying, 3D histograms. In order to decouple the issue of discriminability from the issue of alignment, Figure 2.2 also shows the retrieval results for these histograms.)

The results shown in Figure 2.2 validate the fact that for a shape descriptor to be effective it not only has to provide a descriptive representation of a 3D model, but it also has to be structured in such a manner so that the distance between two descriptors corresponds to a meaningful measure of model similarity. For example, if we consider many of the histogram based descriptors, we find that often these descriptors contain enough information to reconstruct the model. Specifically, the EGI and Complex EGI are invertible for convex models and the Sectors and Shells representation can be used to reconstruct the model (up to the resolution of the bins). However, the normed difference between two such descriptors only compares the values within each bin. Thus, the normed difference between these histogram descriptors is not effective at capturing the intra-class variations that would result in votes being cast into nearby bins. While these limitations could be addressed by using a non-normed measure of similarity, such as the Earth Mover's Distance [50], the resulting comparison becomes much more expensive for spherical and 3D histograms and, as a result, matching can no longer be performed in real time.

Similarly, the Spherical Extent Function captures the maximal extent of the model along rays through the origin. Thus, the normed difference between the Spherical Extent Functions of two different models gives a measure of how far, along the fixed rays, the models need to be deformed in order to transform one model into the other. However, this descriptor does not take into account deformations in the tangential directions and as a result its performance is noticeably worse than the performance of the SSD metric.

The Gaussian Euclidean Distance Transform, by contrast, is designed to give a measure of similarity capturing the proximity of one surface to another. A surface point effects not only the voxel cell corresponding to its location, but also many adjacent voxels (specifically, the voxels in the Voronoi cell associated to the point). As a result, the Euclidean distance between two Gaussian Euclidean Distance Transforms gives a meaningful characterization of the distance between two surfaces, and we find that the precision of this shape descriptor is nearly equivalent to that of the minimum SSD metric. Thus, the Gaussian Euclidean Distance Transform provides an efficient and effective method for representing 3D models — providing the simplicity of a vector-based representation without sacrificing discriminability.

Chapter 3

Shape Registration

One of the principal challenges faced in the area of shape matching is that in many applications, a shape and its image under a similarity transformation are considered to be the same. Thus, the challenge in comparing two shapes is to find the best measure of similarity over the space of all transformations. Three different methods have been proposed to address this challenge:

- **Exhaustive Search**: In order to match two shapes, the shapes are compared at every transformation and the measure of similarity at the optimal transformation (i.e. the one minimizing the similarity measure) is used as the measure of shape similarity. While this approach provides the correct answer, it is often too slow to be of practical use in retrieval tasks.

- **Normalization**: Each shape is placed into a canonical coordinate frame and two shapes are assumed to be optimally aligned when each is in its own frame. In some cases this method can be proven to provide the optimal alignment, thereby giving the optimal measure of similarity without incurring the cost of exhaustive search. However, in the case that normalization does not provide the optimal alignment, this approach hampers retrieval performance because, regardless of the shape metric used, comparing two models at the wrong alignment results in an inaccurate measure of similarity.

- **Invariance**: Shapes are described in a transformation-invariant manner, so that any transformation of a shape will be described in the same way, and the best measure of similarity is obtained at *any* transformation. In general, this method obtains a transformation-invariant representation by discarding alignment-dependent shape information, resulting in smaller shape representations that require less storage and can be compared more efficiently. However, it is often the case that these methods also discard information that is not dependent on the alignment of the models, and the resulting representation is less discriminating.

In this chapter we review a number of approaches for addressing the alignment issue. We show that traditional methods for the normalization of translation and scale can be shown to be provably optimal, while methods for rotation normalization do not have this property. To this end, we review the limitations of traditional methods for rotation-normalization, we describe alternate methods for obtaining rotation-invariant representations, and we present new methods for improving some of the existing approaches.

3.1 Analysis of Related Work

3.1.1 Translation and Scale Normalization

Initial work in 3D surface alignment is motivated by the challenge of aligning point sets: Given two point sets $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ and $Q = \{q_1, \ldots, q_n\} \subset \mathbb{R}^d$ what are the optimal similarity transformations S and T (consisting of translation, scale, and rotation) that minimizes the sum of square distances:

$$\sum_{i=1}^{n} \|S(p_i) - T(q_i)\|^2?$$

Methods for solving for the transformations prove that translation, scale, and rotation components can be solved for independently [23, 24] and in this section we review the solutions for translation and scale.

Optimal Translation

Lemma 3.1.1 *Given two point sets, $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ and $Q = \{q_1, \ldots, q_n\} \subset \mathbb{R}^d$, the sum of squared distances is minimized when each point set is translated so that its centroid is at the origin.*

Proof: Suppose that we have translated the point sets P and Q so that the centroid of each is at the origin. It suffices to show that any (non-trivial) translation of the point sets can only increase the sum of square distances. To show this, we let v and w be any translation vectors and set $d(v, w)$ to be the sum of square distances between the translated point sets $P + v$ and $Q + w$:

$$\begin{aligned} D_{P,Q}(v, w) &= \sum_{i=1}^{n} \|(p_i - v) - (q_i - w)\|^2 \\ &= n\|v - w\|^2 + \sum_{i=1}^{n} \|q_i - p_i\|^2 - 2\left\langle v - w, \sum_{i=1}^{n} q_i - p_i \right\rangle. \end{aligned}$$

Since both P and Q are translated so that their centroid is at the origin, this gives:

$$D_{P,Q}(v, w) = n\|v - w\|^2 + \sum_{i=1}^{n} \|q_i - p_i\|^2$$

so that the minimum sum of squared distances is realized when $v = w$ and in particular, when P and Q are each translated to their centroid they are optimally aligned for translation. Since rotations and scale fix the origin, it follows that the optimal translation can be computed independent of scale or rotation. ▮

Optimal Scale

One of the difficulties with computing the optimal scale for aligning two point sets P and Q is that is not clear how to pose the problem. A direct approach would seek to minimize the sum of square distances over all scalings of the two point sets independently:

$$D_{P,Q}(s,t) = \sum_{i=1}^{n} \|sp_i - tq_i\|^2.$$

However, this function is always minimized at $s, t = 0$, giving a distance of 0 between the two point sets. Thus, the challenge of posing the scale problem is that as opposed to translation and rotation, scaling is not an isometry and as a result the "size" of a point set changes with scale.

In order to address this challenge, we pose the scale problem in a norm-preserving fashion, seeking to find the scales that minimize the distance between two point sets while satisfying the condition that the overall size of the point sets remains constant. This formulation of the problem is similar in motivation to the one presented by Horn *et al.* [24] and results in the same solution — two point sets are optimally scale aligned if each is normalized to have mean variance equal to one. We show this by first proving a more general statement for arbitrary vectors, from which the optimal scale result follows as a corollary.

Lemma 3.1.2 *Given two vectors $\vec{v}$ and $\vec{w}$, if each vector has norm equal to one, then the* *minimum sum of squared distances:*

$$F(s,t) = \|s\vec{v} - t\vec{w}\|^2$$

subject to the norm-preserving constraint:

$$C(s,t) = \|s\vec{v}\|^2 + \|t\vec{w}\|^2 = \|\vec{v}\|^2 + \|\vec{w}\|^2.$$

is realized when $s, t = 1$ *and the vectors are in fact optimally scale-aligned.*

Proof: Applying the method of Lagrange multipliers, it follows that the extrema of the function F, subject to the constraint $C(s,t)$, occur when:

$$st\|\vec{v}\|^2\|\vec{w}\|^2 - t^2\|\vec{w}\|^2\langle\vec{v},\vec{w}\rangle = st\|\vec{v}\|^2\|\vec{w}\|^2 - s^2\|\vec{v}\|^2\langle\vec{v},\vec{w}\rangle.$$

Assuming that $\langle\vec{v},\vec{w}\rangle \neq 0$ this reduces to:

$$s^2 = \frac{\|\vec{w}\|^2}{\|\vec{v}\|^2}t^2$$

and substituting back into the constraint $C(s,t)$ we get:

$$2t^2\|\vec{w}\|^2 = \|\vec{v}\|^2 + \|\vec{w}\|^2.$$

Thus, the function $F(s,t)$, subject to the constraint $C(s,t) = \|\vec{v}\|^2 + \|\vec{w}\|^2$ has an extremum at $s, t = 1$. Since the second derivative of F satisfies:

$$(\alpha,\beta)\nabla^2F(\alpha,\beta)^t = 2F(\alpha,\beta) \geq 0$$

it follows that at the optimal scale occurs when $s, t = 1$ and the vectors $\vec{v}$ and $\vec{w}$ minimize the sum of square differences, subject to the norm-preserving condition.

(Note that if $\langle \vec{v}, \vec{w} \rangle = 0$, we can substitute the constraint $C(s,t) = ||\vec{v}||^2 + ||\vec{w}||^2$ into F to obtain $F(s,t) = ||\vec{v}||^2 + ||\vec{w}||^2$ so that any values of s and t satisfying the constraint $C(s,t)$ also minimize the sum of squared distances. In particular, if $s, t = 1$ then F is minimized and the vectors $\vec{v}$ and $\vec{w}$ are optimally scale-aligned.) ∎

As a corollary of the above lemma, it follows directly that:

Corollary 3.1.3 *Given two point sets, $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ and $Q = \{q_1, \ldots, q_n\} \subset \mathbb{R}^d$. If the mean variance of each point set is equal to 1, the sum of squared distances*

$$F(s,t) = \sum_{i=1}^{n} ||sp_i - tq_d||^2,$$

subject to the norm-preserving constraint:

$$C(s,t) = \sum_{i=1}^{n} ||sp_i||^2 + ||tq_i||^2 = \sum_{i=1}^{n} ||p_i||^2 + ||q_i||^2$$

is minimized when $s, t = 1$ and hence the points are optimally scale-aligned.

These methods for translation and scale normalization have the property that they provide the optimal translation and scale for a single model, independent of the model it will be compared against. Thus, they provide a means for comparing two models at the optimal translation and scale without necessitating an exhaustive search over the space of all possible transformations.

It should be noted that while these methods are guaranteed to minimize the sum of squares distances independent of correspondence, they do not necessarily minimize the difference between two shape descriptors. However, we have found that in whole-object-to-whole-object shape matching applications, these normalizations provide a stable, near-optimal, solution for the transformation minimizing the distance between models.

3.1.2 Rotation

While methods for normalizing for translation and scale on a per-model basis have been described, analogous methods for normalizing for rotation do not exist. Methods for computing the optimal rotational alignment between two models [23, 24] depend on the existence of correspondences between two models, and do not provide a means for putting a single model into a canonical coordinate frame that guarantees optimal alignment for matching. In general, the rotation alignment problem is addressed in one of three ways: (1) exhaustive search, (2) normalization, or (3) invariance. In the following subsections we describe these approaches in more detail.

Exhaustive Search

For shape descriptors that represent a 3D model as either a spherical function, or a function in 3D, one approach for addressing the rotational alignment problem is to exhaustively search for the rotation/reflection minimizing the difference between descriptors. While the method described in Appendix A provides a means for implementing the exhaustive search, it is still too slow to be used in database retrieval applications, where the $O(b^4)$ correlation of a descriptor needs to be computed between the query and every descriptor in the database.

Rotation Normalization with PCA

The traditional method for addressing the alignment problem normalizes a model for rotation by using the principal axes of the model to place it into a canonical coordinate frame. In particular, given a triangulated surface S in 3D, the covariance matrix M is computed by setting

$$M_{i,j} = \sum_{T \in S} \int_T x_i x_j dx$$

where the integral is summed over all triangles in the model. Since this matrix is symmetric, singular value decomposition can be used to compute its eigenvalues (or principal directions), and the model is rotated so that the x-axis maps to the eigenvector with largest eigenvalue, the y-axis maps to the eigenvector with second largest eigenvalue, and the z-axis maps to the eigenvector with smallest eigenvalue.

In our research, we have found rotation normalization via PCA-alignment does not provide a robust normalization for many matching applications. The cause for this is two-fold: First, the eigenvectors are only defined up to a multiple of ± 1. Thus, there is ambiguity in choosing which direction of the eigenvector to choose, and matching performance is hampered if the wrong direction is chosen. Second, there is no guarantee that when two models are each aligned to their own principal axes then they are also optimally pair-wise aligned.

Exhaustive Search for Axial Ambiguity: One approach to addressing the ambiguity in the direction of the principal axes is to search over the 8 possible choices of axes and use the measure of similarity at the best alignment. While this brute force approach provides a resolution to the axial ambiguity, it comes at the cost of increased comparison time. At the end of this chapter we describe an implementation of this exhaustive search approach that gives the measure of model similarity at each of the eight possible axial flips without increasing the computational complexity of model comparison.

Normalizing for Axial Ambiguity: Another approach is to normalize for the ambiguity by using a consistent method for choosing the direction of the axis. Tal *et al.* [15] propose a method for resolving this ambiguity using forward weighting. For each of the three eigenvectors, $\{v_1, v_2, v_3\}$, they compute the area of the intersection of the model with the positive half-space $\{p \in \mathbb{R}^3 | \langle p, v_i \rangle \geq 0\}$ and align with v_i if the area is more than half the area of the model, and with $-v_i$ if the area is less. One of the difficulties of using this method in practice is that often the area in the positive and negative half-spaces are nearly equal, making this approach unstable. Thus, it is often the case that two models that are similar are not aligned similarly, resulting in hampered retrieval performance.

Figure 3.1 demonstrates an example where this method for disambiguating the direction of the principal axes can fail. Though the two models are similar, the direction for the x-axis is chosen inconsistently. As a result, any matching that would occur with the models at this alignment would make the models seem less similar than they are. In this example, the forward weighting approach fails because the area of the half of the model residing over the positive x-axis is nearly equal to the area of the half of the model residing over the negative x-axis.

While these methods provide different approaches to resolve axial ambiguity, they do not address the fact that in some cases the axes defined by PCA are inconsistent. An example of this misalignment is shown in Figure 3.2. Despite the fact that the models in each pair are similar, the principal axes defined by PCA are not consistent, resulting in the incorrect registration of the models in each pair. The figure shows two different types of misalignment that can occur. In the three pairs on the left, the models are misaligned because the x-, y-, and z-axes are permuted. Thus, though the principal axes are computed correctly, their ordering based on the magnitude of the corresponding eigenvalue results

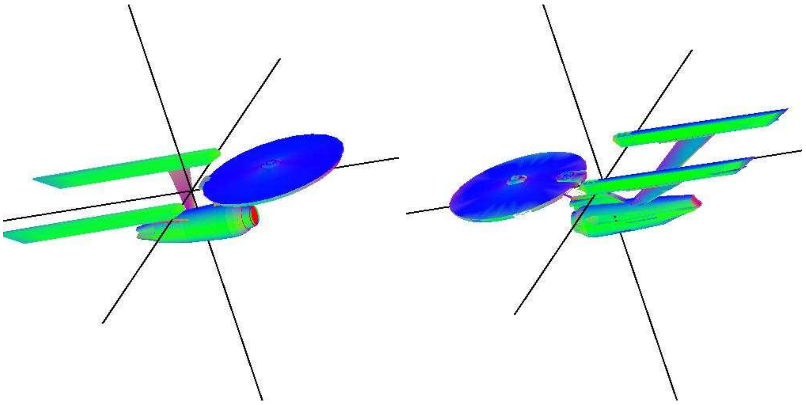


Figure 3.1: An example of a situation in which the forward weighting approach fails. Despite the fact that the two models are similar, the choice of the positive vs. negative direction for the principal axes is inconsistent and the two models are not correctly aligned.

in misaligned models. This type of misalignment can be addressed by searching over the six different permutations for the one minimizing the distance between models. The three pairs on the right demonstrate a more general example of the failure of PCA. For these models, the eigenvectors defined for one model do not overlap with the eigenvectors for the other, and no permutation of axes would give the correct alignment.

One of the reasons for the failure of PCA in aligning 3D models results from the fact that when the covariance matrix is degenerate, (i.e. eigenspaces are multi-dimensional) a unique set of principal axes cannot be established. There are two cases in which the covariance matrix can be degenerate: First, it is possible for all three eigenvectors to have the same eigenvalue. In this case the covariance matrix is a multiple of the identity and any rotation of the model will align the x-, y-, and z-axes with the principal directions. Second, it possible for two eigenvectors, v_1 and v_2 to have the same eigenvalue λ and the third eigenvector v_3 to have a different eigenvalue λ_3. In this case any linear combination of the vectors v_1 and v_2 will also be an eigenvector with eigenvalue λ, and if the model is rotated so that its principal axes align with the x-, y-, and z-axes, then any rotation about the axis v_3 will also align the principal axes of the model with the x-, y-, and z-

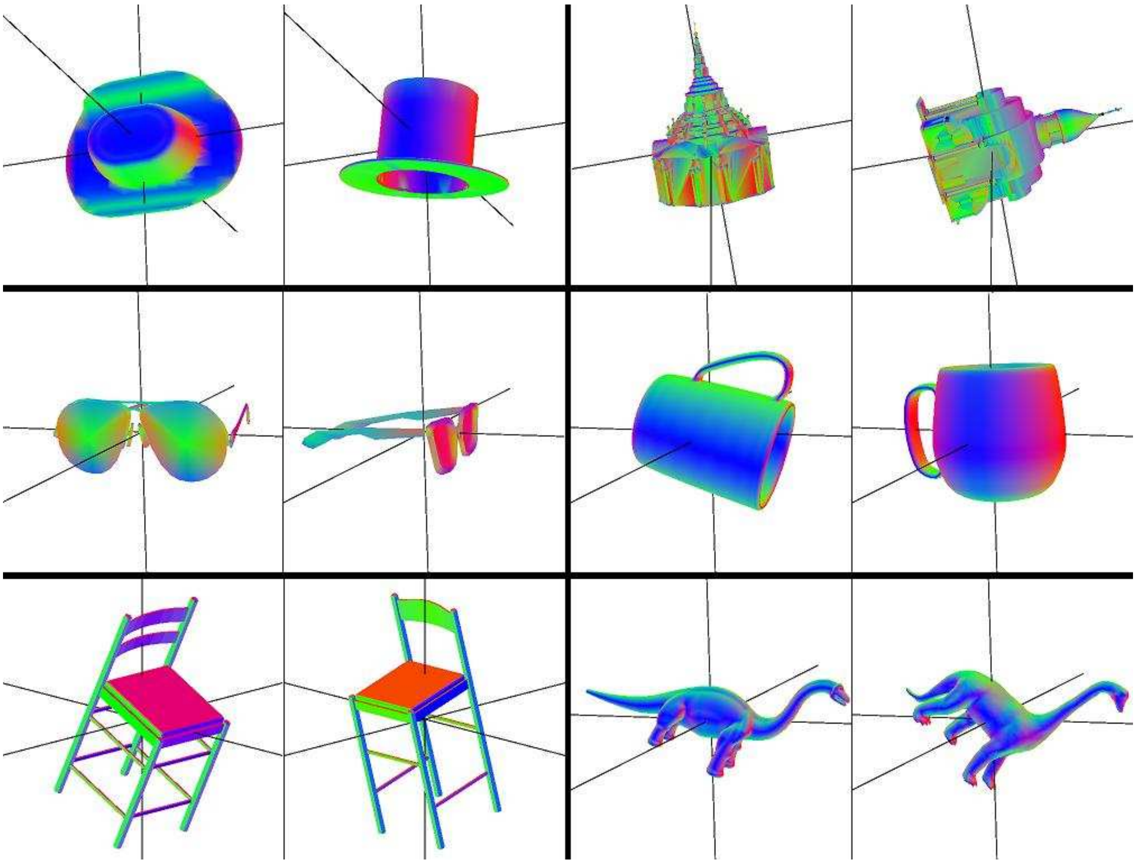


Figure 3.2: An example of a situation in which the PCA alignment fails. Despite the fact that the pairs of models are similar, the principal axes are inconsistent and the models in each pair are not correctly aligned.

axes. Table 3.1 shows the eigenvalues for the covariance matrix of each of the models in Figure 3.2. Note that for most pairs of models for which PCA fails, at least one of the models has the property that two of the eigenvalues have similar value.

Invariance to Rotation

In this section, we review two different methods for addressing the rotational alignment problem by using invariance. The first approach is a hybrid normalization-invariance approach, which uses PCA to establish one axis of alignment, and then gives a rotation-

Model	X-Axis	Y-Axis	Z-Axis	Model	X-Axis	Y-Axis	Z-Axis
Hat 1	0.5391	0.3690	0.0919	Church 1	0.3538	0.3518	0.2944
Hat 2	0.3788	0.3360	0.2852	Church 2	0.4235	0.2954	0.2811
Glasses 1	0.6739	0.2491	0.0770	Cup 1	0.4273	0.3201	0.2526
Glasses 2	0.5060	0.4608	0.0332	Cup 2	0.4063	0.3020	0.2917
Chair 1	0.4910	0.2589	0.2501	Dino 1	0.8412	0.1210	0.0379
Chair 2	0.5543	0.2257	0.2200	Dino 2	0.6981	0.2384	0.0635

Table 3.1: The eigenvalues for the different principal directions for each of the models shown in Figure 3.2. Note that for most pairs of models for which PCA fails, at least one of the models has the property that two of the eigenvalues of the covariance matrix have similar value.

invariant representation of a shape descriptor with respect to all rotations about that axis. The second approach describes a way for obtaining a rotation-invariant representation of a shape descriptor with respect to all rotations. Both approaches utilize the property that the power spectrum of a function – the collection of amplitudes of the different frequency components – discards phase and therefore is invariant to rotation.

Invariance to Axial Rotation: One approach that has been described ([16, 62, 63]) for addressing the limitations of PCA alignment uses the fact that in the case that only two of the eigenvectors have the same eigenvalue, the PCA approach can be used to define one of the principal axes unambiguously and the circular power spectrum can be used to obtain a representation that is invariant to rotations about that principal axis.

The key idea behind this approach is that the spherical harmonic basis functions can be factored as the product of functions of angles of elevation and azimuth:

$$Y_l^m(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)e^{im\phi}$$

where the P_l^m are the associated Legendre polynomials. Thus, if a spherical function is expressed in terms of its spherical harmonics

$$f(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{|m|\leq l} f_{l,m}Y_l^m(\theta,\phi)$$

then a rotation by an angle of ϕ_0 about the North pole maps the function f to a function with spherical harmonic decomposition

$$f(\theta,\phi+\phi_0)=\sum_{l=0}^{\infty}\sum_{|m|\leq l} f_{l,m}Y_l^m(\theta,\phi+\phi_0)=\sum_{l=0}^{\infty}\sum_{|m|\leq l} f_{l,m}e^{im\phi_0}Y_l^m(\theta,\phi).$$

It follows, therefore, that if we represent a spherical function by the complex norms of its harmonic coefficients:

$$f \longrightarrow \{\|f_{l,m}\|\}_{|m|\leq l},$$

we obtain a representation that is invariant to rotation about the North pole.

Since our primary concern is matching shapes, we briefly summarize some of the properties of this representation in so far as they relate to the comparison of different spherical functions.

- **Invariance**: By construction, the representation of a spherical function in terms of the complex norms of its spherical harmonic coefficients is invariant to rotation about the North pole. In particular, it follows that this representation is also invariant to permutations of the two principal axes perpendicular to the North pole.

- **Lower Bound**: Given two spherical functions f and g, the L_2 difference between the collection of their complex norms is a lower bound for the L_2 difference between the two functions, taken over all rotations in the plane perpendicular to the North pole. This property follows from the fact that given any two vectors v and w

in a Euclidean space, it is always true that the L_2 difference between the vectors is always at least as big as the L_2 difference between their norms:

$$\|v - w\| \geq \big|\|v\| - \|w\|\big|.$$

Thus, given two spherical functions f and g, expressed in terms of their spherical harmonic decomposition as:

$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} f_{l,m} Y_l^m(\theta,\phi) \qquad g(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} g_{l,m} Y_l^m(\theta,\phi)$$

we can use the fact that the Y_l^m are orthogonal to show that the L_2 difference between the complex norm representations is a lower bound for the L_2 difference between the spherical functions:

$$\|f - g\|^2 = \sum_{l=0}^{\infty} \sum_{|m|\leq l} \|f_{l,m} - g_{l,m}\|^2 \geq \sum_{l=0}^{\infty} \sum_{|m|\leq l} \left(\|f_{l,m}\| - \|g_{l,m}\|\right)^2.$$

Furthermore, since the complex norm representation is invariant to rotations about the North pole, it follows that the L_2 difference between the complex norm representation is a lower bound for the L_2 difference between the spherical functions, taken over all possible rotations about the North pole.

- **Dependence on PCA**: While this hybrid normalization/invariance approach provides a representation of a spherical function that is invariant to rotations about the North pole, it depends on PCA alignment to effectively determine what the North pole should be. Thus, it only provides a useful rotation-invariant representation of the shape in the case that the North pole can be determined robustly.

- **Information Loss**: Because this approach treats each harmonic coefficient independently, the representation does not capture information characterizing how the different harmonic components of a single function align with respect to each other. As an example, if the spherical function f is the sum of two different harmonics:

  $$f(\theta,\phi) = f_{l,m}Y_l^m(\theta,\phi) + f_{l',m'}Y_{l'}^{m'}(\theta,\phi)$$

  then applying a North pole rotation of ϕ_0 to only one of the two coefficients results in a new spherical function g:

  $$g(\theta,\phi) = f_{l,m}Y_l^m(\theta,\phi+\phi_0) + f_{l',m'}Y_{l'}^{m'}(\theta,\phi)$$

  that is not a rotation of f but still has the same complex norm representation.

- **Choosing the Axis of Rotation-Invariance**: This approach provides a representation of a spherical shape descriptor that is invariant to rotations about the North pole. In implementing this method, models can be consistently rotated so that any of the principal axes are mapped to the North pole and a choice can be made with regards to which principal axis should be the axis of rotation-invariance. In general, the axis is chosen so that it is the most distinguished. Recalling that a principal axis is well-distinguished if it is different from either of the other two motivates choosing either the principal axis with the largest principal eigenvalue or the one with the smallest principal eigenvalue. The choice between using the smallest and largest is generally motivated by the database, so that if most of the models tend to be cylindrical the principal axis with largest eigenvalue is chosen, and if most of the models tend to be disk-shaped the principal axis with smallest eigenvalue is chosen. If we consider as an example the Princeton Shape Benchmark [48], we find that the average triple of eigenvalues is $(6.57, 1.66, 0.32)$.

Thus, models in this database tend to be cylindrical, more surface perturbation is required to transform the ordering of the two largest principal axis than for the two smallest one, and the principal axis with largest eigenvalue should be used as the axis of rotation-invariance. To test this empirically, we computed the precision versus recall for the six rotation varying shape descriptors described in Chapter 2: the Extended Gaussian Image, the Complex Extended Gaussian Image, the Sectors representation, the Sectors and Shells representation, the Radial Spherical Extent Function, and the Gaussian Euclidean Distance Transform. For each descriptor we computed the complex norm representation where the largest (x-axis), middle (y-axis), and smallest (z-axis) principal axis was chosen as the axis of rotation-invariance. The results of the experiment are shown in Figure 3.3. Note that as expected, for all representations, the middle (y-axis) principal axis is the least distinguished and hence gives the worst results. Moreover, since the average eigenvalues for the database indicate that more of the models are cylindrical, the largest (x-axis) principal axis gives the best retrieval results.

Invariance to General Rotation: Another approach, described in [10, 17, 30, 31, 36, 39], generalizes the power spectrum notion defined for functions whose domain is a circle to a notion of power spectrum defined for functions whose domain is a sphere. This makes it possible to obtain a representation of a spherical function that is invariant to all rotations and hence provides a method for matching 3D models that is independent of the robustness of PCA.

The key idea of this approach is based on the fact that the spherical harmonic basis

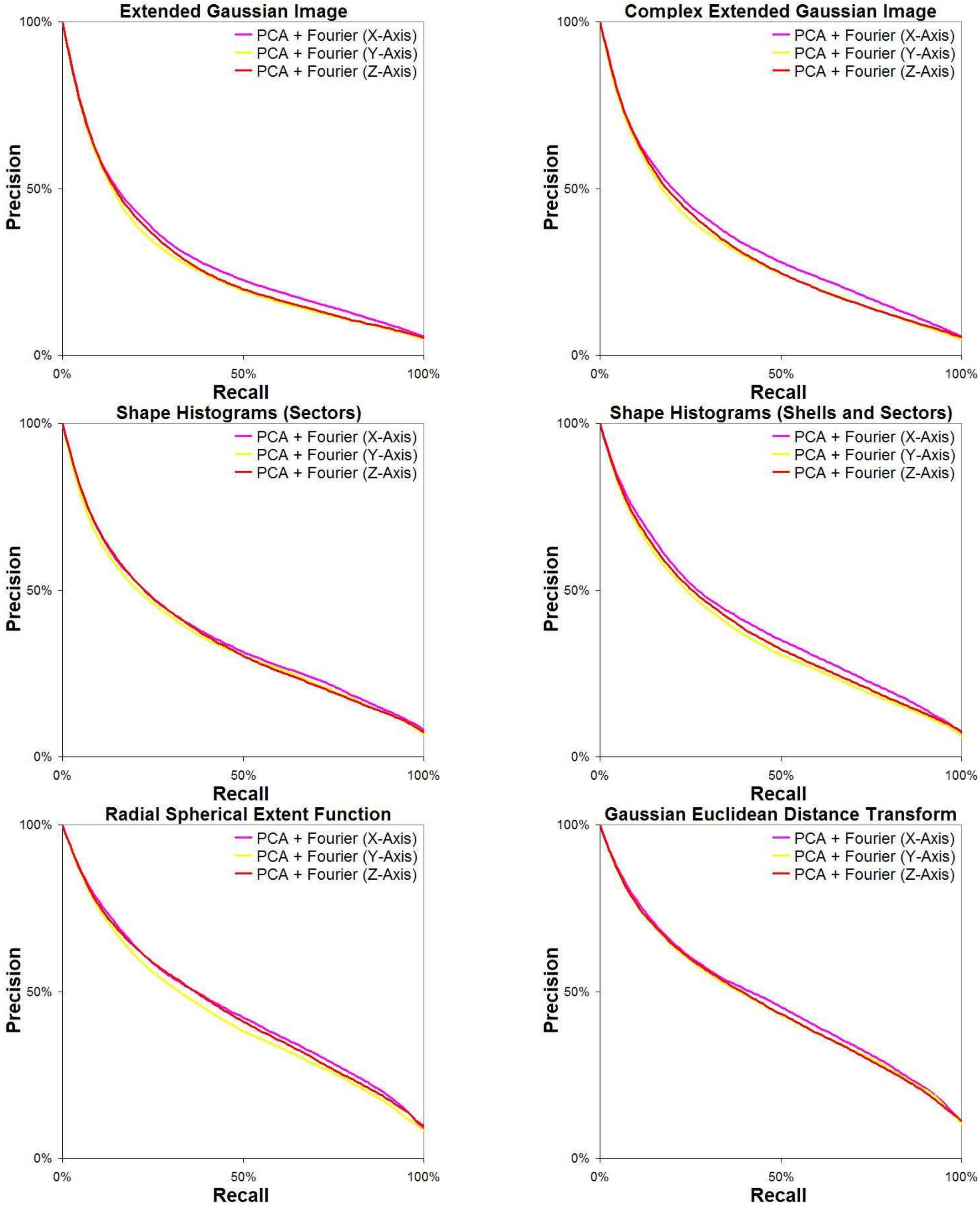


Figure 3.3: Retrieval results with different shape representations, demonstrating the effect of selecting each of the different principal axes as the axis of rotation-invariance for the complex norm representation.

functions provide a representation for the group of rotations (see Appendix A). In particular, if f is a function defined on a sphere, then f can be expressed as the sum of its projections onto the subspaces V_l:

$$f(\theta,\phi)=\sum_{l=0}^{\infty} f_l(\theta,\phi) \qquad \text{with} \qquad f_l=\pi_l(f)=\sum_{|m|\leq l} f_{l,m}Y_l^m$$

where π_l is the projection onto the l-th irreducible representation V_l, the $f_{l,m}$ are the spherical harmonic coefficients of f, and Y_l^m are the spherical harmonics forming the basis for the l-th irreducible representation V_l. Using the fact that the function subspace V_l is a representation space for the rotation group, and using the fact that rotations act linearly on the space of functions, for any rotation ξ we have:

$$\xi(f)=\xi\left(\sum_{l=0}^{\infty}\pi_l(f)\right)=\sum_{l=0}^{\infty}\pi_l\left(\xi(f)\right).$$

Thus, if we use the fact that rotations do not change the L_2 norm of a spherical function — $\|f\|=\|\xi(f)\|$ for all rotations ξ — and we represent a spherical function by the size of its projections onto the representations V_l:

$$f\longrightarrow\left\{\|\pi_l(f)\|\right\}_{l=0}^{\infty}=\left\{\sqrt{\sum_{|m|\leq l}\|f_{l,m}\|^2}\right\}_{l=0}^{\infty}$$

we obtain a representation that is invariant to all rotations.

Since our primary concern is matching shapes, we briefly summarize some of the properties of this representation in so far as they relate to the comparison of different spherical functions.

- **Invariance**: By construction, the representation of a spherical function in terms of the sizes in its frequency components is invariant to all rotations.

- **Lower Bound**: Given two spherical functions f and g, the L_2 difference between their power spectra is a lower bound for the L_2 difference between the two functions, taken over all possible rotations. As with the complex norm representation, we can show this by using the vact that the L_2 difference between two vectors is always at least as big as the L_2 difference between their norms.

  In particular, given two spherical functions f and g, expressed in terms of their frequency components as:

$$f(\theta,\phi)=\sum_{l=0}^{\infty} f_l(\theta,\phi) \qquad g(\theta,\phi)=\sum_{l=0}^{\infty} g_l(\theta,\phi)$$

  we can use the fact that the representations V_l are orthogonal to show that the L_2 difference between the power spectrum representations is a lower bound for the L_2 difference between the spherical functions:

$$\|f-g\|^2=\sum_{l=0}^{\infty}\|f_l-g_l\|^2 \geq \sum_{l=0}^{\infty}\left(\|f_l\|-\|g_l\|\right)^2.$$

  Furthermore, since the power spectrum is invariant to rotation, it follows that the L_2 difference between the power spectrums is a lower bound for the L_2 difference between the spherical functions, taken over all possible rotations.

- **Optimality**: The key idea of the power spectrum approach is to decompose a spherical function into components on which rotations act independently and then to obtain a rotation-invariant representation by storing the L_2 norm of each component. To this end the quality of the rotation-invariant representation is dependent on the fine-ness of the resolution of the space of functions into rotation-independent components. Since the representation spaces V_l are irreducible, it follows that no finer resolution into linear, rotation-independent subspaces is possible. Thus, from a linear perspective, the power spectrum representation gives an optimal rotation-invariant representation.

However, in shape matching applications the linearity condition is not necessary and finer resolutions can be obtained by decomposing each subspace V_l into orbits under the action of the rotation group – subsets (not subspaces) of V_l that are obtained by taking a function $f_l \in V_l$ and looking at the set of functions, obtained by applying all the different rotations to f_l. (We will discuss this in more depth at the end of the chapter.)

- **Information Loss**: Because this approach treats each frequency component independently, the representation does not capture information characterizing how the different frequency components of a single function align with respect to each other. Figure 3.4 shows a visualization of this for two spherical functions. The one on the bottom is obtained from the one on the top by applying different rotations to the different frequency component. Though the two functions differ by more than a single rotation, their power spectrum representations are the same.

  Furthermore, for each frequency component f_l, the power spectrum only stores the energy in that component. For $l \geq 2$ it is *not* true that if $\|f_l\| = \|g_l\|$ then there is a rotation ξ such that $\xi(f_l) = g_l$. Thus knowing only the norm of the l-th frequency component does not provide enough information to reconstruct the component up to rotation. (This form of information loss does not occur for the power spectrum representation of circular functions, as two circular functions with the same amplitude and frequency can only differ by a rotation.) Figure 3.5 shows

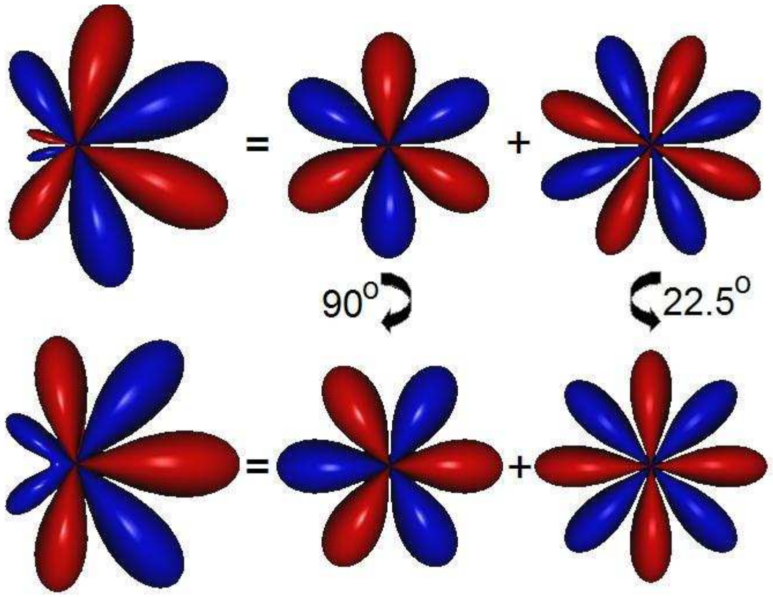


Figure 3.4: The bottom spherical function is obtained by rotating the different frequency components of the function on top by different angles. Although there is no rotation transforming the function on the top to the one on the bottom, their power spectrum representations are the same.

a visualization of this for three spherical functions. The functions are all of the same frequency and have the same amplitude but there is no rotation that can be applied to transform them into each other. In this example, all three are fourth order functions with norm equal to 1. Since the space of fourth order functions is 9-dimensional and since the space of rotations is 3-dimensional, the power spectrum ends up representing six dimensions of rotation invariant information by a single value. As a result, five dimensions of information are lost and the power spectrum representation cannot distinguish between the functions shown in Figure 3.5.

3.2 Improvements and Variations

We now present three novel methods for improving the matching performance of shape descriptors representing a 3D model by either a single spherical function or a collection

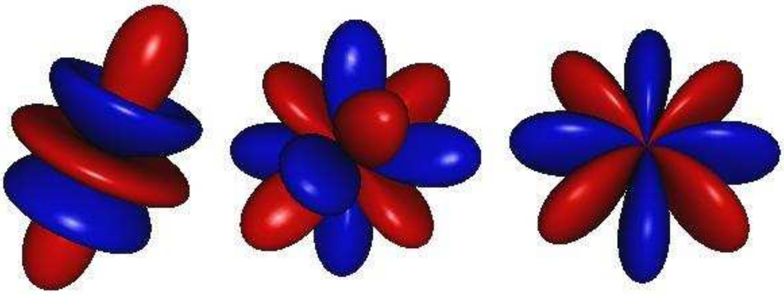


Figure 3.5: These three single-frequency ($l = 4$) spherical functions differ by more than rotation but have the same spherical power spectrum.

of spherical functions. The first method describes a way for comparing two PCA-aligned models, at each of the eight possible axial flips, without necessitating a corresponding eight-fold increase in compare time. The second method describe a simple way for obtaining an axial-flip-invariant representation of these shape descriptors. And the third method provides a way for improving the power spectrum representation by providing a finer resolution of spherical functions into rotation-invariant components, giving an improved measure of model similarity over all possible rotations, while maintaining the lower bound property of the power spectrum representation.

3.2.1 Exhaustive Axial Flip Alignment

Although a brute force approach for computing the measure of similarity over all permutations of axis flips would result in and eight-fold increase in the time complexity for comparing two models, spherical harmonics can be used to perform the exhaustive search without increasing the comparison time. The key idea behind this approach is that if a spherical function is expressed in terms of its spherical harmonics

$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} f_{l,m} Y_l^m(\theta,\phi)$$

then flipping the x-, y-, or z-axis changes the harmonics as follows:

- x-**flip**: Reflecting about the yz-plane so that the x-axis flips to the $-x$-axis sends each harmonic coefficient $f_{l,m}$ to its complex conjugate $\overline{f_{l,m}}$.
- y-**flip**: Reflecting about the xz-plane so that the y axis flips to the $-y$-axis sends the harmonic coefficient $f_{l,m}$ to $(-1)^{l+m} f_{l,m}$.
- z-**flip**: Reflecting about the xy-plane so that the z-axis flips to the $-z$-axis sends the harmonic coefficient $f_{l,m}$ to $(-1)^m \overline{f_{l,m}}$.

The action of the reflections is shown in Figure 3.6 which shows the real and imaginary components of the different harmonics, indicating that the different axial reflections act by multiplying the real and imaginary components of the harmonics by ± 1.

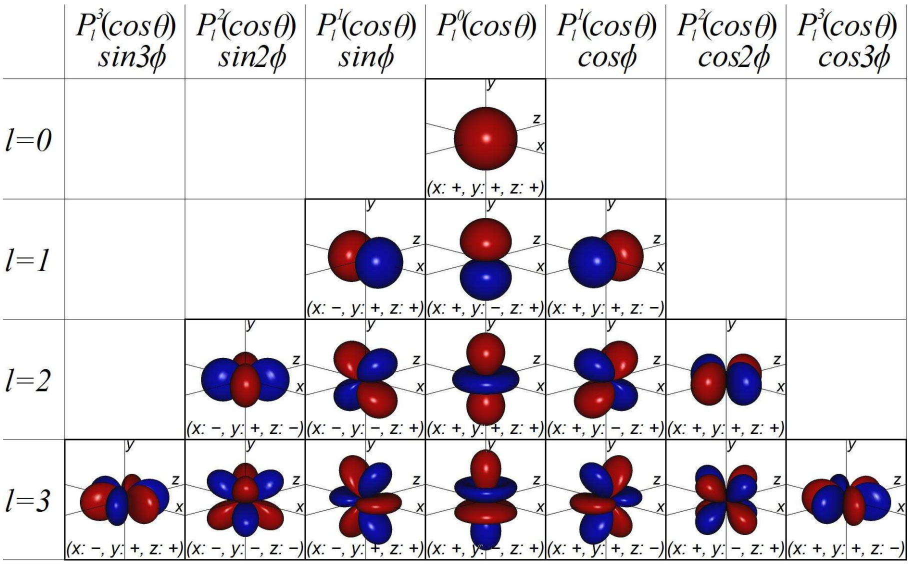


Figure 3.6: The real and imaginary components of the spherical harmonics frequencies, drawn by scaling points on the unit sphere in proportion to their value —drawing points with positive value in red and points with negative value in blue. The image shows the way in which reflections about the x-, y-, and z-axes act on the different harmonics by multiplying the real and imaginary components by ± 1.

Thus, in computing the dot product of f with g, we can consider the contribution of different harmonic components $f_{l,m}$ separately, depending on whether l and m are even or odd and if we are looking at the real or imaginary part of $f_{l,m}$. To do this, we define:

$$d_{pqr} = \sum_{l=0}^{2l+p<b/2} \sum_{|2m+q|\leq 2l+p} \mathrm{R}_r\left(f_{2l+p,2m+q}\right) \cdot \mathrm{R}_r\left(g_{2l+p,2m+q}\right)$$

where $p, q, r \in \{0, 1\}$ and R_r gives the real part of a complex number if $r = 0$ and the imaginary part if $r = 1$. (Note that computing d_{pqr} takes no more time than computing the dot product of f with g.) Then, for any transformation γ composed of axis flips, we can compute the dot product $\langle f, \gamma(g)\rangle$ by summing the values d_{pqr} with the appropriate sign. In particular, if we set:

$$\gamma_{ijk} = \begin{pmatrix} (-1)^i & 0 & 0 \\ 0 & (-1)^j & 0 \\ 0 & 0 & (-1)^k \end{pmatrix}$$

with $i, j, k \in \{0, 1\}$, then the expression for the L_2 difference between the function f with the axially-flipped function g is:

$$\|f - \gamma_{ijk}(g)\|^2 = \|f\|^2 + \|g\|^2 - 2 \sum_{p,q,r\in\{0,1\}} (-1)^{ir+j(p+q)+k(q+r)} d_{pqr}.$$

Thus, by computing the eight contributions d_{pqr} independently, we can reduce the computation of the dot product of f with the different axial flips of g to a signed summation of these eight contribution terms. As a result, computing the distance at all eight axial flips can be done with an additional cost of only 64 arithmetic operations, independent of the size of the shape descriptor.

3.2.2 Invariance to Axial Ambiguity

Using the fact that flipping the x-, y-, and z-axes results in a change of sign in the real and imaginary components of some of the spherical harmonic coefficients, we can obtain a representation that is invariant to axial ambiguity by storing only the absolute values of the real and imaginary components. Furthermore, since the above equations indicate that the contribution of the value d_{000} does not depend on the alignment of the axes, further discrimination can be obtained by storing the actual values of the real components of $f_{l,m}$ for the case that both l and m are even, and storing the absolute value for all other cases.

In addition to being invariant to axial flips, this method is also partially invariant to permutations of the axes. In particular, if we consider a rotation γ by $\pi/2$ (a permutation of the axes) in the plane perpendicular to the North pole, this rotation acts on the spherical harmonic by multiplying by a power of $\sqrt{-1}$:

$$\gamma\left(Y_l^m(\theta,\phi)\right) = Y_l^m(\theta,\phi+\pi/2) = Y_l^m(\theta,\phi)(\sqrt{-1})^m.$$

Thus, in the case that m is even, this amounts to multiplication by -1, so that storing just the absolute value of the real and imaginary coefficients is invariant to this type of rotation. In particular, this demonstrates that the absolute value representation is partially invariant to permutation of the axes that are perpendicular to the North pole.

3.2.3 Quadratic Resolution with PCA

One of the limitations of only storing the energies at the different frequencies of a spherical function is that it does not allow us to reconstruct the frequency components uniquely, up to rotation. In the past, this problem has been addressed by using *algebraic* methods to obtain additional rotation-invariants for the different frequency components[10, 36, 39].

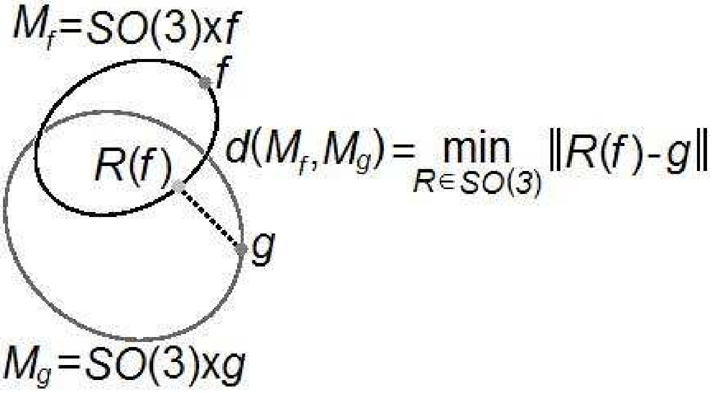


Figure 3.7: The minimum distance between two functions f and g, taken over the space of all rotations, is equal to the distance between the two manifolds M_f and M_g, where M_f and M_g are obtained by tracing out the image of the functions f and g under the action of the rotation group.

The difficulty with these approaches is that the derived 0-th order rotation-invariant tensor are often redundant and consequently, cannot be directly compared to obtain a lower bound for the minimum L_2 difference between two spherical functions.

We present a new *geometric* approach for computing orthogonal invariants and describe an implementation for the quadratic components of a spherical function. This approach is based on the idea that for a spherical function f, we can generate the manifold M_f, defined as the image of f under all rotations. For two spherical functions f and g, the minimum difference between f and g, taken over the space of all rotations, is precisely the distance between the two closest points on the manifolds M_f and M_g, (Figure 3.7). The goal then, is to be able to index these manifolds in such a way that the L_2 difference between two sets of indices is exactly the distance between the two manifolds.

The key idea of our approach is to use PCA to explicitly solve for the alignment that minimizes the L_2 difference between the constant and quadratic components of two spherical functions. This approach provides a representation of the constant and quadratic components that is invertible, up to rotation and eliminates the problem of intra-frequency loss in the second order components that is otherwise suffered by the power spectrum.

Theorem 3.2.1 *If f and g are two spherical functions consisting of only constant and second order harmonics, then the L_2 difference between the two is minimized when each is aligned to its own principal axes.*

Proof: Because f and g consist of only constant and second order terms, we can represent the two functions by symmetric matrices A and B where

$$f(v) = v^t A v \qquad \text{and} \qquad g(v) = v^t B v.$$

If we assume that A and B are already aligned to their principal axes we get:

$$A = \begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{pmatrix}$$

Thus, if ξ is any rotation we get:

$$\langle \xi^t(f), g \rangle = (\alpha - \beta)\text{Trace}(A\xi B\xi^t) + \beta \sum_{i,j=1}^{3} a_i b_j$$

where $\alpha = \int_{S^2} x^4 dx$ and $\beta = \int_{S^2} x^2 y^2 dx$ define the lengths and angles between the functions x_i^2 on the unit sphere. We would like to show that the dot product is maximized when ξ is a permutation matrix so that $\xi A \xi^t$ is diagonal.

Using the fact that the differentials of a rotation ξ are defined by $\xi\psi$ where ψ is a skew-symmetric matrix, it suffices to solve for:

$$0 = \frac{d}{dt}\Big|_{t=0} \text{Trace}(A(\xi + t\xi\psi)B(\xi^t - t\psi\xi^t)) = \text{Trace}(\xi^t A \xi(\psi B - B\psi)).$$

But ψ is a skew-symmetric matrix so that, $\psi B - B\psi$ is a symmetric matrix:

$$\psi B - B\psi = \begin{pmatrix} 0 & (b_2 - b_1)S_{12} & (b_3 - b_1)S_{13} \\ (b_2 - b_1)S_{12} & 0 & (b_3 - b_2)S_{23} \\ (b_3 - b_1)S_{13} & (b_3 - b_2)S_{23} & 0 \end{pmatrix}$$

Thus, if $\xi^t A\xi$ is a diagonal matrix then the derivative is zero, independent of the choice of ψ. Conversely, if the b_i are distinct and $\xi^t A\xi$ is not diagonal, we can always choose values for ψ_{12}, ψ_{13}, and ψ_{23} such that the derivative is non-zero, implying that if $\xi^t A\xi$ is not diagonal it cannot maximize the dot product. (Note that if $b_1 = b_2 = b_3$ then B is a constant multiple of the identity so that the dot product is independent of the choice of rotation. Similarly, if $b_i = b_j$ then rotations in the plane spanned by x_i and x_j also do not change the dot product.)

This shows that the L_2 difference between the functions f and g is at an extremum if and only if A and B are diagonal matrices. The minimum L_2 difference is then attained when $\sum a_i b_i$ is maximal. So, if $a_1 \geq a_2 \geq a_3$ then we must also have $b_1 \geq b_2 \geq b_3$, and the L_2 difference between f and g is minimized precisely when f and g are aligned to their principal axes. ∎

This theorem shows that the L_2 difference between the quadratic components of two spherical functions is minimized when the two functions are aligned with their principal axes. Thus, we can represent the constant and quadratic components by the three scalars $a_1 \geq a_2 \geq a_3$, where after alignment to principal axes:

$$f_0 + f_2 = a_1 x^2 + a_2 y^2 + a_3 z^2.$$

The indices (a_1, a_2, a_3) uniquely define the constant and quadratic function up to rotation, but because the functions $\{x^2, y^2, z^2\}$ are not orthogonal, they do not have the property that the L_2 difference between two sets of indices is the minimum of the L_2 differences between the two functions. To address this, we fix an orthonormal basis $\{v_1, v_2, v_3\}$ for the span of $\{x^2, y^2, z^2\}$ and represent the function $f = f_0 + f_2$ by the three scalars $R^{-1}(a_1, a_2, a_3)$, where R is the matrix whose columns are the orthonormal vectors v_i. This provides the desired orthogonal indexing for the constant and quadratic components of a spherical function, which define the components uniquely, up to rotation.

3.3 Matching Results

In order to evaluate the utility of the different rotation normalization methods, we measured the retrieval performance of the different spherical shape descriptors described in Chapter 2. As in Chapter 2, we used the Princeton Shape Benchmark to obtain precision vs. recall plots comparing the performance of the different shape descriptors when the different methods for addressing the rotational alignment problem were used. In particular, we computed the (1) Extended Gaussian Image, (2) Complex Extended Gaussian Image (3) Sectors representation, (4) Sectors and Shells representation, (5) Radial Spherical Extent Function, and (6) Gaussian Euclidean Distance Transform, for each of the models in the test database. For each model and each type of shape representation, we obtained retrieval results with the different rotational alignment techniques: (1) models were aligned by exhaustively searching over the space of all rotations for the optimal rotation (*Exhaustive Search*), (2) models were aligned with their principal axes and the best of the eight axial flips was chosen for alignment (*PCA + 8x(Axial Flips)*), (3) models were first aligned with their principal axes and then the descriptor were made invariant to the eight axial flips by storing the absolute values of the appropriate real and imaginary components of the spherical harmonic coefficients (*PCA + Abs*), (4) models were first aligned with their principal axes and then the descriptors were made invariant to rotations about the principal axis with largest eigenvalue by storing the complex norm of each spherical harmonic coefficient (*PCA + Fourier (X-axis)*), (5) models were aligned with their principal axes and forward weighting was used to normalize for ambiguity in the axial directions (*PCA (Heaviest Axis)*, (6) models were represented in a rotation-invariant manner by storing the power spectrum of the shape descriptors and using the quadratic resolution method to refine the energy decomposition within the second order frequency (*Harmonic + Quadratic*), (7) models were represented in a rotation-invariant manner by storing the power spectrum of the shape descriptors (*Harmonic*).

The results of the retrieval experiments are shown in Figure 3.8. In general, they indicate that the inability of PCA to resolve the axial ambiguity greatly hampers the matching performance of the different descriptors. Even methods that use forward weighting are not stable and do not consistently align axial directions within a class. Thus, methods that address this issue (e.g. *Abs*, *Fourier*, and *8x(Axial Flips)*) greatly improve the matching performance of the different descriptors and can often be competitive, in precision, with exhaustive search. As a result, we find that (for this dataset) the need for rotation-invariance is not well motivated and representations of shape descriptors in terms of the energies in the different frequencies (e.g. *Harmonic* and *Harmonic + Quadratic*) result in worse retrieval performance as a result of the inherent information loss in the power spectrum. Thus, in order for the rotation-invariant power spectrum representation to be effective in shape retrieval, it is necessary to regain some of the information lost in storing only the size of the different frequency components. (We discuss methods for addressing this issue in the next chapter.)

It is interesting to note that both the *Fourier* and the *Abs* approach are competitive

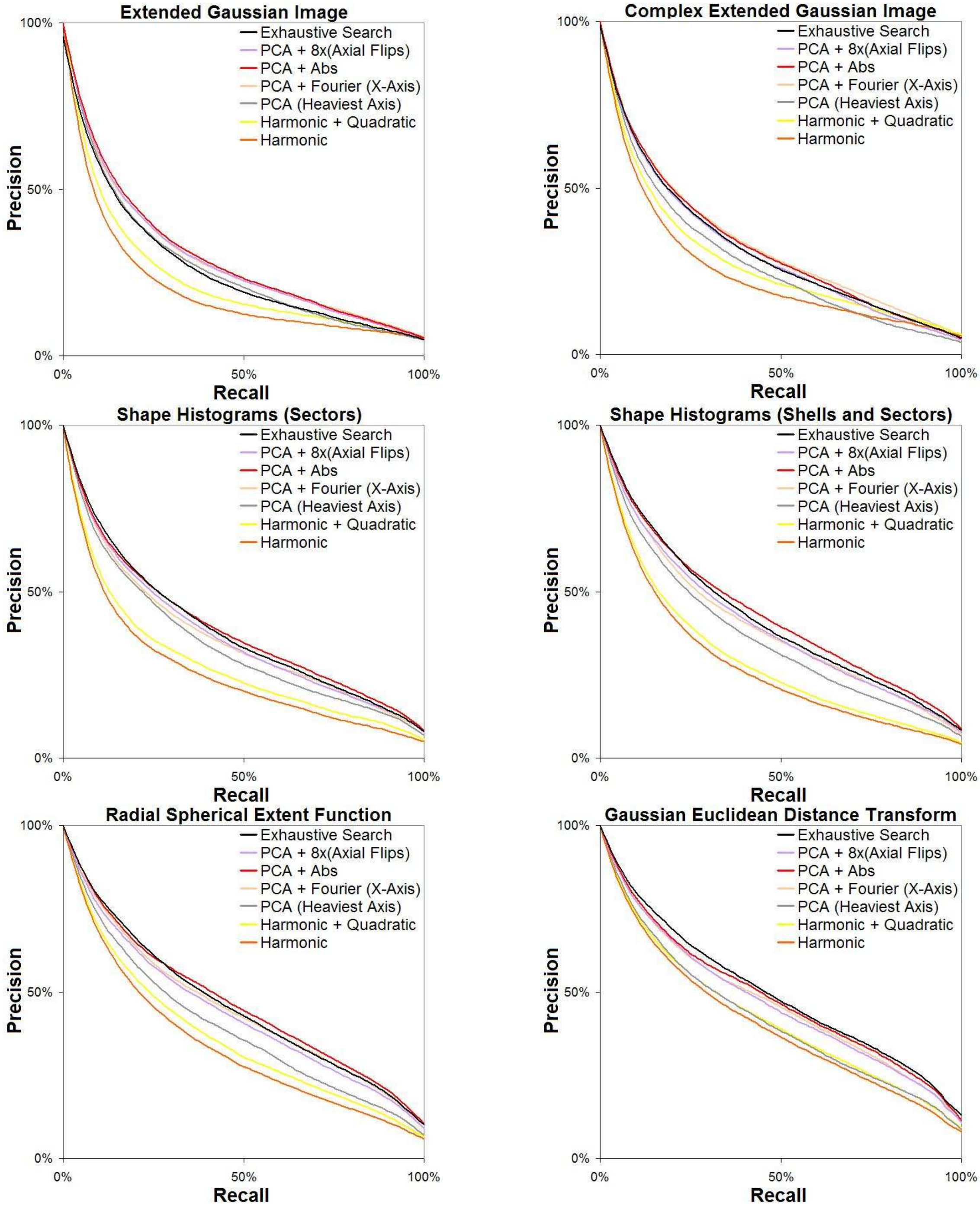


Figure 3.8: Retrieval results with different shape representations, demonstrating the effect of different rotational-alignment techniques on retrieval performance.

with exhaustive search for resolving axial ambiguity. In general, these approaches tend to discard more information and we would expect them to be less discriminating. We believe that part of the explanation for this resides in the limitations of PCA, as indicated in Figure 3.2. In particular, there are a number of classes whose constituent shapes can appear at different anisotropic scales. For these models, PCA alignment can fail not only because it does not correctly choose the correct direction of the principal axis, but also because in some cases the ordering of the principal axes gets changed. The *Fourier* and *Abs* representations begin to address this issue by introducing some invariance to permutation of the axes and we believe that this is part of the explanation for the reason that their retrieval performance is often better than the performance of the approach that exhaustively searches for the best alignment over all possible axial flips.

The sizes of the different descriptors and the comparison time for the different methods are shown in Tables 3.2 and 3.3. These tables highlight several important properties:

- The fact that the *Extended Gaussian Image*, *Complex Extended Gaussian Image*, and *Sectors* representation all represent a 3D model by a single spherical function is reflected in the fact that these descriptors are an order of magnitude smaller than the *Sectors and Shells* representation, the *Radial Extent Function*, and the *Gaussian Euclidean Distance Transform* which represent a 3D model by a collection of spherical functions.

- The rotation-invariant representations are an order of magnitude smaller than the PCA-aligned ones. This is because the power spectrum represents the $2 \cdot l + 1$ dimensions of information contained in the l-th frequency by the single value corresponding to the amplitude. Thus, it represents a two-dimensional spherical function by a one-dimensional array of energy values.

- The axial rotation-invariant representation (*Fourier*) described in [16, 62, 63] is roughly half the size of the initial spherical function because each complex spherical harmonic coefficient is represented by its norm.

- Exhaustively searching for the optimal rotation for model alignment tends to be independent of the size of the descriptor. This is related to the fact that the limiting step in correlating the shape descriptors is computing the Inverse Wigner-D Transform (as described in Appendix A). Since only one such computation needs to be performed, regardless of the number of spherical functions used to represent a model, we find that the average comparison time for exhaustive search is not proportional to the size of the descriptor.

- While a brute force implementation of the exhaustive search for the optimal axial flip would increase the comparison time eight-fold, we find that in practice the method presented for searching over all axial flips is only 54% slower, on average.

	EGI	CEGI	Sectors	Sectors + Shells	REXT	GEDT
Optimal Alignment	256	512	256	8192	8192	8192
PCA + 8x(Axial Flips)	256	512	256	8192	8192	8192
PCA + Abs	256	512	256	8192	8192	8192
PCA + Fourier	136	272	136	4352	4352	4352
PCA (Heaviest Axis)	256	512	256	8192	8192	8192
Harmonic + Quadratic	17	34	17	544	544	544
Harmonic	16	32	32	512	512	512

Table 3.2: Number of floats that need to be stored in order to represent the different shape descriptors shown in Figure 3.8 as a function of the method used for comparing across rotations. Note that the rotation-invariant descriptors are an order of magnitude smaller than the PCA-aligned ones.

	EGI	CEGI	Sectors	Sectors + Shells	REXT	GEDT
Optimal Alignment	15.979	16.407	15.924	20.696	21.470	20.585
PCA + 8x(Axial Flips)	0.031	0.063	0.033	0.972	0.951	0.934
PCA + Abs	0.019	0.037	0.021	0.641	0.680	0.671
PCA + Fourier	0.010	0.020	0.011	0.335	0.354	0.320
PCA (Heaviest Axis)	0.019	0.038	0.020	0.641	0.634	0.628
Harmonic + Quadratic	0.001	0.003	0.002	0.037	0.037	0.039
Harmonic	0.001	0.002	0.001	0.037	0.035	0.035

Table 3.3: Average time to compare two models using the different representations of the different shape descriptors shown in Figure 3.8. Note that, with the exception of the exhaustive search approach, comparison time is proportional to model size. Comparison time is in milliseconds.

3.4 Extending Descriptors

The spherical power spectrum approach described above is a general method for transforming a rotation-varying descriptor into a rotation-invariant one. In this section we show that the two descriptors designed to be rotation-invariant, namely the Shells [4] and the D2 [44] distributions, are actually very specific instances of the power spectrum approach. In particular, we show that using the methodology of the power spectrum approach these two representations can be transformed into more discriminating descriptors without sacrificing rotation-invariance.

Shape Histograms (Shells)

We recall that the Shells representation of a given model is obtained by computing the distribution of the distances of surface points from the center of mass. Another way to obtain this representation is to compute the Shells and Sectors representation, giving the distribution of points as a function of distance from the center of mass and spherical angle, and then extract the rotation-invariant Shells representation by fixing the distance from the center of mass and averaging the values of the distribution over all spherical angles. From a signal processing perspective, this approach is analogous to computing

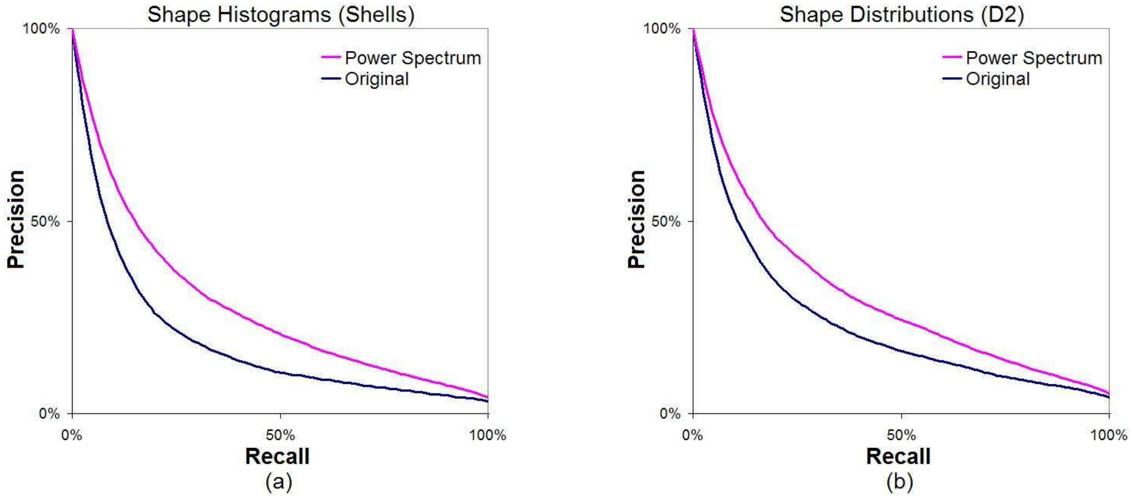


Figure 3.9: Comparison of the Shells and the D2 descriptors with their power spectrum generalization. As less rotation-invariant information is discarded, descriptors are more discriminating.

the Shells and Sectors representation, as a collection of spherical functions, and then obtaining a rotation-invariant representation by computing the spherical harmonic representation of each spherical function and storing only the constant order term. As we had seen above, rotation-invariant information can actually be gleaned from every frequency using the power spectrum. Consequently, we expect the Shells representation to be less discriminating than the more general power spectrum representation of the Shells and Sectors descriptor. This is verified empirically in Figure 3.9(a) which shows that there is a marked improvement in retrieval performance when the entire power spectrum of the Shells and Sectors descriptor is used over the retrieval performance when only the constant order terms represented in the Shells descriptor are used for matching.

Shape Distributions (D2)

As with the Shells descriptor, the D2 descriptor can be realized as the set of constant terms of a collection of spherical functions. In particular, we can represent a model with a 3D histogram, obtained by iterating over all pairs of points on the surface of the model, and for each pair of points p and q, voting with a unit weight in the bin indexed by $p - q$. Restricting the 3D histogram to concentric spheres about the origin, we obtain a translation-invariant representation of the initial model by a collection of spherical functions, giving the distribution of distances between pairs of surface points, as function of spherical angle. In this context, the D2 descriptor can be realized by storing the constant order component of each spherical restriction. Again, we can get a more refined rotation-invariant representation by storing not only the constant order component, but the entire power spectrum. As is shown in Figure 3.9(b), the addition of information about the energy distribution in higher frequencies gives rise to a descriptor that is more discriminating and gives better matching performance in retrieval experiments.

Chapter 4

Symmetry

In this chapter, we present the *symmetry descriptor*, a representation of a 3D model in terms of its reflective and rotational symmetries about all axes passing through the model's center of mass. To each axis of symmetry, the descriptor associates a value representing the extent to which the model is symmetric about that axis. Thus, the descriptor captures global shape information that is determined by the alignment of information across different frequencies and is well suited for regaining some of the information lost by the power spectrum representation.

We begin by reviewing related work in the area of symmetry detection. Then, we present a general method for defining the measure of a model's symmetry and provide an efficient signal processing based algorithm for computing the symmetry descriptor. We describe some of the theoretical properties of the symmetry descriptor and show how these properties can be used to guide methods for augmenting the frequency-local power spectrum representation with frequency-global symmetry information. Finally, we conclude by demonstrating the practical efficacy of the symmetry augmentation approach in shape retrieval experiments.

4.1 Related Work

Early approaches to symmetry detection focused on detecting the symmetries of planar point sets [5, 64, 20]. These methods reduced the symmetry detection problem to a detection of symmetry in circular strings, and used efficient substring algorithms (e.g., [32]) to detect the symmetries by searching for the appearance of a string within its concatenation with itself. While these methods had the theoretical advantage of efficiently evaluating all possible symmetries, they were impractical in empirical settings since they were algorithms that only identify the perfect symmetries of a model. Thus if a symmetric model had even a small amount of noise, these methods would fail to identify its symmetries.

In order to address this issue, Zabrodsky *et al.* [66, 67, 68] defined a continuous measure of symmetry which transformed the discrete question: "Does a model have a given symmetry?" to the continuous question: "How much of a given symmetry does a model have?" The measure of symmetry was defined as the minimum amount of work needed to transform a model into a symmetric model, measured as the sum of the squared distances that points would need to be moved. This approach made it possible to evaluate symmetries in the presence of noise, but suffered from the fact that it depended on the establishment of point correspondences. While this issue could be addressed in the case of 2D curves with uniform sampling, it made it difficult to generalize the method to 3D where uniformly sampling surfaces is often impossible.

With the advent of shape descriptors, a method became available for matching models without explicitly establishing correspondences and the advantage of the canonical parameterization was leveraged in a number of symmetry detection algorithms [43, 59]. These methods used the fact that the covariance ellipsoid of a 3D model rotates with the model, so that a model could only have symmetries where its covariance ellipsoid had them. Since the only axes of symmetry of a non degenerate ellipsoid have to align with its principal axes, this provided an efficient way to identify candidate axes of symmetry. The actual quality of an axis as an axis of symmetry would then be measured by comparing the shape descriptor of the model with the shape descriptors of the rotations and reflections of the model about the candidate axis. This method had the advantage of providing a continuous measure of symmetry for candidate axes of symmetry without necessitating the establishment of point correspondences. Furthermore, the method was a general one that could be applied to wide class of shape descriptors. However, the method's dependence on PCA for the identification of candidate axes could only guarantee the correct identification of symmetry axes for models with perfect symmetry.

Motivated by the ease of evaluating symmetry using shape descriptors, and the efficiency of exhaustive search provided by early substring matching approaches, efficient methods for evaluating the symmetries of a 2D model, at every symmetry, were developed. The key idea of these approaches was the generalization of discrete substring matching to continuous correlation with the Fast Fourier Transform. These methods [58, 40] compute the symmetries of a model by using correlation to compare the shape descriptor of a 2D model with all of its rotations and reflections. This approach was a general one that could be applied to any shape descriptor that represented a model with a function defined either on a circle, or in 2D.

The dependence of these methods on the FFT made them hard to generalize to shape descriptors that represented a 3D model with either a spherical function or a function in 3D. In [28, 29] we presented a method for computing the measure of reflective symmetries for all planes passing through the origin. For a spherical descriptor of size $O(N^2)$ (respectively 3D function of size $O(N^3)$) the method computes the measures of reflective symmetry in $O(N^3 \log N)$ (respectively $O(N^4 \log N)$) time. The efficiency of this approach relies on the use of the FFT to compute correlation with respect to a single axis efficiently and a generalization of this approach to general symmetry detection would result in algorithms that have complexity $O(N^4 \log N)$ for functions defined on the surface of a sphere and $O(N^5 \log N)$ functions defined in 3D.

In this chapter, we show how the analogs of the Fast Fourier Transform and Fast Inverse Fourier Transform on the sphere, namely the Fast Harmonic Transform and the Fast Inverse Wigner-D Transform, can be used to compute the measure of all symmetries efficiently. In particular, we describe a method for computing the measure of all reflective and rotational symmetries of both spherical functions and 3D functions in $O(N^4)$ time, thereby providing a method for computing all the symmetries of a model about its center of mass in less time than previous methods required to compute only the reflective symmetries of the model.

4.2 General Symmetry

Definition: Given a vector space V and a group G that acts on V, we say that $v \in V$ is *symmetric with respect to* G if $\gamma(v) = v$ for all $\gamma \in G$.

Definition: We define the *symmetry distance* of a vector v with respect to a group G as the L_2 distance to the nearest vector that is symmetric with respect to G:

$$\mathrm{sd}_G(v) = \min_{w|G(w)=w} \|v - w\|.$$

Using the fact that the vectors that are invariant to G define a vector subspace of V, it follows that the nearest G-invariant vector w is precisely the projection of v onto the subspace of invariant vectors. That is, if we define π_G to be the projection onto the subspace invariant under the action of G and we define $\pi_G^\perp$ to be the projection onto the orthogonal subspace then:

$$\text{sd}_G(v) = \|v - \pi_G(v)\| = \|\pi_G^\perp(v)\|$$

so that the symmetry distance of v with respect to G is the length of the projection of v onto a subspace indexed by G.

In general, computing the projection of v onto the subspace of vectors invariant under the action of G is a difficult task. However, in our case we can use the fact that the elements of G are orthogonal transformations. In particular, we can apply a theorem from representation theory [51] stating that a projection of a vector onto the subspace invariant under the action of an orthogonal group is the average of the vector over the different elements in the group. Thus, in the case of a vector v and a group G, we get:

$$\text{sd}_G^2(v) = \left\| v - \frac{1}{|G|}\sum_{\gamma \in G} \gamma(v) \right\|^2 = \|v\|^2 - \frac{1}{|G|}\sum_{\gamma \in G} \langle v, \gamma(v) \rangle$$

4.3 Defining the Symmetry Descriptor

In order to evaluate the measure of symmetry of a 3D model, it is necessary to compare a model with its reflections/rotations. A variety of shape descriptors can be used to compare the model with its transformation, and in this paper we focus on those that represent a model by a spherical, or 3D, function that rotates with the model.

Notation: For any integer k and any unit vector p we let G_p^k denote the *k-fold rotational symmetry group with respect to p*. If k is positive, then G_p^k is the group generated by the transformation $r_p^{2\pi/k}$ which is the rotation about the axis p by the angle $2\pi/k$. If k is negative, then G_p^k is the group generated by the transformation $r_p^{2\pi/k} \cdot A$, where A is the antipodal map, sending a point q to the point $-q$.

For example, $G_{(1,0,0)}^3$ is the group generated by rotating by 120° about the x-axis, consisting of 3 elements, while $G_{(0,1,0)}^{-2}$ is the group generated by reflecting through the xz-plane, consisting of two elements.

Definition: Given a shape descriptor f, we define its *k-fold symmetry descriptor* as the function on the sphere whose value at some point p describes the amount of f that is symmetric with respect to G_p^k and the amount of f that is anti-symmetric:

$$\mathrm{SD}_k(f,p) = \left(\|\pi_{G_p^k}(f)\|, \|\pi_{G_p^k}^{\perp}(f)\|\right)$$

where $\pi_{G_p^k}$ is the projection onto the space of functions that are k-fold symmetric about the axis p, and $\pi_{G_p^k}^{\perp}$ is the projection onto the orthogonal complement. (Since $\|f\|^2 = \|\pi_{G_p^k}(f)\|^2 + \|\pi_{G_p^k}^{\perp}(f)\|^2$ it suffices to compute one of $\|\pi_{G_p^k}(f)\|$ and $\|\pi_{G_p^k}^{\perp}(f)\|$. Despite the redundancy, we store both values as they can be used for bounding shape similarity.)

Figure 4.1 shows a visualization of the symmetry descriptors for a number of models. The descriptors are represented by scaling points on the unit sphere in proportion to the measure of symmetry $\|\pi_G(f)\|$, so that points corresponding to axes of near symmetry are pushed out from the origin and points corresponding to axes of near anti-symmetry are pulled in to the origin. Thus, for the 2-fold (respectively k-fold) symmetry descriptors, peaks in the descriptors correspond to axes of near perfect 2-fold (respectively k-fold) rotational symmetry of the models.

Several important properties of the symmetry descriptors are demonstrated in the image. (1) Looking at the cube, we observe that if a model is antipodally symmetric, it has no odd frequency components and the k-fold and $-k$-fold symmetry descriptors are

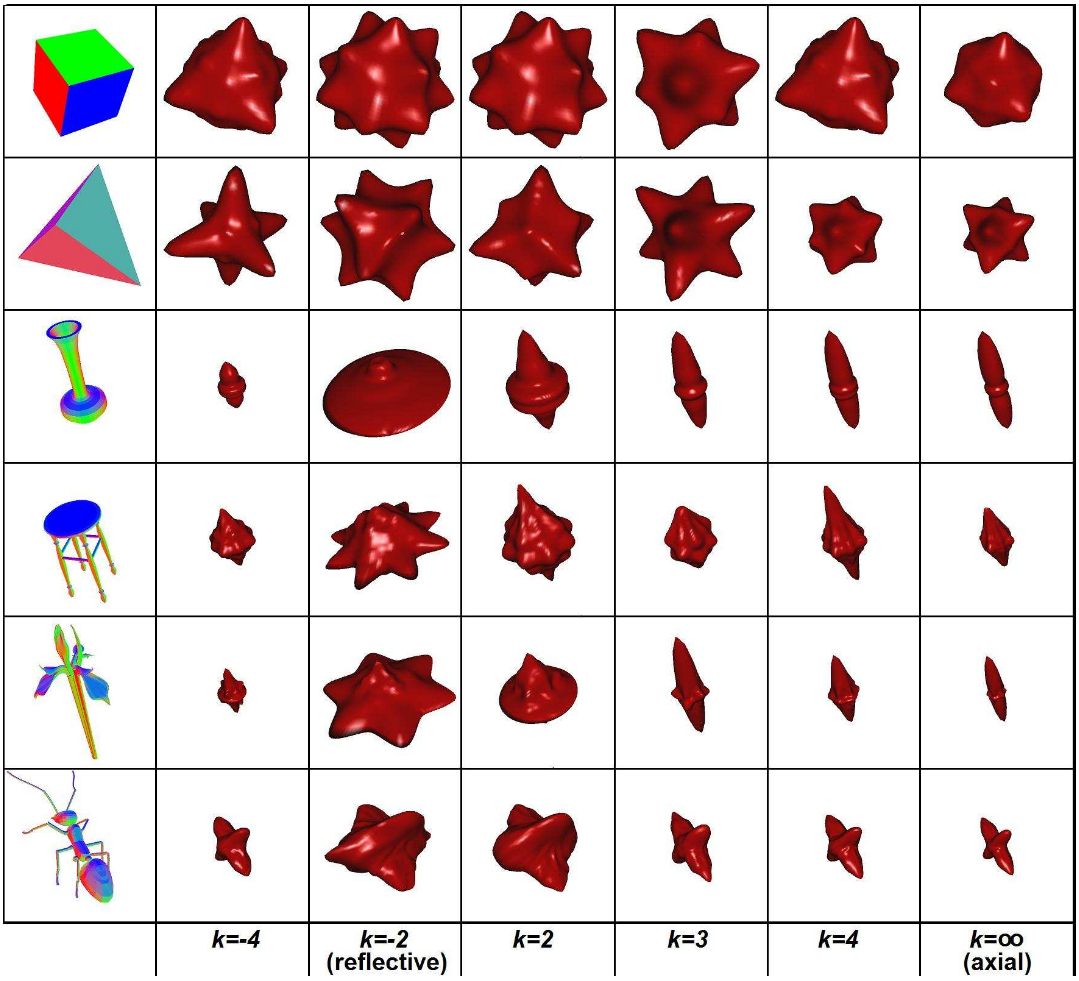


Figure 4.1: A visualization of the symmetry descriptors for a collection of models. The visualization is obtained by scaling unit vectors on the sphere in proportion to the measure of rotational symmetry about the axis through the center of mass, in the direction of the vector.

equal. (2) The example of the tetrahedron shows that a model can have $-k$-fold symmetry even when it is not antipodally symmetric. In particular, though the tetrahedron does not have antipodal symmetry, and it is not 4-fold symmetric about any axis, it does have -4-fold symmetry about the axes passing through the centers of the model's edges. (3) The symmetry descriptors of the vase indicate that the information characterized by the descriptors is not orthogonal. In particular, if a model has axial symmetry, than it must also have all other symmetries. More generally, a model can only have $k \cdot l$-fold symmetries, if it also has k-fold symmetry.

4.4 Computing the Symmetry Descriptor

We will now show how to compute all the k-fold symmetry descriptors of a shape descriptor efficiently. The key idea is that in computing the symmetry descriptor of a shape descriptor f, it is necessary to compare f with its rotations. This amounts to computing the autocorrelation of f across the group of rotations, $\langle f, \gamma(f)\rangle$, and signal processing methods for computing the autocorrelation described in Appendix A can be used to compute the symmetry descriptor efficiently.

In order to compute the symmetry descriptors of a shape descriptor f, it suffices to compute the lengths of the projections:

$$\left\|\pi_{G_p^k}(f)\right\|^2 = \frac{1}{|G_p^k|}\sum_{\gamma\in G_p^k}\langle f, \gamma(f)\rangle$$

for all k and all points p on the unit sphere. When k is positive then the elements in G_p^k are all rotations. Thus, having computed the values the autocorrelation we can reconstruct the symmetry descriptor $\mathrm{SD}_k(f,p)$. However, when k is negative, some elements of G_p^k will be of the form $\gamma = r_p^\alpha \cdot A$ – products of a rotation and the antipodal map. In this case, we can use the same method as in Appendix A, decomposing the function into the sum of its even and components, $f = f^+ + f^-$ with:

$$f^+(p) = \frac{f(p)+f(-p)}{2} \quad \text{and} \quad f^-(p) = \frac{f(p)-f(-p)}{2}.$$

Then, instead of computing the autocorrelation of f, we compute the autocorrelation of the even and odd parts independently to get:

$$\Phi_f^+(\gamma) = \langle f^+, \gamma(f^+)\rangle \qquad \text{and} \qquad \Phi_f^-(\gamma) = \langle f^-, \gamma(f^-)\rangle$$

This provides a general expression for the value of $\left\|\pi_{G_p^k}(f)\right\|$ for all $k \neq 0$ and all axes p:

$$\left\|\pi_{G_p^k}(f)\right\|^2 = \frac{1}{|2k|}\left(\sum_{j=1}^{|2k|} \Phi_f^+(r_p^{2j\pi/k}) + (\mathrm{Sgn}(k))^j \Phi_f^-(r_p^{2j\pi/k})\right).$$

Note that if k is odd and negative, the contributions from Φ_f^- cancel each other out, and the measure of the k-fold symmetry of f is equal to the measure of $-k$-fold symmetry of the even components of f:

$$\|\pi_{G_p^{|k|}}(f)\| = \|\pi_{G_p^{-|k|}}(f^+)\|.$$

Complexity: For both spherical functions and 3D functions the complexity of computing the autocorrelation over all rotations is bounded by $O(b^4)$. Since computing the k-th symmetry descriptor requires $O(k)$ summations for each of $O(b^2)$ points on the sphere the overall complexity of computing the $O(b)$ symmetry descriptors is $O(b^4)$.

4.5 Properties of the Symmetry Descriptors

Work in symmetry detection has been motivated, in part, by the recognition that symmetry is a property characterizing global shape information so that storing a small amount of symmetry information for each model should provide an efficient bound for the similarity of two models. In this section, we formalize this intuition by explicitly describing how the difference in the symmetries of two models relates to their measure of similarity.

4.5.1 Globality

A fundamental property of the symmetry descriptors is that they characterize global properties of a model and hence if the symmetry descriptors of two models differ at even one point, we expect this to imply that the models must be different. This property can be formulated explicitly by stating that the L_∞ difference between symmetry descriptors is a lower bound for the L_2 difference of the models:

$$\max_k \left\|\mathrm{SD}_k(f,p) - \mathrm{SD}_k(g,p)\right\|_\infty \leq \left\|f-g\right\|_2.$$

The explicit proof of this bound derives from the fact that the values of the symmetry descriptors of a function are equal to the lengths of its projections onto two orthogonal subspaces. Hence, for any k-fold symmetry and any axis p we have:

$$\begin{aligned}\left\|f-g\right\|^2 &= \left\|\pi_{G_p^k}(f) - \pi_{G_p^k}(g)\right\|^2 + \left\|\pi_{G_p^k}^\perp(f) - \pi_{G_p^k}^\perp(g)\right\|^2 \\ &\geq \left(\left\|\pi_{G_p^k}(f)\right\| - \left\|\pi_{G_p^k}(g)\right\|\right)^2 + \left(\left\|\pi_{G_p^k}^\perp(f)\right\| - \left\|\pi_{G_p^k}^\perp(g)\right\|\right)^2 \\ &= \left\|\mathrm{SD}_k(f,p) - \mathrm{SD}_k(g,p)\right\|^2\end{aligned}$$

so that the difference between the symmetry descriptors of two models at any point is an explicit bound for the proximity of the two models.

4.5.2 Continuous Symmetry Classification

One of the challenges of shape retrieval stems from the fact that often 3D models are not a priori aligned, and many methods for comparing two models require an initial step of pair-wise registration. For these types of applications, the globality property mentioned above cannot be utilized without first aligning the models. In this section we show how symmetry information can be used for comparing two models without requiring the initial alignment step.

We are motivated in our approach by early work in symmetry detection [5, 20, 64] where the goal was to classify models in terms of the types of symmetry that they have. These methods sought to assign a binary value to each integer k, indicating whether or not a model had k-fold symmetry. Since such a representation did not specify the axis of symmetry, it was inherently rotation-invariant.

Using the symmetry descriptors, we can extend these binary classifications into a continuous framework where for each k, we store the optimal measure of k-fold symmetry, even when the model is not k-fold symmetric. In particular, setting $s_k(f)$ to be the maximal value of k-fold symmetry of f:

$$s_k(f) = \max_{p \in S^2} \left\| \pi_{G_p^k}(f) \right\|$$

we define the optimal k-fold symmetry of f as the pair:

$$\text{Sym}_k(f) = \left(s_k(f), \sqrt{\|f\|^2 - s_k(f)^2} \right).$$

This gives a continuous, rotation-invariant classification of a model in terms of its symmetries. Furthermore, as a direct corollary of the globality property, it follows that the symmetry classification can be used to bound the proximity of two models:

$$\max_k \left\| \text{Sym}_k(f) - \text{Sym}_k(g) \right\| \leq \left\| f - g \right\|.$$

Thus the symmetry classifications can be used for matching models without requiring an initial step of pair-wise registration.

4.6 Symmetry Augmentation

Motivated by the properties described above, we would like to use the continuous symmetry classification for efficiently comparing models in a rotation-invariant manner. In particular, we would like to augment existing shape descriptors with symmetry information, but would like to do so in a manner that is not redundant with information already contained in the descriptors. To this end, we consider the power spectrum representation described in Chapter 3.

The power spectrum representation is a general method for obtaining a rotation-invariant representation of spherical shape descriptors that describes the descriptors in terms of the distribution of energies across different frequencies. The advantages of this representation are two-fold: First, the representation is rotation-invariant by construction, making it possible two compare models without first aligning them. Second, in going from a spherical function to its power spectrum, the dimensionality of the representation is reduced, contracting a 2D spherical function to a 1D array of energy values.

However, this representation treats each frequency component independently and does not capture information characterizing the alignment between different frequency components. Symmetry, by contrast, depends strongly on the manner in which the different frequencies align, and therefore captures information that is missing in the power spectrum representation. Thus, augmenting the power spectrum with symmetry information should provide a more discriminating representation, combining the local (in frequency space) information of the power spectrum with global symmetry information.

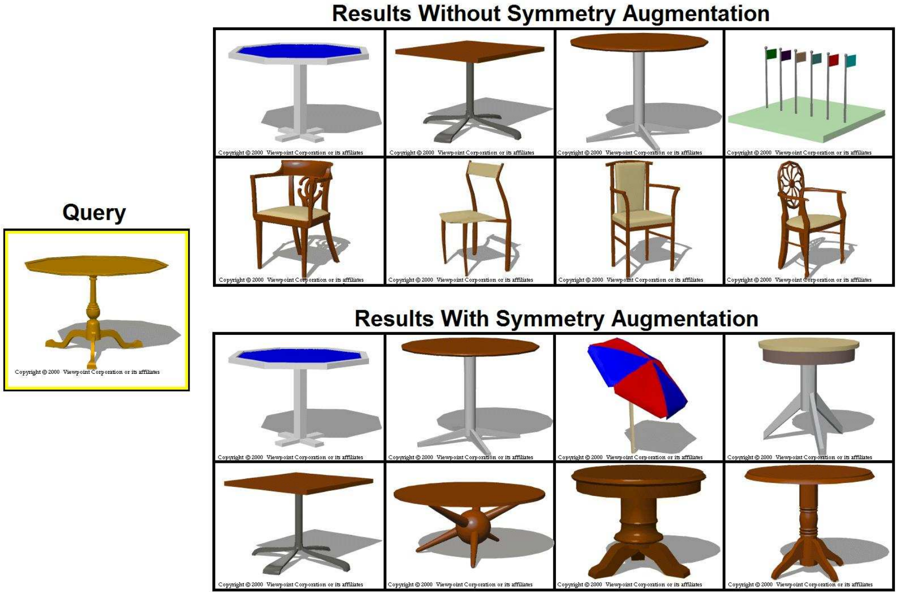

Results Without Symmetry Augmentation
Query
Results With Symmetry Augmentation
Copyright © 2000 Viewpoint Corporation or its affiliates

Figure 4.2: An example of the type of improvement gained by augmenting the power spectrum representation with symmetry information. The database was queried with the near axially-symmetric table on the left and results are shown for retrieval without (top) and with (bottom) symmetry augmentation. Note that symmetry augmentation improves matching performance by introducing a preference for models which have near axial symmetry.

Figure 4.2 demonstrates the motivation for this approach. In this figure, a database is queried with the near-axially symmetric table on the left, and retrieval results are shown without (top) and with (bottom) symmetry augmentation. Note that the addition of symmetry induces a preference for models that are near-axially symmetric, and pushes away models (such as the square table, second model in the non-augmented results) that do not have such a symmetry.

In order to augment the power spectrum representation we make the assumption that symmetry is uniformly distributed across all the non-constant frequencies, so that if f is a shape descriptor and $\text{Sym}_k(f)$ is the measure of the k-fold symmetry of f then:

$$\text{Sym}_k(f_l) = \text{Sym}_k(f) \cdot \frac{\|f_l\|}{\|f\|}$$

where f_l is the l-th frequency component of f. Thus, we replace the original power spectrum representation with the symmetry augmented representations:

$$\mathrm{PSR}_k(f) = \left\{ \|f_0\|, \mathrm{Sym}_k(f) \cdot \frac{\|f_1\|}{\|f\|}, \ldots, \mathrm{Sym}_k(f) \cdot \frac{\|f_b\|}{\|f\|} \right\}.$$

Then, to compare two descriptors, we find the symmetry type for which the two models vary most and compare the corresponding symmetry augmented representations:

$$D(f,g) = \max_k \|\mathrm{PSR}_k(f) - \mathrm{PSR}_k(g)\|.$$

Figure 4.3 demonstrates the process of symmetry augmentation. Given a spherical shape descriptor (shown in the top left), its power spectrum representation is computed by expressing the spherical function in terms of its frequency components, $\{f_0, f_1, \ldots\}$, and storing the norm of each component (shown in the top right). The symmetry descriptors are computed and the continuous, k-fold symmetry of f is extracted (shown in the bottom left). Finally, the power spectrum representation is augmented with symmetry information by scaling with the k-fold symmetry of f to obtain a finer resolution of non-constant frequency information (shown in bottom right).

4.7 Comparing the Symmetry Augmented Descriptor

Despite the fact that the symmetry augmented representation now requires a copy of the spherical harmonic representation for each symmetry type, in theory encumbering both storage and comparison, the symmetry augmented representation is in fact compact and

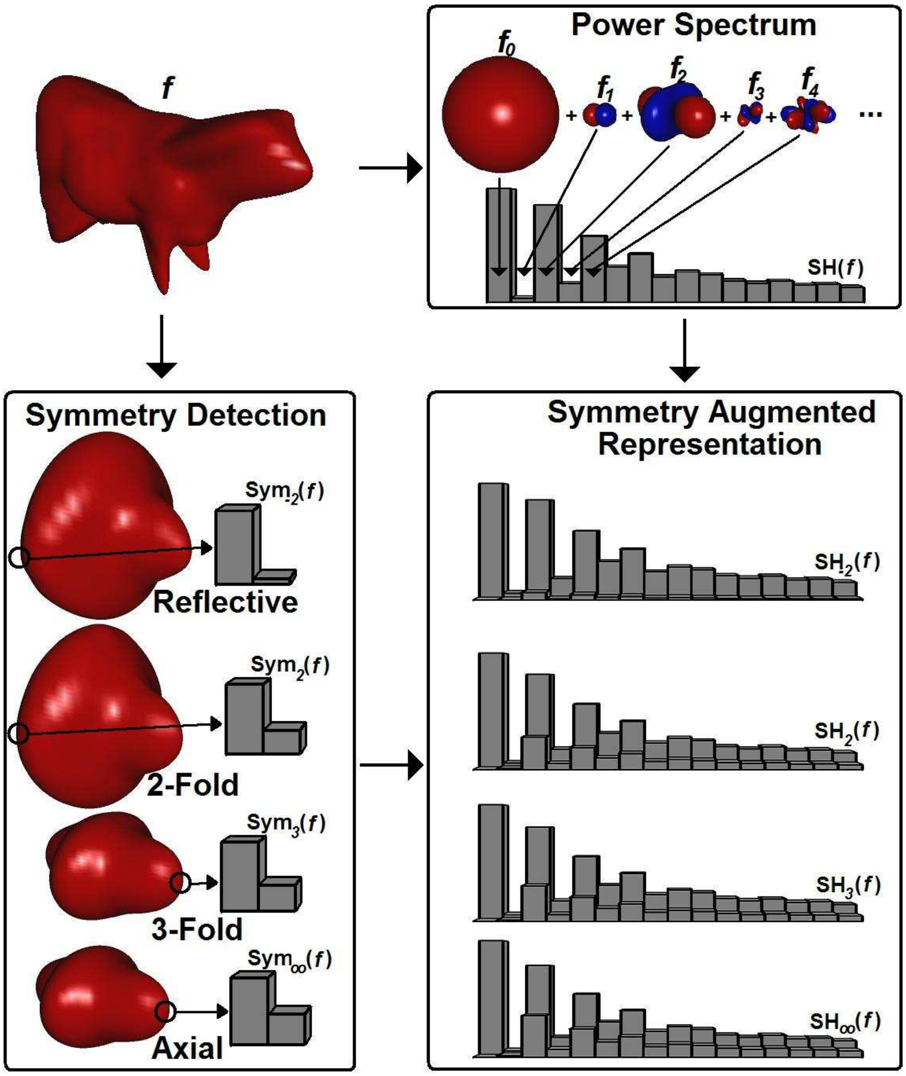


Figure 4.3: The augmented power spectrum representation of a shape descriptor (top left) is obtained by first computing the power spectrum representation (top right) and the k-fold symmetries of f (bottom left). The k-fold symmetries are then used to provide a finer resolution of non-constant frequency information by multiplying each frequency norm by the pair of k-fold symmetry values (bottom right).

compares efficiently. In particular, if we compute the symmetry dot product:

$$\mathrm{SDot}(f,g)=\max_k\left\langle\frac{\mathrm{Sym}_k(f)}{\|f\|},\frac{\mathrm{Sym}_k(g)}{\|g\|}\right\rangle$$

and the frequency dot product:

$$\text{FDot}(f,g)=\sum_{l=1}^{b}\|f_l\|\|g_l\|$$

independently, we can separate the role of symmetry information from frequency information in the measure of shape similarity:

$$D^2(f,g)=\|f\|^2+\|g\|^2-2\|f_0\|\|g_0\|-2\text{SDot}(f,g)\cdot\text{FDot}(f,g).$$

Thus, in comparing two descriptors, the symmetry information is separated from frequency information and only a single copy of the power spectrum representation needs to be stored. Furthermore, the separation of symmetry information from frequency information allows for efficient comparison of two models since the computations of $\text{SDot}(f,g)$ and $\text{FDot}(f,g)$ are both efficient computations that can be performed independently and then combined to give the measure of similarity.

Finally, the separation of symmetry information from frequency information provides an easy method for modulating the importance of symmetry in the measure of model similarity. In particular, we can define the family of shape metrics:

$$D_\alpha^2(f,g)=\|f\|^2+\|g\|^2-2\|f_0\|\|g_0\|-2\text{SDot}^\alpha(f,g)\cdot\text{FDot}(f,g).$$

indexed by the parameter α. When $\alpha=0$ symmetry plays no role in shape comparison and we revert to the power spectrum representation. When $\alpha=1$ we obtain the symmetry augmented representation described above. More generally, as α is increased, symmetry plays a more defining role in evaluation of shape similarity.

We observe that while the motivation for this section was the orthogonality of symmetry information to the information captured by the power spectrum representation, the method can be generalized to augmenting any rotation varying shape descriptors (not just those represented by their power spectrum). Thus, an augmented shape descriptor can be obtained and the value of α can still be used to modulate the importance of symmetry in the evaluation of similarity.

4.8 Matching Results

To measure the efficacy of the symmetry augmented power spectrum representation in tasks of shape retrieval, we measured the retrieval performance of the different spherical shape descriptors described in Chapter 2 comparing the results obtained with and without symmetry augmentation. In particular, we computed the (1) Extended Gaussian Image, (2) Complex Extended Gaussian Image, (3) Sectors representation, (4) Sectors and Shells representation, (5) Radial Spherical Extent Function, and (6) Gaussian Euclidean Distance Transform, for each of the models in the test dataset of the Princeton Shape Benchmark [48]. For each model and each type of shape descriptor, we represented the descriptor with its rotation-invariant power spectrum and its rotation-normalized PCA-aligned representation. We obtained retrieval results with and without symmetry augmentation, where the power spectrum of the shape descriptor was augmented with k-fold symmetry information, with $k = -2, 2, 3, 4, 5, \infty$ corresponding to reflective, 2-fold, 3-fold, 4-fold, 5-fold and axial symmetry information. A value of $\alpha = 2$ was used to amplify the importance of symmetry in retrieval – this was empirically determined to give the best results in classification experiments run on the training dataset.

The results of the retrieval experiments are shown in Figure 4.4, comparing the retrieval performance of augmented and un-augmented descriptors. These results highlight two important properties of the symmetry descriptor:

- First, symmetry augmentation results in markedly improved retrieval performance for the rotation-invariant representations while effecting only negligible improvements for the rotation-normalized ones. This underscores the fact that the improved precision for the power spectrum representation is not simply the result of the amplification of the importance of symmetry in shape matching. Specifically, these results validate the initial motivation for the symmetry descriptor as a frequency-global shape descriptor that can be used to regain some of the information lost by the frequency-local power spectrum.

- Second, we note that symmetry augmentation is less helpful to the Extended Gaussian Image and Complex Extended Gaussian Image than to other shape descriptors. This observation reinforces the fact that the symmetry descriptor is computed by comparing the shape descriptors of a single model, at different rotations. Thus, the quality of the symmetry descriptor is integrally tied to the quality of the underlying shape descriptor. In particular, shape descriptors such as the Extended Gaussian Image and Complex Extended Gaussian Image, which perform poorly in shape retrieval experiments (see Figure 2.2), give rise to less effective augmentation.

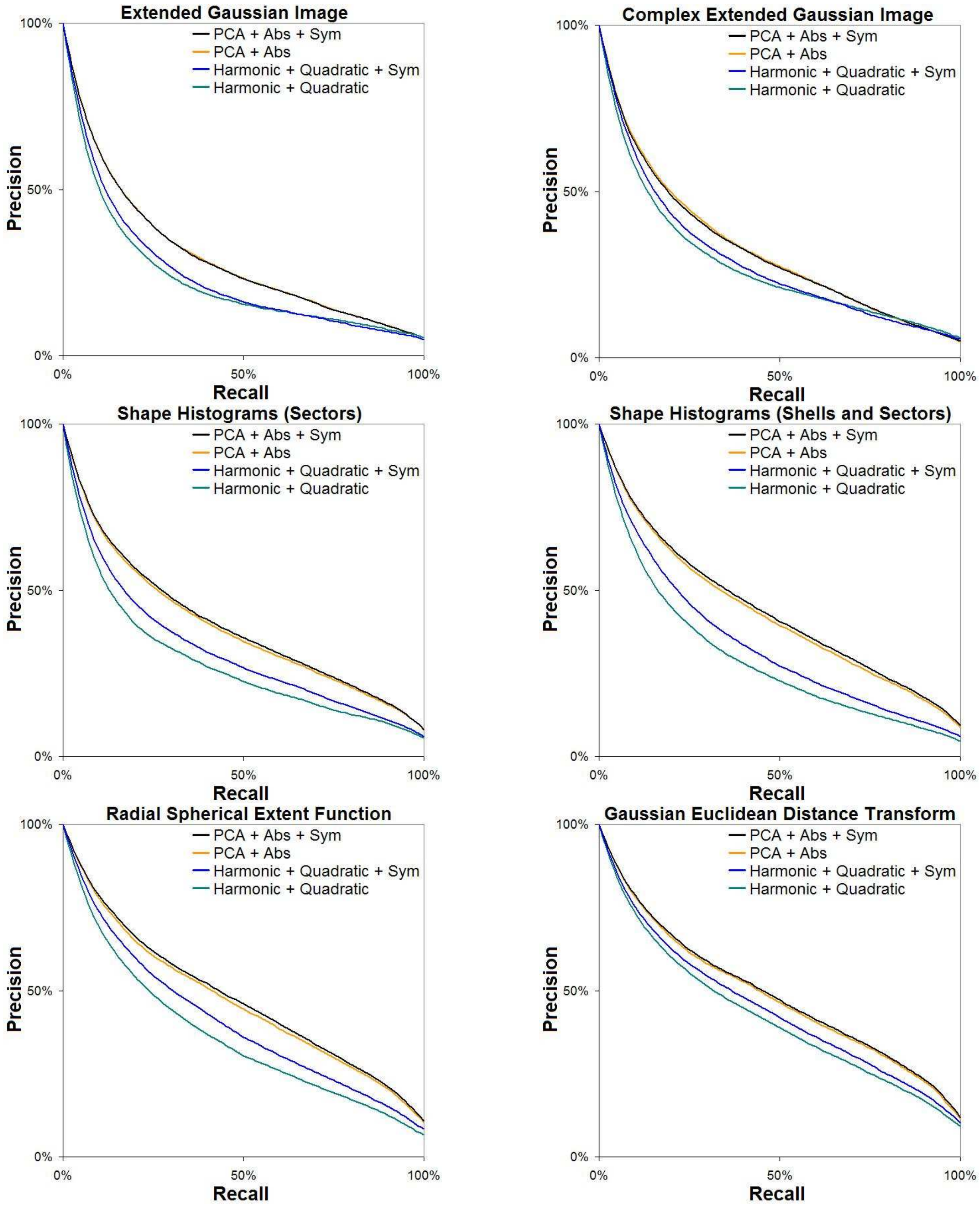


Figure 4.4: Retrieval results with different shape representations, demonstrating the effect of symmetry augmentation on retrieval performance.

Chapter 5

Anisotropy

In this chapter, we consider the implications of anisotropy on shape matching. In particular, we propose a novel method for matching 3D models that factors the shape matching equation as the disjoint product of anisotropy and geometric comparisons. We provide a general method for computing the factored similarity metric and show how this approach can be applied to improve the matching performance of many existing matching methods.

The key idea of our approach is based on the observation that much of the challenge of shape matching is in the establishing of correspondences, and that if two models are both isotropic, then it is easier to establish correspondences between them. Figure 5.1 demonstrates this for models of an armchair and a sofa. When the models are at their initial anisotropic scales (shown on the left), it is difficult to establish correspondences between similar regions. Methods such as associating to a point on one model the nearest point on the other (commonly used in ICP-type approaches [9]) will map points on the corners of the armchair to points in the middle of the sofa, points on the bottom of the arm-rest of the armchair to points on the top of the arm-rest of the sofa, etc. Thus, many poor correspondences will be established, resulting in an inaccurate measure of

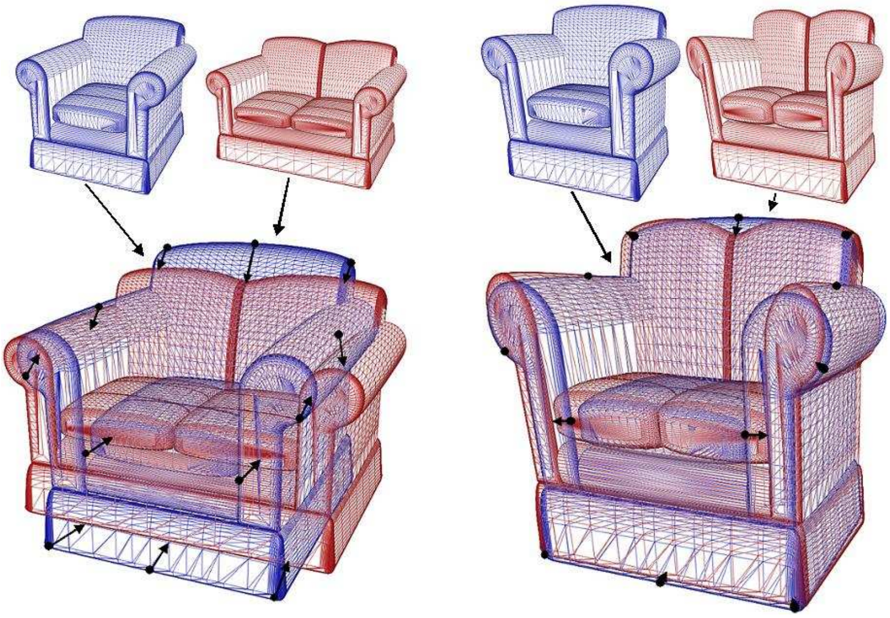


Figure 5.1: When two models have different anisotropic scales (left), it is harder to establish correct correspondences between the two. Thus, matching methods that depend on correspondences for evaluating model similarity will be inaccurate in this case. In contrast, when the models are transformed so that each is isotropic (right), the correspondences are more accurate and, as a result, the measure of shape similarity is more discriminating.

shape similarity. If instead both models are rescaled to be isotropic (right), then the correspondences established reflect corresponding regions in the shape more accurately.

In order to separate anisotropy from the shape matching equation, we propose a method for matching two 3D models that first removes the anisotropy from each of the models, compares the geometry of the isotropic models, and then expresses the measure of similarity of the two models as a function of both geometric and anisotropic similarity. Figure 5.2 demonstrates this process for two different models of a table. Each table is represented by its isotropic version and its initial anisotropic scale, represented by the covariance ellipsoid of the original model. The distance between the two tables is then defined as the combination of the distance between the isotropic tables and the distance between the initial anisotropic scales. This approach is motivated by earlier work in the

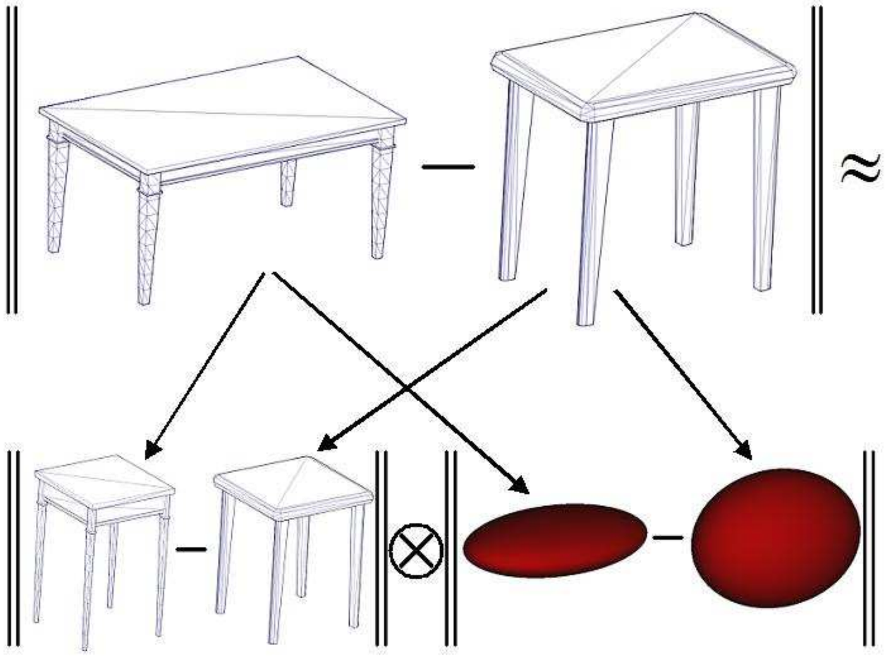


Figure 5.2: Our approach is to compare two models by rescaling each model so that it is isotropic and then defining the distance between two models as the combination of the distance between the isotropic models and the difference between their initial anisotropic scales.

area of isotropic scale normalization, which we reviewed in Chapter 3. We show how these results can be generalized to anisotropic scale and describe a method for removing the anisotropy from models. We conclude by describing a method for comparing two models, providing a family of shape metrics that is parameterized by the importance assigned to anisotropy.

5.1 Anisotropic Scale

In Chapter 3, we had shown that if two point sets are uniformly rescaled so that their mean variance is equal to one, then they are optimally scale aligned. We now show how these results can be generalized to solve for the optimal anisotropic scale. Given two point sets $P = \{p_1, \ldots, p_n\}$, with $p_i = (p_x^i, p_y^i, p_z^i)$, and $Q = \{q_1, \ldots, q_n\}$, with $q_i = (q_x^i, q_y^i, q_z^i)$, the sum of squared differences between the two point sets is given by:

$$\sum_{i=1}^{n}(p_x^i - q_x^i)^2 + (p_y^i - q_y^i)^2 + (p_z^i - q_z^i)^2.$$

It follows from the results of uniform scale alignment that if we search for the optimal anisotropic scale in any single direction v, then this occurs when the point sets P and Q are normalized so that that their variance in the direction v is equal to 1. Consider, for example, the case of solving for the optimal anisotropic scale in the x direction. In this case, we would like to solve for the values of s and t that minimize

$$\sum_{i=1}^{n}(sp_x^i - tq_x^i)^2 + (p_y^i - q_y^i)^2 + (p_z^i - q_z^i)^2$$

subject to the norm-preserving constraint:

$$\sum_{i=1}^{n}(sp_x^i)^2 + (tq_x^i)^2 + (p_y^i)^2 + (q_y^i)^2 + (p_z^i)^2 + (q_z^i)^2 = \sum_{i=1}^{n} \|p_i^2\| + \|q_i\|^2.$$

As in the case of isotropic scale, if the variance in the x-direction is equal to one, then the models are optimally aligned with respect to scale along the x-axis.

More generally, if both point sets satisfy the property that the variance in any direction is equal to 1, it follows that any anisotropic scaling of one of the two points sets will only increase the sum of squared differences and the initial models are in fact optimally anisotropically aligned.

In order to transform an arbitrary point set into one that has unit variance in any direction, it suffices to compute its covariance matrix C and then apply the transformation $C^{-1/2}$ to the point set. (Since we assume that the points are not all coplanar, the matrix C is positive definite and hence can be inverted, and has a real square root.) To see this, note that the covariance matrix of a point set $P = \{p_1, \ldots, p_n\}$ can be defined by:

$$C_P = \sum_{i,j=1}^{n} (p_i - p_j) \cdot (p_i - p_j)^t,$$

where the double summation is taken in order to account for the variance with respect to center of mass. If we set Q to be the transformed point set $Q = \{C_P^{-1/2} p_1, \ldots, C_P^{-1/2} p_n\}$ then the covariance matrix of Q is given by:

$$\begin{aligned} C_Q &= \sum_{i,j=1}^{n} C_P^{-1/2} (p_i - p_j) \cdot (p_i - p_j)^t C_P^{-1/2} \\ &= C_P^{-1/2} \left(\sum_{i,j=1}^{n} (p_i - p_j) \cdot (p_i - p_j)^t \right) C_P^{-1/2} \\ &= C_P^{-1/2} \cdot C_P \cdot C_P^{-1/2} = 1. \end{aligned}$$

Thus, the covariance matrix of the transformed point set is equal to the identity, and the variance in any direction is equal to 1. As with isotropic rescaling, this approach has the advantage that it can normalize for anisotropic scale on a per-model basis, allowing a model to be transformed in a pre-processing stage independent of the model that it will be matched against.

The difficulty with applying this method directly to triangulated models is that the transformation $C_P^{-1/2}$ rescales area patches as a function of their normal direction. Thus points that are uniformly distributed along the untransformed model need no longer be uniformly distributed on the anisotropically rescaled model. This phenomenon is illustrated in Figure 5.3 which shows points uniformly sampled from a model of an iris (left). After an anisotropic transformation is applied to the point set (right), the positions of the points are transformed and they no longer represent a uniform sampling of the surface.

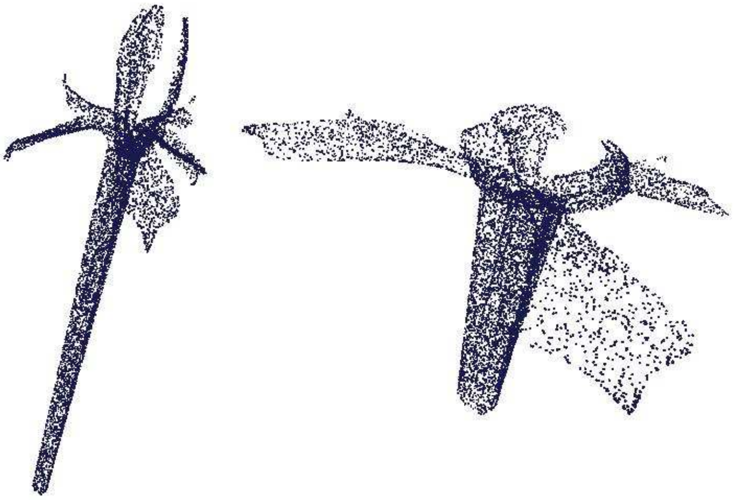


Figure 5.3: Uniform point samples from the surface of an iris model are shown on the left. The same points after anisotropic rescaling, are shown on the right. Though the point set on the right has unit variance in every direction, it no longer represents a uniform sampling from the surface of the rescaled iris.

Note that points on the stem are tightly clustered, while points on the petal become more spread out. This property of 3D meshes results in the undesired effect that transforming a triangulated model with the inverse square root of its covariance matrix need not give rise to an isotropic model.

In order to address this issue, we propose an iterative approach to transform the model. At each step of the iteration, the model is first translated so that its center of mass is at the origin, the covariance matrix is computed, and finally the model is rescaled by the inverse square root of the covariance matrix. In our experiments, we find that this approach converges efficiently to an isotropic model and, in practice, no more than five iterations of this process are necessary to obtain a nearly isotropic shape. Figure 5.4 shows a model of a pen and the transformed model after several steps of the iteration process. The figure also draws the associated covariance ellipsoids, which converge to a sphere as the model becomes isotropic. Note that after the first iteration, the transformed

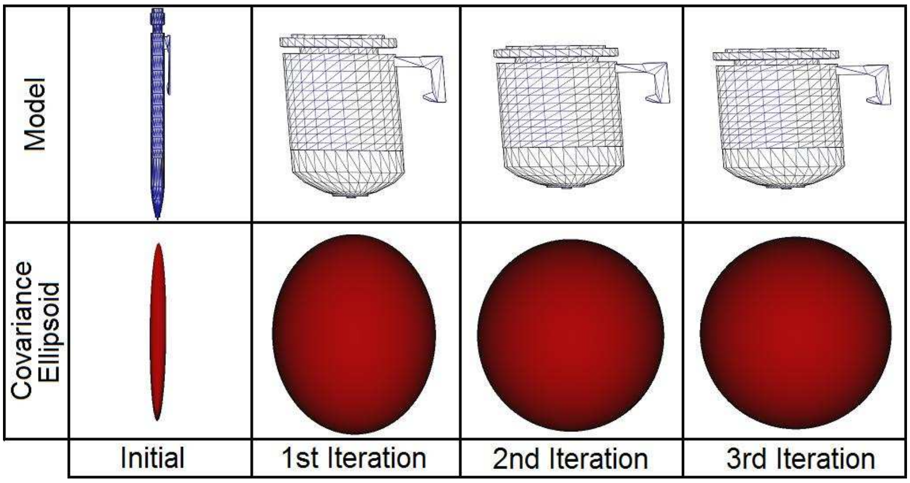


Figure 5.4: A visualization of a pen model and its covariance ellipsoid is shown on the left. The transformed model and its associated covariance ellipsoid, after one, two, and three iterations are shown on the right. Note that though the model is very anisotropic, after the third iteration of anisotropic rescaling we obtain a model that is nearly isotropic, with the covariance ellipsoid converging to a sphere.

model is still not isotropic, though, as the figure indicates, the iterative process converges quickly to an isotropic model. We provide a proof of the convergence of this approach in the next section.

5.2 Iterative Anisotropic Rescaling

In this section, we prove the convergence of the iterative rescaling algorithm for obtaining an isotropic model from an anisotropic one. We assume that the model is not coplanar, so that the variance in any direction is non-zero and we show that iteratively rescaling the model by the inverse square root of the covariance matrix is a process that converges to a model with constant variance, independent of direction. In particular, the steps of each iteration are:

1. Anisotropically scale the model by the inverse square root of the covariance matrix

2. Isotropically scale the model so that the minimum and maximum eigenvalues of the covariance matrix of the new model are reciprocals,

and we show that iterating these steps forces the minimum and maximum eigenvalues of the covariance matrix to converge to 1. To this end, we use the following equation for the covariance matrix of a model M:

$$C_M = \int_M \int_M (p-q) \cdot (p-q)^t dpdq$$

so that the variance of M in a direction v is given by:

$$\text{Var}(M, v) = v^t C_M v = \int_M \int_M \langle p-q, v\rangle^2 dpdq.$$

We will first show that a model with non-zero variance in any direction can always be rescaled so that the minimum and maximum eigenvalues are reciprocals. Next, we prove two lemmas describing the decomposition of M into an even partition and the corresponding decomposition of the variance of M across such a partition. Finally, we use the lemmas to show that the extremal eigenvalues must converge to 1.

Isotropic Rescaling: Given a model M and scale factor s, the covariance matrix of sM is defined as:

$$\begin{aligned} C_{sM} &= \int_{sM} \int_{sM} (p-q) \cdot (p-q)^t dpdq \\ &= s^6 \int_M \int_M (p-q) \cdot (p-q)^t dpdq = s^6 C_M. \end{aligned}$$

Thus, given a model M whose covariance matrix C_M has eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3$, we can rescale the model by $(\sqrt{\lambda_1 \cdot \lambda_3})^{-1/6}$ to obtain a new model whose covariance matrix has as its smallest and largest eigenvalues the reciprocals $\sqrt{\lambda_1/\lambda_3}$ and $\sqrt{\lambda_3/\lambda_1}$.

Lemma 5.2.1 *Given a continuous function f defined on M, there exists an even partition of M into subsets M^+ and M^- and a value μ such that $|M^+| = |M^-|$ and $f(p^+) \geq \mu \geq f(p^-)$ for all $p^+ \in M^+$ and all $p^- \in M^-$.*

Proof: To prove that such a decomposition must exist, we define the function $F(t)$ that gives the area of the subset of M with value less than or equal to t:

$$F(t) = \left|f^{-1}\big((-\infty, t]\big)\right|$$

Then $F(t)$ is a non-decreasing, right-continuous function that starts at 0 and grows to $|M|$, and is discontinuous at points t_0 such that $|f^{-1}(t_0)| > 0$. We set Φ to be the closure of the set of values t for which $F(t) \leq |M|/2$. Since $F(t)$ is monotonic we know that $\Phi = (-\infty, \mu]$, for some value μ. Then for all $t \geq \mu$ we have $F(t) \geq |M|/2$ and for all $t < \mu$ we have $F(t) \leq |M|/2$. If $F(\mu) = |M/2|$ we can set M^- equal to the inverse image of f on the range $(-\infty, \mu]$, and we can set $M^+ = M - M^-$. Otherwise the function $F(t)$ is discontinuous at μ and we must have $|f^{-1}(\mu)| \geq F(\mu) - |M|/2$. Thus we can set M^+ to be the union of the inverse image of f on the range (μ, ∞) and any subset of $f^{-1}(\mu)$ that has area $F(\mu) - |M|/2$. ∎

Lemma 5.2.2 *Given a partition of M into equal sized subsets M^+ and M^-, the variance across M^+ and M^- is at least as large as half the variance within M^+ and half the* *variance within M^-. That is, if*

$$\begin{aligned}
I_M^{++}(v) &= \int_{M^+}\int_{M^+}\langle p-q,v\rangle^2 dpdq \\
I_M^{+-}(v) &= \int_{M^+}\int_{M^-}\langle p-q,v\rangle^2 dpdq \\
I_M^{--}(v) &= \int_{M^-}\int_{M^-}\langle p-q,v\rangle^2 dpdq
\end{aligned}$$

then we must have: $I_M^{++}(v) \leq 2I_M^{+-}(v)$ and $I_M^{--}(v) \leq 2I_M^{+-}(v)$.

Proof: Integrating the function $I_M^{+-}(v)$ over the space M^-, we get:

$$2I_M^{+-}(v) = \frac{1}{|M^-|}\int_{M^-}\int_{M^-}\int_{M^+}\left(\langle p^- - p^+, v\rangle^2 + \langle q^- - p^+, v\rangle^2\right) dp^+dp^-dq^-$$

By the triangle inequality, we know that $\langle p^- - p^+, v\rangle^2 + \langle q^- - p^+, v\rangle^2 \geq \langle p^- - q^-, v\rangle^2$ so that we can get a bound for $I_M^{+-}(v)$:

$$\begin{aligned}
2I_M^{+-}(v) &\geq \frac{1}{|M^-|}\int_{M^-}\int_{M^-}\int_{M^+}\langle p^- - q^-, v\rangle^2 dp^+dp^-dq^- \\
&= \frac{|M^+|}{|M^-|}\int_{M^-}\int_{M^-}\langle p^- - q^-, v\rangle^2 dp^-dq^- \\
&= I_M^{--}(v)
\end{aligned}$$

as desired. The proof for $I_M^{++}(v) \leq 2I_M^{+-}(v)$ is analogous. ■

Anisotropic Rescaling: Given a model M, we set C_M to be the covariance matrix of M, $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 = 1/\lambda_1$ to be the eigenvalues of C_M, and $B_M = C_M^{-1/2}$ to be the inverse square root of C_M. Applying B_M to the model M, we obtain a model whose variance in direction v is given by:

$$\begin{aligned} \text{Var}(B_M(M), v) &= \int_{B_M(M)} \int_{B_M(M)} \langle p - q, v \rangle^2 dpdq \qquad (5.1) \\ &= \int_M \int_M \langle p - q, B_M(v) \rangle^2 b(p) b(q) dpdq \end{aligned}$$

where $b(p)$ is the differential change of area at the point M and must satisfy:

$$\frac{\sqrt{\lambda_1}}{\sqrt{\lambda_2}} \leq b(p) \leq \frac{1}{\sqrt{\lambda_1 \cdot \lambda_2}}.$$

Using the fact that each summand in Equation 5.1 is positive, we can apply the above inequalities to get:

$$\int_M \int_M \langle p - q, B_M(v) \rangle^2 dpdq \frac{\lambda_1}{\lambda_2} \leq \text{Var}(B_M(M), v) \leq \int_M \int_M \langle p - q, B_M(v) \rangle^2 dpdq \frac{1}{\lambda_1 \cdot \lambda_2}.$$

Using the fact that $B_M = C_M^{-1/2}$ it follows that

$$\int_M \int_M \langle p - q, B_M(v) \rangle^2 dpdq = \|v\|^2$$

and we know that the variance of the transformed model in any direction ($\|v\| = 1$) can be bounded by

$$\frac{\lambda_1}{\lambda_2} \leq \text{Var}(B_M(M), v) \frac{1}{\lambda_1 \cdot \lambda_2}.$$

We observe that when we rescale the model so that minimal and maximal eigenvalues of the covariance matrix are reciprocals, the minimal eigenvalue is no smaller than λ_1 so that transforming M by B_M cannot make the minimal variance smaller, nor can it make the maximal variance larger.

To show that the minimal and maximal eigenvalues must actually converge to 1, we use the lemmas above. To do this, we use the function $b(p)$ and Lemma 5.2.1 to evenly partition M into M^+ and M^- and obtain a value μ satisfying:

$$\sqrt{\frac{\lambda_1}{\lambda_2}} \leq b(p^-) \leq \mu \leq b(p^+) \leq \sqrt{\frac{1}{\lambda_1 \cdot \lambda_2}}$$

for all $p^- \in M^-$ and $p^+ \in M^+$. (Though $b(p)$ is not continuous on M, it is only discontinuous on a closed subset with 0 area, so Lemma 5.2.1 still holds.) Expressing the variance of $B_M(M)$ in the direction v in terms of this partition we get:

$$\begin{aligned}
\text{Var}(B_M(M), v) \;=&\\
&\int_{M^-}\int_{M^-} \langle p^- - q^-, B_M(v)\rangle^2 b(p^-)b(q^-)dp^-dq^-\\
&+\int_{M^+}\int_{M^+} \langle p^+ - q^+, B_M(v)\rangle^2 b(p^+)b(q^+)dp^-dp^+\\
&+2\int_{M^-}\int_{M^+} \langle p^- - p^+, B_M(v)\rangle^2 b(p^-)b(q^+)dp^-dp^+
\end{aligned}$$

This allows us to bound the minimal variance by:

$$\text{Var}(B_M(M), v) \geq I_M^{--}(B_M(v))\frac{\lambda_1}{\lambda_2} + 2I_M^{+-}(B_M(v))\mu\sqrt{\frac{\lambda_1}{\lambda_2}} + I_M^{++}(B_M(v))\mu^2.$$

Because B_M is the inverse square root of the covariance matrix of M we know that,

$$I_M^{--}(B_M(v)) + 2I_M^{+-}(B_M(v)) + I_M^{++}(B_M(v)) = \text{Var}(M, B_M(v)) = 1.$$

Additionally, since we know that $\sqrt{\lambda_1/\lambda_2} \leq \mu$, and $2I_M^{+-}(B_M(v)) \geq I_M^{--}(B_M(v))$ (by Lemma 5.2.2) it follows that the minimum variance is bounded by:

$$\frac{\frac{\lambda_1}{\lambda_2} + \mu\sqrt{\frac{\lambda_1}{\lambda_2}}}{2} \leq \text{Var}(B_M(M), v).$$

In a similar manner we can get an upper bound for the variance:

$$\frac{\frac{\lambda_1}{\lambda_2} + \mu\sqrt{\frac{\lambda_1}{\lambda_2}}}{2} \leq \text{Var}(B_M(M), v) \leq \frac{\frac{1}{\lambda_1 \cdot \lambda_2} + \mu\sqrt{\frac{1}{\lambda_1 \cdot \lambda_2}}}{2}$$

Isotropically rescaling $B_M(M)$ to get a model $\tilde{M}$ with minimal and maximal variances that are reciprocals, we get:

$$\lambda_1\sqrt{f(\mu)} \leq \text{Var}(\tilde{M}, v) \qquad \text{with} \qquad f(t) = \frac{1 + t\sqrt{\lambda_2/\lambda_1}}{1 + t\sqrt{\lambda_2 \cdot \lambda_1}}.$$

To find the minimal and maximal variances, we compute the derivative of $f(t)$:

$$f'(t) = \frac{\sqrt{\lambda_2/\lambda_1} - \sqrt{\lambda_2 \cdot \lambda_1}}{(1 + t\sqrt{\lambda_2 \cdot \lambda_1})^2}.$$

Since $\lambda_1 \leq 1$, the derivative is never negative, and hence the variance of $\tilde{M}$ is minimized when μ is as small as can be, which is to say $\mu = \sqrt{\lambda_1/\lambda_2}$. In this case we get:

$$\lambda_1\sqrt{\frac{2}{1+\lambda_1}} \leq \text{Var}(\tilde{M}, v) \leq \frac{1}{\lambda_1}\sqrt{\frac{1+\lambda_1}{2}}.$$

Thus, the minimal and maximal variances of the model are guaranteed to converge to 1, and the iterative method described in the previous section is guaranteed to converge to a model with variance 1 in every direction. ∎

In order to evaluate the empirical performance of this approach, we measured the efficiency of convergence by computing the magnitude of the largest eigenvalue as a function of the number of iterations. In particular, for each model in the Princeton Shape Benchmark [48], we iteratively (1) computed the covariance matrix of the model, (2) transformed the model by the inverse square root of the covariance matrix, and (3) rescaled the models so that its smallest and largest eigenvalues were reciprocals. We measured the efficiency of the iterative approach by evaluating how quickly the largest eigenvalue would converge to 1. (Since the model is uniformly rescaled so that the smallest eigenvalue is the reciprocal of the largest eigenvalue, when the largest eigenvalue is equal to 1, the covariance matrix reduces to the identity and the model is isotropic.) Figure 5.5 shows a plot of the magnitude of the largest eigenvalue as a function of the number of iterations, with the red plot showing the average value of the largest eigenvalue, averaged over all the models in the benchmark, and the yellow plot showing the maximal value of the largest eigenvalue, after each step of the iteration. Note that even though some of the models are initially quite anisotropic, after five iterations even the most anisotropic of the models is transformed into a nearly isotropic model, indicating that the iterative approach converges very efficiently in practice.

5.3 Anisotropy Factoring

The method that we propose for anisotropy factoring is a general one that can be applied to any of the many methods [4, 17, 22, 27, 44, 63] that matches two models by independently representing each one by a shape descriptor, and then defining the measure of model similarity as the L_2 difference between the corresponding descriptors. In particular, we anisotropically rescale a model M to obtain an isotropic model $\tilde{M}$, storing the

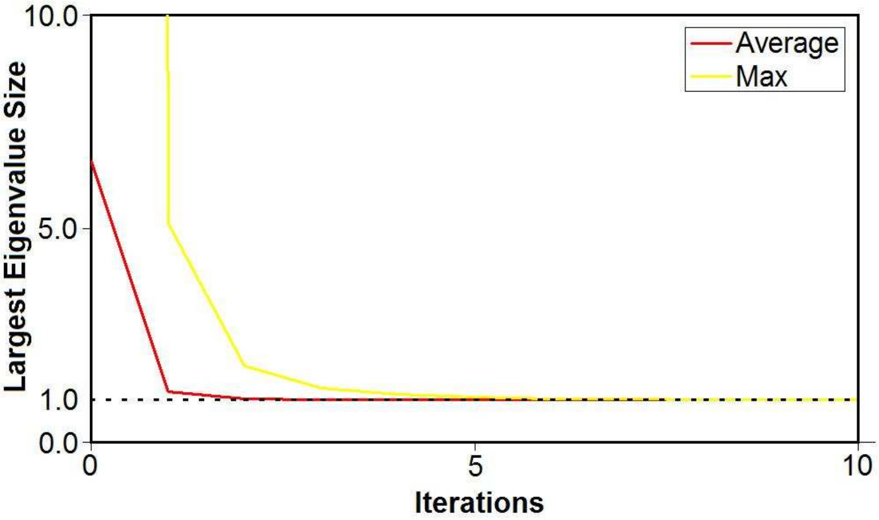


Figure 5.5: A graph of the magnitude of the largest eigenvalue as a function of the number of anisotropic rescaling iterations performed. The red plot shows the average value of the largest eigenvalue for a database of roughly 1800 models, while the yellow plot shows the maximum value. Note that even though some of the models are initially quite anisotropic, in practice the iterative approach converges efficiently and a nearly isotropic model is obtained within very few iterations.

sorted triplet of eigenvalues $\lambda_M = (\lambda_1^M, \lambda_2^M, \lambda_3^M)$ of the matrix transforming M into $\tilde{M}$. The triplet λ_M is a rotation-invariant representation of the anisotropy of M and, for simplicity, we normalize the triplet so that $\|\lambda_M\| = 1$. We compute the shape descriptor $v_{\tilde{M}}$ of the isotropic model and, using the fact that the information contained in $v_{\tilde{M}}$ is orthogonal to the information contained in λ_M, we represent the initial model M by the new shape descriptor $v_{\tilde{M}} \times \lambda_M$, as shown in Figure 5.6.

At runtime, when a query model Q is presented to the database, we compute the anisotropy factorization of Q and define the measure of similarity between Q and a database model M to be the value:

$$D_\alpha(M,Q) = \|v_{\tilde{M}}\|^2 + \|v_{\tilde{Q}}\|^2 - 2\langle v_{\tilde{M}}, v_{\tilde{Q}}\rangle\langle\lambda_M, \lambda_Q\rangle^\alpha.$$

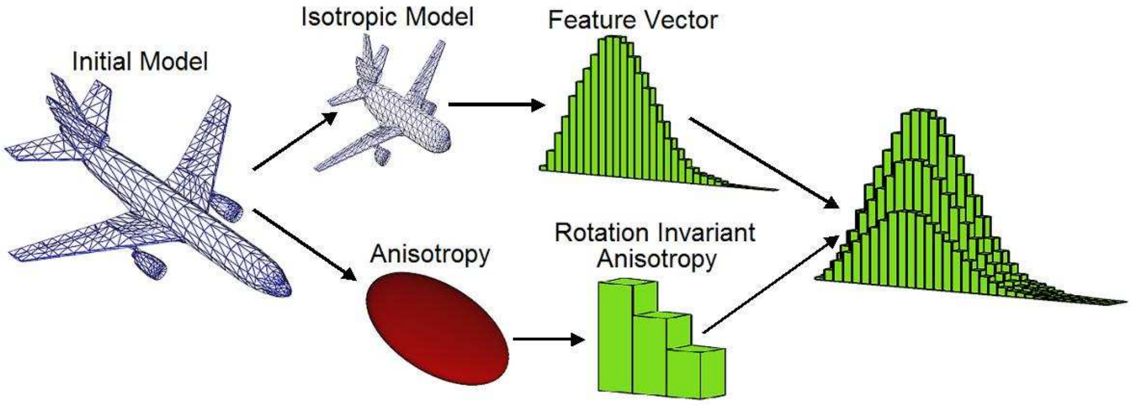


Figure 5.6: We create a new shape descriptor by computing the outer product of the rotation-invariant anisotropic scales with the shape descriptor of the isotropic model.

If $\alpha = 1$ then $D_\alpha(M, Q)$ is the L_2 difference between the vectors $v_{\tilde{Q}} \times \lambda_Q$ and $v_{\tilde{M}} \times \lambda_M$. More generally, α can be treated as a fixed constant representing the importance of anisotropy information in the context of shape matching. Thus, in the case that $\alpha = 0$, anisotropy information plays no role in the matching and the matching method is invariant to anisotropic scale. If additionally the shape descriptor is itself rotation-invariant [4, 44, 17] then we obtain a matching method that is invariant to all *affine* transformations.

The advantage of this matching approach is that the shape metric defines similarity as the product of the similarity of the shape descriptors and the similarity of the anisotropies. Thus, the new shape descriptor only needs to store three additional values, corresponding to the normalized eigenvalues of the symmetric matrix transforming an anisotropic model into an isotropic one. This means that neither the storage nor the comparison time of the anisotropy factorized shape descriptor is significantly larger than the corresponding storage and comparison time for the original shape descriptor.

5.4 Matching Results

To measure the efficacy of the anisotropy augmented descriptor in tasks of shape retrieval, we measured the retrieval performance of the different shape descriptors described in Chapter 2, comparing the results obtained with and without anisotropy augmentation. In particular, we computed the (1) D2 distribution, (2) Shells distribution, (3) Extended Gaussian Image, (4) Complex Extended Gaussian Image (5) Sectors representation, (6) Sectors and Shells representation, (7) Radial Spherical Extent Function, and (8) Gaussian Euclidean Distance Transform, for each of the models in the test dataset of the Princeton Shape Benchmark. For each model and each type of shape representation, we obtained retrieval results with and without anisotropy factorization. A value of $\alpha = 3$ was used to amplify the importance of anisotropy in retrieval – this was the values that was empirically determined to give the best results in classification experiments run on the training dataset.

The results of the retrieval experiments for the histogram descriptors are shown in Figure 5.7. The figure shows plots of the retrieval results comparing the retrieval performance of the descriptor without anisotropy factorization to the performance of the descriptor with anisotropy factorization. For the anisotropy factorization we show two different plots. In the first plot, results were obtained by setting $\alpha = 3$, amplifying the importance of anisotropy (*Histogram* + *Scale*). In the second, results were obtained by setting $\alpha = 0$, ignoring the initial difference in anisotropy scales (*Histogram – Scale*).

Combining the approaches described in this chapter and the previous one, we obtain a method for augmenting shape descriptors with both symmetry and anisotropy information. This approach can be applied to any spherical descriptor that rotates with the model and we show the results for this hybrid approach for descriptors that represent a model by

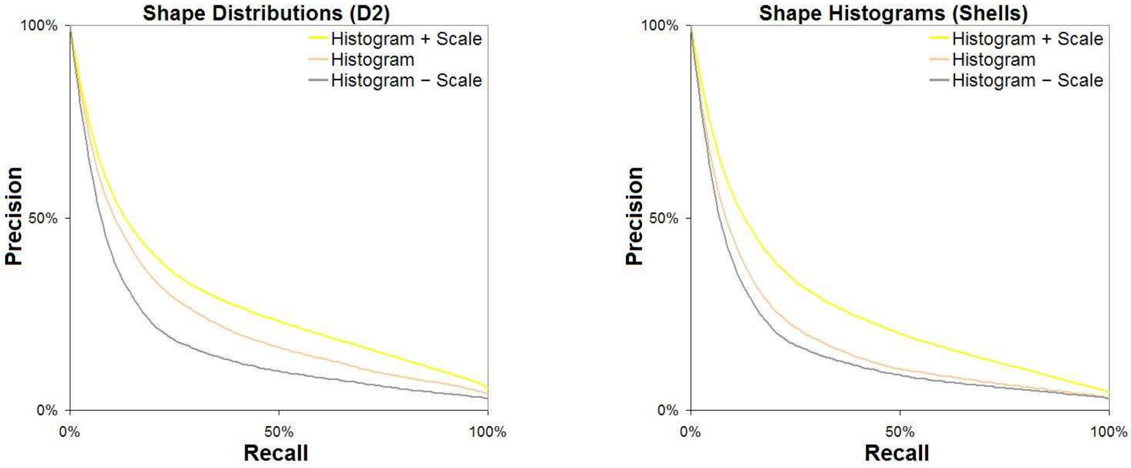


Figure 5.7: Retrieval results with shape descriptors representing a 3D model by 1D histograms, demonstrating the effect of anisotropy factorization on retrieval performance.

a single spherical function in Figure 5.8, and for descriptors that represent a model with multiple spherical functions in Figure 5.9.

Several properties of symmetry augmentation can be observed from the results of the retrieval experiments for both the rotation-invariant descriptors shown in Figure 5.7 and the rotation-varying descriptors shown in Figures 5.8 and 5.9.

- First, we note that amplifying a shape descriptor's anisotropy information improves the performance of the shape descriptor, independent of the representation. Thus, anisotropy factorization provides a simple an effective method for comparing 3D models in a more meaningful manner.

- Second, we note that anisotropy factorization improves the matching performance of all descriptors, independent of their retrieval quality. This observation reinforces the fact that anisotropy is a property of the geometry of the model and does not depend on the method used to represent shapes.

- Third, we note that in the case that anisotropy factorization is performed and no weight is assigned to anisotropy comparison ($\alpha = 0$ for *Histogram – Scale*, *PCA + Abs – Scale*, and *Harmonic + Quadratic – Scale*) the matching performance does not generally improve and for the rotation-invariant representations (*Histogram* and Harmonic + Quadratic) the matching performs actually exhibits a marked deterioration in performance. We believe that the reason for this is associated with the manner in which rotation alignment is addressed. Specifically, the rotation-invariant methods address rotation after the model has been normalized for anisotropy while the PCA-normalization methods first normalize for rotation and only then normalize for anisotropy. Thus, the rotation-invariant approaches actually match models across a wider class of transformations, finding the best alignment over the space of all affine transformations, while PCA-normalization methods do not actually match over shears. Since shears are not a type of transformation associated with intra-class variation, the rotation-invariant methods match over too large a class of transformations and consequently are less capable of distinguishing between models in different classes.

- Finally, we note that when the power spectrum is used to represent a shape descriptor in a rotation-invariant manner, anisotropy and symmetry augmentation can be combined to amplify two important characteristics of 3D models. In this case, the improved precision obtained using both methods of augmentation simultaneously is roughly the sum of the improved precisions obtained using each augmentation method separately. Thus, these augmentations approaches are orthogonal, providing independent methods for amplifying essential shape properties.

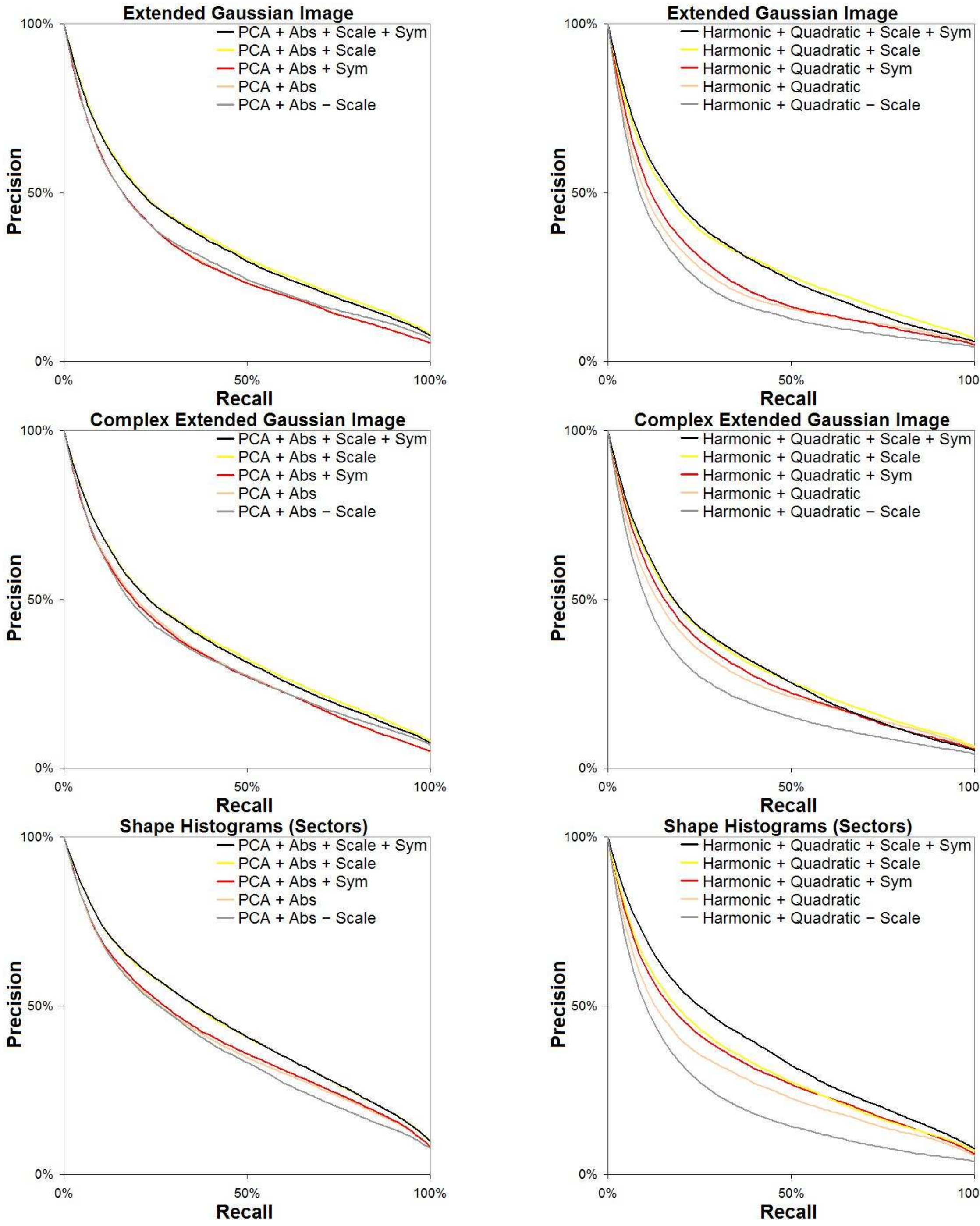


Figure 5.8: Retrieval results with spherical shape representations, demonstrating the effect of the combination of anisotropy and symmetry augmentation.

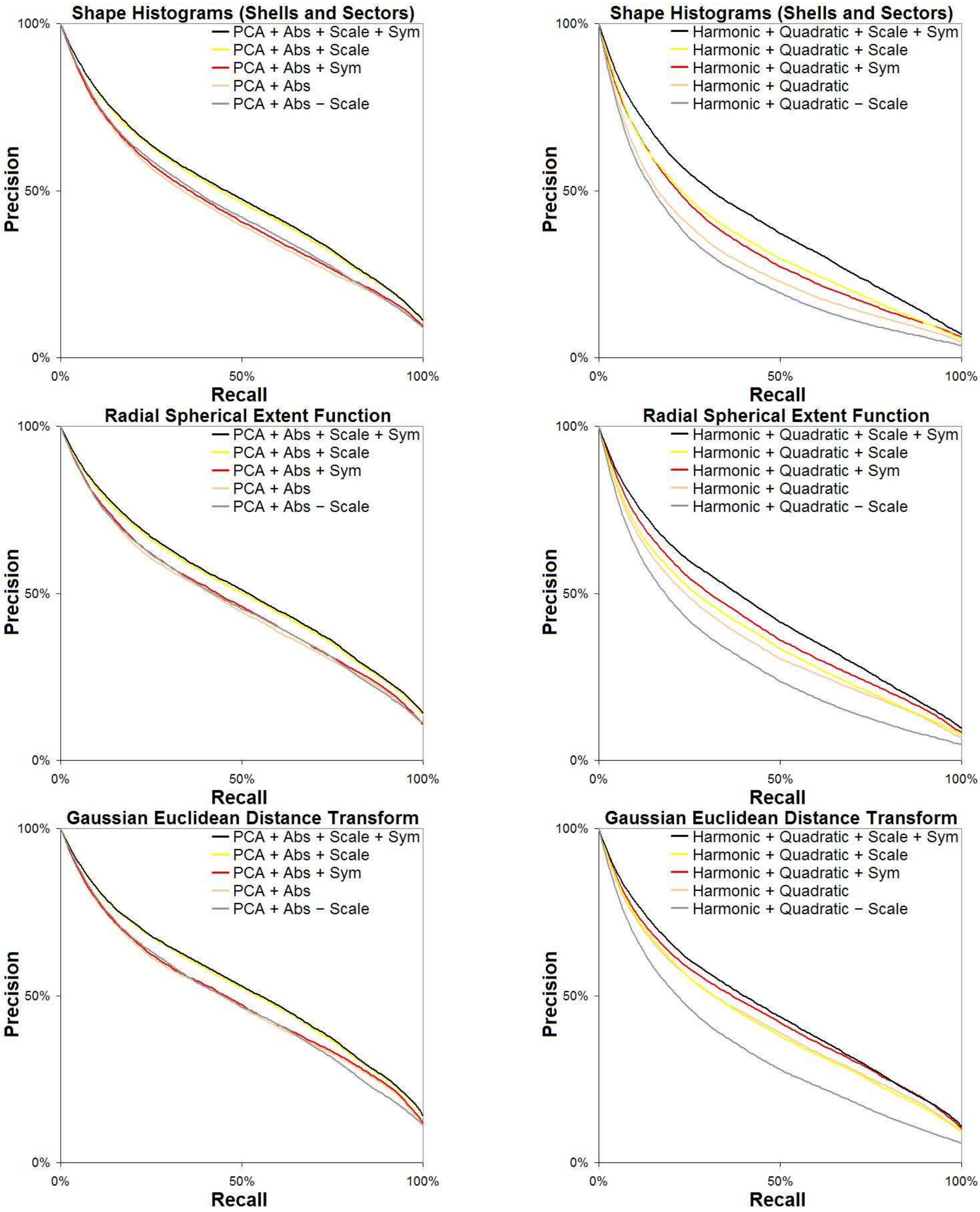


Figure 5.9: Retrieval results with multi-spherical shape representations, demonstrating the effect of the combination of anisotropy and symmetry augmentation.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we explored a number of the issues associated with the challenge of retrieving shapes from large repositories of 3D models and we have focused on general methods for addressing some of the central challenges of shape matching. In particular, the contributions of this thesis are four-fold.

First, we introduced the Gaussian Euclidean Distance Transform, a new shape descriptor characterizing a model by a function that peaks at the surface of the model and falls off exponentially away from it. We designed the descriptor so that the difference between the descriptors of two models approximates the minimum sum of squared distances between the two corresponding surfaces. In empirical evaluations, we have demonstrated that the new descriptor is more effective in retrieval tasks than existing shape descriptors and have shown that the retrieval performance of the GEDT is nearly identical to the performance obtained when comparing models by explicitly computing the minimum sum of squared distances between their surfaces. Thus, this new descriptor provides the discriminability of a shape metric, while maintaining the ease of comparison of a vector-based shape representation, making it particularly useful for shape matching applications designed for interactive search.

Second, we presented a general framework for addressing the rotation-alignment problem in shape matching. We reviewed the standard PCA method for aligning models into a canonical coordinate system and the power spectrum approach for transforming rotation-varying shape representations into rotation-invariant ones. We presented an analysis of the limitations of these approaches and presented novel methods for addressing these limitations. In particular, we presented two methods for addressing the axial ambiguity of PCA alignment, (1) giving a method for efficiently performing an exhaustive search over the space of all axial flips, and (2) describing an approach for representing shape descriptors in an axial-flip-invariant manner. We have also shown how some of the information loss occurring in the power spectrum can be ameliorated by providing a full resolution of the second order frequency information into rotation-invariant components.

Third, we introduced the symmetry descriptor, a representation of the reflective and rotational symmetries of spherical shape descriptors. We have shown how spherical signal processing techniques can be used to compute the symmetry descriptor efficiently and we have shown how the obtained symmetry information can be used to augment shape descriptors, giving rise to new shape descriptors which exhibit improved retrieval performance without sacrificing efficiency. While this approach is a general one that can be applied to many types of shape descriptors, we have demonstrated that it is particularly well-suited for methods that use the power spectrum representation. This allows many existing shape descriptors to leverage the advantages of the power spectrum representation (e.g. rotation-invariance and compactness) without suffering from the impaired matching performance previously associated with the information loss inherent in this approach.

Finally, we introduced the anisotropy factorization method, a general approach for transforming anisotropic models into isotropic models that are better suited for shape matching. By contrast with the symmetry descriptor, anisotropy factorization acts on the model directly, rather than on the shape descriptor and hence is a technique that can be applied to a more general class of shape retrieval methods. We provided a general approach for augmenting existing shape representations with anisotropy information and shown that anisotropy augmented representations gives rise to shape descriptors with improved retrieval performance without sacrificing efficiency in matching. We have also shown that for applications that use the power spectrum to represent a model in a rotation-invariant manner, the two augmentation methods — symmetry and anisotropy — can be combined to give still more discriminating results than either method alone.

6.2 Future Work

The work presented in this thesis suggests a number of different venues for future work:

6.2.1 Alignment

In Chapter 3 we presented a variety of approaches for addressing the challenge of rotation alignment for shape matching. Though we have found that the power spectrum approach provides a conservative estimate of the measure of model similarity at the best possible orientation while simultaneously reducing the dimensionality of the information needed to represent a 3D model, we have also seen that this representation discards too much information and is consequently less discriminating than PCA-normalization methods. In the future, we would like to consider the possibility of extending the power spectrum representation to contain more rotation-invariant information, thereby obtaining a more discriminating representation that is still more compact than the underlying shape descriptor, allowing for efficient retrieval of 3D shapes in real world applications.

6.2.2 Symmetry

The method for computing the symmetry descriptor that we presented in Chapter 4 takes as its input a representation of a 3D model and returns the measure of the k-fold symmetries of the model with respect to all axes passing through the center. As such, this approach is focused on the detection of symmetries of whole objects. In the future, we would like to consider applications of this method to the symmetry detection of partial objects. The advantage of such an approach are two-fold. First, it would allow for the possibility of detecting local symmetries in a model, guiding methods for segmenting models into symmetric components. Second, it could be used to guess at the symmetries of models with missing data (e.g. parts that are occluded in the course of 3D scanning) thereby suggesting approaches for reconstructing the missing regions.

More generally, both local and global symmetry capture the redundancy of information within a model. As such, it is natural to consider methods for utilizing symmetry to guide compression algorithms for providing more compact representations of 3D shapes.

6.2.3 Anisotropy

In Chapter 5 we provided an iterative approach for transforming anisotropic 3D models into isotropic ones and proved that this approach is guaranteed to converge to a model with unit variance in every direction. In experimental evaluation, we have found that the convergence to an isotropic model is substantially more efficient than the one suggested by the proof. In the future, we would like to examine the possibility of the existence of a proof with tighter convergence bounds that more closely approximate the efficient convergence exhibited in practice. We would also like to investigate the possibility of directly transforming an anisotropic model into an isotropic one, without requiring multiple iterations. Because the convergence of the iterative process depends on the distribution of triangle normals over the surface of the model, we believe that it may be possible to use the Extended Gaussian Image [22] to design a method for directly transforming a 3D surface into an isotropic one.

6.2.4 Shape Descriptors

Finally, this thesis has described two general methods for extending existing shape representations by augmenting them with anisotropy and symmetry information. The value of these approaches depends fundamentally on the quality of the underlying shape representation, so that if the underlying representation is not discriminating enough to distinguish between shapes, the obtained augmented representation will also not be useful. Thus, one of the challenges in shape matching is the design of an efficient and effective shape representation. In the future, we would like to consider the more general questions of: “What does it mean for two models to be similar?” and “What is a good shape representation for capturing this notion of shape similarity” We believe that answers to these question exist both in the general area of shape matching, and in more domain-specific applications where the notion of model similarity may be more analytically defined. In either case, we believe that a good shape representation is fundamental to the task of shape matching and analysis and is a rich area for future research.

Bibliography

[1] 3D Cafe. `http://www.3dcafe.com`.

[2] 3D Model Retrieval System. `http://3d.csie.ntu.edu.tw/~dynamic/`.

[3] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation: A survey. Technical Report B 96-11, EVL-1996-142, Institute of Computer Science, Freie Universität Berlin, 1996.

[4] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In R. H. Güting, D. Papadias, and F. H. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99, Hong Kong, China, July 20-23, 1999, Proceedings*, volume 1651 of *Lecture Notes in Computer Science*, pages 207–226. Springer, 1999.

[5] M. J. Atallah. On symmetry detection. *IEEE Trans. on Computers*, c-34(7):663–666, July 1985.

[6] D. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE PAMI*, 12:111–122, 1981.

[7] S. Belongie and J. Malik. Matching with shape contexts. *IEEE Workshop on Content-based access of Image and Video-Libraries*, 2000.

[8] P. Besl. Triangles as a primary representation. *Object Recognition in Computer Vision*, LNCS 994:191–206, 1995.

[9] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE PAMI*, 14:239–256, 1992.

[10] G. Burel and H. Henocq. Three-dimensional invariants and their application to object recognition. *Signal Processing*, 45(1):1–22, 1995.

[11] Content-based Classification of 3D-models by Capturing Spatial Characteristics. `http://merkur01.inf.uni-konstanz.de/`.

[12] De Espona 3D Models Enciclopedia. `http://www.deespona.com`.

[13] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10:132–144, 1992.

[14] J. Driscoll and D. Healy. Computing Fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15:202–250, 1994.

[15] M. Elad, A. Tal, and S. Ar. Directed search in a 3d objects database using SVM. Technical Report HPL-2000-20R1, HP Labs, 2000.

[16] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[17] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, pages 83–105, 2003.

[18] Google. `http://www.google.com`.

[19] D. Healy, D. Rockmore, P. Kostelec, and S. Moore. FFTs for the 2-sphere – improvements and variations. *The Journal of Fourier Analysis and Applications*, 9:341–285, 2003.

[20] P. Highnam. Optimal algorithms for finding the symmetries of a planar point set. *Information Processing Letters*, 22:219–222, 1986.

[21] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. *Computer Graphics (Proceedings of SIGGRAPH 01)*, pages 203–212, 2001.

[22] B. Horn. Extended Gaussian images. In *Proceedings of the IEEE*, volume 72, pages 1656–1678, 1984.

[23] B. Horn. Closed form solutions of absolute orientation using unit quaternions. *Journal of the Optical Society*, 4:629–642, 1987.

[24] B. Horn, H. Hilden, and S. Negahdaripour. Closed form solutions of absolute orientation using orthonormal matrices. *Journal of the Optical Society*, 5:1127–1135, 1988.

[25] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. ACM STOC*, pages 604–613, 1998.

[26] A. E. Johnson and M. Hebert. Using spin-images for efficient multiple model recognition in cluttered 3-D scenes. *IEEE PAMI*, 21(5):433–449, 1999.

[27] S. Kang and K. Ikeuchi. Determining 3-d object pose using the complex extended Gaussian image. *CVPR*, pages 580–585, June 1991.

[28] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser. A reflective symmetry descriptor. *European Conference on Computer Vision (ECCV)*, pages 642–656, 2002.

[29] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A reflective symmetry descriptor for 3d models. *Algorithmica: Special Issue*, 2004.

[30] M. Kazhdan and T. Funkhouser. Harmonic 3d shape matching. Sketches and Applications, SIGGRAPH 2002.

[31] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. *Symposium on Geometry Processing*, pages 167–175, 2003.

[32] D. Knuth, J. J.H. Morris, and V. Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6(2):323–350, 1977.

[33] P. Kostelec and D. Rockmore. FFTs on the rotation group. Technical Report 03-11-060, Santa Fe Institute's Working Paper Series, 2003.

[34] J. Kovacs and W. Wriggers. Fast rotational matching. *Acta Cryst. D*, 58:1282–1286, 2002.

[35] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. *Proceedings of the 2nd International Conference on Computer Vision*, pages 238–249, 1988.

[36] C. Lo and H. Don. 3D moment forms: Their construction and application to object identification and positioning. *IEEE PAMI*, 11(10):1053–1064, October 1989.

[37] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[38] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface reconstruction algorithm. *SIGGRAPH Conference Proceedings*, pages 163–169, 1987.

[39] A. Makadia and K. Daniilidis. Direct 3D-rotation estimation from spherical images via a generalized shift theorem. *IEEE CVPR*, June 2003.

[40] G. Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE PAMI*, 11(1):104–108, January 1989.

[41] G. Mori, S. Belongie, and H. Malik. Shape contexts enable efficient retrieval of similar shapes. *Computer Vision and Pattern Recognition*, 1:723–730, 2001.

[42] Mr. Furniture. `http://www.mr-cad.com`.

[43] D. O'Mara and R. Owens. Measuring bilateral symmetry in digital images. In *TENCON Digital Signal Processing Applications*, pages 151–156. IEEE, 1996.

[44] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Shape Modeling International*, pages 154–166. IEEE Computer Society, 2001.

[45] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *Transactions on Pattern Analysis and Machine Intelligence*, pages 1105–1120, 1999.

[46] A. R. Pope. Model-based object recognition: A survey of recent research. Technical Report TR-94-04, University of British Columbia, January 1994.

[47] Princeton 3D Model Search Engine. `http://shape.cs.princeton.edu`.

[48] Princeton Shape Benchmark.
`http://shape.cs.princeton.edu/benchmark`.

[49] Protein Data Bank. `http://www.rcsb.org`.

[50] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. *IEEE ICCV*, 1998.

[51] J. Serre. *Linear Representations of Finite Groups*. Springer-Verlag, New York, 1977.

[52] ShapeSifter. `http://www.shapesearch.net/`.

[53] A. Shokoufandeh, S. Dickinson, C. Jonsson, L. Bretzner, and T. Lindeberg. On the representation and matching of qualitative shape at multiple scales. *European Conference on Computer Vision*, pages 759–775, 2002.

[54] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *Sixth International Conference on Computer Vision*, pages 222–229, 1998.

[55] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zukcer. Shock graphs and shape matching. *International Journal of Computer Vision*, pages 1–24, 1999.

[56] SOFT 1.0. `http://www.cs.dartmouth.edu/˜geelong/soft/`, 2003.

[57] SpharmonicKit 2.5.
`http://www.cs.dartmouth.edu/˜geelong/sphere/`, 1998.

[58] C. Sun. Symmetry detection using gradient information. *Pattern Recognition Letters*, 16:987–996, 1995.

[59] C. Sun and J. Sherrah. 3d symmetry detection using the extended Gaussian image. *IEEE PAMI*, 19(2):164–168, 1997.

[60] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Shape Modeling International*, May 2004.

[61] Viewpoint Data Labs. `http://www.viewpoint.com`, 2001.

[62] D. Vranic. An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres. In *IEEE International Conference on Image Processing*, September 2003.

[63] D. Vranic and D. Saupe. 3d model retrieval with spherical harmonics and moments. *Proceedings of the DAGM*, pages 392–397, 2001.

[64] J. D. Wolter, T. C. Woo, and R. A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, 1:37–48, 1985.

[65] Yahoo. `http://www.yahoo.com`.

[66] H. Zabrodsky, S. Peleg, and D. Avnir. A measure of symmetry based on shape similarity. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pages 703–706, Los Alamitos, California, 15–18 1992. IEEE Press.

[67] H. Zabrodsky, S. Peleg, and D. Avnir. Continuous symmetry for shapes. *2nd Intl. Workshop on Visual Form*, pages 594–613, 1994.

[68] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE PAMI*, 17(12):1154–1156, December 1995.

[69] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13(2):119–152, October 1994.

Appendix A

Signal Processing

In this appendix, we review three different aspects of spherical signal processing. First, we describe the spherical harmonics, a basis for the space of functions on the sphere. Second, we describe the Wigner-D functions, a basis of functions on the group of 3D rotations that describe how rotations interact with the spherical harmonics. Finally, we show how the fast spherical harmonic transform and the fast Wigner-D transform can be used to find the measure of similarity between two models, over all rotations and reflections, in an efficient manner.

A.1 Spherical Harmonics

One of the challenges of studying the action of the rotation group on the space of spherical functions results from the fact that this space is not finite-dimensional. Spherical harmonics provide a solution to this problem by decomposing the space of spherical functions into a sum of simple, rotationally-independent, finite-dimensional subspaces. Thus, they reduce the problem of understanding the action of rotations on the entire space of spherical functions to the problem of understanding the action of rotations on each of these subspaces independently.

More specifically, the spherical harmonics are an orthonormal basis for the space of functions defined on the surface of the sphere:

$$\{Y_l^m(\theta,\phi)\}_{l\geq 0,|m|\leq l} \quad \text{with} \quad Y_l^m(\theta,\phi)=\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)e^{im\phi}$$

(the P_l^m are the associated Legendre polynomials). They define the frequency subspaces:

$$V_l = \text{Span}(Y_l^{-l}, Y_l^{-l+1}, \ldots, Y_l^{l-1}, Y_l^l)$$

which has the following two essential properties:

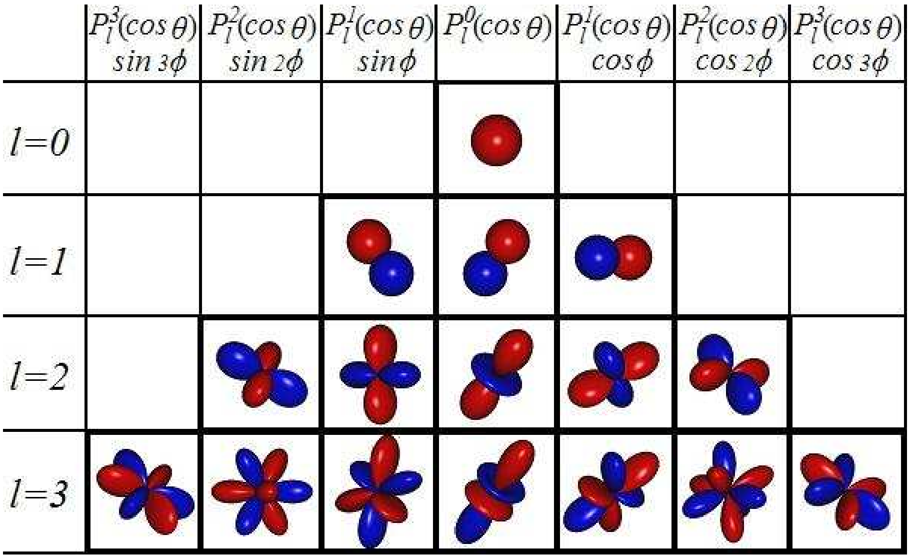

$P_l^3(cos\,\theta)$ $sin\,3\phi$
$P_l^2(cos\,\theta)$ $sin\,2\phi$
$P_l^1(cos\,\theta)$ $sin\,\phi$
$P_l^0(cos\,\theta)$
$P_l^1(cos\,\theta)$ $cos\,\phi$
$P_l^2(cos\,\theta)$ $cos\,2\phi$
$P_l^3(cos\,\theta)$ $cos\,3\phi$
l=0
l=1
l=2
l=3

Figure A.1: The spherical harmonics. The functions are visualized by scaling points on the sphere in proportion to the magnitude of the function at that point, where points with positive value are drawn in red, and points with negative value are drawn in blue.

1. V_l is a *representation*: The subspace V_l is closed under the action of rotation, so that for any rotation $\xi \in SO(3)$ and any function $f_l \in V_l$, we have $\xi(f_l) \in V_l$.

2. V_l is *irreducible*: The subspace V_l cannot be decomposed further as the sum of (non-trivial) representations.

Figure A.1 shows a visualization of the spherical harmonics. The functions are visualized by scaling points on the sphere in proportion to the magnitude of the function at that point, where points with positive value are drawn in red, and points with negative value are drawn in blue. Each row represents a different frequency l, with functions within a row corresponding to the different Y_l^m, for fixed l. Note that as the frequency increases the number of different Y_l^m increases, and the number of lobes, or undulations, of the constituent functions also increases.

Efficient methods for computing the forward and inverse spherical harmonic transform of a spherical function have been developed [14, 19, 57], performing the transformation in $O(b^2 \log b^2)$ time for functions defined on the surface of a sphere, sampled on a regular $O(b^2)$ grid.

A.2 Wigner-D Functions

While the spherical harmonics provide a decomposition of the space of spherical functions into rotation-independent components, the Wigner-D functions describe how rotations act on the spherical harmonics. These are an orthonormal basis for the space of functions defined on the group of rotations:

$$D_l^{m,n}(\xi) = \langle Y_l^m, \xi(Y_l^n)\rangle.$$

Thus, given the spherical harmonic decomposition of a function f as:

$$f = \sum_{l=0}^{\infty} \sum_{|m|\leq l} f_{l,m} Y_l^m,$$

the coefficients of any rotation of f can be expressed in terms of the spherical harmonic coefficients of f and the Wigner-D functions. In particular, for any rotation $\xi \in SO(3)$:

$$\xi(f) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} f_{l,m} \xi(Y_l^m) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} \left(\sum_{|n|\leq l} f_{l,n} D_l^{n,m}(\xi) \right) Y_l^m.$$

Efficient methods for computing the forward and inverse Wigner-D transform of a function defined on the group of rotations have been developed [14, 19, 33, 56], performing the transformation in $O(b^4)$ time for functions defined on the group of rotations, sampled on a regular $O(b^3)$ grid.

A.3 Correlation

In this thesis, we are primarily concerned with matching 3D models across different rotations and reflections. To this end, the spherical harmonics transform and the Wigner-D transform provide tools for the efficient computation of the distance between two models over the space of all possible rotations/reflections [34]. In particular, given two spherical functions f and g, our goal is to compute:

$$\|f - \zeta(g)\|^2 = \|f\|^2 + \|g\|^2 - 2\langle f, \zeta(g)\rangle$$

for every rotation/reflection $\zeta \in O(3)$. This reduces to the problem of computing the correlation of the two functions $\langle f, \zeta(g)\rangle$ over the space of all rotations/reflections $\zeta \in O(3)$. Using the forward spherical harmonic transform to decompose the spherical functions in terms of their harmonics, cross multiplying harmonic coefficients within each frequency, and then applying the inverse Wigner-D transform, provides an efficient method for computing the correlation of the two functions at every rotation. In particular, if we express the functions f and g in terms of their harmonics:

$$f = \sum_{l=0}^{\infty} \sum_{|m| \leq l} f_{l,m} Y_l^m \qquad \text{and} \qquad g = \sum_{l=0}^{\infty} \sum_{|m| \leq l} g_{l,m} Y_l^m$$

then the correlation at a rotation $\xi \in SO(3)$ can be obtained by cross-multiplying harmonic terms within each frequency:

$$\langle f, \xi(g) \rangle = \sum_{l=0}^{\infty} \sum_{|m|,|n| \leq l} f_{l,m} \overline{g_{l,n}} \langle Y_l^m, \xi(Y_l^n) \rangle = \sum_{l=0}^{\infty} \sum_{|m|,|n| \leq l} f_{l,m} \overline{g_{l,n}} D_l^{m,n}(\xi).$$

This expresses the correlation as the linear sum of the Wigner-D functions and the inverse Wigner-D transform can be used to get the value of the correlation at every rotation.

If $\zeta \in O(3)$ and ζ is not a rotation (i.e. ζ has determinant -1) then ζ is the product of some rotation $\xi \in SO(3)$ and the antipodal map A – the transformation that sends a point p to the point $-p$. In order to compute the value of the correlation of f with g at ζ, we observe that the antipodal map acts on a function f as follows:

1. If f is an even function then the antipodal map leaves f unchanged
2. If f is an odd function then the antipodal map sends the function f to $-f$.

Thus, we can compute the correlation $\langle f, \xi(A(g)) \rangle$ if we address the even and odd frequencies of f and g independently. In particular, we express f and g as the sum of their even and odd components, $f = f^+ + f^-$ and $g = g^+ + g^-$, with:

$$f^+(p) = \frac{f(p) + f(-p)}{2} \qquad \text{and} \qquad f^-(p) = \frac{f(p) - f(-p)}{2}$$

$$g^+(p) = \frac{g(p) + g(-p)}{2} \qquad \text{and} \qquad g^-(p) = \frac{g(p) - g(-p)}{2}$$

Then, instead of computing the correlation of f with g over all rotations, we use the inverse Wigner-D transform to compute the correlation of the even and odd parts independently to get:

$$\langle f^+, \xi(g^+) \rangle \qquad \text{and} \qquad \langle f^-, \xi(g^-) \rangle$$

This gives an expression for the correlation of f with g over all rotations/reflections, as:

$$\langle f, \xi(g) \rangle = \langle f^+, \xi(g^+) \rangle + \langle f^-, \xi(g^-) \rangle \qquad \langle f, \xi(A(g)) \rangle = \langle f^+, \xi(g^+) \rangle - \langle f^-, \xi(g^-) \rangle$$

for all $\xi \in SO(3)$. Since any $\zeta \in O(3)$ is either itself a rotation, or the product of some rotation $\xi \in SO(3)$ and the antipodal map, this provides a method for computing the correlation of f with g over all rotations/reflections.

Complexity: If the spherical functions are band-limited with band-width b and are represented as regular samples on a $2b \times 2b$ grid of spherical angles, then the spherical harmonic transform can be computed in $O(b^2 \log^2 b)$ time, the cross multiplication can be done in $O(b^3)$ time, and the inverse Wigner-D transform can be performed in $O(b^4)$ time, resulting in a total running time of $O(b^4)$ to compute the correlation at each of $O(b^3)$ rotation. (Note that a brute force algorithm would compute the dot-product at each of $O(b^3)$ rotations by performing an $O(b^2)$ comparison of f with the rotation of g, and would result in an overall running time of $O(b^5)$.)

Surprisingly, increasing the complexity of the function from a single spherical function to a collection of spherical functions, does not change the time for computing the correlation if the number of spherical functions is order $O(b)$. In particular, given the collection of spherical functions $\{f_1, \ldots, f_r\}$ and $\{g_1, \ldots, g_r\}$, the L_2 difference between the two sets of functions, at any rotation/reflection ζ, is given by:

$$\sum_{j=1}^{r} \|f_j - \zeta(g_j)\|^2 = \sum_{j=1}^{r} \left(\|f_j\|^2 + \|g_j\|^2\right) - 2\sum_{j=1}^{r} \langle f_j, \zeta(g_j)\rangle.$$

Thus, as in the case of a single spherical function, computing the difference over all rotations/reflections reduces to the problem of computing the correlation. Expressing each spherical function in terms of its harmonic decomposition:

$$f_j = \sum_{l=0}^{b} \sum_{|m|\leq l} f_{l,m}[j] Y_l^m \qquad \text{and} \qquad g_i = \sum_{l=0}^{b} \sum_{|m|\leq l} g_{l,m}[j] Y_l^m$$

allows us to define the correlation of the functions $\{f_1, \ldots, f_r\}$ and $\{g_1, \ldots, g_r\}$ at every rotation/reflection $\zeta \in O(3)$ as:

$$\sum_{j=1}^{r} \langle f_j, \zeta(g_j)\rangle = \sum_{l=0}^{\infty} \sum_{|m|,|n|\leq l} \left(\sum_{j=1}^{r} f_{l,m}[j] \overline{g_{l,n}[j]} \right) D_l^{m,n}(\zeta).$$

Complexity: If the spherical functions are band-limited with band-width b and there are $O(b)$ of them, then the correlation can be computed in $O(b^4)$ time. In particular, the harmonic transform is computed in $O(b) \cdot O(b^2 \log^2 b)$ time, the cross multiplication can be done in $O(b) \cdot O(b^3)$ time, and the single inverse Wigner-D transform is still performed in $O(b^4)$ time. Thus, the limiting complexity of $O(b^4)$ for computing the inverse Wigner-D transform is not exceeded and computing the correlation of two collections of spherical functions can still be done in $O(b^4)$ time. (Note that a brute force algorithm would compute the dot-product at each of $O(b^3)$ rotations by performing an $O(b^3)$ comparison of $\{f_1, \ldots, f_r\}$ with the rotations of $\{g_1, \ldots, g_r\}$, and would result in an overall running time of $O(b^6)$.)