An Approximate and Efficient Method for Optimal Rotation Alignment of 3D Models

Michael Kazhdan

Michael Kazhdan is with the Johns Hopkins University.

Abstract

In many shape analysis applications, the ability to find the best rotation that aligns two models is an essential first step in the analysis process. In the past, methods for model alignment have either used normalization techniques, such as PCA alignment, or have performed an exhaustive search over the space of rotation to find the best optimal alignment. While normalization techniques have the advantage of efficiency, providing a quick method for registering two shapes, they are often imprecise and can give rise to poor alignments. Conversely, exhaustive search is guaranteed to provide the correct answer, but even using efficient signal processing techniques, this type of approach can be prohibitively slow.

In this paper we present a new method for aligning two 3D shapes. We show that the method is markedly faster than existing approaches based on efficient signal processing and we provide registration results demonstrating that the alignments obtained using our method have a high degree of precision and are markedly better then those obtained using normalization.

Index Terms

Alignment, Matching, Retrieval, Shape Descriptors, Signal Processing

I. Introduction

In many shape analysis applications, finding the rotation that best aligns two 3D models is a necessary step in the analysis process: Systems for matching 3D models [22], [4] generally assume that the shape of a model is unchanged when acted on by a rotation, so that comparing two models requires finding the rotation at which the two models are optimally pairwise aligned. In systems allowing a user to construct new models by transferring parts from existing ones [23], [3], an alignment step is necessary before replacing the old part with the new. Methods for transfering texture, segmentation, and surface detail, [14] first need to establish correspondences between the two surfaces and solving for the optimal rigid body alignment is a necessary first step in this process. Often, hole-filling methods [16] use intact patches from the surface of a model to fill in parts with missing data, and again finding the transformation that best aligns the complete patch to the patch with missing data is a necessary step in the process.

One approach to addressing this problem is to represent each model by a function (defined either on the sphere or in 3D) and then to find the rotation that minimizes the distance between the two functions. This approach transforms the alignment problem into a correlation problem, making it possible to use recently developed, efficient signal processing techniques to find the optimal rotation. However, while these signal processing techniques have running times that are markedly faster than a brute force search over the space over all rotations, (taking $O(b^4)$ time to align two spherical or 3D functions of band-width b, as compared to the brute force approaches which takes order $O(b^5)$ and $O(b^6)$ respectively), they are still too slow for applications in which many candidate models need to be compared against a single query.

To address this concern, previous research has approached the correlation problem using parameter factorization techniques: The space of rotations is parameterized by two variables, the value of the correlation is optimized over the first variable, and then refined by holding the first variable fixed and optimizing over the second. Thus, rather than performing a computationally expensive search over the entirety of the three-dimensional space of rotations, models can be efficiently aligned by performing independent optimizations over two smaller spaces.

An example of such an approach is proposed by Makadia *et al.* [11], [12]) where the space of rotations is parameterized using the axis/angle parameterization, representing each rotation by the axis about which the rotation occurs and the angle of the rotation. The initial axis optimization step can then be performed using the Fast Spherical Harmonic Transform and the angular refinement can be performed using the Fast Fourier Transform.

The key challenge in this type of approach results from the fact that the values of the correlation are not separable. Thus, the value of the axis parameter optimizing the initial alignment step is not guaranteed to be equal to, or even close to, the value of the axis parameter at the optimizing the full correlation.

In this paper, we show that the initial step of optimizing over the axis of rotation can be interpreted as a *global* optimization of an axially symmetric subset of the model. Thus, by pre-processing the models so as to ensure that the axially symmetric subsets are as large as possible, we obtain an axis parameter in the first step whose value is more likely to be close to the value of the axis parameter optimizing the full correlation. Although this approach is only guaranteed to give the correct alignment when the initial model is axially symmetric, the pre-processing steps ensures that even when the model is not axially symmetric, the axis parameter obtained in the first step is as meaningful as possible. This results in an alignment algorithm that is more accurate than previous methods, even for classes of models that do not have axial symmetry, closely approximating the results of exhaustive search without incurring the prohibitive computational cost.

The remainder of the paper is structured as follows. In Section II we review a number of exisiting shape descriptors that represent a 3D model as a function defined, either on the surface of a sphere, or in three-dimensions and in Section III we review the standard signal processing techniques that have been used to align two such shape descriptors. In Section IV, we present our approach for efficiently finding the rotation aligning two models and discuss the theoretical bounds of this approach. We present empirical results in Section V demonstrating the efficacy of our method and conclude by summarizing our approach in Section VI.

II. Related Work

In order to find the rotation that best aligns two 3D models, it is first necessary to define a measure of model similarity. Then, the optimal rotation can be defined as the rotation minimizing the distance between two models. In practice, directly computing the distance between two surfaces is a difficult and time consuming task. Instead, the approach that is commonly taken is to represent a 3D surface by a structured representation, or *shape descriptor*, that captures much of the salient model information. Two models can then be compared by independently computing each of their shape descriptors and then defining the measure of model similarity in terms of the distance between the descriptors.

Because of the importance of model comparison to many different areas of shape analysis, a large number of different shape descriptors have been proposed, ranging from structural descriptors that represent a 3D model by a graph and define the distance between two models in terms of the topological similarity of the associated graphs (e.g. [18], [17], [15], [6]) to statistical shape descriptors that represent a model as a vector in a fixed dimensional space and define the measure of model similarity in terms of the norm of the difference of the vectors.

In general, the structural shape descriptors are designed to be invariant to transformations and while well suited for model comparison, cannot be used for aligning two shapes. Similarly, statistical shape descriptors that are designed to be rotation invariant (e.g. [13], [8]) cannot be used for alignment because the measure of similarity between two models does not change when either of the models is rotated. Instead, in this work, we will focus on the statistical shape descriptors that represent a model by rotation-varying functions (e.g. [7], [1], [22], [21], [4], [3]). These are descriptors that represent a 3D model by a function defined either on the surface of the sphere or in three-space, satisfying the property that rotating a model and then computing its shape descriptor is equivalent to first computing the shape descriptor and then applying the rotation to the descriptor.

III. Signal Processing Review

In this section we present a review of the signal processing framework that serves as the foundation for our research. We begin by establishing notation. We present a review of the spherical harmonic and Wigner-D decompositions. Finally, we describe how these tools can be used to solve for the optimal alignment of two spherical functions.

A. Notation

- *Parameterizing the Sphere*: We parameterize points on the surface of the sphere, S^2, using angles of elevation $\theta \in [0, \pi]$ and azimuth $\phi \in [0, 2\pi)$:

$$\Phi(\theta, \phi) \longrightarrow (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta)).$$

Note that this parameterization has the property that a rotation about the z-axis by ϕ_0 degrees, corresponds to a shift in the azimuthal value:

$$(\theta, \phi) \mapsto (\theta, \phi + \phi_0).$$

- *Parameterizing Rotations*: We parameterize the space of rotations, $SO(3)$, in terms of the triplet of Euler angles $\theta \in [0, \pi]$ and $\phi, \psi \in [0, 2\pi)$, expressing every rotation as a rotation about the y-axis, multiplied on the left and right by rotations about the z-axis:

$$R(\phi, \theta, \psi) \rightarrow \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that this parameterization has the property that the rotation $R(\phi, \theta, \psi)$ can be decomposed as the product of rotations:

$$R(\phi, \theta, \psi) = R(\phi)R(\theta, \psi)$$

where $R(\theta, \psi)$ is a rotation mapping the point $\Phi(\theta, \pi - \psi)$ to the vector $(0, 0, 1)$ on the z-axis and $R(\phi)$ is a rotation by ϕ degrees about the z-axis.

- *Action of Rotations on Spherical Functions*: Given a function on the surface of a sphere f and given a rotation $R \in SO(3)$ the rotated function $R(f)$ is defined as the function whose value at a point p is:

$$R(f)(p) = f(R^{-1}(p)).$$

B. Spherical Harmonics

Given a function $f(\theta, \phi)$ defined on the surface of the sphere, sampled on an $n \times n$ grid, then it can be expressed in terms of the orthonormal spherical harmonic basis

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

where the P_l^m are the associated Legendre polynomials, $b = n/2$ is the band-width of the spherical signal, $0 \leq l < b$ and $|m| \leq l$. In particular, there are complex coefficients f_l^m such that f can be expressed as the sum:

$$f(\theta, \phi) = \sum_{l=0}^{b-1} \sum_{|m| \leq l} f_l^m Y_l^m(\theta, \phi). \tag{1}$$

The spherical harmonics $\{Y_l^m\}$ are solutions to Laplace's Equation in spherical coordinates and satisfy the condition that for any rotation R, the functions Y_l^m and $R(Y_{l'}^{m'})$ are orthogonal whenever $l \neq l'$:

$$\langle Y_l^m, R(Y_{l'}^{m'}) \rangle = 0 \qquad \forall l \neq l'. \tag{2}$$

Furthermore, because the associated Legendre polynomials P_l^m are independent of ϕ, the spherical harmonics have the property that a rotation about the z-axis by ϕ_0 degrees acts on the harmonics by multiplication:

$$Y_l^m(\theta, \phi - \phi_0) = e^{-im\phi_0} Y_l^m(\theta, \phi).$$

While a brute-force computation of the forward and inverse spherical harmonic expansions would require $O(b^4)$ time, (computing an $O(b^2)$ dot-product for each of $O(b^2)$ coefficients), recent developments in signal processing have provided efficient algorithms that compute the transforms in $O(b^2 \log^2 b)$ time [2], [5], [20].

C. Wigner-D Functions

Given a function $F(\phi,\theta,\psi)$ defined on the space of rotations, if the function has band-width b, then it can be expressed in terms of the orthonormal Wigner-D basis $\{D^l_{m,m'}(\phi,\theta,\psi)\}$, with $0 \leq l < b$ and $|m|, |m'| \leq l$, with complex coefficients $F^{m,m'}_l$:

$$F(\phi,\theta,\psi) = \sum_{l=0}^{b-1} \sum_{|m|,|m'|\leq l} F^{m,m'}_l D^l_{m,m'}(\phi,\theta,\psi). \tag{3}$$

The Wigner-D functions $\{D^l_{m,m'}\}$ express the correlation of the spherical harmonics over the space of rotations, so that for any rotation $R \in SO(3)$ the Wigner-D functions satisfy:

$$D^l_{m',m}(R) = \langle R(Y^m_l), Y^{m'}_l \rangle. \tag{4}$$

While a brute-force computation of the forward and inverse Wigner-D expansions would require $O(b^6)$ time, (computing an $O(b^3)$ dot-product for each of $O(b^3)$ Wigner-D coefficients), recent developments in signal processing have provided efficient algorithms that compute the transforms in $O(b^4)$ time [10], [19].

D. Optimal Alignment

In many applications, it is often necessary to find the rotation $R \in SO(3)$ that minimizes the L_2-distance between two spherical functions f and g:

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

Since rotations do not change the size of a function ($\|R(f)\| = \|f\|$) finding the rotation minimizing the L_2-distance is equivalent to finding the rotation maximizing the dot-product:

$$\langle R(f), g \rangle.$$

If the functions f and g are expressed in terms of their spherical harmonics, then applying Eq. 2 and Eq. 4, the function giving the dot product at every rotation R can be expressed as:

$$\sum_{\substack{l=0\\|m|,|m'|\leq l}}^{b-1} f^m_l \overline{g^{m'}_l} \langle R(Y^m_l), Y^{m'}_l \rangle = \sum_{\substack{l=0\\|m|,|m'|\leq l}}^{b-1} f^m_l \overline{g^{m'}_l} D^l_{m',m}(R). \tag{5}$$

Since the equation on the right expresses the dot product as a linear sum of the Wigner-D functions, the rotation maximizing the dot product can be obtained by computing the inverse

Wigner-D transform to get the value of the dot product at each (discretized) Euler angle and then finding the Euler angle with largest value.

While a brute-force computation of the dot-product at every rotation would require $O(b^5)$ time, (computing an $O(b^2)$ dot-product for each of $O(b^3)$ rotations), using the spherical harmonic transform to get the harmonic coefficients of f and g (order $O(b^2 \log^2 b)$ time), cross-multiplying the spherical harmonic coefficients in Eq. 5 (order $O(b^3)$ time), computing the inverse Wigner-D transform (order $O(b^4)$ time) and sampling the obtained dot-product function (order $O(b^3)$ time) gives an order $O(b^4)$ algorithm for finding the rotation that minimizes the L_2-distance between two spherical functions.

E. Extending to 3D Functions

We can generalize this method to functions defined on the interior of the unit sphere. If F is a function defined on the interior of a unit sphere, we can represent F as a collection of spherical functions $\{F_r\}$, by restricting the function F to spheres with different radii:

$$F_r(p) = F(rp)\sqrt{4\pi r^2}$$

where the square-root accounts for the change of area, so that for any two functions F and G defined in the interior of the sphere, we have:

$$\int_{\|x\|\leq 1} F(x)G(x)dx = \int_0^1 \int_{p\in S^2} F_r(p)G_r(p)dpdr.$$

Now, we can apply the method described above, expressing the dot product of F with G as:

$$\langle R(F), G\rangle = \sum_{l=0}^{b-1} \sum_{|m|,|m'|\leq l} \left(\int_0^1 (F_r)_l^m \overline{(G_r)_l^{m'}} dr \right) D_{m',m}^l(R).$$

If we sample the radii at $O(b)$ different points, computing the harmonic coefficients of the spherical restrictions takes $O(b^3 \log^2 b)$ time, computing the inverse spherical harmonic transform takes $O(b^2 \log^2 b)$ time, cross-multiplying the spherical harmonic coefficients takes order $O(b^4)$ time, computing the inverse Wigner-D transform takes order $O(b^4)$ time and sampling the obtained dot-product function takes $O(b^3)$ time, giving an order $O(b^4)$ algorithm for finding the rotation that minimizes the L_2-distance between two functions defined on the interior of the unit sphere.

IV. Approximating Optimal Alignment

In this section we present our approach for efficiently finding the rotation that aligns two models. Given a shape descriptor that represents the model as a spherical function, we describe a method for aligning two models in a way that minimizes the L_2-distance between the shape descriptors. We begin by showing that in the case that one of the two models is axially symmetric (i.e. obtained by rotating a curve about some fixed axis) the optimal rotation can be found efficiently. Then, we show how the method for axially symmetric models can be generalized to models that are not axially symmetric, to give an approximate and efficient alignment method.

A. Axially Symmetric Alignment

In general, when we consider the correlation of the function f with the function g, the resulting correlation function is a function defined on the three-dimensional space of rotation, for every $R \in SO(3)$ giving the value of the dot-product $\langle R(f), g\rangle$. However, in the case that the function g is axially symmetric with respect to the z-axis, (i.e. rotations about the z-axis leave g unchanged) the correlation function becomes much simpler. If we express the rotation $R(\phi,\theta,\psi)$ as:

$$R(\phi,\theta,\psi) = R(\phi)R(\theta,\psi)$$

where $R(\theta,\psi)$ is a rotation taking the point $\Phi(\theta,\pi-\psi)$ to the positive z-axis and $R(\phi)$ is a rotation by ϕ degrees about the z-axis, then the value of the dot product $\langle R(f), g\rangle$ is independent of the value of ϕ:

$$\begin{aligned}\langle R(\phi,\theta,\psi)(f), g\rangle &= \langle R(\phi)R(\theta,\psi)(f), g\rangle = \langle R(\theta,\psi)(f), R(-\phi)(g)\rangle \\ &= \langle R(\theta,\psi)(f), g\rangle.\end{aligned}$$

Thus the correlation function becomes a function defined on the two-dimensional sphere, for every point $p \in S^2$ giving the value of the dot-product $\langle R_p(f), g\rangle$ where R_p is any rotation mapping the point p to the positive z-axis.

We can leverage the simplicity of the correlation function in the case that the function g is axially symmetric with respect to the z-axis to give an alignment method that is markedly faster than the general method requiring the use of the inverse Wigner-D transform. In particular, if the function g is axially symmetric with respect to the z-axis, the spherical harmonics coefficients of g satisfy the condition:

$$g_l^m = 0 \qquad \forall m \neq 0$$

In this case, the expression for the dot product of g with every rotation of f becomes:

$$\langle R(f), g \rangle = \sum_{l=0}^{b-1} \sum_{|m| \leq l} f_l^m \overline{g_l^0} D_{0,m}^l(R). \tag{6}$$

Taking advantage of the identity

$$D_{0,m}^l(\theta, \phi, \psi) = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\theta, \pi - \psi)$$

we express the initial correlation function as:

$$\langle R(\phi, \theta, \pi - \psi)(f), g \rangle = \sum_{\substack{l=0 \\ |m| \leq l}}^{b} \sqrt{\frac{4\pi}{2l+1}} f_l^m \overline{g_l^0} Y_l^m(\theta, \psi),$$

giving an expression for the dot product as a linear sum of spherical harmonics. Thus, instead of requiring the use of the inverse Wigner-D transform (which takes order $O(b^4)$ time to compute), in the case that the function g is axially symmetric with respect to the z-axis we can compute the correlation function by simply computing the inverse spherical harmonic transform (taking $O(b^2 \log^2 b)$ time).

B. General Alignment

In general, however, a function need not be axially symmetric with respect to the z-axis and the method described above cannot be applied directly to find the optimal alignment between two functions. We propose a novel method for aligning two spherical functions that aligns the functions f and g in four steps:

1) Compute the z-axially symmetric component of g:

$$\tilde{g}(\theta, \phi) = \sum_{l=0}^{b} g_l^0 Y_l^0(\theta, \phi) = \int_0^{2\pi} g(\theta, \phi) d\phi.$$

2) Use the method described above to find the best rotation $R_2 = R(\theta, \psi)$, aligning the function f with the function $\tilde{g}$.
3) Find the optimal rotation about the z-axis $R_1 = R(\phi)$ aligning $R_2(f)$ with g.
4) Set the aligning rotation to be the product of the two rotations $R = R_1 R_2$.

The idea behind this approach is to factor the alignment problem so that instead of directly searching over the three-dimensional space of rotations for the optimal alignment, we first search over the two-dimensional sphere to find the best rotation aligning f with the z-axially symmetric component of g and then perform a search over the one-dimensional space of rotations about the z-axis. Since aligning f with the z-axially symmetric component of g can be performed in $O(b^2 \log^2 b)$ time (using the fast spherical harmonic transform) and since aligning about the z-axis can be done in $O(b^2 \log b)$ time (using the fast Fourier transform) the overall running time of our method is $O(b^2 \log^2 b)$. This is particularly meaningful when we consider the fact that the input functions are of size $O(b^2)$ and the space of rotations is of size $O(b^3)$, as we provide an alignment method whose complexity is smaller than the space of rotations.

In general, the method we propose is not guaranteed to give the optimal alignment as the optimal rotation $R(\phi, \theta, \psi)$ aliging f with g does not necessarily have the property that $R(\theta, \psi)$ is the optimal rotation aligning f with the axially symmetric $\tilde{g}$. In particular, in the case that the L_2-norm of $\tilde{g}$ is much smaller than the L_2-norm of g, aligning f with $\tilde{g}$ we ignore much of the information contained in g and the obtained alignment $R(\theta, \psi)$ may not be meaningful.

To address this problem we propose a pre-processing step in which the function g is first aligned so as to maximize the amount of information in $\tilde{g}$. To do this, we use the symmetry descriptor of [9]. In particular, using the method of Kazhdan *et al.*, we compute the spherical function $\left(SD^2_\infty(g)\right)(p)$ giving the square L_2-norm of the axially symmetric component of g with respect to the axis p. Finding the axis p_0 at which the function $\left(SD^2_\infty(g)\right)(p_0)$ is maximized, we rotate the function g so that p_0 maps to the positive z-axis. In the case that the initial function g was axially symmetric about some axis (not necessarily the z-axis), this pre-processing step is guaranteed to align g so that it is axially symmetric about the z-axis. In general, this method guarantees that the amount of information contained in the z-axially symmetric $\tilde{g}$ is maximized. Thus, even in the case that g is not axially symmetric, or has multiple axes with equal symmetry values, we ensure that the amount of information in $\tilde{g}$ is maximized, and hence the alignment $R(\theta, \psi)$ obtained in the first step of our method is as meaningful as possible.

Note that though computing the symmetry descriptor $SD^2_\infty(g)$ takes order $O(b^4)$ time, it needs to be done on a per-model basis and hence can be performed in a pre-processing step, so as not to affect the overall running time of our method.

Figure 1 outlines the steps of our approach: In a pre-processing stage, the axial symmetry descriptor of each target model is computed (1). In this visualization, the axial symmetry descriptor is visualized by scaling points on the unit sphere in proportion to the measure of axial symmetry about the corresponding axes. The model is then rotated so that the axis of maximal symmetry (highlighted in gray) is aligned with the z-axis (2). Then, when a query is presented, the query is correlated with the z-axially symmetric component of each target model (2), the query is then rotated so that the axis of maximal symmetry (highlighted in gray) is aligned with the positive z-axis (2), the correlation of the query with the target about the z-axis is computed (3), and the query is rotated about the z-axis by the angle that maximizes the correlation (4).

C. *Bounding the Approximation*

One of the advantages of our alignment method is that it provides a method for efficiently culling a large subset of rotations which are guaranteed not to provide the optimal alignment. In particular, if $R(\theta,\psi)$ is any rotation mapping the point $p=\Phi(\theta,\pi-\psi)$ to the positive z-axis, we can use the values of the axial correlation function $\langle R(\theta,\psi)(f),\tilde{g}\rangle$ to bound the correlation value $\langle R(\phi,\theta,\psi)(f),g\rangle$ for any rotation by ϕ degrees about the z-axis. Specificaly, if $\left(SD^2_\infty(f)\right)(p)$ is the function giving the square L_2-norm of the axially symmetric component of f about p, we know that L_2-norm of the non-axially symmetric component of f about p is:

$$\sqrt{\|f\|^2-\left(SD^2_\infty(f)\right)(p)}.$$

Thus the product of the L_2-norm of the non-axially symmetric component of f about p with the L_2-norm of the non-axially symmetric component of g about the z-axis is equal to:

$$ASymDot(p)=\sqrt{\|f\|^2-\left(SD^2_\infty(f)\right)(p)}\left(\|g-\tilde{g}\|\right)$$

and we can bound the value of the correlation $\langle R(\phi,\theta,\psi)(f),g\rangle$ in terms of the value of the axial correlation function $\langle R_p(f),\tilde{g}\rangle$ and the product of the norms of the non-axially symmetric components of f and g:

$$ASymDot(p)\geq\|\langle R(\phi,\theta,\psi)(f),g\rangle-\langle R(\theta,\psi)(f),\tilde{g}\rangle\|.$$

Note that the smaller the value of the non-axially symmetric component of g about the z-axis (expressed as $\|g-\tilde{g}\|$) the smaller the value $ASymDot(p)$, and hence the tighter the bound on

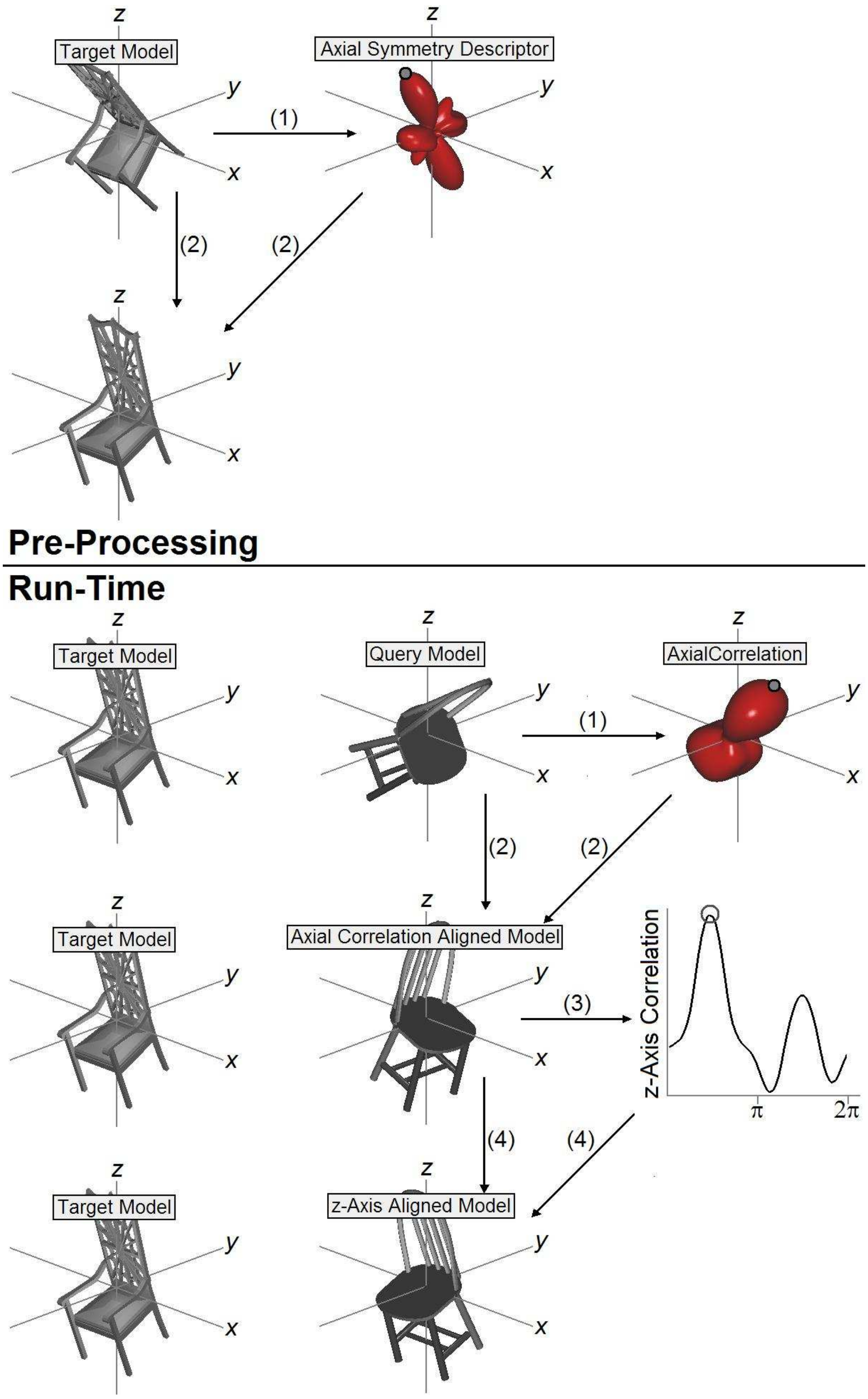

Fig. 1. The steps of our alignment approach: In a preprocessing stage, the axial symmetry descriptor is computed (1), and the model is rotated so that the axis with largest symmetry value maps to the z-axis (2). Then, when a query model is presented, the query is correlated with the z-axially symmetric component of the target model (1), the query is rotated so that the axis with largest symmetry values maps to the positive z-axis (2), the correlation of the models about the z-axis is computed (3), and the query is rotated about the z-axis by the angle that maximizes the correlation (4).

the value of the correlation $\langle R(\phi,\theta,\psi)(f),g\rangle$. Thus, by aligning g, in the pre-processing step so as to to maximize the size of the z-axially symmetric component $\tilde{g}$, we encourage the obtained approximate alignment to be a good one.

D. Extending to 3D Functions

As in Section III-E the method for finding the approximate alignment of two functions defined on the surface of a sphere can be generalized to functions defined on the interior of the sphere. In particular, defining the restriction of F to the sphere with radius r as F_r:

$$F_r(p) = F(rp)\sqrt{4\pi r^2}$$

we can apply the method described above, defining the spherical harmonic expansion of the axial correlation function as:

$$\sum_{l=0}^{b-1}\sum_{|m|\leq l}\sqrt{\frac{4\pi}{2l+1}}\left(\int_0^1 (F_r)_l^m\overline{(G_r)_l^0}dr\right)Y_l^m(\theta,\psi).$$

If we sample the radii at $O(b)$ different points, computing the harmonic coefficients of the axial correlation function takes $O(b^3)$ time, computing the inverse spherical harmonic transform takes $O(b^2\log^2 b)$ time, and correlating the functions about the z-axis takes $O(b^3\log b)$ time. Thus, the total running time for finding the approximate alignment of two functions defined in 3D is $O(b^3\log b)$ time. Since the complexity of each function is itself $O(b^3)$, this provides an alignment method that is only marginally slower than the time required to read in the functions.

V. EXPERIMENTAL RESULTS

To evaluate the efficacy of our approach, there are two qestions that we need to consider. First, how efficient is our method in practice? And second, how good is the obtained alignment? In this section we present the results of experiments designed to address these two concerns and we provide a brief discussion comparing our method to previous work.

A. Efficiency

To evaluate the efficiency of our method, we generated spherical and 3D functions of different band-widths, and compared the alignment times of our method with the alignment times of the fast signal processing approach described in Section III.D. These experiments were run on a PC

	Spherical Functions		(3D) Voxel Functions	
b	Fast SP	Our Method	Fast SP	Our Method
8	0.003	< 0.001	0.003	< 0.001
16	0.023	0.001	0.026	0.005
32	0.288	0.005	0.487	0.048
64	3.001	0.020	3.445	0.265
128	26.698	0.090	34.589	3.135

TABLE I

TIME, IN SECONDS, FOR ALIGNING TWO FUNCTIONS DEFINED ON A SPHERE AND ON A 3D VOXEL GRID: COMPARING THE EFFICIENCY OF THE FAST SIGNAL PROCESSING APPROACH WITH OUR ALIGNMENT METHOD AT DIFFERENT BAND WIDTHS.

with a 2.0GHz Pentium M processor with 2 GB of RAM and the results of these experiments are shown in Table I.

As expected, the $O(b^2 \log b)$ complexity of our approach for spherical functions makes it markedly faster than the fast signal processing method with $O(b^4)$ complexity, giving alignment times that are more than 100 times faster at high band widths. Also, we can observe that since the fast signal processing method has complexity $O(b^4)$ when aligning both spherical and 3D function, there is only a small increase in compute time for 3D functions. In our approach, however, aligning 3D functions changes the computation bottle-neck from the calculation of the Spherical Harmonic Transform, to the correlation of 3D functions about a fixed axis. As a result, the overall complexity of the approach changes from $O(b^2 \log^2 b)$ to $O(b^3 \log b)$. None-the-less, our approach still remains noticeably faster than the fast signal processing method, with alignment times that are more than 10 times faster at high band widths.

B. 3D Model Alignment

The alignment algorithm we propose is a general scheme that can be applied to any shape spherical or 3D shape descriptor that rotates with the model. As such, the accuracy of the alignment will depend on the specific shape descriptor used to represent the 3D model. In order to evaluate how well our method works in practice, we use two different shape descriptors:

- **Spherical Extent Function**: [22] This a shape descriptor that represents each model by a function defined on the surface of a sphere. The value of the function at any direction is defined by the distance, or extent, of the shape in that direction. Specifically, for any direction v, the value of the function is computed by intersecting the model with the ray, starting at the origin and having direction v, and computing the distance to the last point of intersection. In the case that the ray never intersects the model, the value of at v is set to 0. Computing the L_2-difference between two spherical extent functions, one gets the sum of squared distances that points on one model need to be moved in order to lie on the second model. (Note that since points can only be moved along rays through the origin, the defined measure of similarity is not necessarily the *minimum* sum of squared distances.)
- **Minimum Sum of Squared Distance**: [3] More recently, a shape descriptor has been proposed that enables the efficient computation of the minimum sum of squared distances between two models. This descriptor represents a model by two 3D functions. The first is a rasterization function, with value equal to 1 at any point intersecting the surface of the model and value 0 everywhere else. The second function is the squared Euclidean Distance Transform (EDT), whose value at any point is equal to the squared distance to the surface of the model. Given two such shape descriptors, the measure of similarity is defined as the dot product of the rasterization of the first model with the squared EDT of the second, plus the dot product of the rasterization of the second model with the squared EDT of the first. Since the dot product of the rasterization of one model with the squared EDT of the second gives the sum of the values of the squared EDT of the first model over all the points on the surface of the second model, the resulting measure of similarity is precisely the minimum sum of squared distances between the two surfaces.

  Although this method does not define the measure of similarity as the L_2-difference between two functions, we can still apply our approach to find the rotation minimizing the sum of the dot products.

We define the alignment error as the probability that a random rotation would align better. Specifically, given the function $D_{M,N}(R)$ returning the distance between two models M and N at rotation R, we define the alignment error of a given rotation R_0 as the fraction of rotations that align better:

$$\frac{\|\{R \in SO(3) \mid D_{M,N}(R) < D_{M,N}(R_0)\}\|}{\|SO(3)\|}$$

where the measure of rotations aligning better is computed by explicitly performing the full correlation using the Wigner-D transform (as described in Section III-D). Thus, if R_0 is the optimal rotation computed using exhaustive search, we always have $\text{Error}_{M,N}(R_0) = 0$.

Figure 2 gives an example of this error metric for the case when we would like to align a model of a dog with itself. The graph on the left shows the error values for the subset of rotations obtained by rotating the model about the view direction while the images on the right show the alignments at some of these rotations.

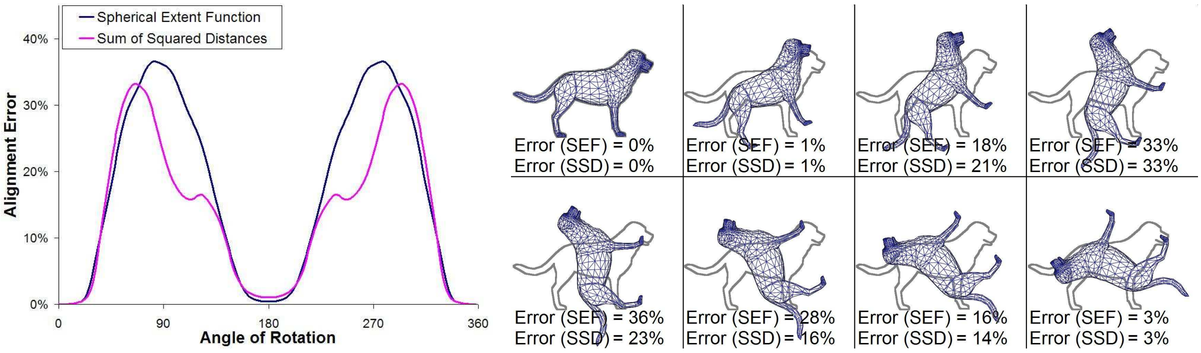

Fig. 2. Aligning a model of a dog with itself. The graph on the left shows the error values for the subset of rotations obtained by rotating the dog about the view direction while the figures on the right show the alignments at some of these rotations. Note that since we are aligning a model with itself, rotating by α and $360° - \alpha$ degrees give the same alignment error.

We evaluated the performance of our alignment method by generating a dataset consisting of 120 classified models with 10 models in each class. We then used our method to align every pair of models within a class and computed the alignment error. Tables II and III give the average and worst alignment error for each class using the spherical extent function and the minimum sum of squared distances to compare models. The tables also compare the results of our method to two existing methods: (1) PCA alignment, which aligns the model so that the eigen-values of the covariance matrix are aligned with coordinate axes, and (2) Makadia *al.*'s rotation estimation method [11], [12] which does not take into account maximal symmetry information.

As the table indicates, for both types of shape descriptors, our approach accurately finds the aligning rotation, with alignment errors that never excceed 1%. In contrast, aligning with PCA

	PCA Alignment		Makadia *et al.*		Our Method	
Class	Average /	Max	Average /	Max	Average /	Max
Cars	$< 0.01\%$ /	$< 0.01\%$	1.93% /	14.84%	$< 0.01\%$ /	$< 0.01\%$
Humans	$< 0.01\%$ /	0.01%	1.39% /	7.23%	$< 0.01\%$ /	0.05%
Swords	$< 0.01\%$ /	$< 0.01\%$	0.71% /	5.04%	$< 0.01\%$ /	0.01%
Pistols	0.01% /	0.07%	1.51% /	7.08%	0.01% /	0.11%
Planes	0.11% /	3.10%	1.11% /	5.11%	0.01% /	0.27%
Boats	0.19% /	1.11%	2.95% /	8.93%	$< 0.01\%$ /	0.03%
Shelves	0.19% /	6.93%	6.43% /	17.17%	$< 0.01\%$ /	0.04%
Helicopters	0.48% /	6.83%	1.54% /	7.94%	$< 0.01\%$ /	0.04%
Molecules	1.97% /	9.95%	1.99% /	10.10%	0.15% /	0.75%
Heads	3.02% /	14.75%	1.69% /	5.21%	0.03% /	0.56%
Chairs	3.73% /	29.45%	0.41% /	5.60%	0.08% /	0.93%
Plants	4.79% /	26.86%	1.59% /	5.45%	0.01% /	0.17%

TABLE II

ALIGNMENT ERRORS USING SPHERICAL EXTENT FUNCTIONS: A COMPARISON OF THE ACCURACY OF PCA, MAKADIA *et al.*'S ROTATION ESTIMATION METHOD, AND OUR APPROACH.

can give quite noticeable error values. Specifically, these results accenuate a limitation of PCA alignment: PCA alignment will work well for classes of models in which the principal axes are well distinguished. For classes of more isotropic models, such as vases (which have a strong axis of axial symmetry), and heads, plants, and molecules (which do not have a strong axis of axial symmetry), PCA is less robust and the resulting alignment is less accurate. In contrast, the results of our experiments indicate that our method aligns models well, independent of class and the extent of axial symmetry contained in the model.

C. Comparison to Previous Methods

The alignment algorithm we have presented in this paper takes a parameter factorization approach similar to that proposed by Makadia *et al.* – parameterizing rotations by the axis and

	PCA Alignment		Makadia *et al.*		Our Method	
Class	Average /	Max	Average /	Max	Average /	Max
Cars	$< 0.01\%$ /	$< 0.01\%$	7.32% /	21.63%	$< 0.01\%$ /	$< 0.01\%$
Humans	$< 0.01\%$ /	$< 0.01\%$	3.98% /	22.51%	$< 0.01\%$ /	0.03%
Swords	$< 0.01\%$ /	0.01%	11.64% /	30.34%	$< 0.01\%$ /	$< 0.01\%$
Pistols	0.03% /	0.56%	2.82% /	14.52%	$< 0.01\%$ /	0.05%
Planes	$< 0.01\%$ /	0.04%	4.65% /	24.60%	$< 0.01\%$ /	0.05%
Boats	0.05% /	0.21%	10.39% /	27.31%	$< 0.01\%$ /	0.01%
Shelves	0.49% /	4.49%	5.12% /	23.30%	$< 0.01\%$ /	0.03%
Helicopters	0.22% /	2.77%	2.09% /	19.36%	0.02% /	0.21%
Molecules	3.31% /	20.09%	2.19% /	16.06%	0.08% /	0.70%
Heads	1.44% /	10.85%	1.39% /	7.58%	0.01% /	0.21%
Chairs	0.87% /	7.38%	0.91% /	3.25%	$< 0.01\%$ /	0.01%
Plants	4.49% /	24.47%	0.73% /	10.53%	0.01% /	0.08%

TABLE III

ALIGNMENT ERRORS USING MINIMUM SUM OF SQUARED DISTANCES: A COMPARISON OF THE ACCURACY OF PCA, MAKADIA *et al.*'S ROTATION ESTIMATION METHOD, AND OUR APPROACH, WITH BAND-WIDTH $b = 64$.

angle of rotation, using the Fast Spherical Harmonic Transform to compute the optimal axial alignment, and then refining the alignment by using the Fast Fourier Transform to find the optimal angle of rotation.

The key contribution of our approach is the observation that the initial axial alignment can be interpreted as the optimal alignment of the z-axially symmetric component of the query model to the target. Thus, by introducing a pre-processing step that aligns the query model so at to maximize the z-axial symmetry, we obtain a parameter factorization approach that forces the optimization over the first parameter to return a parameter value that is more consistent with the value of the parameter optimizing the full alignment.

This improves on the alignment results of previous approaches in two ways:

- **Accuracy**: Since our pre-processing step maximizes the amount of model information used in the first optimization step, we obtain alignment results that are more accurate than those of previous methods. This is evidenced by the results in Tables II and III which compare the inter-class alignment performance of our method to the method of Makadia *et al.* [11], [12]. In these experiments, we find that though Makadia *et al.*'s method can out-perform PCA in the case of isotropic model alignment, it's inaccurate estimation of the axial correlation parameters gives alignment results that are almost 50 times less accurate in experiments with the Spherical Extent Function and almost 500 times less accurate in experiments with the Minimum Sum of Squared Distances descriptor.
- **Consistency**: The incorporation of the pre-processing step in our algorithm guarantees that the alignment of two shapes computed by our method will be independent of the initial poses of the models. Since the query model will always be aligned so that its axis of maximal symmetry aligns with the z-axis, after the pre-processing step, the pose of the query can only vary by a reflection through the origin and/or a rotation about the z-axis. Since the angular correlation will resolve differences in rotation about the z-axis and axial correlation will resolve reflections through the origin, the final alignment of the query will be unaffected by its pose. In contrast, alignment methods such as those proposed by Makadia *et al.*, perform the axial and angular correlation without a pre-processing step. Consequently the alignment results will vary with the initial pose of the model.

Figure 3 highlights the difference between our method and the method of Makadia *et al.* in an example of aligning a horse to a cow using the spherical extent function descriptor. The middle column shows the results of the axial alignment and angular refinement steps when Makadia *et al.*'s alignment method is used. The right column shows the results of the axial alignment and angular refinement steps when our pre-processing step is used to align the cow so that it's axis of maximal rotational symmetry aligns with the z-axis.

As the figure indicates, without a pre-processing step, the axial alignment step finds a poor alignment between the horse and the cow models. As a result, after performing the axial alignment, there is no rotation about the z-axis that can rotate the horse to the cow, and the parameter factorization fails to return the proper alignment. In contrast, when our pre-processing step is applied, the axial alignment step aligns the horse to the cow so that the poses of the two models only differ by a rotation about the z-axis. The angular refinement then finds this rotation, and the parameter factorization returns the proper alignment of the horse to the pre-processed cow model.

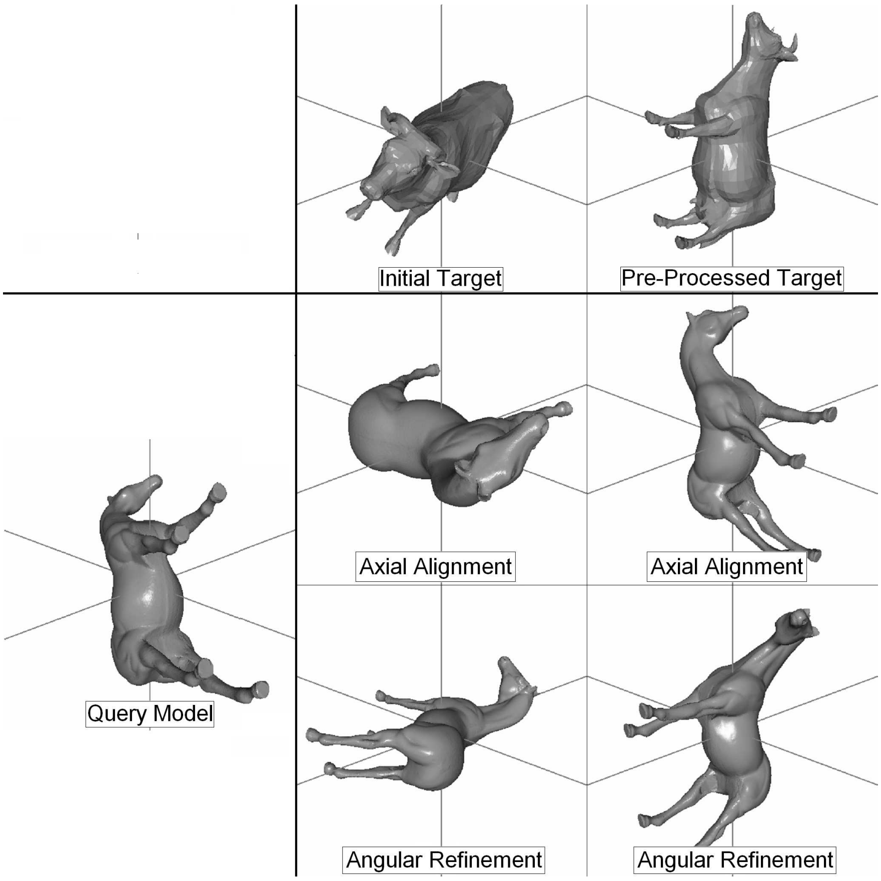

Fig. 3. An example of the results of using the parameter factorization approach to align a model of a horse to a model of a cow. The middle columns shows the results of the axial alignment and angular refinement steps when the parameter factorization is applied without a pre-processing step. The left column shows the results of the axial alignment and angular refinement steps when our pre-processing step is applied to align the cow so that the axis of dominant rotational symmetry is aligned with the z-axis.

VI. Conclusion

In conclusion, this paper present a novel approach that can be applied to a wide class of shape representation to facilitate the task of registration. In this paper, we have provided a method for efficiently finding the rotation aligning two models by transforming a search over the three-dimensional space of rotations into a search over the two-dimensional surface of the sphere, followed by a one-dimensional search over the circle. We have demonstrated that this approach provides a method for aligning models that is more efficient than standard fast signal processing techniques and gives rise to alignment results that are more accurate than those obtained using standard normalization methods. Thus, we have provided a novel and general tool that can be applied to the wide class of shape analysis problems in which the ability to register two 3D models plays a fundamental role.

References

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

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

[3] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., and Dobkin, D. 2004. Modeling by example. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 652–663.

[4] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. 2003. A search engine for 3d models. *ACM Transactions on Graphics 22*, 83–105.

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

[6] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)*, 203–212.

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

[8] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 167–175.

[9] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. 2004. Symmetry descriptors and 3d shape matching. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 116–125.

[10] Kovacs, J. and Wriggers, W. 2002. Fast rotational matching. *Acta Crystallographica 58*, 1282–1286.

[11] Makadia, A. and Daniilidis, K. 2003. Direct 3D-rotation estimation from spherical images via a generalized shift theorem. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 217–224.

[12] Makadia, A., Sorgi, L., and Daniilidis, K. 2004. Rotation estimation from spherical images. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 590–593.

[13] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. 2001. Matching 3d models with shape distributions. In *Proceedings of the SMI 2001 International Conference on Shape Modeling and Applications*, 154–166.

[14] Praun, E., Sweldens, W., and Schroder, P. 2001. Consistent mesh parameterizations. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)*, 179–184.

[15] SEBASTIAN, T., KLEIN, P., AND KIMIA, B. 2001. Recognition of shapes by editing shock graphs. In *Proceedings of the 8th International Conference on Computer Vision*, 755–762.

[16] SHARF, A., ALEXA, M., AND COHEN-OR, D. 2004. Context-based surface completion. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 878–887.

[17] SHOKOUFANDEH, A., DICKINSON, S. J., SIDDIQI, K., AND ZUCKER, S. W. 1999. Indexing using a spectral encoding of topological structure. In *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 491–497.

[18] SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S., AND ZUCKER, S. 1999. Shock graphs and shape matching. *International Journal of Computer Vision 35*, 13–32.

[19] SOFT 1.0:
`www.cs.dartmouth.edu/~geelong/soft/`.

[20] SPHARMONICKIT 2.5:
`www.cs.dartmouth.edu/~geelong/sphere/`.

[21] VRANIC, D. 2003. An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres. In *Proceedings of the 2003 International Conference on Image Processing*, 757–760.

[22] VRANIC, D. AND SAUPE, D. 2001. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM Symposium on Pattern Recognition*, 392–397.

[23] YU, Y., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 644–651.