Computing Spherical Transform and Convolution on the 2-Sphere

Boon Thye Thomas Yeo ythomas@mit.edu

May 10, 2005

Abstract

We propose a simple extension to the Least-Squares method of projecting samples of an unknown spherical function onto the spherical harmonics. Using Gram-Schmidt orthogonalization, our spherical transform (unlike previous algorithms) guarantees that the inverse transform converges to sample points without any assumption about sampling density or grid (such as the latitude-longitude grid), allowing us to perform spherical convolution with non lat-lon samples. Although we have not derived theoretical proofs, we will demonstrate that our algorithm achieves low interpolation errors on the uniformly random grid even when sampling below the theoretical non-aliasing rate of the lat-lon grid. While we will only consider real-valued spherical functions, our method should be readily extensible to complex-valued functions.

1 Introduction

We define spherical functions, $L^2(S^2)$ to be the Hilbert space of square integrable functions on the two dimensional sphere, S^2. In spherical coordinates, the usual inner product on the sphere is given by:

$$\langle f, h \rangle = \int_0^{\pi} \int_0^{2\pi} f(\theta, \phi) \overline{h(\theta, \phi)} \, sin\, \theta d\phi d\theta$$

where by convention, $0 \leq \theta \leq \pi$ (measured down from the z-axis) and $0 \leq \phi < 2\pi$ measured counterclockwise off the x-axis.

Spherical functions arise in many fields, such as shape modeling in computer vision [2] and medical imaging [9], computer graphics [7, 8], astrophysics [11] and geophysics [5]. The filtering of spherical functions is therefore an active area of research.

However, even simple linear space-invariant filtering is hard because it is not possible to discretize the surface of the sphere evenly such that the neighborhood of each point is the same. Therefore a popular method of performing spherical convolution is to first project the discretized spherical function and filter onto the span of spherical harmonics and perform the convolution in the fourier domain via simple multiplications.

The sampling theorem, convolution theorem and fast spherical transform presented in Dricoll and Healy's seminal paper [4] provides theoretical guarantees for this approach. The sampling theorem ensures that the discretization of the spherical functions is reversible (i.e. no aliasing), and the fast spherical transform allows for fast computation. Since then, there have been many improvements in creating faster spherical transform algorithms [3, 1].

However, the sampling theorem only holds for sampling on the lat-lon grid, which suffers from excessive sampling near the poles. In practice, the bandwidth of our function might not be known in advance, and so it is conceivably better if we spread out our measurements over the sphere.

In addition to the convolution theorem mentioned above, another reason for representing functions as a linear combination of spherical harmonics is the uniform resolution property of the spherical harmonics (see section 2) [4].

Our own motivation comes from the representation of brain cortical surfaces as functions on spheres. Neurobiologists believe that most of our higher cognitive abilities originate from the cerebral cortex, and that neurological growth or diseases significantly alter the structure of the cortex. Currently, we are working with MRI-segmented cortical surfaces that are mapped onto the topologically-equivalent spherical surfaces. In this case, the sampling is clearly of a more uniform nature than the lat-lon grid and the sampling density is outside our control.

In this paper, we present a method of computing the harmonic decomposition of a spherical function, not necessarily sampled on the lat-lon grid. Our transform guarantees that the inverse transform converges to the sample points. We will demonstrate empirically that our algorithm performs better than the usual Least-Squares method and fast spherical transform on the lat-lon grid. On the uniformly random grid, both our algorithm and Least-Squares achieve low interpolation errors when sampling below the theoretical non-aliasing rate of the lat-lon grid.

This paper is organized as follows. Section 2 goes through the basic definitions and properties of spherical harmonics, while section 3 formulates the Least-Squares approximation to the harmonic decomposition problem. This leads to the discussion of Gram-Schmidt orthogonalization in section 4 and computational issues in section 5, followed by experimental results and conclusions in section 6 and 7.

For completeness, we will mention that there are other competing methods of spherical filtering. Among these are adhoc techniques designed for particular filters, and therefore not extensible to all filters. A notable example is Swelden's lifting scheme [10], which constructs second generation wavelets capable of handling non-Euclidean manifolds, such as the sphere [7], [8]. Spherical wavelets utilizes the geodesic grid, which is obtained by projecting the icosahedron onto the sphere and then continually subdividing its faces into smaller triangles. The advantage of

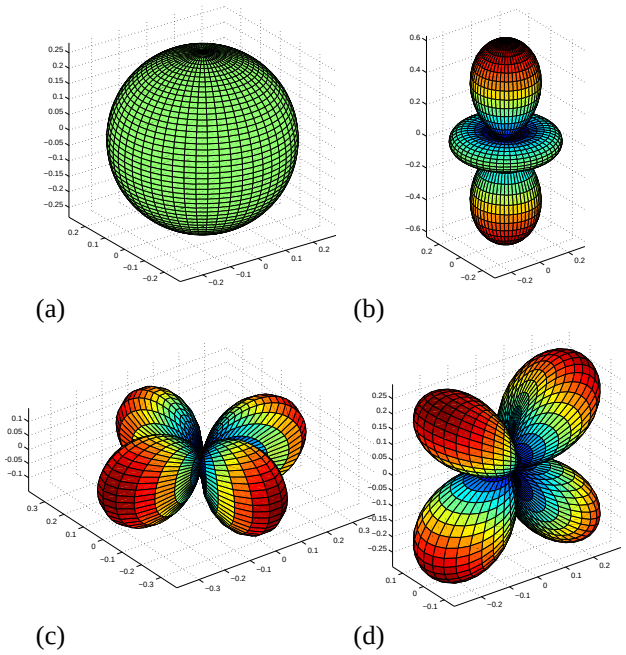

Figure 1: Low-order spherical harmonics. Only the real part is plotted for complex-valued harmonics. (a) $Y_0^0(\theta,\phi) = \frac{1}{2\sqrt{\pi}}$ (b) $Re\{Y_2^0(\theta,\phi)\} = \frac{1}{4}\sqrt{\frac{5}{\pi}}(3cos^2\theta - 1)$ (c) $Re\{Y_2^2(\theta,\phi)\} = \frac{1}{4}\sqrt{\frac{15}{2\pi}}sin^2\theta cos2\phi$ (d) $Re\{Y_2^1(\theta,\phi)\} = -\frac{1}{2}\sqrt{\frac{15}{2\pi}}sin\theta cos\theta cos\phi$. The color of the surface represents the value of the function. For example, in (a), $Y_0^0(\theta,\phi)$ is actually a sphere and hence all surface points are equidistant from the origin, and thus are of the same color.

starting with the icosahedron is that the resulting triangulation has the least imbalance in area between its constituent triangles [8]. Indeed, besides the lat-lon grid, the geodesic grid is often the spherical grid of choice. It is for example used in the iterative solution of partial differential equations that simulate ocean dynamics [5]. However, as a sampling grid, it suffers from the problem that we are unable to sample at intermediate number of points, as the number of vertices jump sharply with each subdivision. At level 7, there are already 163842 vertices.

2 Spherical Harmonics Preliminaries

For any integer $l \geq 0$ and integer m, $|m| \leq l$, the spherical harmonic of degree l and order m, $Y_l^m(\theta,\phi)$ takes the form for $m \geq 0$:

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

and for $m < 0$:

$$Y_l^m(\theta,\phi) = (-1)^m \overline{Y_l^{-m}(\theta,\phi)} \qquad (2)$$

where $P_l^m(x)$ are the associated Legendre Polynomials:

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}(1-x^2)^{m/2}\frac{d^{m+l}}{dx^{m+l}}(x^2-1)^l \qquad (3)$$

The spherical harmonics equations can be overwhelming at first, so we display a few of the harmonics in figure 1.

The spherical harmonics, $Y_l^m(\theta,\phi)$ form an orthonormal basis for $L^2(S^2)$. Therefore, we can expand a function, $f \in L^2(S^2)$ as

$$f = \sum_{l\geq 0}\sum_{|m|\leq l} \widehat{f}(l,m) Y_l^m(\theta,\phi) \qquad (4)$$

where $\widehat{f}(l,m)$ denotes the (l,m) harmonic coefficient, equal to $\langle f, Y_l^m\rangle$. Because of eq (4), approximating the harmonics coefficients is equivalent to interpolating our samples.

We state the following theorems without proof [4, 3]:

Theorem 1 *(Convolution Theorem [4]). For functions $f, h \in L^2(S^2)$, the transform of the convolution is a pointwise product of the transforms:*

$$\widehat{(f * h)}(l,m) = 2\pi\sqrt{\frac{4\pi}{2l+1}}\widehat{f}(l,m)\widehat{h}(l,0)$$

Observe the asymmetry of the convolution theorem: $\widehat{h}(l,0)$ vs $\widehat{f}(l,m)$. This comes from the definition of convolution as a left-convolution (averaging left translations). There is a similar theorem for right convolution (averaging right translations).

Another important thing to note about the convolution theorem is that it is independent of the sampling. Hence, as long as we can project our samples onto the span of spherical harmonics accurately, we can perform convolution via the fourier domain accurately, regardless of the sampling grid.

Theorem 2 *(Sampling Theorem [3]). Let $f \in L^2(S^2)$ be a bandlimited function of bandwith b, i.e. $\widehat{f}(l,m) = 0$ for $l \geq b$. Then:*

$$\widehat{f}(l,m) = \frac{\sqrt{2\pi}}{2b}\sum_{j=0}^{2b-1}\sum_{k=0}^{2b-1} a_j^{(b)} f(\theta_j,\phi_k)\overline{Y_m^l(\theta_j,\phi_k)}$$

for $l < b$ and $|m| \leq l$. Here $\theta_j = \pi\frac{2j+1}{4B}$, $\phi_k = \pi k/b$ and $a_j^{(b)}$ are deterministic weights designed to compensate for the oversampling at the poles.

Note that the proof only showed $(2b)^2$ samples to be sufficient (not necessary), although empirically our methods cannot handle less than $(2b)^2$ samples on the lat-lon grid (see section 6.2). While the sampling theorem only applies to the lat-lon grid, for the sake of comparison, we will still consider a function of bandwidth b to be undersampled when less than $(2b)^2$ samples are taken.

Theorem 3 *(Uniform Resolution [4]). Under a rotation, each spherical harmonic of degree l is transformed into a linear combination of only those harmonics of the same degree.*

Hence, the level of resolution of bandlimited functions obtained through the truncation of the harmonics is uniform all

over the sphere, thus providing an avenue of getting around the inability to discretize the sphere evenly [1].

We will now establish some properties of real-valued spherical functions. Consider a real-valued function f and define c_l^m to be the projection coefficient of the function f onto the spherical harmonic of degree l and order m. Since $c_l^m = \langle f, Y_l^m \rangle$, we have

$$\begin{aligned} c_l^{-m} &= \int_0^{\pi}\int_0^{2\pi} f(\theta,\phi)\overline{Y_l^{-m}(\theta,\phi)}\, sin\ \theta d\phi d\theta \\ &= \int_0^{\pi}\int_0^{2\pi} f(\theta,\phi)[(-1)^m Y_l^m(\theta,\phi)]\, sin\ \theta d\phi d\theta \\ &= (-1)^m \overline{\int_0^{\pi}\int_0^{2\pi} f(\theta,\phi)\overline{Y_l^m(\theta,\phi)}\, sin\ \theta d\phi d\theta} \\ &= (-1)^m \overline{c_l^m} \end{aligned} \tag{5}$$

where we used eq (2) and $f(\theta,\phi) = \overline{f(\theta,\phi)}$ since f is real-valued. Hence, when trying to find the spherical transform coefficients c_l^m, we only need to consider $m \geq 0$. Therefore, using the above, for $m > 0$:

$$\begin{aligned} c_l^{-m}Y_l^{-m} + c_l^m Y_l^m &= (-1)^m\overline{c_l^m}(-1)^m\overline{Y_l^m} + c_l^m Y_l^m \\ &= \overline{c_l^m Y_l^m} + c_l^m Y_l^m \\ &= 2Re\{c_l^m Y_l^m\} \\ &= 2(a_l^m R_l^m + b_l^m(-I_l^m)) \end{aligned} \tag{6}$$

where $c_l^m = a_l^m + ib_l^m$ and R_l^m, I_l^m are the real and imaginary parts of Y_l^m. Note that for $m = 0$, Y_l^m is real, and therefore c_l^0 must also be real in order for f to be real-valued.

3 Least-Squares Approximation

While $c_l^m = \langle f, Y_l^m \rangle$, in practice, we are unable to evaluate the inner product because we only know the values of f at its sample points. Because the sampling theorem only holds for lat-lon grid, we are unsure what to do for non lat-lon samples. Suppose we sample f at (θ_i, ϕ_i), $1 \leq i \leq n$, not necessarily on the lat-lon grid, we could try to discretize the integral [2]:

$$c_l^m \approx \frac{4\pi}{n}\sum_{i=0}^{n-1} f(\theta_i,\phi_i)\overline{Y_l^m(\theta_i,\phi_i)}$$

Unfortunately, while the spherical harmonics are orthonormal, their values evaluated at some set of parameter pairs (θ_i, ϕ_i) will generally not form an orthonormal set of vectors. However, what we really want is a spherical harmonic series that passes near the sample points [2], based on the assumption that a series that passes near the sample points will be close to the original function everywhere.

Here, we deviate from the formulation of [2], adapting their method to our problem at hand. Once again, letting R_l^m, I_l^m be the real and imaginary parts of Y_l^m, we define

$$\mathbf{B}^T = \begin{pmatrix} R_0^0(\theta_1,\phi_1) & R_0^0(\theta_2,\phi_2) & \dots & R_0^0(\theta_n,\phi_n) \\ R_1^0(\theta_1,\phi_1) & R_1^0(\theta_2,\phi_2) & \dots & R_1^0(\theta_n,\phi_n) \\ 2R_1^1(\theta_1,\phi_1) & 2R_1^1(\theta_2,\phi_2) & \dots & 2R_1^1(\theta_n,\phi_n) \\ -2I_1^1(\theta_1,\phi_1) & -2I_1^1(\theta_2,\phi_2) & \dots & -2I_1^1(\theta_n,\phi_n) \\ R_2^0(\theta_1,\phi_1) & R_2^0(\theta_2,\phi_2) & \dots & R_2^0(\theta_n,\phi_n) \\ \vdots & \vdots & \vdots & \vdots \\ -2I_l^l(\theta_1,\phi_1) & 2I_l^l(\theta_2,\phi_2) & \dots & -2I_l^l(\theta_n,\phi_n) \end{pmatrix} \tag{7}$$

Hence, B is a n x l^2 matrix containing the spherical harmonics up to degree l, evaluated at our sample points (with some scaling). Only the real part of order 0 harmonics are included because order 0 harmonics are real. The factor 2 and minuses are included here due to eq (6).

We denote the samples of f by the n x 1 vector $\vec{f}$ and define $\vec{c}^T = [a_0^0\ a_1^0\ a_1^1\ b_1^1\ a_2^0\ \dots\ b_l^l]$, where $c_l^m = a_l^m + ib_l^m$. Then our objective of finding a harmonic series that passes near the sample points is equivalent to finding a $\vec{c}$ that minimizes the error vector $\vec{e}$, such that $\vec{f} = B\vec{c} + \vec{e}$. If we adopt the l^2-norm as our measure of $\vec{e}$, then

$$\vec{c} = (B^T B)^{-1} B^T \vec{f} \tag{8}$$

is the Least-Squares solution that minimizes our error measure.

Hence, to project our sampled function onto the spherical harmonics up to degree l, we form the n x l^2 matrix, B and estimate the coefficients using eq (8). In practice, we can increase the degree l progressively, until a pre-determined accuracy is achieved.

The problem with the Least-Squares method is that $B^T B$ might not be invertible even when $l^2 < n$ or $l^2 < b$, where b is the bandwidth of our function. While $B^T B$ seems to be mostly invertible for uniformly random sampling (see section 6), there is no theoretical guarantees. In fact, when sampling on the lat-lon grid, the columns of B are often not linearly independent, when we have $< (2b)^2$ samples.

4 Gram-Schmidt Orthogonalization

Notice that in the Least-Squares approximation, as we progressively increase the degree of the spherical harmonics to improve our approximation, the number of columns of B increases but not the rows. Hence, to avoid having to worry about linear independence of B's columns, we can perform Gram-Schmidt orthogonalization. Since Gram-Schmidt orthogonalization is done in an iterative fashion, it is just right for the progressive addition of columns to B. The columns of B that are already in the span of previous columns are discarded.

Formally, we let $\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \dots\}$ denote the columns of B. Our aim is to produce an orthogonal sets of vectors $W = \{\vec{w}_1, \vec{w}_2, \vec{w}_3, \dots\}$ spanning the column space of B, giving priorities to the earlier columns (lower harmonics). We do this iteratively:

$$\begin{aligned} \vec{w}_1 &= \vec{v}_1 \\ \vec{w}_2 &= \vec{v}_2 - \frac{\langle \vec{v}_2, \vec{w}_1 \rangle}{\langle \vec{w}_1, \vec{w}_1 \rangle}\vec{w}_1 \end{aligned}$$

$$
\begin{aligned}
\vec{w}_3 &= \vec{v}_3 - \frac{\langle \vec{v}_3, \vec{w}_1 \rangle}{\langle \vec{w}_1, \vec{w}_1 \rangle}\vec{w}_1 - \frac{\langle \vec{v}_3, \vec{w}_2 \rangle}{\langle \vec{w}_2, \vec{w}_2 \rangle}\vec{w}_2 \\
\vec{w}_4 &= \vec{v}_4 - \frac{\langle \vec{v}_4, \vec{w}_1 \rangle}{\langle \vec{w}_1, \vec{w}_1 \rangle}\vec{w}_1 - \frac{\langle \vec{v}_4, \vec{w}_2 \rangle}{\langle \vec{w}_2, \vec{w}_2 \rangle}\vec{w}_2 - \frac{\langle \vec{v}_4, \vec{w}_3 \rangle}{\langle \vec{w}_3, \vec{w}_3 \rangle}\vec{w}_3 \\
\vdots \quad & \quad \vdots
\end{aligned} \tag{9}
$$

Note that if $\vec{v}_i$ lies in the span of the previous $\vec{v}_j$'s, then $\vec{w}_i = \vec{0}$ and we should discard it. In practice, we discard $\vec{w}_i$ if its norm is below a certain threshold. It is easy to verify that the $\vec{w}_i$'s are by construction orthogonal. The projection of $\vec{f}$ on the span of W is therefore given by $\vec{f} = \sum_i a_i \vec{w}_i = \sum_i \frac{\langle \vec{f}, \vec{w}_i \rangle}{\langle \vec{w}_i, \vec{w}_i \rangle}\vec{w}_i$. We can summarize our algorithm as follows:

1. Set $W = \{\}$ (the empty set), $l = -1$ and residual vector $\triangle\vec{f} = \vec{f}$.
2. Evaluate $|\triangle\vec{f}| = \frac{\langle \triangle\vec{f}, \triangle\vec{f} \rangle}{\langle \vec{f}, \vec{f} \rangle}$, the normalized residual error. Normalization by $\langle \vec{f}, \vec{f} \rangle$ is neccessary to eliminate scaling effects.
3. While ($|\triangle\vec{f}| >$ threshold),
   (a) Set $l = l + 1$.
   (b) Use Gram-Schmidt orthogonalization to check if the sampled harmonics of degree l is in the span of W
   (c) Add only the set of new orthogonal vectors, $\triangle W$ (if any) to W.
   (d) Set $\triangle\vec{f} = \triangle\vec{f} - \sum_{i \in \triangle W} \frac{\langle \vec{f}, \vec{w}_i \rangle}{\langle \vec{w}_i, \vec{w}_i \rangle}\vec{w}_i$ and record $\frac{\langle \vec{f}, \vec{w}_i \rangle}{\langle \vec{w}_i, \vec{w}_i \rangle}$.
   (e) Evaluate $|\triangle\vec{f}|$

In our experiments, threshold of $|\triangle\vec{f}|$ is set to 10^{-10}.

Observe that our algorithm has only computed the projection coefficients of $\vec{f}$ on W, but what we really want are the corresponding projection coefficients of $\vec{f}$ on the independent spherical harmonics. In theory, these can be recovered from the projection coefficients of $\vec{f}$ on W and the projection coefficients of $\vec{v}_i$'s on $\vec{w}_i$'s obtained from the orthogonalization process.

However, the inversion process can be quite messy to program and so in practice, we keep track of the independency between the sampled spherical harmonics during the orthogonalization process. When we complete the algorithm from above, we assemble the set of independent sampled harmonics and perform a least-squares projection of $\vec{f}$ on that set. The coefficients of the dependent harmonics are set to 0. Note that the coefficients obtained this way will be exactly the same as those obtained by the inversion process and will in fact avoid numerical issues associated with a progressively tinier residual vector, $\triangle\vec{f}$.

5 Evaluating the Harmonics

To evaluate the spherical harmonics, we make use of the three-term recurrence for $P_l^m(x)$, the associated Legendre Polynomials [1]:

$$xP_l^m = \alpha_l^m P_{l-1}^m + \alpha_{l+1}^m P_{l+1}^m \tag{10}$$

where,

$$\alpha_l^m = \sqrt{\frac{(l-m)(l+m)}{(2l-1)(2l+1)}} \tag{11}$$

and hence to calculate $P_l^m(x)$, we begin with $P_{m-1}^m(x) = 0$ and

$$P_m^m(x) = (-1)^m \frac{(1-x^2)^{\frac{m}{2}}}{2^m m!}\sqrt{\frac{2m+1}{2}(2m)!} \tag{12}$$

and use the recurrence eq. (10) to obtain

$$P_{m+1}^m = \frac{1}{\alpha_{m+1}^m}(xP_m^m - \alpha_m^m P_{m-1}^m) \tag{13}$$

and repeat till we achieve the desired $P_l^m(x)$.

Care must be taken to evaluate the constant $\frac{1}{2^m m!}\sqrt{\frac{2m+1}{2}(2m)!}$. If 2^m, $m!$ or $(2m)!$ is evaluated directly, the floating-point range can be easily exceeded even with moderate values of m. To avoid overflow, we begin with $\sqrt{\frac{2m+1}{2}}$, and multiply a factor from the denominator when the constant is greater than 1 and multiply a factor from the numerator when the constant is less than 1. When all the factors from either the numerator or denominator have been used up, we multiply the remaining factors [1].

6 Experiments

In this section, we compare the performance of the Gram-Schmidt projection, the Least-Squares approximation and the Fast Spherical Transform on both the lat-lon grid and the uniformly random grid. For the fast spherical transform, we actually utilize the non-speedup version, S2kit [6] because the fast spherical transform package, SpharmonicKit [6] only handles powers-of-2 bandwidths. However, this is fine since they both use the sampling theorem and we are only testing for accuracy not speed.

The uniformly random grid is defined as follows. Given an input N, we sample $(2N)^2$ random points uniformly around the sphere. This can be implemented as follows: For each sample, we generate the coordinate (θ, ϕ) by sampling ϕ uniformly in the range $[0, 2\pi]$ and θ from the probability density distribution $\frac{1}{2}sin\,\theta$, $0 \leq \theta \leq \pi$. The "sin" serves to compensate for the smaller area near the poles and plays an analogous role to the sin factor arising from integration on the sphere.

We generate two random real-valued bandlimited functions, f (bandwidth 20) and g (bandwidth 5), by setting the real and imaginary parts of the spherical harmonic coefficients uniformly in the range $[-1, 1]$, for $m \geq 0$. This determines the coefficients of negative order harmonics since $\overline{c_l^m} = c_l^{-m}(-1)^m$ (see eq. 2) for real-valued functions.

6.1 Toy Example

As a sanity check, we sample f on the lat-lon grid using $(2 * 20)^2 = 1600$ samples, and apply Least-Squares, Gram-Schmidt and S2kit transforms to the samples. We then compare the coefficients obtained by the transforms with the original known coefficients. The results are summarized in the following table:

	Max Absolute Difference
Least-Squares	4.0523e-15
Gram-Schmidt	4.0523e-15
S2Kit	2.5077e-14

We repeat this with 1600 uniformly random samples, but without S2kit since it only handles lat-lon samples:

	Max Absolute Difference
Least-Squares	3.6637e-15
Gram-Schmidt	3.7894e-15

As expected, our algorithms work well with $(2b)^2$ samples. Note that in the second table, Least-Squares and Gram-Schmidt should have the same errors except that we generate the 1600 randomly uniform samples on the fly, and so the algorithms were running on different set of samples.

6.2 Undersampling on the Lat-Lon Grid

Next, we undersample f on the lat-lon grid with varying number of samples, and apply Least-Squares, Gram-Schmidt and S2kit transforms. This time, the transform coefficients are definitely different from the original coefficients, and so to measure interpolation errors between sample points, we reconstruct our function, $\tilde{f}$ from our transform coefficients using eq (4). We then compare the values of $\tilde{f}$ and f at 10000 uniformly random points. Denoting $\vec{\tilde{f}}$ and $\vec{f}$ as the sample vectors, we define the normalized difference between $\vec{\tilde{f}}$ and $\vec{f}$ to be $||\vec{\tilde{f}} - \vec{f}||/||\vec{f}||$, where the norm $||\cdot||$ is taken to be the usual inner product norm, $\sqrt{\langle \cdot, \cdot \rangle}$. We can interpret the normalized difference to be the test error because it measures how well the coefficients we found generalize to unseen sample points. We summarize the results in figure 2.

From the figure, notice that the normalized difference (test error) of the Gram-Schmidt is in general lower than Least-Squares and S2kit. We can also conclude from the figure that Gram-Schmidt suffers from overfitting, since test error is a lot higher than training error (almost 0). Indeed, in the case of 100 samples (Gram-Schmidt suffers from the worst test error), setting the threshold for $|\triangle \vec{f}|$ to be 0.5 instead of 10^{-10} increases the training error to 0.3393, while decreasing the test error to 1.2284 (not shown in figure).

An important observation is that although Gram-Schmidt manages to beat Least-Squares and S2kit, in general, the results are still considerably weaker than the almost-zero errors obtained in the previous section, when there's sufficient sampling. We shall see in the next section that on the uniformly random grid, we can achieve low errors despite undersampling.

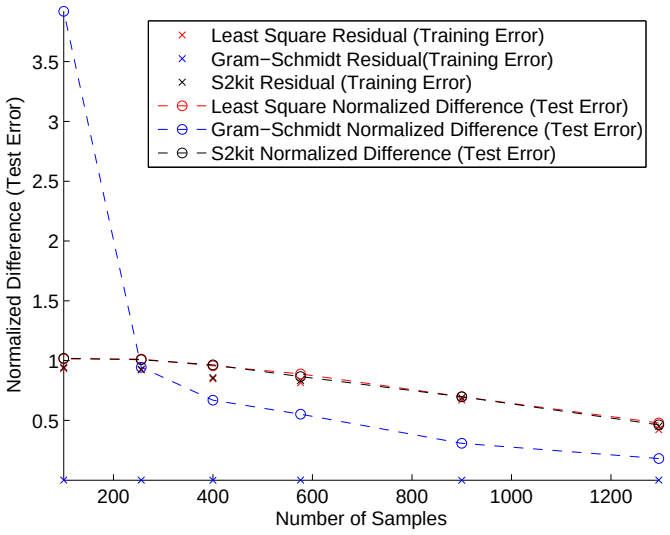

Figure 2: Performing Spherical Transform on the Undersampled Lat-Lon Grid: Training (Normalized Residual) Error and Test (Normalized Difference) Error of Least-Squares, Gram-Schmidt and S2kit.

6.3 Undersampling on the Uniformly Random Grid

In this section, we undersample f using the uniformly random grid with varying number of samples. Once again we use normalized difference to measure our generalization error. The results are shown in figure 3.

From the figure, we find that as long as we have at least 400 samples, both Least-Squares and Gram-Schmidt yield almost 0 error. Interestingly, once we have less than 400 samples, the errors simply blow up. Indeed, when we compare the experimental errors between 399 and 400 samples, we notice a big jump in error. This is summarized in the table below:

Number of Samples	400	399
Least-Squares $\|\|\vec{\tilde{f}} - \vec{f}\|\|/\|\|\vec{f}\|\|$	2.2057e-08	8.8130
Gram-Schmidt $\|\|\vec{\tilde{f}} - \vec{f}\|\|/\|\|\vec{f}\|\|$	1.7346e-11	159.5883

To ensure that our results are not dependent on the error measure, we try a different error measure. We define the maximum normalized absolute difference between $\vec{\tilde{f}}$ and $\vec{f}$ to be $max(\frac{abs(\vec{\tilde{f}} - \vec{f})}{abs(\vec{f})})$. It basically measures the largest percentage error in the approximation. The following table summarizes the result:

Number of Samples	400	399
Least-Squares	1.2619e-04	1.62535e+04
Gram-Schmidt	1.1150e-07	8.6322e+05

We do not think that it is a coincidence that $400 = 20^2$, where 20 is the bandwidth of our sampled function, f. It might be that for a bandlimited spherical function of bandwidth b that is uniformly sampled, the "Nyquist" rate might be b^2, rather than $(2b)^2$ on the lat-lon grid. Also observe

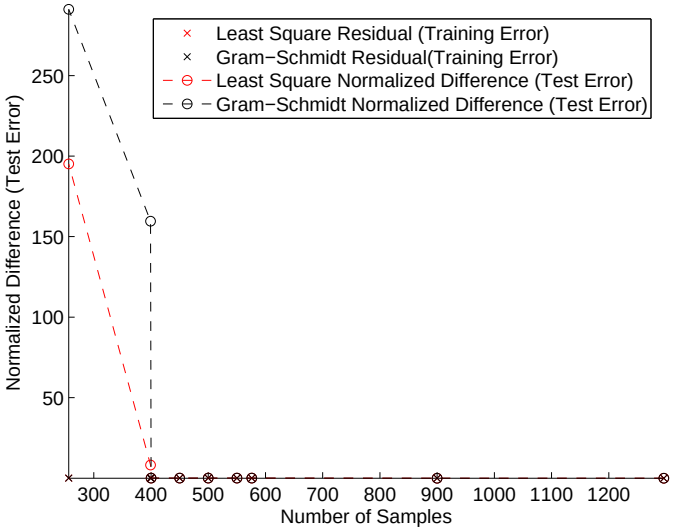

Figure 3: Performing Spherical Transform on Undersampled Uniformly Random Grid: Training (Normalized Residual) Error and Test (Normalized Difference) Error of Least-Squares and Gram-Schmidt.

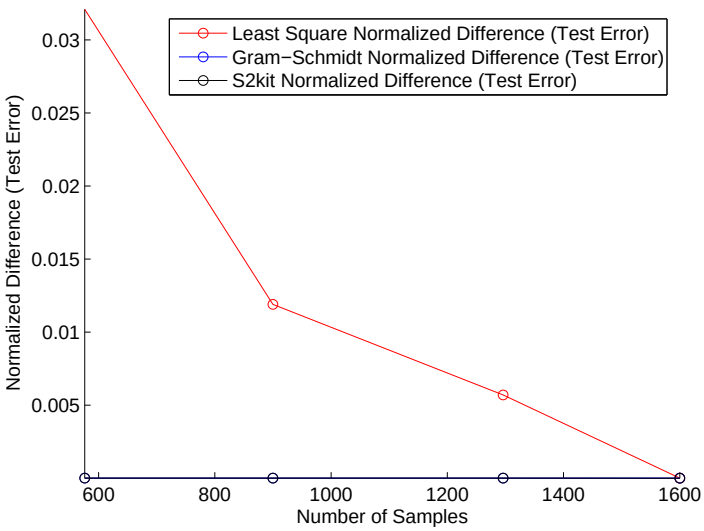

Figure 4: Performing Convolution on the Undersampled Lat-Lon Grid: Training (Normalized Residual) Error and Test (Normalized Difference) Error of Least-Squares and Gram-Schmidt. Note that the black line lies exactly on the blue line, so you can't see the blue line.

that 20^2 corresponds to the degree of freedom we have in determining our harmonic coefficients (where we consider a single complex coefficient to contain 2 degrees of freedom).

One might wonder why Least-Squares has similar performance to Gram-Schmidt on the randomly uniform grid, but not on the lat-lon grid. The problem is that for the lat-lon grid, $B^T B$ (see eq. 8) is often non-invertible even when there are fewer columns than there are samples.

6.4 Convolution: Undersampling on the Lat-Lon Grid

In this experiment, we convolve our random bandlimited function, f with our random filter, g of bandwidth 5 to obtain h, i.e. $f * g = h$. We assume that we have sufficient resources to represent the filter, so we will use its true harmonic coefficients. The approximation comes from estimating f's coefficients, using Least-Squares, Gram-Schmidt and S2kit with varying number of samples. We then use the convolution theorem to obtain $\tilde{h}$, our approximation of h. Once again, we measure our interpolation error by comparing h and $\tilde{h}$ at 10000 uniformly random points using normalized difference as our error measure. Note that we can calculate h exactly since we generated f and g and so can apply the convolution theorem directly.

We summarize our results in figure 4. Both S2kit and Gram-Schmidt have almost zero errors up till 576 ($(2(12))^2$) samples. While it is not shown in the figure, the accuracy deteriorates rapidly when we use < 576 samples. On the other hand, Least Squares perform badly the moment we undersample f (less than 1800 samples). However, notice that despite this, all three methods have significantly lower errors in figure 4 compared with figure 2. A possible explanation is this: because our filter is of bandwidth 5, it acts as a low pass filter and remove the higher harmonics of f by multiplying them with zero. This tends to reduce the errors because the higher frequencies are less reliable due to undersampling.

What is surprising however, is that while all three methods have approximately the same errors when estimating the spherical harmonics coefficients on the undersampled lat-lon grid (see figure 2), S2kit and Gram-Schmidt perform so much better than Least-Squares when performing convolution.

6.5 Convolution: Undersampling on the Uniformly Random Grid

We repeat the convolution experiment from the previous section, but this time using the uniformly random grid. From section 6.3, we expect our convolution to be accurate when we have at least 400 samples, since Least-Squares and Gram-Schmidt Spherical Transforms are accurate over that range. Figure 5 shows the result of our convolution experiment, and our predictions are indeed correct. However, unlike the lat-lon grid, we do not get any boost in accuracy due to the low-pass filter. Despite this, we note that the uniformly random grid still performs better, requiring only 400 samples compared with 576 samples on the lat-lon grid.

6.6 Final Toy Example

We will end our experimental section with a final toy example. Figure 6a shows a spherical image with a noisy protrusion (whose harmonic coefficients we know), and figure 6b shows a spherical filter with a smooth symmetric bump on

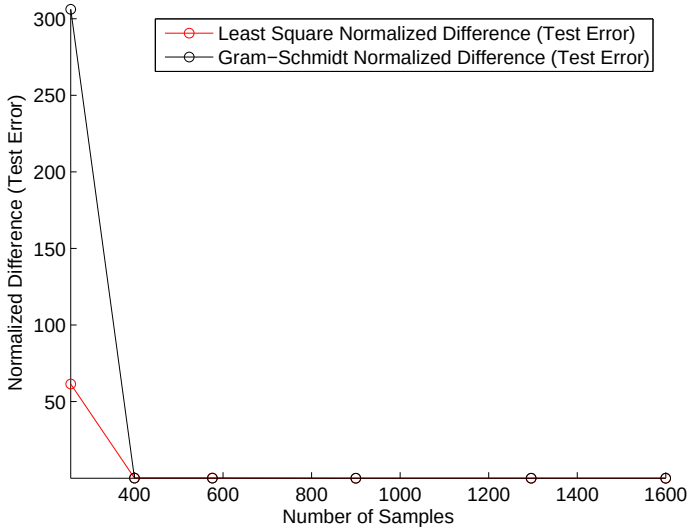

Figure 5: Performing Convolution on the Undersampled Uniformly Random Grid: Training (Normalized Residual) Error and Test (Normalized Difference) Error of Least-Squares and Gram-Schmidt.

the North Pole (whose harmonic coefficients we also know). Both the image and filter have bandwidths of 64. Convolving our image with the filter should ideally yield an output that has a maximum at the bump (see figure 6c). In our experiment, we sample our filter and image at 64^2 uniformly random points (instead of the required $(2(64))^2$ points on the lat-lon grid). The result is shown in figure 6d. The maximum normalized absolute difference between the ideal output and our result over 40000 uniformly random points is $1.9416e-08$.

7 Conclusions and Future Work

In this paper, we proposed and experimented with an extension to the Least-Squares method of projecting samples of an unknown spherical function onto the span of spherical harmonics. The use of Gram-Schmidt orthogonalization is motivated by the fact that while Least-Square aims to find a spherical harmonic decomposition that passes near the sample points, there's no guarantee that the columns of B (refer to eq. 7) will be independent even when there are more rows than columns in B (i.e. more samples than the degree of freedom we have in the harmonic coefficients). Hence B^TB could become non-invertible before we could attain a good fit. Indeed, as shown in section 6.2, the columns of B quickly became dependent when sampling on the lat-lon grid, resulting in poor training and test error. On the other hand, Gram-Schmidt was able to fit the samples completely, although there were obvious signs of overfitting.

We like to comment on the discarding of dependent spherical harmonics in the Gram-Schmidt orthogonalization process. Since these (sampled) spherical harmonics already lie in the span of previously sampled harmonics, adding them will not yield a better approximation to the sampled points. By throwing them away, we are in fact favoring the harmonics we process earlier, i.e. the lower-order harmonics. Hence, our algorithm has an implicit preference for smooth (lower-order) harmonics that explain our sample points.

We also demonstrated that we needed less samples for the randomly uniform grid compared with the lat-lon grid. Empirically, we found that a function of bandwidth b required b^2 samples on the uniformly random grid compared with $(2b)^2$ samples on the lat-lon grid.

Finally, our experiments on convolution (section 6.4 and 6.5) showed that if we filtered a function of bandwidth b with a filter of lower bandwidth, we might require less than $(2b)^2$ samples on the lat-lon grid in order to get accurate results. This was not true for the uniformly random grid, although overall, the uniformly random grid still performed better. Such a property is less surprising if one considers a 1-D signal. Remember that sampling a 1-D signal is equivalent to replicating the fourier transform of the signal periodically in the fourier domain. As sampling becomes more sparse, these replicas come closer and closer together. When the replicas overlap, we have aliasing. Since overlaps first occur in the higher frequencies, it means that aliasing effects tend to first show up in the higher frequency bands. Applying a low-pass filter will eliminate the problematic higher frequency bands.

The advantage of Least-Squares and Gram-Schmidt is that they handle the case of non lat-lon sampling. However the fast spherical transform is simply much faster ($O(nlog^2n)$), where n is the number of samples. In comparison, our algorithm is at least $O(n^3)$ (due to matrix inversion) assuming we use just enough samples to prevent aliasing. The less sampling we require on the uniformly random grid only comes up as an algorithmic constant. A possible alternative might be to perform a one-time decomposition of our image using Least-Squares or Gram-Schmidt. Then for any filter or cascade of filters we would like to apply to our image, we can perform fast spherical transform to obtain the filters' coefficients assuming we know the analytic expression of the filters.

There are a lot more work to be done:

1. We would like to test our algorithm on other grids and investigate their sampling threshold.

2. We would like to investigate a more theoretical basis of sampling limits on a non lat-lon grid, besides the intuitive reason that the lat-lon grid wastes too many samples near the poles. A possible source of inspiration could come from non-uniform sampling of 1-D signals. It is known that if non-uniform sampling is done correctly, the nyquist rate can be violated for 1-D signals.

3. While projecting our image onto the span of spherical harmonics provides a convenient way of performing convolution, we are still unable to handle non-linear filters. There are many useful (even simple) non-linear filters in image processing (such as the median filter) that cannot be implemented via convolution. For such

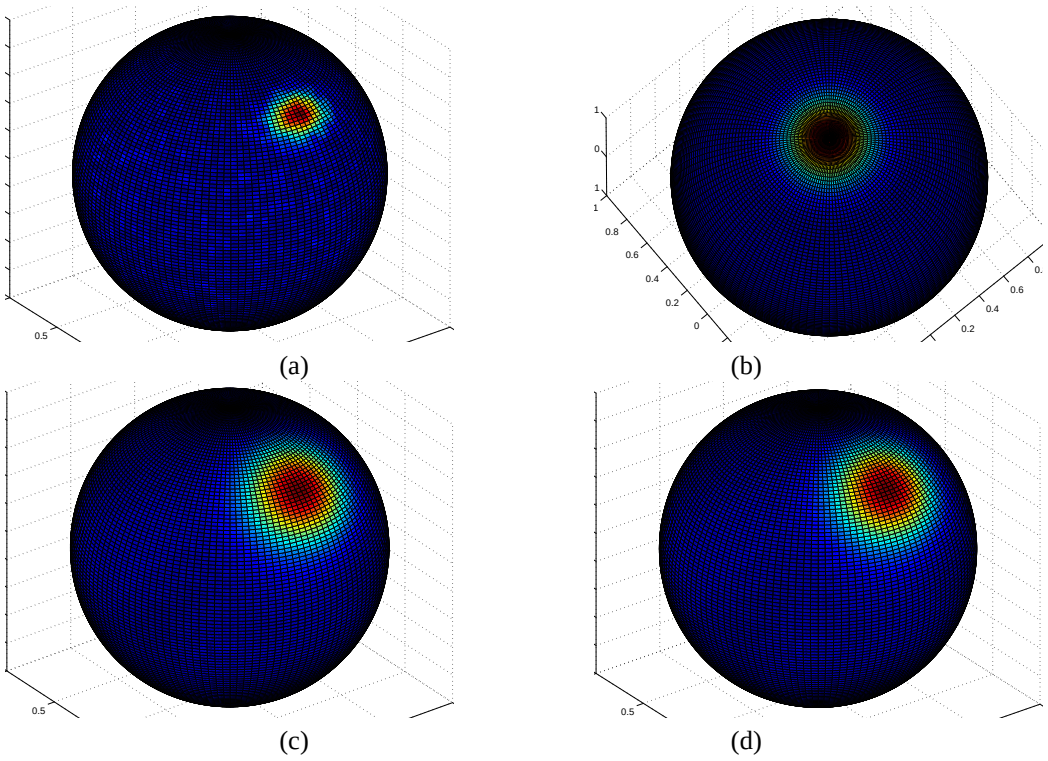

Figure 6: (a) Spherical function, Bandwidth = 64, with a noisy protrusion. (b) Spherical Filter, Bandwidth = 64, with smooth symmetric bump on the North Pole. (c) Ideal Output. (d) Output found by estimating harmonic coefficients using Gram-Schmidt over 64^2 samples over the uniformly random grid.

filters, we probably have to employ brute force methods, such as projecting our spherical function onto the plane and doing image processing there. Since the sphere and plane are geometrically and topologically different, we will need to warp our image and filter appropriately. We will also need to interpolate the pixels of the 2D-grid, introducing possibly interpolation errors. It will be interesting to compare the accuracy of the brute force method with our algorithm applied to linear filtering.

4. We have not considered the case when our samples are noisy. For example, we would not expect our segmented cortical surfaces to be free of noise. Hence we would probably not want to fit our data samples completely. This is related to the overfitting problem that Gram-Schmidt suffers from. However, for real data, we actually do not know the underlying function, so it will be hard to test for overfitting. Perhaps we can use techniques such as leave-one-out cross-validation, although that will slow down our algorithms even further.

References

[1] Martin Bohme. A fast algorithm for filtering and wavelet decomposition on the sphere. *Thesis (Diploma)*, June 2002.

[2] G. Gerig CH. Brechbuhler and O. Kubler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, pages Vol. 61, No.2, 154–170, Mar, 1995.

[3] D.N. Rockmore P.J. Kostelec D.M. Healy, Jr. and S. Moore. Ffts for 2-sphere-improvements and variations. *The Journal of Fourier Analysis and Applications*, pages Vol. 9, No. 4, 341–385, 2003.

[4] James R. Driscoll and JR Dennis M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, pages Vol. 15, 202–250, 1994.

[5] Francis X. Giraldo. Lagrange-galerkin methods on spherical geodesic grids. *Journal of Computational Physics*, pages 136: 197–213, 1997.

[6] Peter J. Kostelec and Daniel N. Rockmore. S2kit: A lite version of spharmonic kit. *http://www.cs.dartmouth.edu/ geelong/sphere/*.

[7] Peter Schroder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *In Computer Graphics Proceedings (SIGGRAPH)*, pages 161–172, 1995.

[8] Peter Schroder and Wim Sweldens. Spherical wavelets: Texture processing. *Rendering Techniques*, pages 252–263, Aug 1995.

[9] Lawrence H. Staib and James S. Duncan. Model-based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, pages Vol. 15, No. 5, 720:731, Oct 1996.

[10] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *Siam Journal of Mathematical Analysis*, pages Vol 29, No. 2, pp. 551–546, Mar 1998.

[11] L. Jacques Y. Wiaux and P. Vandergheynst. Corresponding principle between spherical and euclidean wavelets. *Submitted to Astrophysics*, 2005.