# Large-Scale Modeling of Parametric Surfaces using Spherical Harmonics

Li Shen
Dept. of Computer and Info. Science
University of Massachusetts Dartmouth
N. Dartmouth, MA 02747
lshen@umassd.edu

Moo K. Chung
Department of Statistics
University of Wisconsin Madison
Madison, WI 53706
mchung@stat.wisc.edu

## Abstract

*We present an approach for large-scale modeling of parametric surfaces using spherical harmonics (SHs). A standard least square fitting (LSF) method for SH expansion is not scalable and cannot accurately model large 3D surfaces. We propose an iterative residual fitting (IRF) algorithm, and demonstrate its effectiveness and scalability in creating accurate SH models for large 3D surfaces. These large-scale and accurate parametric models can be used in many applications in computer vision, graphics, and biomedical imaging. As a simple extension of LSF, IRF is very easy to implement and requires few machine resources.*

## 1. Introduction

Parametric surface modeling is an important research topic in 3D data processing, where a pre-defined mathematic model is used to describe a surface. Typical techniques include spherical harmonics (SHs) [2, 4], hyperquadrics [14], and superquadrics [23]. Recently, the SH method has received a lot attention, and has been studied and applied to several areas including computer vision [2, 4], computer graphics [5, 10, 24], medical image analysis [11, 12, 16, 20, 21], and computational biology [19].

Spherical harmonics were first used as a type of parametric surface representation for radial or stellar surfaces $r(\theta, \phi)$ by Schudy and Ballard [2, 20], where the harmonics were used as basis functions to expand $r(\theta, \phi)$. Recently, an extended method, called SPHARM, was proposed by Brechbühler, Gerig and Kubler [4] to model more general shapes, where three functions of $\theta$ and $\phi$ were used to represent a surface. SPHARM can deal with protrusions and intrusions and can model arbitrarily shaped but simply connected 3D objects.

Although SPHARM has been successfully applied to many applications in biomedical imaging [11, 12, 16, 21], it is still difficult to use it to accurately model large and complicated surfaces such as brain cortex. In these studies, a least square fitting (LSF) method [4] is usually used for spherical harmonic expansion. In typical cases, a 3D model with a few thousand vertices is expanded using spherical harmonics up to degree 15-25 and the SPHARM reconstruction can only roughly approximate the original model.

To the best of our knowledge, although SPHARM has been available for a decade, no attempts have been made to build complicated SPHARM models using a large-scale LSF method. One possible reason is that most methods for solving large linear systems [3, 9] are either designed for sparse or symmetric matrices or not easy to implement. In this paper, we propose an iterative residual fitting (IRF) algorithm, which can perform large-scale spherical harmonic expansion (*e.g.*, up to degree $100$) and generate accurate models for complicated 3D surfaces (*e.g.*, with $> 40,000$ vertices) efficiently. As a simple extension of LSF, IRF is very easy to implement and requires few machine resources. As a result, large-scale SPHARM modeling can be easily done on standard workstations with average configuration using a standard linear solver. We also believe that much larger SPHARM models can be created if one combines the IRF method with an enhanced large-scale linear solver.

Besides LSF, spherical harmonic expansions can also be evaluated using numeric integrations [15, 19]. One of the best approaches in this category was proposed by Healy *et al.* [15]. Healy's algorithm was used by several studies [5, 13, 24] to process 3D models. Bulow [5] used spherical harmonics to describe star-shaped surfaces and developed a surface smoothing method based on linear diffusion. Zhou *et al.* [24] developed a 3D surface filtering method using spherical harmonics that worked for arbitrary surfaces. Gu *et al.* [13] used the conformal mapping for spherical harmonic transformation for the purpose of brain surface compression and rotation-invariant shape analysis. The spherical harmonic expansions used in these studies were typically up to degree 30 or 60, and the corresponding reconstructions did not seem to capture all the surface details of the original models. We expect that adding more higher degree harmonics into the models can improve the representation accuracy. Thus, it is an interesting future topic to compare Healy's method with our IRF algorithm concerning both modeling accuracy and running time.

## 2. Preliminary

### 2.1. Surface Parameterization

We use the following convention for spherical coordinates $(\theta, \phi)$ to match the definition of spherical harmonics: $\theta$ is taken as the polar (colatitudinal) coordinate with $\theta \in [0, \pi]$, and $\phi$ as the azimuthal (longitudinal) coordinate with $\phi \in [0, 2\pi)$. To create a SPHARM model for a 3D closed surface, we first need to perform surface parameterization [8] that establishes a bijective mapping between each vertex $\mathbf{v} = (x, y, z)^T$ on a surface and a pair of spherical coordinates $(\theta, \phi)$. We use $\mathbf{v}(\theta, \phi)$ to denote such a mapping, meaning that, $\mathbf{v}$ is parameterized with $(\theta, \phi)$. Taking into consideration the $x$, $y$, and $z$ coordinates of $\mathbf{v}$ in object space, the mapping can be represented as: $\mathbf{v}(\theta, \phi) = (x(\theta, \phi), y(\theta, \phi), z(\theta, \phi))^T$.

A satisfactory mapping often requires a minimization of some types of distortions such as length distortion, angle distortion, or area distortion. Here we list a few interesting spherical mapping methods. (1) Brechbühler *et al.* [4] developed an equal area mapping technique for voxel surfaces by solving a constrained optimization problem. (2) We [22] developed a spherical mapping technique for triangle surfaces to control both area and length distortion. (3) We [6] created a spherical mapping by deforming an ellipsoid mesh to cortical surfaces as well as to the unit sphere. (4) Gu *et al.* [13] developed a conformal mapping method to minimize angle distortions. (5) Praun and Hoppe [18] developed a spherical mapping method that minimizes distortion in vector length. In our experiments, we used five 3D models, and their spherical parameterizations were created using Method 1 (Hipp), Method 2 (Bowl, Part, and Head), and Method 3 (Cort). Figure 1 shows the Bowl model and its spherical parameterization.

### 2.2. Spherical Harmonic Expansion

Spherical harmonics are a natural and convenient choice of basis functions for representing any twice-differentiable spherical function [1, 2]. They are an infinite set of complex functions that are continuous, orthonormal, single-valued, and complete on the sphere. Spherical harmonics $Y_l^m(\theta, \phi)$ of degree $l$ and order $m$ are defined as follows:

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

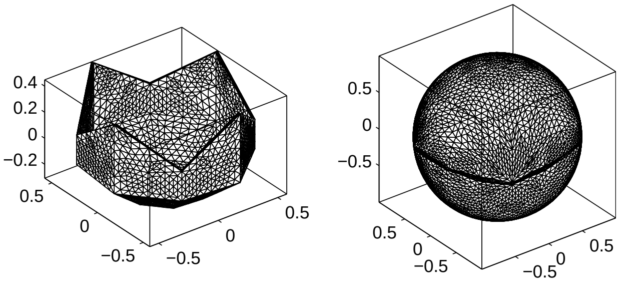


**Figure 1. Sample spherical parameterization.**

where $l$ and $m$ are integers with $|m| \leq l$, and the associated Legendre polynomial $P_l^m$ is defined by the differential equation

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

Any twice-differentiable spherical function $f(\theta, \phi)$ can be represented by a linear combination of spherical harmonics $Y_l^m(\theta, \phi)$ as follows:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta, \phi), \tag{3}$$

where the coefficients $a_l^m$ are uniquely determined by [19]

$$a_l^m = \int_0^{\pi} \int_0^{2\pi} Y_l^m(\theta, \phi)^* \, f(\theta, \phi) \, sin\theta \, d\phi \, d\theta. \tag{4}$$

Here $Y_l^m(\theta, \phi)^*$ is the complex conjugate of $Y_l^m(\theta, \phi)$.

The spherical harmonic expansion described above is essentially the Fourier transform for functions defined on the sphere; and it transfers spherical scalar signals into its frequency spectrum. Spherical harmonics have several favorable properties such as orthonormality, completeness, and coarse-to-fine hierarchy, which make them a nature choice of basis functions to represent radial surfaces $r(\theta, \phi)$ [2, 19]. Recently, Brechbühler *et al.* [4] have extended the spherical harmonics expansion technique to more general shapes by representing a surface using three functions of $\theta$ and $\phi$. This technique has been referred to as the SPHARM surface modeling in previous studies [11, 12]; and can be applied to arbitrarily shaped but simply-connected objects.

The SPHARM expansion requires a spherical parameterization performed in advance, as described in Section 2.1. The parameterization has the following form of $\mathbf{v}(\theta, \phi) = (x(\theta, \phi), y(\theta, \phi), z(\theta, \phi))^T$, where $x(\theta, \phi)$, $y(\theta, \phi)$, and $z(\theta, \phi)$ are three functions defined on the sphere. Thus, in order to describe the object surface, we just need to expand these three spherical functions using spherical harmonics:

$$\mathbf{v}(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\mathbf{c}_l^m Y_l^m(\theta,\phi), \tag{5}$$

where $\mathbf{c}_l^m=\left(c_{lx}^m, c_{ly}^m, c_{lz}^m\right)^T$.

The task of the SPHARM expansion is to compute the coefficients $\mathbf{c}_l^m$ up to a user-desired degree, which we will discuss in detail in Section 3. Note that a degree $L_{max}$ expansion involves $3\times(L_{max}+1)^2$ coefficients. The object surface can be reconstructed using these coefficients. Since spherical harmonics form a complete set of orthonormal basis functions with a coarse-to-fine hierarchy, using more coefficients leads to a more accurate reconstruction.

## 3. Methods

In this section, we present a new method for spherical harmonic expansion. We focus our discussion on expanding a spherical scalar signal $f(\theta,\phi)$. For the SPHARM case, we can apply the same method three times and expand $x(\theta,\phi)$, $y(\theta,\phi)$, and $z(\theta,\phi)$ separately. Given a function $f(\theta,\phi)$ and a user-specified maximum degree $L_{max}$, our task is to extract coefficients $a_l^m$ in Eq. (3) for $l\leq L_{max}$ and $|m|\leq l$.

There are two types of approaches for computing $a_l^m$: one uses numerical integration according to Eq. (4) [15, 19]; the other formulates a linear system and solves it using least square fitting (LSF) [4]. The LSF approach is easy to implement if some linear solver is available (*e.g.*, we are using one provided by Matlab). Our method belongs to this category and overcomes its limitation of being unscalable. An interesting future topic is to compare our method with Healy's algorithm [15], which incorporates a divide-and-conquer idea and becomes one of the best methods in the numerical integration category. Here we first briefly describe the LSF approach and then present our method.

### 3.1. Least Square Fitting

The input data for spherical harmonic expansion contain a spherical function $f(\theta,\phi)$ and a user-specified maximum degree $L_{max}$. The spherical function is described by a set of spherical samples $(\theta_i,\phi_i)$ and their function values $f_i=f(\theta_i,\phi_i)$, for $1\leq i\leq n$. According to Eq. (3), we can formulate a linear system as follows.

$$\begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,k}\\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,k}\\ \vdots & \vdots & \vdots & & \vdots\\ y_{n,1} & y_{n,2} & y_{n,3} & \cdots & y_{n,k}\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\ \vdots\\ x_k\end{pmatrix}=\begin{pmatrix}f_1\\f_2\\ \vdots\\ f_n\end{pmatrix}$$

where $y_{i,j}=Y_l^m(\theta_i,\phi_i)$, $j=l^2+l+m+1$, and $k=(L_{max}+1)^2$. Note that we use an indexing scheme that assigns a unique index $j=l^2+l+m+1$ to every pair $(l,m)$. Least square fitting is used to solve the above system for $(x_1,x_2,...,x_k)^T$, since $n\neq k$ in almost all the cases. Because each $x_j\equiv\hat{a}_l^m$ is an estimate of the original coefficient $a_l^m$ for $j=l^2+l+m+1$, we can reconstruct the original function as follows:

$$\hat{f}(\theta,\phi)=\sum_{l=0}^{L_{max}}\sum_{m=-l}^{l}\hat{a}_l^m Y_l^m(\theta,\phi)\approx f(\theta,\phi). \tag{6}$$

The more degrees one uses, the more accurate the reconstruction $\hat{f}(\theta,\phi)$ is. See Figure 5 for some examples.

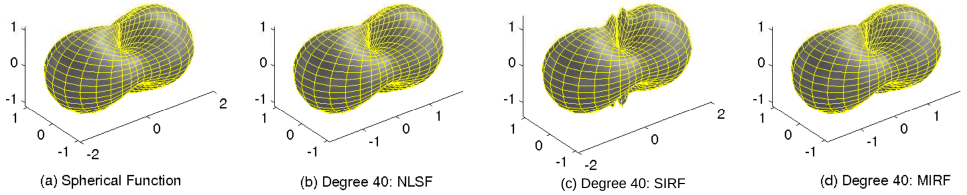
We refer to this method as **naive least square fitting (NLSF)**. There are many linear solvers available and we use Matlab in this study. Thus, NLSF is very easy to implement. However, it is not suitable for large values of $L_{max}$ and $n$, since the above linear system cannot even be loaded into the memory. To model large 3D surfaces, we need to increase $n$ to get enough surface sampling resolution and capture all the surface details. At the same time, we also need to pick a big value of $L_{max}$ to capture geometry information in the high end of the frequency spectrum. These observations motivate us to develop a modified NLSF approach that can not only handle large-scale spherical harmonic expansions but also retain the property of being easy to implement.

### 3.2. Simple Iterative Residual Fitting

The basic idea behind our method is simple and follows the properties of spherical harmonic (SH) transform. First, SHs form a coarse-to-fine hierarchy. If we just use a few low degree SHs to expand a spherical function $f(\theta,\phi)$, we get a low-pass filtered reconstruction. If we use more degrees, more details are included in the reconstruction. Our approach takes advantage of this coarse-to-fine hierarchy. We start from a low degree reconstruction and then iteratively adding more details into our model by involving higher degree SHs. Second, SHs form an orthonormal basis and geometric information is stored in different frequency channels. Thus, if we first extract information from low frequency channels, the residual (*i.e.*, $f(\theta,\phi)-$ its reconstruction) will exactly contain information in high frequency channels. To add in more details to our model, we can simply use a few higher degree harmonics to fit the residual.

Let us first introduce some notation:

$$\begin{aligned} A_l &= \begin{pmatrix} Y_l^{-l}(\theta_1,\phi_1) & Y_l^{-l+1}(\theta_1,\phi_1) & \cdots & Y_l^{l}(\theta_1,\phi_1)\\ Y_l^{-l}(\theta_2,\phi_2) & Y_l^{-l+1}(\theta_2,\phi_2) & \cdots & Y_l^{l}(\theta_2,\phi_2)\\ \vdots & \vdots & & \vdots\\ Y_l^{-l}(\theta_n,\phi_n) & Y_l^{-l+1}(\theta_n,\phi_n) & \cdots & Y_l^{l}(\theta_n,\phi_n)\end{pmatrix},\\ \mathbf{b}_l &= (\hat{a}_l^{-l},\hat{a}_l^{-l+1},...,\hat{a}_l^{l})^T,\\ \mathbf{f} &= (f(\theta_1,\phi_1),f(\theta_2,\phi_2),...,f(\theta_n,\phi_n))^T,\\ \mathbf{m} &= (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_{L_{max}}^T)^T,\end{aligned}$$



(a) Spherical Function

(b) Degree 40: NLSF

(c) Degree 40: SIRF

(d) Degree 40: MIRF

**Figure 2. (a) Spherical function: radius is plotted as a function of angle. (b-c) Degree 40 reconstructions generated by NLSF (b), by SIRF (c), and by a four pass MIRF (d).**

where $A_l$ is a matrix of degree $l$ spherical harmonic values for $n$ sampling points, $\mathbf{b}_l$ contains the estimated degree $l$ coefficients, $\mathbf{f}$ is a vector of spherical function values at $n$ sampling points, and $\mathbf{m}$ is a spherical harmonic model containing coefficients up to degree $L_{max}$.

Now we can formally present our algorithm as follows.

1. Solve the linear system: $(A_0\ A_1\ ...\ A_s)\ (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_s^T)^T = \mathbf{f}$.
2. Calculate the residual: $\mathbf{r} = \mathbf{f} - (A_0\ A_1\ ...\ A_s)\ (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_s^T)^T$.
3. Iteratively fit the residual:

```
for ($l = s+1$; $l \leq L_{max}$; $l++$) do
    solve for $A_l \mathbf{b}_l = \mathbf{r}$
    update residual $\mathbf{r} = \mathbf{r} - A_l \mathbf{b}_l$
```

4. Return the spherical harmonic model $\mathbf{m}$: $\mathbf{m} \equiv (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_{L_{max}}^T)^T$.

We refer to this method as **simple iterative residual fitting (SIRF)**. In SIRF, we first create an SH model using coefficients up to degree $s$. In each iteration, we estimate coefficients for one more degree by fitting relevant SHs to the residual. SIRF stops when $L_{max}$ is reached. In NLSF, we need to load $(A_0\ A_1\ ...\ A_{L_{max}})$ into memory, which may cause "out of memory" for large $L_{max}$ and $n$. In SIRF, we only need to load $A_i$'s one at a time. This solves the memory overflow problem and makes the method scalable. To further improve the performance, all the $A_i$'s can be precomputed and stored on disk for a fixed sampling scheme.

### 3.3. Multipass Iterative Residual Fitting

Although SIRF runs much faster than NLSF for large $L_{max}$ and $n$, it sometimes creates a less accurate reconstruction. Figure 2 shows such an example: the NLSF reconstruction (b) matches the original function (a) better than the SIRF reconstruction (c) does. To improve the reconstruction accuracy, we propose a **multipass iterative residual fitting (MIRF)** approach as follows.

1. Create an SH model $\mathbf{m}$ for $\mathbf{f}$ using SIRF.
2. Compute the residual: $\mathbf{r} = \mathbf{f} - (A_0\ A_1\ ...\ A_{L_{max}})\ \mathbf{m}$.
3. Fit the residual iteratively in $p$ passes and update $\mathbf{m}$:

```
for ($i = 1$; $i \leq p$; $i++$) do
    create an SH model $\mathbf{m}_r$ for $\mathbf{r}$ using SIRF
    update residual $\mathbf{r} = \mathbf{r} - (A_0\ A_1\ ...\ A_{L_{max}})\ \mathbf{m_r}$
    update model $\mathbf{m} = \mathbf{m} + \mathbf{m_r}$
```

4. Return the spherical harmonic model $\mathbf{m}$

Since a single pass of SIRF is not enough to model the function accurately, MIRF employs multiple passes to fit the residual iteratively and aims to improve the modeling accuracy. The effectiveness of this approach is shown in Figure 2, where the MIRF reconstruction (d) using four passes matches the original function (a) very well.

### 3.4. Generalized Iterative Residual Fitting

**Algorithm 1** Iterative Residual Fitting (IRF).

**Input:** A spherical function $f$ with $n$ sampled values $\{\theta_i, \phi_i, f(\theta_i, \phi_i) \mid 1 \leq i \leq n\}$, the maximal SH expansion degree $L_{max}$, the number of passes $p$, and a granularity parameter $g$.

**Output:** An SH model $\mathbf{m}$ of $f$, where $\mathbf{m} = (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_{L_{max}}^T)^T$ and $\mathbf{b}_l = (\hat{a}_l^{-l}, \hat{a}_l^{-l+1}, ..., \hat{a}_l^{l})^T$. Note that the model $\mathbf{m}$ minimizes $\sum_{i=1}^{n}(f(\theta_i, \phi_i) - \sum_{l=0}^{L_{max}} \sum_{m=-l}^{l} \hat{a}_l^m Y_l^m(\theta_i, \phi_i))^2$.

```
$\mathbf{r} = (f(\theta_1,\phi_1), f(\theta_2,\phi_2), ..., f(\theta_n,\phi_n))^T$
$\mathbf{m} = (0, 0, ..., 0)^T$ {Initialization, $|\mathbf{m}| = (1+L_{max})^2$}
for $i = 1$ to $p$ do {Perform $p$ passes}
    $d = 0$
    while $d \leq L_{max}$ do
        $d' = \arg\max_{d' \in [d, L_{max}]} |(\mathbf{b}_d^T\ \mathbf{b}_{d+1}^T\ ...\ \mathbf{b}_{d'}^T)| \leq g$
        {Involve as many basis functions as possible until the number exceeds $g$}
        solve the following linear system
        $(A_d\ A_{d+1}\ ...\ A_{d'})\ (\mathbf{b}_d^T\ \mathbf{b}_{d+1}^T\ ...\ \mathbf{b}_{d'}^T)^T = \mathbf{r}$
        $\mathbf{r} = \mathbf{r} - (A_d\ A_{d+1}\ ...\ A_{d'})\ (\mathbf{b}_d^T\ \mathbf{b}_{d+1}^T\ ...\ \mathbf{b}_{d'}^T)^T$
        {Update the residual $\mathbf{r}$}
        $d = d' + 1$ {Let $d$ be the next unprocessed degree}
    $\mathbf{m} = \mathbf{m} + (\mathbf{b}_0^T\ \mathbf{b}_1^T\ ...\ \mathbf{b}_{L_{max}}^T)^T$ {Model update}
return the SH model $\mathbf{m}$
```

Besides multiple passes, we observe that the modeling accuracy can also be improved by involving as many basis functions as possible at each iteration of least square residual fitting. If we can load in all the basis functions at the beginning, this approach becomes NLSF that only requires one iteration. Figure 2(b) shows that NLSF derives an accurate reconstruction. For large-scale modeling problems, we cannot load in all the basis functions. But in each iteration, we can involve more basis functions than SIRF does, reduce the iteration number, and improve the modeling accuracy.

With this observation, we introduce a granularity parameter $g$, which can be specified by a user to control the number of basis functions being involved in each iteration. Our approach aims to involve as many basis functions as possible in an iteration until the number exceeds $g$. Algorithm 1 describes a generalized iterative residual fitting method that incorporates the above idea.

We refer to Algorithm 1 as **iterative residual fitting (IRF)** directly. This method is generalized because NLSF, SIRF and MIRF are just special cases of IRF. If we set $p = 1$ and $g = \infty$, IRF performs only one pass and in this pass it can involve all the basis functions due to $g = \infty$. Thus, IRF becomes the same as NLSF. If we let $s$ be 0 in the first step of SIRF, IRF becomes SIRF when $p = 1$ and $g = 1$, and becomes MIRF when $p \geq 1$ and $g = 1$.

## 4. Experimental Results

We have performed extensive experiments on several 3D models by varying the values of $p$ and $g$ in Algorithm 1. The basic information about these 3D models can be found in Table 1 and Figure 4. Since these 3D surfaces are not star-shaped, in our experiments, we create a SPHARM model for each of them. Thus, we need to perform three SH expansions for each surface using its spherical parameterization. The spherical parameterization comes with each model and see Section 2.1 for how it is created. Our algorithm is implemented in Matlab. The experiments are performed on a DELL workstation PWS670 with a 3 GHz Xeon CPU and 2 GB of RAM, running WinXP and Matlab 7.

### 4.1. Computing Basis Functions

In order to use least square fitting, we need to compute SH basis function values of sampled spherical coordinates $(\theta_i, \phi_i)$, where $1 \leq i \leq n$. These values form matrix $A = (A_0\ A_1\ ...\ A_{L_{max}})$, where $A_l$ is defined in Section 3. $A$ has a dimension of $n \times (L_{max} + 1)^2$ and so cannot be completely loaded into memory for large $n$ and $L_{max}$. Our approach is to precompute $A$ and store it on disk. At the actual modeling stage, we only load the relevant part of $A$ as needed. Figure 3 shows the running time for creating $A$, which is clearly quadratic with respect to $L_{max}$ and linear to $n$. Table 1 Column 4 shows the space required for storing $A$ for $L_{max} = 100$. For a given model, this preprocessing step is a one-time operation. The same $A$ can be reused for different parameter settings at the modeling stages. The same $A$ can even be reused for different models if they share the same sampling scheme in the parameter space.

### 4.2. Modeling Accuracy

Let $\mathbf{v}(\theta, \phi)$ be an original 3D surface and $\hat{\mathbf{v}}(\theta, \phi)$ be its SPHARM reconstruction:

$$\hat{\mathbf{v}}(\theta, \phi) = \sum_{l=0}^{L_{max}} \sum_{m=-l}^{l} \hat{\mathbf{c}}_l^m Y_l^m(\theta, \phi),$$

where $\hat{\mathbf{c}}_l^m = \left(\hat{c}_{lx}^m, \hat{c}_{ly}^m, \hat{c}_{lz}^m\right)^T$ are SPHARM coefficients calculated by one of our algorithms. In order to measure the SPHARM modeling accuracy, we calculate the mean squared distance ($msd$) and the maximal squared distance ($xsd$) between the $n$ mesh vertices on the original surface and their reconstructions as follows:

$$\begin{aligned} msd &= \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{v}(\theta_i, \phi_i) - \hat{\mathbf{v}}(\theta_i, \phi_i)||^2, \\ xsd &= \max\{\, ||\mathbf{v}(\theta_i, \phi_i) - \hat{\mathbf{v}}(\theta_i, \phi_i)||^2 \ \mid 1 \leq i \leq n\}. \end{aligned}$$

To make the accuracy comparison consistent across different 3D surfaces, *we have done isotropic scaling for each of these surfaces so that the volume of its bounding box is 1.*

We first test NLSF on five 3D models. The fourth column of Table 1 shows the maximal $L_{max}$ our machine can handle without running out of memory. Figure 4 shows sample visualization between the original surfaces and their reconstructions, and the corresponding $msd$ and $xsd$ values. For small models like Bowl and Part, NLSF can create almost perfect reconstructions. This indicates that SPHARM is an excellent surface modeling technique when a sufficient number of basis functions can be involved. But for large models like Hipp, Head and Cort, NLSF can only involve a limited number of basis functions due to the memory limitation and result in only suboptimal reconstructions.

Our second experiment is to test if SIRF can help improve the modeling accuracy for large models, since SIRF is able to involve more basis functions, *e.g.*, up to degree 100 in our experiment. Figure 5 shows the visualization results together with $msd$ and $xsd$ for $L_{max} = 10, 20, 60, 100$ for Hipp and Cort. Both degree 100 reconstructions look satisfactory. However, Figure 6(a) shows the degree 100 reconstruction of Head using SIRF and this is obviously not an accurate reconstruction.

In order to create a more accurate SPHARM model for Head, we employ the generalized IRF algorithm and vary

| Model | Vertex # | Face # | $S_{bf}$ | $D_{max}^{NLSF}$ | Sample satisfactory result generated by IRF | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | $L_{max}$ | $(p, g)$ | Max squared distance ($xsd$) | Mean squared distance ($msd$) | Running time (sec.) |
| Bowl | 3458 | 6912 | 539 MB | 78 | 60 | $(1, \infty)$ | 1.80e-025 | 1.41e-026 | 119 |
| Part | 4422 | 8840 | 688 MB | 71 | 70 | $(1, \infty)$ | 1.29e-017 | 6.28e-019 | 265 |
| Hipp | 10082 | 20160 | 1.53 GB | 40 | 100 | $(2, 1)$ | 1.63e-003 | 2.33e-005 | 210 |
| Head | 27618 | 55232 | 4.19 GB | 29 | 100 | $(1, 400)$ | 2.46e-004 | 1.16e-005 | 568 |
| Cort | 40962 | 81920 | 6.22 GB | 24 | 100 | $(1, 100)$ | 6.54e-004 | 4.57e-005 | 555 |

**Table 1. 3D model information and sample results generated by IRF. $S_{bf}$ indicates the space required to store SPHARM basis functions up to degree 100. $D_{max}^{NLSF}$ indicates the maximal $L_{max}$ NLSF can handle without running out of memory.**

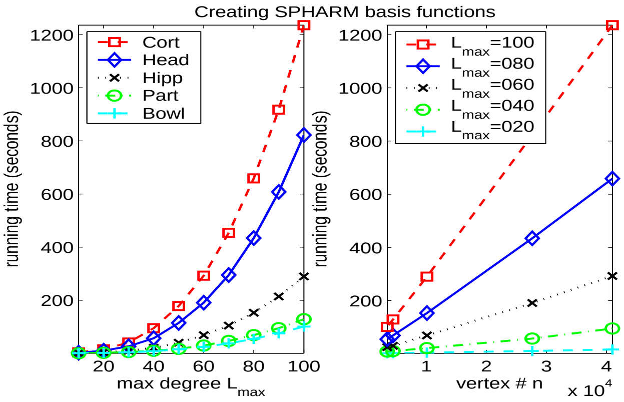


**Figure 3. Time for computing basis functions.**

the values of $p$ (number of passes) and $g$ (granularity parameter). Figure 6 shows sample results. In this case, a satisfactory model can be created when either $p \geq 4$ or $g \geq 400$. Figure 7 contains more accuracy results for Hipp, Head and Cort, where $-\log_{10} msd$ is plotted on the y-axis. Clearly, increasing $p$, $g$ or both can improve the modeling accuracy.

### 4.3. Computational Cost

Let $n$ be the number of mesh vertices on a 3D surface, $L_{max}$ the maximal degree of a SPHARM expansion, $g$ the granularity parameter, $p$ the number of passes. Solving an overdetermined set of $m_1$ equations with $m_2$ unknowns can be done in time $O(m_1\,m_2^2)$ [7]. Thus, if $g \leq L_{max}$, IRF needs to solve $O(L_{max}\,p)$ linear systems and each requires time $O(n\,L_{max}^2)$. If $g > L_{max}$, IRF needs to solve $O(\frac{L_{max}^2}{g}\,p)$ linear systems and each requires time $O(n\,g^2)$. The total running time is $O(n\,L_{max}^2\,L_{max}\,p) = O(n\,L_{max}^3\,p)$ for $g \leq L_{max}$, and $O(n\,g^2\,\frac{L_{max}^2}{g}\,p) = O(n\,L_{max}^2\,g\,p)$ for $g > L_{max}$. These two cases can be combined and the time complexity of our algorithm becomes $O(n\,L_{max}^2\ \max\{L_{max}, g\}\,p)$.

Figure 7 shows the running time and modeling accuracy (measured in $msd$) for various experiments on Hipp, Head, and Cort, where $n \in \{10082, 27618, 40962\}$, $L_{max} = 100$, $g \in \{1, 100, 200, 400\}$, and $1 \leq p \leq 8$. The experimental running time matches our analysis. In practice, we can control the values of $p$ and $g$ to be relatively small and still obtain accurate results. For example, Table 1 shows sample satisfactory result for each surface used in our experiment and all these results can be generated within 10 minutes even for large models like Head and Cort.

## 5. Discussions and Conclusions

The proposed IRF algorithm makes the spherical harmonic method applicable to large surfaces and thus can derive many new applications in related areas. For example, in medical image analysis, it can create accurate parametric models for cortical surfaces and facilitate statistical surface analysis to localize shape changes in certain diseases [6]. In computer graphics, using IRF, large-scale expansion can be performed to generate accurate models; and these parametric models can then be used to help many operations in digital geometry processing such as 3D surface filtering, compression, morphing, remeshing, and shape matching.

Here we list a few examples. *Remeshing*: Using a uniform parameter mesh, a SPHARM reconstruction can form a uniformly remeshed geometry for the original object. *Morphing*: If different objects are reconstructed using the same mesh topology, we can morph one to another. *Multiresolution modeling*: The coarse-to-fine hierarchy of a SPHARM model makes it suitable for multiresolution modeling and geometric compression. *Other processing*: Using SPHARM, many tasks can be accomplished in the frequency domain more efficiently, such as shape matching, surface denoising, shape analysis [17] and surface filtering [24]. Many graphical models do not have genus-zero surfaces. But using a method described in [24], SPHARM can be extended to process arbitrary manifold meshes .

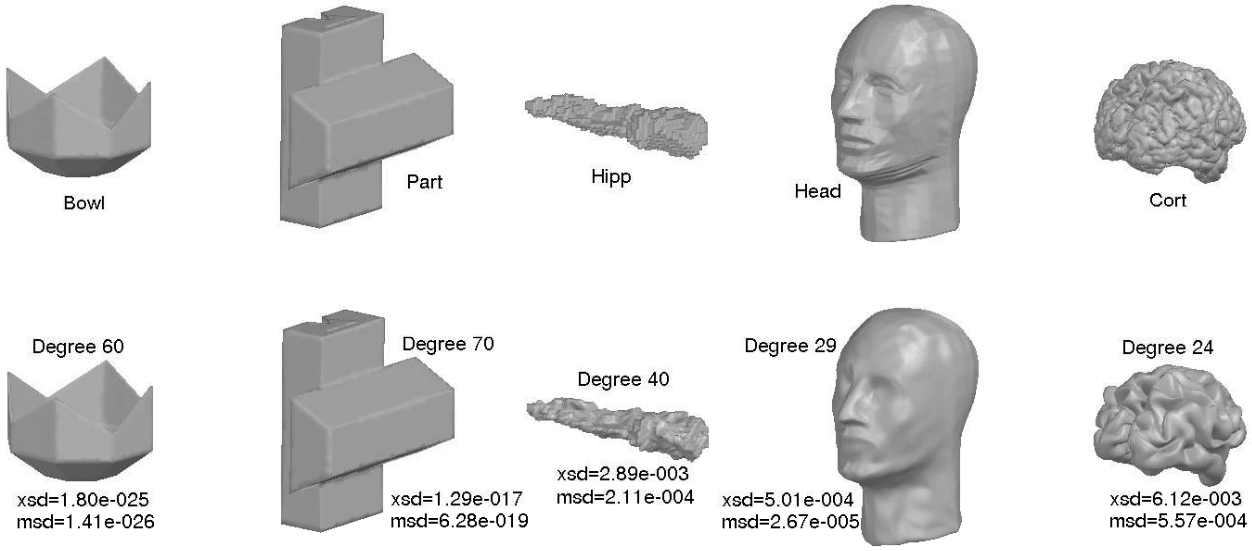


**Figure 4. Five 3D models and their SPHARM reconstructions created by NLSF.**

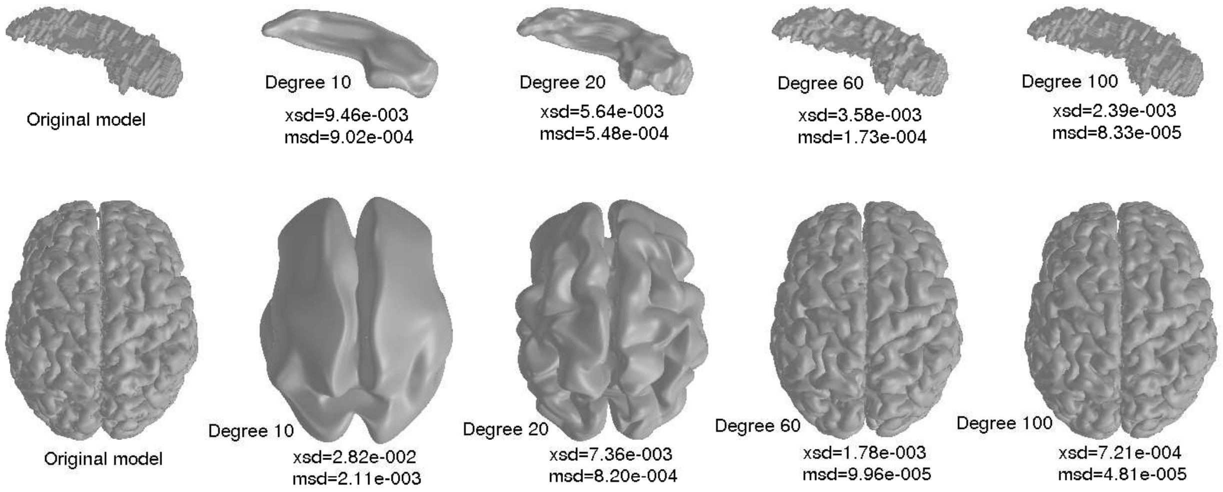
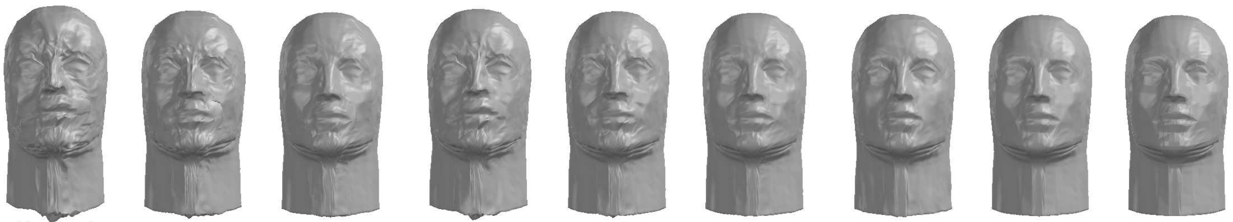


**Figure 5. Simple iterative residual fitting (SIRF) results for Hipp (top) and Cort (bottom).**



(a) g = 1, p = 1
xsd=1.05e-002
msd=2.22e-004

(b) g = 1, p = 2
xsd=4.56e-003
msd=4.73e-005

(c) g = 1, p = 4
xsd=1.60e-003
msd=8.66e-006

(d) g = 100, p = 1
xsd=4.30e-003
msd=8.78e-005

(e) g = 100, p = 2
xsd=1.72e-003
msd=1.81e-005

(f) g = 100, p = 4
xsd=9.12e-004
msd=3.78e-006

(g) g = 400, p = 1
xsd=2.46e-004
msd=1.16e-005

(h) g = 400, p = 8
xsd=7.95e-005
msd=1.18e-006

(i) Original model

**Figure 6. IRF result, where two parameters $g$ (granularity) and $p$ (number of passes) vary.**

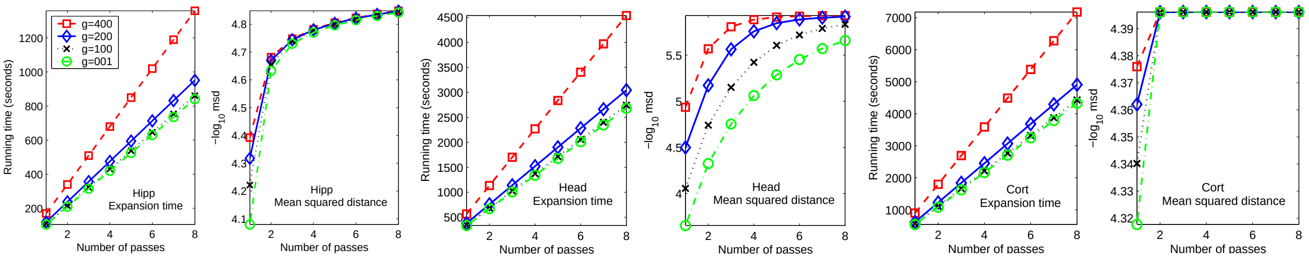


**Figure 7. Running times and modeling accuracies of IRF experiments on Hipp, Head and Cort.**

To sum up, we have proposed an iterative residual fitting (IRF) algorithm for large-scale modeling of parametric surfaces using spherical harmonics. The key ideas of IRF includes: (1) instead of solving a large linear system, IRF solves a series of smaller linear systems iteratively, and the size of the smaller system is controlled by the granularity parameter $g$; (2) IRF uses multiple passes of LSF to fit the residual and aims to improve the modeling accuracy, and the number of passes $p$ can be specified by a user. As a simple extension of LSF, our IRF algorithm is very easy to implement and does not require additional machine resources.

The IRF algorithm is also a generalization of the LSF method, since it turns into LSF when $g = \infty$ and $p = 1$. This setting works perfectly for small models, but not for large models due to the memory limitation. To perform large-scale expansion of complicated surfaces, we can adjust $g$, $p$, or both to get a desired result.

Our extensive experimental results show that the IRF algorithm can not only create accurate SPHARM models for large surfaces but also do it efficiently. For example, IRF can easily and accurately model a complicated surface with more than 40,000 vertices using spherical harmonics up to degree 100 within 10 minutes. Since our implementation is Matlab-based, we expect that the IRF algorithm has a potential to be implemented more efficiently using a lower level language such as C. The large-scale and accurate parametric models generated by IRF can derive many new applications in computer vision, graphics, and biomedical imaging.

## References

[1] G. B. Arfken. *Mathematical Methods for Physicists*. Academic Press, Orlando, 3rd edition, 1985.

[2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, N.J., 1982.

[3] R. Barrett, M. Berry, et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[4] C. Brechbühler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3D shape description. *Computer Vision and Image Understanding*, 61(2):154–170, 1995.

[5] T. Bulow. Spherical diffusion for 3D surface smoothing. *IEEE Trans. on PAMI*, 26(12):1650–1654, 2004.

[6] M. Chung, S. Robbins, et al. Model building in two-sphere via gauss-weierstrass kernel smoothing and its application to cortical analysis, Part 2. TR1116, Stat., UW-Madison, 2006.

[7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, McGraw-Hill, NY, 1990.

[8] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Multiresolution in Geometric Modelling*. Springer, 2004.

[9] R. Freund, G. Golub, and N. Nachtigal. Iterative solution of linear systems. pages 57–100, 1992.

[10] T. Funkhouser, P. Min, et al. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[11] G. Gerig and M. Styner. Shape versus size: Improved understanding of the morphology of brain structures. In *MICCAI'2001, LNCS 2208*, pages 24–32, 2001.

[12] G. Gerig, M. Styner, et al. Shape analysis of brain ventricles using SPHARM. In *IEEE MMBIA*, pages 171–178, 2001.

[13] X. Gu, Y. Wang, et al. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. on MI*, 23(7):1–10, 2004.

[14] A. J. Hanson. Hyperquadrics: smoothly deformable shapes with convex polyhedral bounds. *Comput. Vision Graph. Image Process.*, 44(2), 1988.

[15] 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(4):341–385, 2003.

[16] H. Huang, L. Shen, et al. Surface alignment of 3D spherical harmonic models: Application to cardiac MRI analysis. In *MICCAI'2005, LNCS 3749*, pages 67–74, 2005.

[17] S. C. Joshi, M. I. Miller, and U. Grenander. On the geometry and shape of brain sub-manifolds. *Int. J. of Pat. Rec. & Art. Int., Special Issue on MRI*, 11(8):1317–1343, 1997.

[18] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Trans. on Graphics*, 22(3):340–349, 2003.

[19] D. W. Ritchie and G. J. Kemp. Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces. *Journal of Comp. Chem.*, 20:383–395, 1999.

[20] R. Schudy and D. Ballard. Towards an anatomical model of heart motion as seen in 4-D cardiac ultrasound data. In *6th Conf. on Comp. App. in Rad. & Anal. of Rad. Im.*, 1979.

[21] L. Shen, J. Ford, F. Makedon, and A. Saykin. A surface-based approach for classification of 3D neuroanatomic structures. *Intelligent Data Analysis*, 8(6):519–542, 2004.

[22] L. Shen and F. Makedon. Spherical mapping for processing of 3-d closed surfaces. *Im. & Vis. Comp.*, 2006. in print.

[23] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. on PAMI*, 13(7):703–714, 1991.

[24] K. Zhou, H. Bao, and J. Shi. 3D surface filtering using spherical harmonics. *CAD*, 36(4):363–375, 2004.