

Efficient Spherical Harmonics Representation of 3D Objects

M. MOUSA R. CHAINE S. AKKOUCHE E. GALIN
LIRIS, Claude Bernard University Lyon1 - France
{mmousa, rchaine, sakkouch, egalin}@liris.cnrs.fr

Abstract

In this paper, we present a new and efficient spherical harmonics decomposition for spherical functions defining 3D triangulated objects. Such spherical functions are intrinsically associated to star-shaped objects. However, our results can be extended to any triangular object or oriented point set surface after segmentation into star-shaped surface patches and recombination of the results in the implicit framework. There is thus no restriction about the genus number of the object. We demonstrate that the evaluation of the spherical harmonics coefficients can be performed by a Monte Carlo integration over the edges, which makes the computation more accurate and faster than previous techniques, and provides a better control over the precision error in contrast to the voxel-based methods. We present several applications of our research, including fast surface reconstruction from point clouds, local surface smoothing and interactive geometric texture transfer.

1. Introduction

In computer graphics community, the spherical harmonics have gained much of interest due to their contribution in many fields such as global illumination [5], shape descriptors [11, 12], shape reconstruction of star-shaped point set [22], frequency-based representations and filtering [25, 16]. The spherical harmonics decomposition is naturally applied to spherical domains (star-shaped objects) and may be extended to non spherical domains using a prior spherical parameterization. This can be used in particular to zero genus 3D objects described by triangular meshes [25].

Current methods for computing the spherical harmonics transform often sample these spherical functions over a regular grid on the sphere or a 3D grid surrounding the object (voxelization) and apply a discrete spherical harmonics transform algorithm [6, 14, 20] to this regular grid to compute the harmonics coefficients. Spatial grid-based methods [4, 11, 19] are prone to numerical errors associated with the size of the cells. It is very difficult to predict the size of the

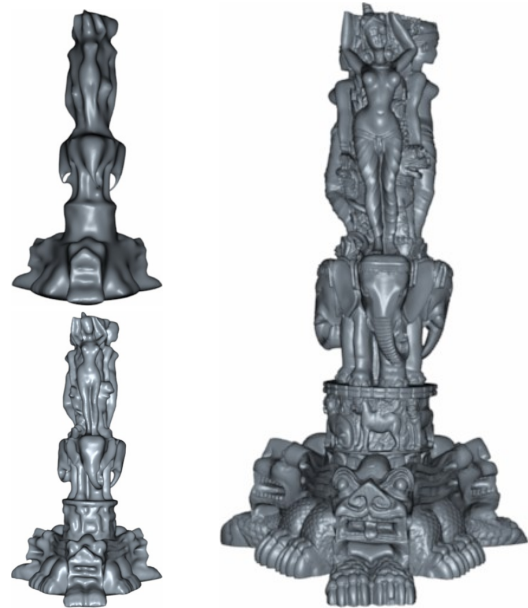


Figure 1. The model of Thai statue at different levels of details represented by our spherical harmonics representation using a bandwidth of 24, 64 and 480. The initial data set of the model is given as an oriented set of 3000k points without any information about the connectivity.

voxel which satisfies a given error tolerance. In addition, the complexity of the method increases tremendously as the voxel size decreases.

In this paper, we present a new and efficient algorithm for computing the spherical harmonics decomposition of spherical functions measuring the radial extent of the points of 3D objects represented as meshes or oriented point sets. The main contribution of our approach is as follows:

Efficient spherical harmonics The evaluation of the spherical harmonics coefficients is distributed over the edges thanks to the Curl theorem that permits to reduce a

first formulation over the triangles.

Error Control The evaluation of the spherical harmonics coefficients is made through a Monte Carlo method that generates a uniformly distributed set of random points along the edges. The precision of each coefficient is also directly binded on the number of points on each edge. Moreover, when the spherical harmonics transform is used for filtering purpose we add more control on the global geometric approximation error since the spherical harmonics coefficients are computed progressively until the appropriate bandwidth is reached.

Processing general objects If the object is not star-shaped, we segment it into star-shaped surface patches and recompose the local results on each patch. This process can also be extended to handle point clouds directly without a prior global mesh reconstruction step, and there is no restriction about the genus number of the objects.

Local spherical harmonics The computations are performed directly and locally on the surface description of the object without neither voxelization nor spherical parameterization.

Our computation mainly depends on the geometry and the normal direction over the surface. If the input object is described as mesh, the normal information is implicitly embedded in the description. In the case of point cloud, we assume that each point is supplied with the corresponding normal. This is a quite usual assumption when working with point-based objects, otherwise, the normal direction can be estimated using least square fitting and a consistent normal orientation can be propagated via a minimum spanning tree for the point set [7].

The remainder of the paper is organized as follows. Section 2 presents an overview of related work. Section 3 presents fundamental concepts and notations. Section 4 presents our method for accelerating the computation of the spherical harmonics transform by distributing the computation over the edges of star-shaped meshes. Section 5 presents an extension of our algorithm for computing the spherical harmonics transform for non star-shaped meshes and surface point clouds. Section 6 presents some applications of our approach to demonstrate its efficiency.

2. Related work

The extension of the spectral concepts to the geometry of 3D models have gained much of interest during the last decade. Conceptually, this generalization is achieved thanks to the Laplace-Beltrami operator. Taubin [21] has proposed a spectral method to fair triangular meshes by coding the

cartesian coordinates into differential coordinates at each vertex using the discrete Laplacian. This method has been improved by Desbrun et al. [2] by introducing the curvature flow on the surface to remove noise. Although, these methods are based on the methodology of signal processing and they do not compute a real frequency-based representation of the surface of the objects. Therefore the corresponding filters should be applied in the spatial domain.

A real frequency-based representation can be obtained by considering the eigenfunctions of the Laplace-Beltrami operator as a set of basis functions. Karni et al. [8] have used this decomposition to compress triangular meshes. One of the drawback of their approach is that the basis eigenvectors must be computed separately for each mesh, as pointed in [9]. For more detail about Laplace-Beltrami eigenfunctions as a basis for surface based function spectral decomposition, we recommend the reader to refer to [13]. The eigenfunctions of the Laplacian of the mesh can be used also for surface quadrangulation [3].

The planar and spherical domains provide us with a well known set of eigenfunctions: Fourier exponentials and the spherical harmonics, respectively. The frequency-based representations, using Fourier transforms and spherical harmonics, are used for retrieval, surface reconstruction, compression, progressive transmission and filtering purposes.

Fourier transform Vranic et al. [23] have applied 3D Fourier transforms to a function induced from the 3D voxelized mesh to obtain shape descriptor for retrieval purposes. Kazhdan [10] has used 3D Fourier transform of the indicator function of an oriented set of points to reconstruct a surface that approximates these points. An interesting point of his approach is that he has converted the volumetric integrations of the Fourier transform of the indicator function into surface integrations. These surface integrations can be computed directly on the set of points. However, he has used a posterior voxelization to convert the set of the obtained frequencies into an implicit representation of the object in the spatial domain.

In general, it is more difficult to control the numerical errors from voxelized-based computations. For a given error tolerance, it is very difficult to predict the optimal size of the voxel. Pauly et al. [18] have used Fourier analysis to achieve common surface processing operations as noise removal, enhancement, restoration and subsampling. They have interpolated the data onto a regular grid, in order to make a discrete Fourier transform applicable. However, sampling the data over a regular grid on the sphere ignores some important regions and oversamples others.

Spherical harmonics transform The spherical harmonics transform of spherical functions [4, 19] induced from the mesh can be calculated using a voxelization of the mesh as

a preprocessing step to obtain a shape descriptor. In [25], Zhou et al. have applied the spherical harmonics transform to three independent spherical functions representing the spherical parametrization of the local coordinates x , y and z over the original mesh. However, using three independent functions ignores the correlation between these coordinates on the surface.

An interesting idea on efficient computations of geometrical properties of triangulated meshes has been presented by Zhang et al. [24]. They have shown that the computation of the volumetric moments of a triangulated mesh can be decomposed into a signed sum of the volumetric moments computed over the geometric primitives composing the mesh. The geometric primitives are triangles or tetrahedra for meshes embedded in 2D or 3D respectively. Mousa et al. [15] have extended this idea to calculate the spherical harmonics transform of the indicator function representing the intersection of the volume enclosed by a triangulated mesh and a given sphere. The computation consists of a signed sum of 2D Monte-Carlo integrations on a set of spherical triangles. A similar technique was used in [16] to obtain a frequency-based representation of triangular meshes using the spherical harmonics.

3. Notations

The spherical harmonics $\{Y_l^m(\theta, \varphi) : |m| \leq l \in \mathbb{N}\}$ are special functions defined on the unit sphere \mathbb{S}^2 as :

$$Y_l^m(\theta, \varphi) = k_{l,m} P_l^m(\cos \theta) e^{im\varphi} \quad (1)$$

where $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi]$, $k_{l,m}$ is a constant, and P_l^m is the associated Legendre polynomial. The spherical harmonics are orthonormal functions such that ¹:

$$\int_0^{2\pi} \int_0^\pi Y_l^m(\theta, \varphi) \overline{Y_{l'}^{m'}}(\theta, \varphi) \sin(\theta) d\theta d\varphi = \delta_{l,l'} \delta_{m,m'} \quad (2)$$

where $\delta_{u,v}$ is the Kronecker delta function defined as the following :

$$\delta_{u,v} = \begin{cases} 1 & \text{if } u = v \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Therefore, a spherical function $g : \mathbb{S}^2 \rightarrow \mathbb{R}$ can be expanded as a linear combination of spherical harmonics :

$$g(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_{l,m} Y_l^m(\theta, \varphi) \quad (4)$$

where the coefficients $c_{l,m}$ are uniquely determined by :

$$c_{l,m} = \int_0^{2\pi} \int_0^\pi g(\theta, \varphi) \overline{Y_l^m}(\theta, \varphi) \sin(\theta) d\theta d\varphi \quad (5)$$

¹The bar over the variable means the complex conjugate of this variable.

4. Spherical harmonics computations

Given a star-shaped triangular mesh, we present in this section a new algorithm to compute the spherical harmonics decomposition of the spherical function that measures the distance from the surface to a center c . Firstly, we show how such spherical function can be expanded as a sum of spherical functions defined over the local triangles. Secondly, we show how to evaluate spherical harmonics coefficients by a sole integration along the edges of these triangles.

4.1. Spherical harmonics decomposition

Consider the spherical function f that measures the radial extent of the points of a given star-shaped triangulated object M . We will show now how to expand this spherical function over the triangles of M .

Assuming that $f_i(\theta, \varphi)$ is the restriction of f to the triangle T_i , i.e. f_i has the same value than f for the points belonging to the triangle T_i and has the value zero otherwise. The global spherical function $f(\theta, \varphi)$ can be decomposed in terms of $f_i(\theta, \varphi)$ as follows :

$$f(\theta, \varphi) = \sum_{T_i \in T} f_i(\theta, \varphi) \quad (6)$$

where T is the set of triangles of M . The spherical harmonics decomposition of each f_i is given by :

$$f_i(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_{l,m}^i Y_l^m(\theta, \varphi) \quad (7)$$

Therefore, the expansion of $f(\theta, \varphi)$ in terms of the spherical harmonics can be rewritten as follows :

$$f(\theta, \varphi) = \sum_{T_i \in T} \left(\sum_{l=0}^{\infty} \sum_{m=-l}^l c_{l,m}^i Y_l^m(\theta, \varphi) \right) \quad (8)$$

Let T'_i denote the projection of T_i on the unit sphere centered at c . Each $c_{l,m}^i$ is computed using the following equation:

$$c_{l,m}^i = \int \int_{T'_i} f_i(\theta, \varphi) \overline{Y_l^m}(\theta, \varphi) \sin(\theta) d\theta d\varphi \quad (9)$$

The above equation evaluates the coefficients $c_{l,m}^i$ as a surface integration over the spherical triangle T'_i . Then, the evaluation of $c_{l,m}^i$ is reduced to the evaluation of the corresponding $c_{l,m}^i$ over the primitive forms of the mesh. Hence, the main computation of the spherical harmonics decomposition of the function f is simplified to computation of the spherical harmonics decomposition of the functions f_i .

4.2. SHT over the triangle edges

In this subsection, we show how to reduce the evaluation of $c_{l,m}^i$ as a linear integration over the boundaries of the spherical triangle T'_i . For that, we rely on the Curl theorem that permits to reduce some 2D integrations to 1D integrations.

4.2.1 The Curl theorem

The Curl operator is a vector operator that measures a vector field's rate of rotation: the direction of the axis of rotation and the magnitude of the rotation. Let V denote a volume element delimited by a closed surface element S . The Curl of a vector field \mathbf{F} denoted as $\text{Curl}(\mathbf{F}) = \nabla \times \mathbf{F}$ is defined as :

$$\text{Curl}(\mathbf{F}) = \lim_{V \rightarrow 0} \frac{1}{V} \oint_S \mathbf{n} \times \mathbf{F} dS$$

where \mathbf{n} is the normal of S .

The Curl theorem is an extension of the fundamental theorem of calculus which provides a method for expressing the integral of some function over the interior of a region S as an integral over its boundary ∂S . Given a 3D vector field function \mathbf{F} and a surface S with boundaries, the Curl theorem states :

$$\int_S (\nabla \times \mathbf{F}) \cdot d\mathbf{a} = \int_{\partial S} \mathbf{F} \cdot d\mathbf{s} \quad (10)$$

In the spherical coordinates system, the Curl of a vector function $\mathbf{F} \equiv (F_r, F_\theta, F_\varphi)$ is defined as :

$$\nabla \times \mathbf{F} = \begin{bmatrix} \vec{r} & \vec{\theta} & \vec{\varphi} \end{bmatrix} \begin{bmatrix} \frac{1}{r \sin \theta} \left[\frac{\partial(\sin \theta F_\varphi)}{\partial \theta} - \frac{\partial F_\theta}{\partial \varphi} \right] \\ \frac{1}{r} \left[\frac{1}{\sin \theta} \frac{\partial F_r}{\partial \varphi} - \frac{\partial(r F_\varphi)}{\partial r} \right] \\ \frac{1}{r} \left[\frac{\partial(r F_\theta)}{\partial r} - \frac{\partial F_r}{\partial \theta} \right] \end{bmatrix} \quad (11)$$

where r, θ, φ are spherical coordinates and $\vec{r}, \vec{\theta}, \vec{\varphi}$ the local unit vectors basis. Moreover, the area and the line elements are given by :

$$d\mathbf{a} = r^2 \sin \theta d\theta d\varphi \vec{r} \quad (12)$$

$$d\mathbf{s} = dr \vec{r} + r d\theta \vec{\theta} + r \sin \theta d\varphi \vec{\varphi} \quad (13)$$

Using equations 11, 12 and 13, the left hand side of equation 10 is rewritten as follows :

$$(\nabla \times \mathbf{F}) \cdot d\mathbf{a} = r \left[\frac{\partial(\sin \theta F_\varphi)}{\partial \theta} - \frac{\partial F_\theta}{\partial \varphi} \right] d\theta d\varphi \quad (14)$$

Therefore, in the spherical coordinates system, we can reformulate the Curl theorem as the following :

$$\int_S \left[\frac{\partial(\sin \theta F_\varphi)}{\partial \theta} - \frac{\partial F_\theta}{\partial \varphi} \right] d\theta d\varphi = \int_{\partial S} [F_r dr + r F_\theta d\theta + r \sin \theta F_\varphi d\varphi] \quad (15)$$

4.2.2 SHT computation

To calculate the spherical harmonics coefficients $c_{l,m}^i$ as a linear integration along the edges of the spherical triangle T'_i , we need a vector field function $\mathbf{F}^i = (F_r^i, F_\theta^i, F_\varphi^i)$ such that the following equality is satisfied :

$$(\nabla \times \mathbf{F}^i) \cdot d\mathbf{a} = f_i(\theta, \varphi) \overline{Y}_l^m(\theta, \varphi) \sin(\theta) d\theta d\varphi \quad (16)$$

Any vector field \mathbf{F}^i satisfying equation 16 gives the same evaluation of the coefficients $c_{l,m}^i$. We will show that it is possible to find a vector field function of the form $\mathbf{F}^i = (0, F_\theta^i, 0)$, i.e. such that the field lines of \mathbf{F}^i are geodesics along the $\vec{\theta}$ direction. This allows a simplified expression of the Curl operator. Therefore, on the unit sphere \mathbb{S}^2 ($r = 1$), the above equation reduces to :

$$\text{Curl}(\mathbf{F}^i) \cdot d\mathbf{a} = -\frac{\partial F_\theta^i}{\partial \varphi} d\theta d\varphi = f_i(\theta, \varphi) \overline{Y}_l^m(\theta, \varphi) \sin(\theta) d\theta d\varphi \quad (17)$$

And hence,

$$-\frac{\partial F_\theta^i}{\partial \varphi} = f_i(\theta, \varphi) \overline{Y}_l^m(\theta, \varphi) \sin(\theta) \quad (18)$$

$$F_\theta^i = -\int f_i(\theta, \varphi) k_{l,m} P_l^m(\cos \theta) \sin(\theta) e^{-im\varphi} d\varphi \quad (19)$$

Using the integration by parts (see appendix A), we get :

$$F_\theta^i(r, \theta, \varphi) = \frac{f_i(\theta, \varphi) k_{l,m} P_l^m(\cos \theta) \sin(\theta) e^{-im\varphi}}{-\frac{k_{l,m} P_l^m(\cos \theta) \sin(\theta)}{im} \int e^{-im\varphi} \frac{\partial f_i(\theta, \varphi)}{\partial \varphi} d\varphi} \quad (20)$$

Using this expression of F_θ^i and the Curl theorem, the determination of $c_{l,m}^i$ amounts to :

$$c_{l,m}^i = \int_{T'_i} (\nabla \times \mathbf{F}^i) \cdot d\mathbf{a} \quad (21)$$

Therefore,

$$c_{l,m}^i = \int_{\partial T'_i} F_\theta^i d\mathbf{s} \quad (22)$$

where $\partial T'_i$ is the boundary of the spherical triangle T'_i . Thus, we can calculate the spherical harmonics coefficient $c_{l,m}^i$ by applying the Curl theorem to the vector field function \mathbf{F}^i . The computation of $c_{l,m}^i$ reduces to a 1D integration of \mathbf{F}^i along the edges of the triangle T'_i . However, the right most integration of equation 20 is not easy to calculate and need to be simplified. Consider the quantity $g_1(p) = \overrightarrow{pp_i} \cdot \overrightarrow{n_i}$ where $\overrightarrow{n_i}$ is the normal of T_i and p_i is the projection of p on the plane supporting T_i .

$$\nabla g_1 = \overrightarrow{n_i} \quad (23)$$

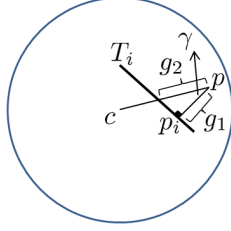


Figure 2.

Similarly, let $g_2(p) = r - f_i(\theta, \varphi)$ where r, θ and φ are the spherical coordinates of p . From Figure 2, $g_2(p) \cos \gamma = g_1(p)$, therefore

$$\nabla g_2 \times \cos \gamma + g_2 \times \nabla \cos \gamma = \nabla g_1 \quad (24)$$

For all points p lying on T_i , $g_2(p) = 0$. Then

$$\nabla g_2(p) \times \cos \gamma = \nabla g_1(p) \quad (25)$$

Moreover, $\vec{r} \cdot \vec{n}_i = \cos \gamma$.

$$\nabla(r - f_i) \times (\vec{r} \cdot \vec{n}_i) = \vec{n}_i \quad (26)$$

$$\left(\vec{r} - \frac{\partial f_i}{\partial r} \vec{r} - \frac{1}{r} \frac{\partial f_i}{\partial \theta} \vec{\theta} - \frac{1}{r \sin(\theta)} \frac{\partial f_i}{\partial \varphi} \vec{\varphi} \right) \times (\vec{r} \cdot \vec{n}_i) = \vec{n}_i \quad (27)$$

Multiplying by $\vec{\varphi}$

$$-\frac{1}{r \sin \theta} \frac{\partial f_i}{\partial \varphi} \vec{r} \cdot \vec{n}_i = \vec{n}_i \cdot \vec{\varphi} \quad (28)$$

Taking $\kappa = -\frac{r \sin \theta}{\vec{r} \cdot \vec{n}_i}$ we get

$$\frac{\partial f_i}{\partial \varphi} = \kappa \vec{n}_i \cdot \vec{\varphi} \quad (29)$$

The right most term of equation 20 simplifies to :

$$\int e^{-im\varphi} \frac{\partial f_i}{\partial \varphi} d\varphi = \kappa \int (\vec{n}_i \cdot \vec{\varphi}) e^{-im\varphi} d\varphi \quad (30)$$

The normal \vec{n}_i is constant for all points on the triangle T_i . Moreover, $\vec{\varphi}$ is the unit vector in the direction of φ and is equal to $(-\sin(\varphi), \cos(\varphi), 0)$. Therefore, we have :

$$\int e^{-im\varphi} \frac{\partial f_i}{\partial \varphi} d\varphi = \kappa \vec{n}_i \cdot \int \vec{\varphi} e^{-im\varphi} d\varphi \quad (31)$$

To evaluate the integration presented in equation 31, the following sub-integrations should be evaluated :

$$I_1 = \int -\sin(\varphi) e^{-im\varphi} d\varphi \quad (32)$$

$$I_2 = \int \cos(\varphi) e^{-im\varphi} d\varphi \quad (33)$$

Using the integration by parts (see Appendix B), we get :

$$I_1 = \begin{cases} \frac{e^{-im\varphi}}{1-m^2} [\cos \varphi + im \sin \varphi] & m \neq \pm 1 \\ \frac{1}{4} \cos 2\varphi + \frac{im}{2} [\varphi - \frac{1}{2} \sin 2\varphi] & m = \pm 1 \end{cases} \quad (34)$$

$$I_2 = \begin{cases} \frac{e^{-im\varphi}}{1-m^2} [\sin \varphi - im \cos \varphi] & m \neq \pm 1 \\ \frac{im}{4} \cos 2\varphi + \frac{1}{2} [\varphi + \frac{1}{2} \sin 2\varphi] & m = \pm 1 \end{cases} \quad (35)$$

Therefore, given the values of the surface normal at any given point, we can evaluate F_θ^i for any value of l, m, θ and φ using equations 20, 31, 34 and 35, and hence $c_{l,m}^i$.

5. Algorithm

The spherical harmonics computations are naturally adapted to spherical domains. Non spherical objects should be preprocessed such that they are decomposed into spherical subparts. In this section, we give an algorithm that generalizes our approach to any mesh of point set surface. Our method operates in five steps as illustrated in Figure 3 :

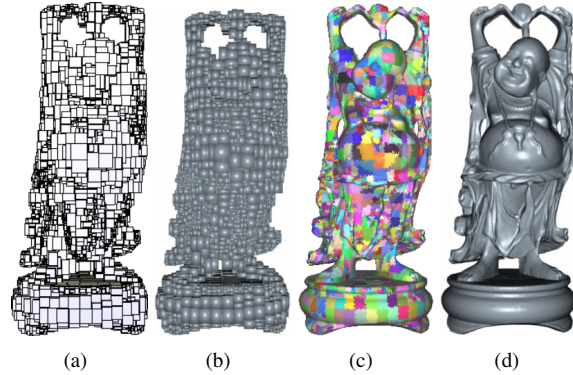


Figure 3. The main steps of the application of our approach to a non spherical object. (a) The kd-tree decomposition. (b) The best fitting spheres for the cells of the kd-tree. (c) The triangulation of the local spherical patches. (d) The corresponding object after the spherical harmonics computations.

1. Partition the object using a kd-tree and an unfolding split condition.
2. For each cell of the kd-tree find the best fitting sphere on which we apply the spherical harmonics computations.
3. If the object is already meshed, ignore this step. Otherwise, triangulate each local patch of the object.

4. Apply the spherical harmonics computations for the local polar function representing this patch.
5. Blend together the spherical harmonics representations of the subparts to reconstruct the overall object.

We present the details of those steps in the next subsections.

5.1. Partitioning

Starting from the initial bounding box of the set of points of the mesh, we adopt the division criterion presented by Boubekeur et al. [1] : each cell of the kd-tree hierarchy, \mathbf{C}_i , is split if the following condition is not satisfied :

$$\forall p \in \mathbf{C}_i \implies \vec{n}_p \cdot \vec{n}_i > \delta_a \quad \delta_a \in [0, 1] \quad (36)$$

where \vec{n}_p is the surface normal at the point p and \vec{n}_i is the average normal vector in \mathbf{C}_i . This condition ensures that there is no folding inside the cell \mathbf{C}_i . Therefore projecting the local points of the cell \mathbf{C}_i on the local sphere will keep the local neighborhood relationship between the points. This process creates locally spherical geometry which is the natural domain of the spherical harmonics computations.

5.2. Best fitting sphere

Now, we show how we can find a suitable sphere that provides a good spherical approximation for the spatial locations for the points $p \in \mathbf{C}_i$. The objective is to find a sphere centered at \mathbf{c} and having a radius r satisfying as much as possible :

$$\|p_j - \mathbf{c}\|^2 - r^2 = 0 \quad \forall p_j \in \mathbf{C}_i \quad (37)$$

Introducing the variable $\rho = \mathbf{c}^2 - r^2$ into equation (37), the previous set of quadratic equations becomes a system of linear equations for the unknown coordinates x_c, y_c, z_c of \mathbf{c} and ρ :

$$(2x_j x_c + 2y_j y_c + 2z_j z_c) - \rho - (x_j^2 + y_j^2 + z_j^2) = 0 \quad \forall p_j \in \mathbf{C}_i \quad (38)$$

We solve this system of linear equations using the least square method to give the best values of the center and the radius of the best fitting sphere.

5.3. Local triangulation

The decomposition criterion presented in the previous subsection ensures that the spherical patch may not contain any folding. This yields that we can project the local patch within the cell \mathbf{C}_i on its best fitting sphere. This projection will maintains the neighborhood properties of the points inside \mathbf{C}_i . Therefore, the local triangulation of the points inside each kd-tree cell can be approximated by triangulating their projection of the fitting sphere.

5.4. Numerical evaluation

In the previous section, we have seen how to find the vector field function $\mathbf{F}^i = (F_r^i, F_\theta^i, F_\varphi^i)$ that enables us to use the Curl theorem and to reduce the integration over the triangles along their edges. To evaluate the value of the harmonic coefficients $c_{l,m}^i$, we need to compute the line integral of \mathbf{F}^i along the edges of the triangle T_i :

$$c_{l,m}^i = \int_{\partial T_i} F_\theta^i ds \quad (39)$$

We use a Monte Carlo integration to evaluate this integral over each edge $e_1 e_2$ of the triangles as summarized in the following steps :

1. Pick n randomly distributed points p_1, p_2, \dots, p_n over the edge $e_1 e_2$.
2. Determine the average value of the function

$$\langle F_\theta^i \rangle = \frac{1}{n} \sum_{i=1}^n F_\theta^i(p_i)$$

3. Compute the approximation of the integral

$$\int_{e_1}^{e_2} F_\theta^i d\theta = (\|e_1 e_2\|) \langle F_\theta^i \rangle$$

The standard deviation of the above evaluation is equal to :

$$\sigma = \sqrt{\langle (F_\theta^i)^2 \rangle - \langle F_\theta^i \rangle^2}$$

where

$$\langle (F_\theta^i)^2 \rangle = \frac{1}{n} \sum_{i=1}^n (F_\theta^i(p_i))^2$$

This yields a convergence of the order $O(n^{-\frac{1}{2}})$. σ can interactively be computed during the evaluation of F_θ^i over the sample points, so that the number of Monte Carlo samples can be tuned to reach some fixed deviation for each degree l spherical harmonics coefficient. Moreover, generating evenly distributed sample points along the edge $e_1 e_2$ is straightforward from the parametric equation of the line $e_1 e_2$:

$$p = e_1 + \lambda(e_2 - e_1) \quad \lambda \in [0, 1] \quad (40)$$

The computation of the spherical harmonics coefficients $c_{l,m}$ of the spherical function f is summarized in Algorithm 1.

Algorithm 1 Computation of the spherical harmonics coefficients $c_{l,m}$ of the distance function $f(\theta, \varphi)$

Require: a triangulated mesh M oriented consistently;
Ensure: SHT coefficient $c_{l,m}$ of $f(\theta, \varphi)$ representing M ;
 $E \Leftarrow$ the set of edges of M ;
 $T \Leftarrow$ the set of triangles of M ;
 $c_{l,m} \Leftarrow 0$;
for all $T_i \in T$ **do**
 $c_{l,m}^i \Leftarrow 0$;
end for
for all $e \in E$ **do**
 $T_{i_1}, T_{i_2} \Leftarrow$ the triangles sharing the edge e ;
 $f_{i_1} \Leftarrow$ the distance function parametrizing T_{i_1} ;
 $f_{i_2} \Leftarrow$ the distance function parametrizing T_{i_2} ;
 $c_{l,m}^{i_1} \Leftarrow c_{l,m}^{i_1} + \frac{\|e\|}{n} \mathbf{F}_\theta^{i_1} |_e$
 ▶ $\mathbf{F}_\theta^{i_1}$ computed using equations 20, 31, 34 and 35
 $c_{l,m}^{i_2} \Leftarrow c_{l,m}^{i_2} + \frac{\|e\|}{n} \mathbf{F}_\theta^{i_2} |_e$
 ▶ $\mathbf{F}_\theta^{i_2}$ computed using equations 20, 31, 34 and 35
end for
for all $T_i \in T$ **do**
 $c_{l,m} \Leftarrow c_{l,m} + c_{l,m}^i$
end for
return $c_{l,m}$

5.5. Error control

The quality of the spherical harmonics representation of the objects depends on the number of Monte Carlo samples used to evaluate each coefficient and on the number of spherical harmonics coefficients to be computed for each distance function f . Noting that the approximated distance function \hat{f} is given by :

$$\hat{f}(\theta, \varphi) = \sum_{l=0}^{\beta} \sum_{m=-l}^l c_{l,m} Y_l^m(\theta, \varphi) \quad (41)$$

To determine the bandwidth β interactively, we introduce an error tolerance ε_β between the approximated function \hat{f} and the original function f . Of course, the deviation on the estimation of each coefficient should be negligible with regards to the authorized error tolerance. ε_β is defined as follows:

$$\varepsilon_\beta = \sqrt{\sum_{j \in V} [f(\theta_j, \varphi_j) - \hat{f}(\theta_j, \varphi_j)]^2} \quad (42)$$

where V denotes the set of local points.

5.6. Blending local patches

After computing the local frequency-based representation of each local patch of the object, we need to blend all

these representations of the object. To do that, we associate for each distance function f a function g which is expressed in the same local spherical coordinates system defined within the spatial cell \mathbf{C} containing f . The functions g are defined as follows :

$$g(p) = \text{sign}(\vec{n} \cdot \vec{cp}) (f(\theta, \varphi) - r)$$

where (r, θ, φ) are the spherical coordinates of p , c is the center of the coordinate system and \vec{n} is the average normal vector inside the cell \mathbf{C} .

Each triangulated patch divides the containing cell into two parts. One is contained inside the object and the other is outside. g has positive values for points located within the surface, has negative values for those located outside and has zero values on the surface. Therefore, g can be considered as the potential function of the local surface patch located inside the cell \mathbf{C} . We can obtain the entire surface by blending all the potential functions g and extracting the corresponding 0-level surface using a marching cubes algorithm. The standard form of the blending operator is expressed as a weighted sum of the potential functions g :

$$G(p) = \sum w_g(p)g(p) \quad (43)$$

In this paper, we use quadratic B-spline $b(t)$ [17] to generate the blending weight functions;

$$w_g(p) = b\left(\frac{3\|p - c\|}{2R}\right) \quad (44)$$

centered at c and having a spherical support of radius R , where R is the radius of the cell \mathbf{C} .

6. Applications and discussion

Our method is implemented in C++; we have used a 3GHz Pentium IV PC with 1GB memory for the experiments. The input objects are given as a set of oriented points. The normal directions of the points are assumed to be oriented consistently.

6.1. Spectral representation and filtering for star-shaped meshes

Figure 4 shows a vase model reconstructed using the approach presented in [16] in comparison with our approach at the same bandwidth $\beta = 200$. The same number of Monte Carlo samples were used in both cases. The difference is that the Monte Carlo samples are distributed over the triangles and along the edges respectively. Figure 5 show the representation of Max Plank using our spherical harmonics decomposition. We can see that when the value of ε_β is large, the object becomes smoother and it becomes more detailed when ε_β is smaller. The timings are given in Table 1.

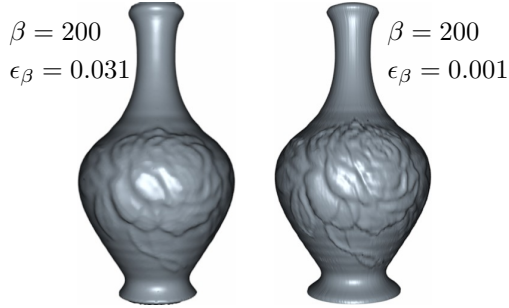


Figure 4. A vase model reconstructed using [16] (left) and using our approach (right) at the same bandwidth β and the same number of Monte Carlo samples. The corresponding error is 31 time less using our approach.

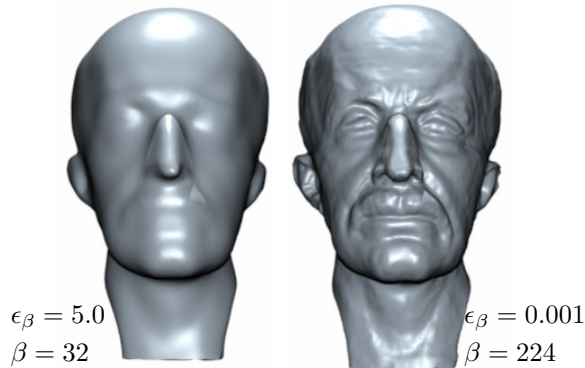


Figure 5. The model of Max Plank represented using our spherical harmonics decomposition.

6.2. Surface reconstruction

Given a set of oriented points, we can reconstruct a surface that approximates the given points using our spherical harmonics decomposition. We can do that by simply computing the spherical harmonics representation of the set of points as described in section 5 and apply a marching cubes algorithm to reconstruct the surface.

The spatial decomposition and the local triangulation steps take no more that 5 seconds for the objects presented in this paper. The models shown in Figures 1 and 3(d) are reconstructed in this way. Figure 6 shows the reconstruction of 3D objects using the spherical harmonics representation. The precision error ϵ_β is fixed to 0.001. Table 1 summarizes the timing of the computation of the spherical harmonics decomposition for the models presented in this paper. The last two columns show the corresponding time and ϵ_β if we apply the evaluation over the whole triangle [16] with the

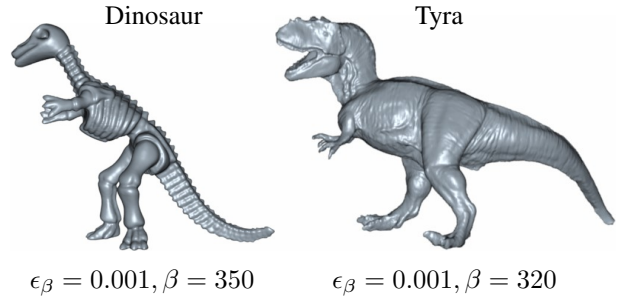


Figure 6. Reconstruction of 3D objects using our spherical harmonics representation. The models of Dinosaur and Tyra are decomposed into 250 and 200 patches respectively.

Model	Points	β	Time	Time [16]	ϵ_β [16]
Igea	135k	256	5.5	140	0.012
Max-Plank	200k	224	4.2	220	0.01
Dinosaur	56k	350	4.4	59	0.018
Tyra	100k	320	6.8	130	0.01
Buddha	543k	450	10.7	480	0.009
Thai	3000k	480	20.2	1200	0.019
Vase	68k	200	4.7	65	0.031

Table 1. Summary of the timing (in seconds) for the models used in this paper.

same β .

The spherical harmonics decomposition presented in this paper has many advantages. One of them is that we can control number of computed spherical harmonics coefficients. This can be useful in some applications.

6.3. Local mesh smoothing

We can repair cracks found on some 3D surfaces by eliminating the higher frequency components from their spherical harmonics decomposition at the local patches containing the cracks, Figure 7. This is not equivalent to the filtering of the surface. In fact, eliminating high frequency components in the filtering process affects the geometric details over all the surface while in our case the effect is local.

6.4. Texture transfer

Given a source surface patch S containing the geometric texture and a target surface patch U , we compute their spherical harmonics representation as described in section 4. As we have mentioned, the higher frequency components

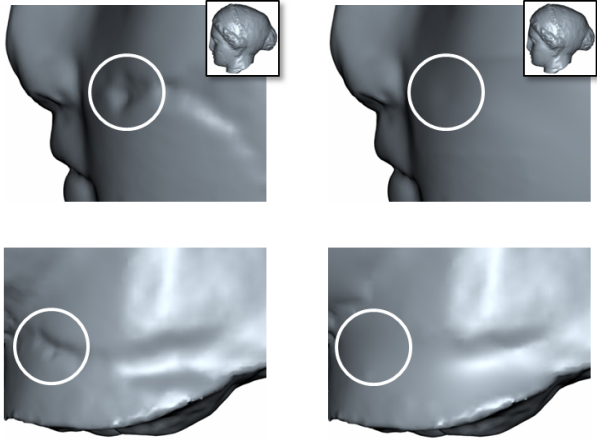


Figure 7. Smoothing parts of the Igea model using local filtering of our spherical harmonics representation.

encode the geometric details on the surface S . The number of these components that we will add to the target U is a user defined parameter that identifies the level of details to be extracted from the surface S . An alignment of the region of interest is performed before this transfer of coefficients.

Figure 8 shows an example of transferring a part of Armadillo’s coating onto the Bunny’s back using the spherical harmonics representation.

7. Conclusions

Traditionally the spherical harmonics decomposition has been applied to 3D objects by employing a spherical parametrization of the mesh representing the surface as a preprocessing step. In this paper, we defend the idea that a single spherical parametrization should be used instead. It enables an exploitation of the geometric correlation between the x,y,z coordinates over the surface. The polar parametrization provides a good candidate for star-shaped meshes. However it can be extended to any triangular object after segmentation into star-shaped sub volumes and recombination of the results in the implicit framework. In the case of oriented point set surfaces, we propose to use some more local segmentation that also provides an atlas of local triangular patches. Given a spherical distance function defining a star-shaped triangulated shape, the principal point of our approach is that the computation of the spherical harmonics decomposition can be performed directly from the oriented edges of the triangles. More precisely, the usage of the Curl theorem permits to distribute the computation of the spherical harmonics coefficients over geodesic arcs on the sphere. This gives faster convergence and more control on the eval-

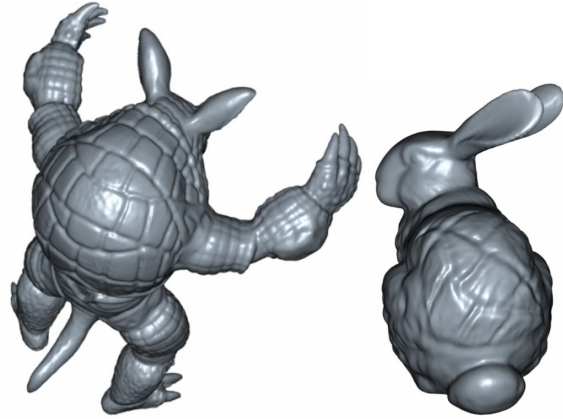


Figure 8. Geometric texture transfer using the spherical harmonics.

uation of the integrations determining the coefficients.

Using the proposed spherical harmonics decomposition, we can add and remove details from local patches on the surface without affecting the entire surface at interactive rates. We can also transfer geometric texture locally from a given model to another. This makes our approach suitable for applications such as local filtering, mesh editing and surface repairing.

References

- [1] T. Boubekeur, P. Reuter, and C. Schlick. Visualization of point-based surfaces with locally reconstructed subdivision surfaces. In *Shape modeling international*, 2005.
- [2] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99*, pages 317–324, 1999.
- [3] S. Dong, P. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *SIGGRAPH'06*, pages 1057–1066, 2006.
- [4] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.
- [5] R. Green. Spherical harmonic lighting: The gritty details. In *Game Developers Conference*, March 2003.
- [6] D. M. Healy, D. N. Rockmore, and S. S. B. Moore. Ffts for the 2-sphere-improvements and variations. Technical report, Hanover, NH, USA, 1996.
- [7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92*, pages 71–78, 1992.
- [8] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH '00*, pages 279–286, 2000.
- [9] Z. Karni and C. Gotsman. 3d mesh compression using fixed spectral bases. In *GRIN'01*, pages 1–8, 2001.
- [10] M. Kazhdan. Reconstruction of solid models from oriented point sets. In *SGP '05*, pages 73–82, 2005.

- [11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03*, pages 156–164, 2003.
- [12] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Symmetry descriptors and 3d shape matching. In *SGP '04*, pages 116–125, 2004.
- [13] B. Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications*, 2006.
- [14] M. J. Mohlenkamp. A fast transform for spherical harmonics. *Journal of Fourier Analysis and Applications*, 5:159–184, 1999.
- [15] M. Mousa, R. Chaine, and S. Akkouche. Direct spherical harmonic transform of a triangulated mesh. *JGT: Journal of Graphics Tools*, 11(2):17–26, 2006.
- [16] M. Mousa, R. Chaine, and S. Akkouche. Frequency-based representation of 3d models using spherical harmonics. In *WSCG'06*, volume 14, pages 193–200, 2006.
- [17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.
- [18] M. Pauly and M. Gross. Spectral processing of point-sampled geometry. In *SIGGRAPH '01*, pages 379–386, 2001.
- [19] D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 392–397, 2001.
- [20] R. Suda and M. Takami. A fast spherical harmonics transform algorithm. *Math. Comput.*, 71(238):703–715, 2002.
- [21] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH '95*, pages 351–358, 1995.
- [22] I. Tosic and P. Frossard. Fst-based reconstruction of 3d-models from non-uniformly sampled datasets on the sphere. In *Proceedings of the Picture Coding Symposium*, 2006.
- [23] D. Vranic and D. Saupe. 3d shape descriptor based on 3d fourier transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*, pages 271–274, 2001.
- [24] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *ICIP '01*, pages 935–938, October 2001.
- [25] K. Zhou, H. Bao, and J. Shi. 3d surface filtering using spherical harmonics. *Computer-Aided Design*, 36(4):363–375, 2004.

A Integration by parts (1)

Let u and v two functions, the general form of the integration by parts states :

$$\int u dv = uv - \int v du$$

Taking $u = f_i(\theta, \varphi)$ and $dv = e^{-im\varphi} d\varphi$, we get :

$$\begin{aligned} du &= \frac{\partial f_i(\theta, \varphi)}{\partial \varphi} d\varphi \\ v &= \frac{-1}{im} e^{-im\varphi} \end{aligned}$$

Therefore, by substituting in equation 19, we get :

$$F_{\theta}^i = \frac{f_i k_{l,m} P_l^m(\cos \theta) \sin(\theta) e^{-im\varphi}}{im} - \frac{k_{l,m} P_l^m(\cos \theta) \sin(\theta)}{im} \int e^{-im\varphi} \frac{\partial f_i}{\partial \varphi} d\varphi$$

B Integration by parts (2)

$$I_1 = \int -\sin(\varphi) e^{-im\varphi} d\varphi \quad (45)$$

Taking $u = -\sin(\varphi)$ and $dv = e^{-im\varphi} d\varphi$, we get :

$$\begin{aligned} du &= -\cos(\varphi) d\varphi \\ v &= \frac{-1}{im} e^{-im\varphi} \end{aligned}$$

Therefore, by applying the integration by parts to equation 45, we get :

$$I_1 = \frac{\sin(\varphi) e^{-im\varphi}}{im} - \frac{1}{im} \int \cos(\varphi) e^{-im\varphi} d\varphi \quad (46)$$

Taking $u = \cos(\varphi)$ and $dv = e^{-im\varphi} d\varphi$, we get :

$$\begin{aligned} du &= -\sin(\varphi) d\varphi \\ v &= \frac{-1}{im} e^{-im\varphi} \end{aligned}$$

Again, by applying the integration by parts to equation 46, we get :

$$\begin{aligned} I_1 &= \frac{\sin(\varphi) e^{-im\varphi}}{im} - \frac{1}{im} \left[\frac{-\cos(\varphi) e^{-im\varphi}}{im} + \frac{1}{im} \int -\sin(\varphi) e^{-im\varphi} d\varphi \right] \\ &= \frac{\sin(\varphi) e^{-im\varphi}}{im} - \frac{\cos(\varphi) e^{-im\varphi}}{m^2} + \frac{1}{m^2} I_1 \end{aligned}$$

Therefore,

$$I_1 = \frac{e^{-im\varphi}}{1-m^2} [\cos(\varphi) + im \sin(\varphi)] \quad m \neq \pm 1 \quad (47)$$

While the intermediate steps to solve equation 45 requires a subdivision by m , the final solution of I_1 is still valid for $m = 0$. When $m = \pm 1$, we can rewrite equation 45 as follows :

$$\begin{aligned} I_1 &= \int -\sin(\varphi) e^{-im\varphi} d\varphi \\ &= \frac{1}{4} \cos(2\varphi) + \frac{im}{2} (\varphi - \frac{1}{2} \sin(2\varphi)) \end{aligned}$$

In a similar manner, we can find the analytical solution of the integration I_2 .