FST-based Reconstruction of 3D-models from Non-Uniformly Sampled Datasets on the Sphere*

Ivana Tosic and Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute
CH-1015 Lausanne

Abstract. This paper proposes a new method for reconstruction of star-shaped 3D surfaces from scattered datasets, where such surfaces are considered as signals living in the space of square integrable functions on the unit sphere. We first propose a generalization of the Fourier transform on the sphere. A practical reconstruction method is then presented, which interpolates a spherical signal on an equiangular grid, from non-uniformly sampled dataset representing a 3D point cloud. The experiments show that the proposed interpolation method results in smoother surfaces, and higher reconstruction PSNRs than the nearest neighbor interpolation method.

1 Introduction

Compression of 3D models represents an important and still open problem in communications, due to the high amount of data needed for their rendering. Moreover, transmission of this data over lossy networks requires multiresolutional representations of 3D models and scalable coding methods. An intuitive way to satisfy these requirements is to apply signal processing tools to a 3D model projected on an appropriate signal space. The most natural projection space for genus-zero closed 3D surfaces is certainly the sphere. Recent development of signal processing tools in the spherical framework, like the Fast Fourier Transform on the Sphere [1, 2] and spherical wavelets [3], opened a way to a new class of 3D model compression methods based on spherical parametrization. Most of these methods employ the spherical wavelet transform, like the Progressive Geometry Compression [4], or shape compression using spherical geometry images [5]. An efficient alternative for low bit-rate 3D coding is the Spherical Matching Pursuit based compression method [6]. However, mapping a 3D model on the signal space, while keeping the signal properties, is not a trivial task and it surely influences the compression efficiency.

In this paper, we propose a method that takes the 3D point cloud of a star-shaped[1] model and reconstructs the underlying spherical signal on the equiangular spherical grid, without cutting or unfolding the surface. We define this problem as an interpolation problem on the sphere from non-uniformly sampled datasets. Unfortunately, signal interpolation on the sphere has not been much investigated; some theoretical frameworks have been developed ([7, 8]), but none of these works actually consider the practical implementation issues of signal reconstruction from scattered data points. On the other hand, non-uniform sampling of signals on 2D plane has been quite deeply investigated. Gröchenig et al. [9] showed that a 2D signal can be reconstructed by a generalization of a Fourier expansion. Using a similar approach, we propose here a reconstruction method for signals defined on a unit sphere, based on the spherical Fourier transform. We show how a generalization of Fourier expansions can be applied on the sphere, and we propose a method for reconstruction of signals from non-uniformly sampled datasets.

This paper is organized as follows: in the Section 2 we introduce a new method for spherical interpolation from scattered data, while in the Section 3 we explain how the complexity of the proposed method can be reduced. Experimental results are given in the Section 4. With Section 5 we conclude the paper.

2 FST-based interpolation from non-uniformly sampled datasets

The signal on the unit sphere is a two-dimensional signal, described with two spherical coordinates θ

* This work has been supported by the Swiss National Science Foundation under grants 20001-107970/1 and PP002-68737.

[1] Star-shaped 3D models are those models for which every radial line originating from the center of the point cloud has only one intersection point with the surface of the model.

and φ, i.e., it is of a form $f(\theta, \varphi)$. Fast Spherical Fourier Transform (FST) [2] decomposes a signal that belongs to the Hilbert space of square-integrable functions on the two-dimensional sphere $L^2(S^2, d\omega)$ into a series of spherical harmonics:

$$f(\theta, \varphi) = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} \hat{f}(l, m) Y_l^m(\theta, \varphi), \tag{1}$$

where Fourier coefficients $\hat{f}(l, m)$ are given with:

$$\hat{f}(l, m) = \int_{S^2} f \bar{Y}_l^m(\theta, \varphi) d\omega, \tag{2}$$

and $d\omega(\theta, \varphi) = d\cos\theta d\varphi$ is the rotation invariant Lebesgue measure on the sphere.

Spherical harmonics of order (l, m), i.e., Y_l^m, are given with the following expression:

$$Y_l^m(\theta, \varphi) = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(cos\theta) e^{im\varphi}, \tag{3}$$

where $P_l^m(cos\theta)$ are the associated Legendre functions (see [1] for more details on their construction).

The sampling theorem for uniformly placed samples on the sphere has been established by Driscol and Healy [2]. It states that if a signal on the sphere is bandlimited, i.e., if $\hat{f}(l, m) = 0$ for $l \geq N$, then it can be perfectly recovered from uniformly sampled data $\theta_j = \pi j / 2N, \varphi = \pi k / N; j, k = 0, ..., 2N - 1$.

When the sampling is non-uniform, a similar framework can be used. Let P_M denote the space of polynomials on the sphere, given by :

$$p(\theta, \varphi) = \sum_{l=0}^{N-1} \sum_{|m| \leq l} a(l, m) Y_l^m(\theta, \varphi). \tag{4}$$

An arbitrary sampling problem can then be considered as a discretization of the here-above polynomials on the sphere, and one can thus reconstruct a bandlimited signal on the unit sphere from non-uniformly sampled data points $S = (r_j, \theta_j, \varphi_j), j = 1, ..., q$. Using the formula (4), for each sample $(r_j, \theta_j, \varphi_j), j = 1, ..., q$ we have:

$$r_j = f(\theta_j, \varphi_j) = \sum_{l=0}^{N-1} \sum_{|m| \leq l} a(l, m) Y_l^m(\theta_j, \varphi_j). \tag{5}$$

We can rewrite eq. (5) under a matrix form as:

$$V \cdot \mathbf{a} = \mathbf{r}, \tag{6}$$

with

$$V = \{Y_l^m(\theta_j, \varphi_j)\}_{q \times N^2} \tag{7}$$

$$\mathbf{a} = \{a(l, m)\}_{N^2 \times 1} \tag{8}$$

$$\mathbf{r} = \{r_j\}_{q \times 1}, \tag{9}$$

where the subscripts denote the size of these matrices.

By solving this linear system we obtain the values for coefficients $a(l, m)$, which are the approximates of Fourier coefficients for the observed signal on the sphere f:

$$\hat{f}(\theta, \varphi) \approx \mathbf{a} = V^{-1} \cdot \mathbf{r}. \tag{10}$$

Since V is not a square matrix, finding its inverse is not an easy task. The pseudo-inverse will give a minimal norm solution, resulting in a stable reconstruction, but it is computationally very expensive. Instead of directly solving eq. (6), we propose to multiply each side of the relation with $V^* = conj(V^T)$:

$$V^* \cdot V \cdot \mathbf{a} = V^* \cdot \mathbf{r}, \tag{11}$$

or equivalently:

$$T \cdot \mathbf{a} = \mathbf{R}, \tag{12}$$

with $T = V^* \cdot V$ and $\mathbf{R} = V^* \cdot \mathbf{r}$.

As the matrix T is a square matrix of size N^2, solving the system of eq. (12) instead of eq. (6) drastically decreases the computational complexity. Note that since the system (12) is complex, the total number of unknowns amounts to $2N^2$.

Finally, after the estimation of the Fourier coefficients, we can substitute them in the equation (1) and reconstruct the continuous function on the unit sphere.

3 Symmetry-based system complexity reduction

Due to the symmetric structure of associated Legendre functions, spherical harmonics satisfy the following property:

$$\overline{Y_l^m} = (-1)^m Y_l^{-m}. \tag{13}$$

Combining this property and the equation (2), we can express Fourier coefficients $\hat{f}(l, -m)$ as functions of $\hat{f}(l, m)$:

$$\hat{f}(l, -m) = (-1)^m \overline{\hat{f}(l, m)}, \tag{14}$$

having in mind that we are considering real functions on the sphere ($f(\theta, \varphi) \in \mathbb{R}$).

Since the solution of (12) is an approximated vector of Fourier coefficients, its values will satisfy a similar relation, i.e.:

$$a(l,-m) = (-1)^m \overline{a(l,m)}. \quad (15)$$

Thus, we can rewrite the equation (5), for each sample $j = 1,...,q$, in the following form:

$$\begin{aligned} r_j &= \sum_{l=0}^{N-1} \{a_l^0 Y_l^0(j) + \sum_{m=1}^{l} [a_l^m Y_l^m(j) \\ &+ (-1)^m \overline{a_l^m} (-1)^m \overline{Y_l^m}(j)]\} \\ &= \sum_{l=0}^{N-1} \{a_l^0 Y_l^0(j) + 2 \sum_{m=1}^{l} [Re\{a_l^m\} Re\{Y_l^m(j)\} \\ &+ Im\{a_l^m\} Im\{Y_l^m(j)\}]\}, \end{aligned} \quad (16)$$

where $a_l^m \equiv a(l,m)$ and $Y_l^m(j) \equiv Y_l^m(\theta_j, \phi_j)$, and Re and Im denote the real and the imaginary part of a complex number, respectively.

Consider further a new matrix V_r obtained from the matrix V by replacing the spherical harmonics $Y_l^m(j)$ for $m = -l,...,-1$ with $2Im\{Y_l^m(j)\}, m = 1,...,l$ and similarly, replacing $Y_l^m(j), m = 1,...,l$ with $2Re\{Y_l^m(j)\}, m = 1,...,l$; for $\forall j$. So, the obtained matrix V_r is of the following form:

$$V_r = \begin{bmatrix} Y_0^0(1) & 2Im\{Y_1^1(1)\} & Y_1^0(1) & 2Re\{Y_1^1(1)\} & ... \\ Y_0^0(2) & 2Im\{Y_1^1(2)\} & Y_1^0(2) & 2Re\{Y_1^1(2)\} & ... \\ \vdots & & & & \\ Y_0^0(q) & 2Im\{Y_1^1(q)\} & Y_1^0(q) & 2Re\{Y_1^1(q)\} & ... \end{bmatrix}$$

By solving the following real-valued linear system of equations:

$$V_r \cdot \mathbf{a}_r = \mathbf{r}, \quad (17)$$

we obtain a vector $\mathbf{a}_r$, which is a modification of the vector $\mathbf{a}$ in (5), where coefficients $a(l,m)$ for $m = -l,...,-1$ are substituted with $Im\{a_m^l\}, m = 1,...,l$ and coefficients $a(l,m)$ for $m = 1,...,l$ are substituted with $Re\{a_m^l\}, m = 1,...,l$; i.e., it is of the form:

$$[\, a_0^0 \quad Im\{a_1^1\} \quad a_1^0 \quad Re\{a_1^1\} \quad ... \,]. \quad (18)$$

The equivalence of systems (5) and (17) follows directly from the relation (16). Therefore, we can solve the real system (17), instead of solving the complex system (5), and thus reduce the number of unknown variables from $2N^2$ to N^2. Similarly to (11)-(12), we propose to multiply the system (17) with V_r^T and further reduce the problem to the square real system:

$$T_r \cdot \mathbf{a}_r = \mathbf{R}_r, \quad (19)$$

with $T_r = V_r^T \cdot V_r$ and $\mathbf{R}_r = V^T \cdot \mathbf{r}$.

Finally, from the obtained vector $\mathbf{a}_r$ we can calculate the complex values $a_l^m, l = 0,...,N-1; m = 1,...,l$ as:

$$a_l^m = Re\{a_l^m\} + iIm\{a_l^m\}. \quad (20)$$

The rest of the values a_l^m for $m = -l,...,-1$ follow from (15).

4 Experimental results

4.1 Preliminaries

Implementation A given 3D point cloud is first mapped to spherical coordinates with the center in the point mass, thus obtaining non-uniformly sampled data on the sphere. Afterwards, the FST-based interpolation scheme is used to reconstruct the underlying spherical signal, evaluated on the equiangular spherical grid. This grid type is defined as follows:

$$\mathcal{G} = \{(\theta_j, \varphi_k) \in S^2 : \theta_j = \tfrac{(2j+1)\pi}{4N}, \varphi_k = \tfrac{k\pi}{N}\}, \quad (21)$$

with $j, k \in \mathcal{N} \equiv \{n \in \mathbb{N} : n < 2N\}$.

Rendering The interpolated 3D point cloud, defined as a spherical function, can be also adapted to suite the existing rendering techniques. For this purpose, the connectivity information of the obtained model has to be defined. Using the fact that interpolated points lie on the equiangular spherical grid, we can generate the connectivity by dividing the spherical grid into rings limited with two successive values of θ, and then triangulate each ring to produce a triangular strip.

4.2 Numerical results

The performance of the proposed scheme has been evaluated on two datasets: the point cloud of 16258 points corresponding to the geometry information of the Venus model, and the point cloud of 47763 points representing the Male model[2]. These two point clouds are shown on the Figures 1 and 2 for Venus and Male models, respectively.

Figures 3a and 4a display the reconstruction of two observed models, using the FST-based interpolation method on an equiangular grid, with $N =$

[2] both models were downloaded from the Cyberware models database (http://www.cyberware.com/samples/)

64. Reconstructed models were rendered using the method described in 4.1. For comparison, on the Figures 3b and 4b we have shown these two models reconstructed using the nearest neighbor method with the same resolution. Value of the spherical function at each point in the equiangular grid is interpolated by averaging 4 nearest neighbors. As mentioned in the introduction, no other practical interpolation methods for spherical data exist, thus we constrain our comparison to this simple nearest neighbor method. We can see that in the in the case of the proposed FST based method, visual quality of the reconstruction is better and the object is smoother. This is because the FST based method gives priority to the lower frequencies of the signal ($l < N$), while it cuts off the high frequencies ($l \geq N$), unlike nearest neighbor method which does not take into account the frequency characteristics of the signal. As a result, models reconstructed using the nearest neighbor method have unpleasant aliasing artifacts. Nevertheless, these artifacts can be removed using a posteriori filtering, but at the expense of smoothing other model features as well. On the Figures 3c) and 4c) we have shown the reconstructed models after nearest neighbor interpolation and LF filtering, where we used the spherical low-pass filter with a cutoff frequency N=64 (for details on this filter see [10]). The quantitative comparison of these three methods is given in Table 1 by means of the PSNR, which is simply given by 20 $log\left(\frac{1}{L^2}\right)$, where relative L^2 error is a ratio of RMS - Root Mean Square Error (that measures the squared symmetric distance between two surfaces averaged over the first surface) relative to a bounding box diagonal. We can see that the PSNR obtained by the FST scheme is 1.3dB (Venus) and 3.2dB (Male) higher than the PSNR for the nearest neighbor scheme. Moreover, we notice that the LF filtering after the nearest neighbor interpolation drastically decreases the PSNR value.

Table 1. PSNR comparison of FST-based and Nearest neighbor methods

	PSNR[dB]	
	Venus	Male
FST	63.3593	61.5904
Nearest neighbor	62.0576	58.3444
Nearest neighbor & filtering	57.3411	51.1856

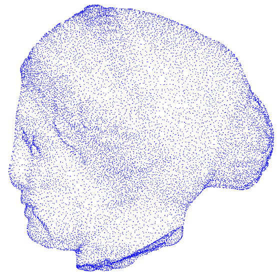
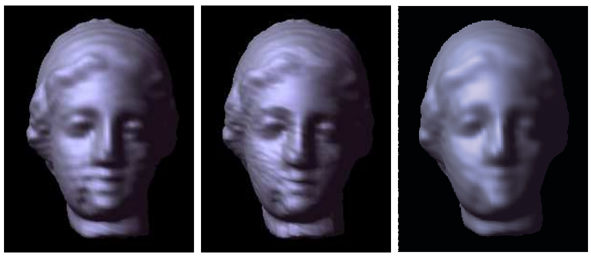


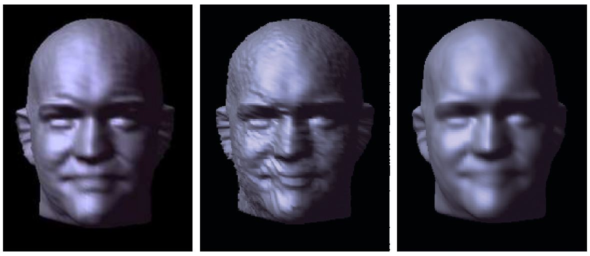
Fig. 1. Point Cloud of the model Venus

Fig. 2. Point Cloud of the model Male



(a) **(b)** **(c)**

Fig. 3. a) Venus reconstructed using the FST method, with resolution 128x128 b) Venus reconstructed using the nearest neighbor method, with resolution 128x128 c) Filtered model of Venus from b)



(a) **(b)** **(c)**

Fig. 4. a) Male reconstructed using the FST method, with resolution 128x128 b) Male reconstructed using the nearest neighbor method, with resolution 128x128
c) Filtered model of Male from b)

5 Discussion and conclusions

In this paper we propose a method for reconstruction of star-shaped 3D models as functions defined on a unit sphere, from 3D point clouds. The reconstruction problem is defined as an interpolation problem from non-uniformly sampled data on the sphere. Therefore, a practical method for reconstruction of spherical signals from non-uniform samples is developed, which uses a spherical harmonics base to evaluate the estimates of the Fourier coefficients on the sphere, and then reconstructs a signal using an inverse FST. The complexity of the proposed method is linear with respect to the number of scattered points, so the size of the point cloud does not influence the algorithm efficiency. However, the reconstruction quality increases with the density of the point cloud and with the number of estimated Fourier coefficients, i.e. with the desired resolution N. We have shown that the reconstruction quality of the FST-based method outperforms the nearest neighbor method, both visually and quantitatively.

Finally, the proposed reconstruction of 3D models as functions on the sphere offers a lot of flexibility and freedom in design and implementation of arbitrary 3D data processing tasks. Therefore, we believe that the developed FST-based 3D model reconstruction method could be of great benefit to applications like: various 3D model deformations, representation of 3D models in a multiresolution fashion, efficient compression of 3D models, etc.

References

1. J.R. Driscoll and D. Healy: Computing Fourier transforms and convolutions on the 2-sphere. Adv. in Appl. Math **15** (1994) 202–250
2. J.R. Driscoll and D.M. Healy, Jr.: Asymptotically fast algorithms for spherical and related transforms. 30th Annual Symposium on Foundations of Computer Science (1989) 344 – 349
3. J-P. Antoine and P. Vandergheynst: Wavelets on the 2-sphere : a group theoretical approach. Applied and Computational Harmonic Analysis **7** (1999) 1–30
4. A. Khodakovsky, P. Schröder and W. Sweldens: Progressive geometry compression. Siggraph '00 Conference Proceedings (2000)
5. H. Hoppe and E. Praun: Shape compression using spherical geometry images. Symposium on Multiresolution in Geometric Modeling. Cambridge (2003)
6. I. Tosic, P. Frossard and P. Vandergheynst: Progressive coding of 3d objects based on overcomplete decompositions. EPFL-ITS Technical Report TR-ITS-2005.026, 1015 Ecublens (2005)
7. F. Narcowich and J.D. Ward: Scattered data interpolation on spheres: error estimates and locally supported basis functions. SIAM Journal on Mathematical Analysis **33** (2002) 1393–1410
8. S. Hubbert and T.M. Morton: L_p-error estimates for radial basis function interpolation on the sphere. Journal of Approximation Theory (2004) 58–77
9. F. Marvasti: 6. In: Nonuniform sampling: Theory and Practice. Kluwer Academic/Plenum Publishers (2000) 284–290
10. I. Tosic, I. Bogdanova, P. Frossard and P. Vandergheynst: Multiresolution Motion Estimation for Omnidirectional Images. In: Proceedings of EUSIPCO. (2005)