Poster:Continuous FVR of Irregularly Sampled Data Using Gaussian RBFs

H.Quynh Dinh[*]
Department of Computer Science
Stevens Institute of Technology

Neophytos Neophytou[†]
Department of Computer Science
Stony Brook University

Klaus Mueller[‡]
Department of Computer Science
Stony Brook University

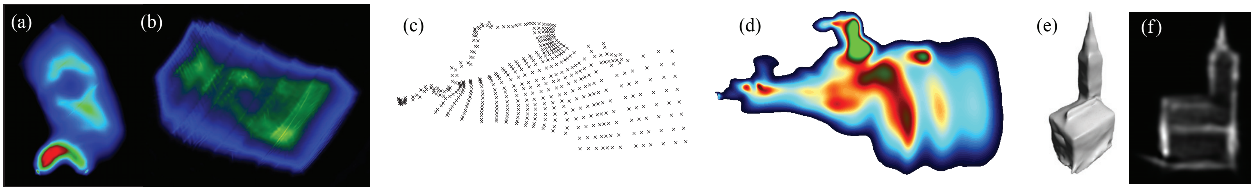

Figure 1: Continuous FVR of irregularly sampled datasets: (a) SPX, (b) Combustion Engine, (c) and (d) computational fluid dynamics (CFD) particle concentrations, (e) and (f) Ghiradelli Square. Input data for CFD data and Ghiradelli Square are shown to the left of their X-rays.

Abstract

We describe a Fourier Volume Rendering (FVR) algorithm for datasets that are irregularly sampled and require anisotropic (*e.g.*, elliptical) kernels for reconstruction. We sample the continuous frequency spectrum of such datasets by computing the continuous Fourier transform of the spatial interpolation kernel which is a radially symmetric Gaussian basis function (RBF) that may be anisotropically scaled. While in the frequency domain, we can apply signal processing filters to the dataset before performing an inverse 2D Fourier transform to obtain the X-ray projection.

Index Terms: I.3.3 [Computing Methodologies]: Picture/Image Generation —Display algorithms

1 Introduction

Fourier Volume Rendering (FVR) was presented separately by Dunne [4], Levoy [6], and Malzbender [7]. FVR generates an X-ray projection (summed volume rendering) of a 3D volume using the Fourier Projection-Slice Theorem. The theorem states that a 2D slice passing through the origin of the 3D Fourier transform is the 2D Fourier transform of the projection of the 3D space in the direction orthogonal to the slice. In conventional FVR, the 3D Fourier Transform of the volume is computed in a pre-processing step via the discrete FFT. Interpolating a 2D slice and transforming it back to the spatial domain via an inverse 2D DFT is an $O(I^{2} log I)$ operation that requires careful design of a frequency domain interpolation kernel to avoid ghosting and attenuation in the spatial domain [7]. The Non-uniform Discrete Fourier Transform (NDFT) extends the Fourier transform to irregularly sampled data, but with limits on the degree of irregularity [8]. [9] uses the NDFT to regrid originally irregular datasets into a regular domain but does not anisotropically scale the kernels, resulting in gaps in the X-rays.

Our contribution is an algorithm that samples at arbitrary resolution the continuous frequency spectrum of irregularly sampled datasets. Instead of using the discrete Fourier transform, we sample the continuous Fourier transform of the spatial interpolation kernel – a Gaussian radial basis function (RBF) which may be anisotropically scaled and located at irregular spatial intervals. We then use the inverse 2D DFT to obtain the X-ray image. By doing so, we defer discretization which, as suggested in [1], can reduce propagation of the associated errors in the visualization pipeline. We call this approach *continuous Fourier Volume Rendering*. With a continuous frequency spectrum, our FVR algorithm does not need to interpolate discrete frequency samples or deal with aliasing in the frequency domain. We describe how to properly sample the frequency spectrum of anisotropic RBFs to avoid aliasing and present an optimal GPU algorithm for FVR of irregularly sampled datasets.

1.1 Computing the Continuous Frequency Spectrum

A summation of weighted RBFs has been used as a data interpolant in many domains, including volume rendering:

$$f(\vec{x}) = \sum_{i=1}^{n} w_i \phi_i(\vec{x} - \vec{c}_i) \qquad (1)$$

In the above equation, ϕ_i are the interpolation kernels; n is the number of scattered samples; c_i are the data points (centers of the kernels); and w_i are densities associated with each data point. The continuous Fourier transform of the above, as derived in [2], is:

$$F(\omega) = \sum_{i=1}^{n} w_i e^{-j2\pi\omega\vec{c}_i} \Phi_i(|\omega|) \qquad (2)$$

$\Phi_i(|\omega|)$ is the Fourier transform of ϕ_i, and $e^{-j2\pi\omega\vec{c}_i}$ is due to applying the Fourier Shift Theorem to RBFs centered at $\vec{c}_i$. The RBF we use is the Gaussian kernel used in splatting ($\phi(r) = 2^{-r^2}$) whose generalized Fourier Transform is: $\frac{1}{\ln^{\frac{3}{2}} 2} \pi^{\frac{3}{2}} e^{-\pi^2 \frac{r^2}{\ln 2}}$.

For regularly sampled datasets, ϕ_i is isotropic and identical (homogeneous) for all data points. The i subscript can be dropped and $\Phi(|\omega|)$ can be pulled outside of the summation. In this case, $\Phi(|\omega|)$ is a radially symmetric function and essentially behaves as a low-pass filter. In a more general approach, we allow ϕ_i to be rotated and scaled uniquely and anisotropically for each data point. Anisotropic scaling may be the result of sampling data on curvilinear or non-uniform rectilinear grids or from kernel fitting optimization techniques [3, 5] that incorporate more advanced data-sensitive constraints. To compute the frequency spectrum for anisotropic RBFs, we apply the Fourier Scaling Theorem.

2 Frequency Spectrum of Anisotropic RBFs

We must address two key issues to achieve continuous FVR on the general anisotropic formulation: (1) define the sampling requirements in frequency space to ensure that high frequency information is retained while preventing aliasing in the spatial domain (resulting X-ray), and (2) optimize sampling of the frequency spectrum

[*] e-mail: quynh@cs.stevens.edu
[†] e-mail:nneophyt@gmail.com
[‡] e-mail:mueller@cs.stoneybrook.edu

since $\Phi_i(|\omega|)$ cannot be pulled out of the equation. The minimum number of frequency samples Ị depends on the period size T and the highest frequency component ω_{max} which is inversely related to the smallest spatial interval t ($\omega_{max} = 1/t$). Thus, the number of required freqency samples is Ị $= T/t$.

For anisotropic RBFs, the required sampling density depends on RBF scaling. When the scaling is less than 1 (compressed RBFs), the data points are, in effect, brought closer together. The minimum spatial interval is further reduced, requiring a denser sampling of the frequency spectrum: Ị $= T/(s*t)$ and $\omega_{max} = 1/(s*t)$ where s is the minimum RBF scaling. As a result, Ị and ω_{max} increase as s approaches 0. Most datasets do not exhibit excessive compression, and Ị and ω_{max} do not become prohibitively large.

To efficiently compute the frequency spectrum using the GPU, we precompute Φ and store it in a texture. We then compute a slice of the frequency spectrum by rendering a quad to accumulate the shift factor $e^{-j2\pi\omega\vec{c}_i}$ on a per-fragment basis. To further optimize the GPU implementation, we can rotate and anisotropically scale the texture coordinates used to texture map the RBF footprint in the CPU. This reduces per-fragment computations and relies on the GPU's linear interpolation of the texture coordinates to sample the RBF footprint rather than computing the rotation and scaling in the fragment shader. The optimization does reduce the computation time for the frequency spectrum as shown in Figure 3, but some blurring is apparent in the resulting X-rays due to the interpolation.

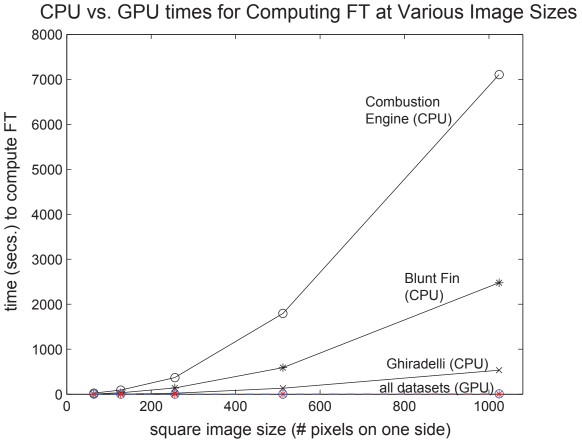

Figure 2: CPU and GPU times for computing a 2D frequency spectrum of anisotropic datasets. All GPU times are under 10 secs.

3 Results

We apply continuous FVR to three types of irregularly sampled datasets. The SPX (4,011 pts, Fig. 1a), Blunt Fin (15,256 pts), and Combustion Engine (46,805 pts, Figs. 1b and 4) datasets were from fitting elliptical basis functions to a local Delauney triangulation of the dataset [5]. The CFD data (3,652 pts, Figs. 1c and 1d) is a volume of particle concentrations of the New York harbor sampled on a curvilinear grid. The Ghiradelli Square dataset (2,762 pts, Figs. 1e and 1f) came from anisotropic RBF fitting [3]. The colors in the X-rays are generated by applying a transfer function to the resulting 2D X-ray images, not to the original volumetric datasets.

3.1 X-rays and Timing Results

Visually, our X-ray results are similar to splatting and more continuous than Non-uniform DFT. An advantage of our algorithm is that it is highly amenable to computation on the GPU (GPU times are orders of magnitude faster than CPU times). In Figure 2, we compare timing results for continuous FVR on the CPU (Intel 2 Quad 2.4 GHz) versus the GPU (Nvidia GeForce 8800 GT).

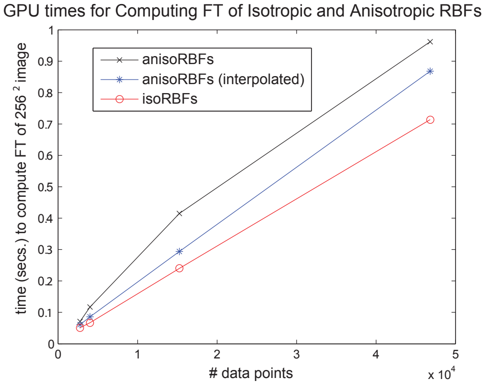

Figure 3: Timing results for different GPU algorithms.

3.2 Filtering, Zooming, and Frequency-sensitive X-rays

We can apply low, band, and high-pass filters to the frequency spectrum before restoring the data to the spatial domain via the inverse DFT. These operations are implemented via a multiplication of the spectrum with a box filter which retains the desired frequencies and zeros out the remaining frequencies. Depending on the filtering or zooming operation, this can be achieved on the graphics hardware by scaling up or down the texture coordinates of the RBF spectral footprint or the quadrilateral to which the texture is mapped.

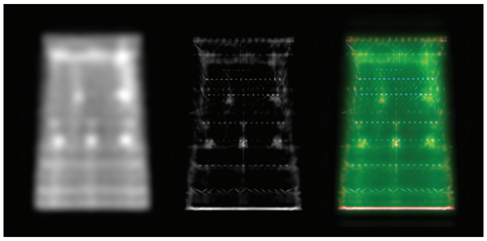

Figure 4: Left and center: Low and high-pass filtering of the Combustion Engine. Far right: Color is applied based on frequency band instead of opacity for a frequency-sensitive transfer function.

With a frequency spectrum, we are able to generate *frequency-sensitive* X-rays by applying a transfer function based on frequency bands rather than opacity values. We do so by transforming each band back into the spatial domain via the inverse DFT, applying a transfer function to the resulting filtered X-ray, and blending all X-ray bands into a final image as shown in Figure 4.

References

[1] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.

[2] M. Buhmann. *Radial Basis Functions*, 2003.

[3] H. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces using anisotropic basis functions. *IEEE ICCV*, pages 606–613, 2001.

[4] S. Dunne, S. Napel, and B. Rutt. Fast reprojection of volume data. *1st Conf on Visualization in Biomedical Computing*, pages 11–18, 1990.

[5] W. Hong, N. Neophytou, K. Mueller, and A. Kaufman. Constructing 3d elliptical gaussians for irregular data. *Math Found. of Sci. Visualization, Computer Graphics, and Massive Data Exploration*, 2006.

[6] M. Levoy. Volume rendering using the fourier projection-slice theorem. *ACM SIGGRAPH*, pages 61–69, 1992.

[7] T. Malzbender. Fourier volume rendering. *ACM Trans. on Graphics (TOG)*, 12(3):233–250, 1993.

[8] F. Marvasti. *Ị onuniform Sampling: Theory and Practice*, 2001.

[9] P. Stark. Fourier volume rendering of irregular data sets. *MS Thesis, Simon Fraser University*, 2002.