Towards High Quality Gradient Estimation on Regular Lattices

Zahid Hossain* Usman R. Alim† Torsten Möller‡

Graphics, Usability, and Visualization Laboratory
School of Computing Science
Simon Fraser University

Abstract

We present two methods for accurate gradient estimation from scalar field data sampled on regular lattices. The first method is based on the multi-dimensional Taylor series expansion of the convolution sum and allows us to specify design criteria such as compactness and approximation power. The second method is based on a Hilbert space framework and provides a minimum error solution in the form of an orthogonal projection operating between two approximation spaces. Both methods lead to discrete filters which can be combined with continuous reconstruction kernels to yield highly accurate estimators as compared to the current state of the art. We demonstrate the advantages of our methods in the context of volume rendering of data sampled on Cartesian and Body-Centered Cubic lattices. Our results show significant qualitative and quantitative improvements for both synthetic and real data, while incurring a moderate preprocessing and storage overhead.

1 Introduction

Volumetric data, typically given on a discrete lattice, is perceived as a continuous data type and therefore requires the algorithms working on them to model the data as if it were given in a continuous domain. Hence, interpolation and reconstruction are the key aspects of any volumetric manipulation and have a tremendous impact on the quality and efficiency of the underlying visualization task. While there has been a large body of work on interpolation and reconstruction filter design, in many tasks we also need secondary information of the volumetric data such as histograms for data exploration or gradients for shading. One could simply take an interpolation filter and consider its analytical derivative as a proper derivative filter. However, this approach is not only computationally inefficient since it requires additional interpolations to be performed, it also unnecessarily constrains the conditions on accuracy and smoothness. Hence, a separate design of gradient estimation schemes can lead to much better results.

In this paper, we shall consider the design of gradient estimation schemes for data sampled on a regular lattice. In particular, we consider two competing designs. On the one hand, we explore a design based on a Taylor series expansion which leads to computationally efficient discrete derivative kernels that can be used on the fly without pre-computing gradients in a gradient volume. On the other hand, we consider the idea of prefilters which typically yield tremendous improvement of image quality, but have to be applied in a pre-processing step and need to be stored in a gradient volume in order to be computationally feasible.

*e-mail: zha13@cs.sfu.ca
†e-mail: ualim@cs.sfu.ca
‡e-mail: torsten@cs.sfu.ca

2 Gradient Estimation Schemes

2.1 Taylor Series Approach

This approach is motivated by the 1D analysis of Möller et al. [1,2]. Particularly, we extend their analysis to multiple dimensions in order to facilitate the design of derivative filters for arbitrary lattices.

Let $\mathcal{L}$ denote the d-dimensional lattice generated by the matrix $\boldsymbol{L}$. We can decompose a multidimensional function f at a lattice site, $\boldsymbol{Lk}$ ($\boldsymbol{k} \in \mathbb{Z}^d$), about $\boldsymbol{x} \in \mathbb{R}^d$ into a Taylor series as

$$f(\boldsymbol{Lk}) = \sum\nolimits_{\boldsymbol{n}} \frac{(\boldsymbol{Lk}-\boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} D^{\boldsymbol{n}} f(\boldsymbol{x}). \tag{1}$$

where $\boldsymbol{n} \in \mathbb{N}^d$ and $D^{\boldsymbol{n}}$ is a cascaded partial differential operator defined as $D^{\boldsymbol{n}}(\cdot) := \left(\frac{\partial^{n_1}}{\partial {x_1}^{n_1}} \frac{\partial^{n_2}}{\partial {x_2}^{n_2}} \cdots \frac{\partial^{n_d}}{\partial {x_d}^{n_d}}\right)(\cdot)$. Vector factorial and vector exponent have the usual multi-index interpretation, i.e. $\boldsymbol{n}! := \prod_{k=1}^{d} n_k!$, and $\boldsymbol{v}^{\boldsymbol{n}} := \prod_{k=1}^{d} v_k^{n_k}$. The usual convolution representation of the function reconstructed from its samples is defined as

$$f_r^w(\boldsymbol{x}) := \sum\nolimits_{\boldsymbol{k}} f(h\boldsymbol{Lk}) \cdot w(\frac{\boldsymbol{x}-h\boldsymbol{Lk}}{h}), \tag{2}$$

where h is a scalar parameter that controls the sampling rate and w is a reconstruction filter defined for $\mathcal{L}$. Substituting (1) into (2) we obtain

$$f_r^w(\boldsymbol{x}) = \sum\nolimits_{\boldsymbol{n}} D^{\boldsymbol{n}} f(\boldsymbol{x}) \cdot a_{\boldsymbol{n}}^w(\boldsymbol{x}), \tag{3}$$

where $a_{\boldsymbol{n}}^w(\boldsymbol{x})$, hereinafter referred to as Taylor-coefficient, is defined as

$$a_{\boldsymbol{n}}^w(\boldsymbol{x}) := \sum\nolimits_{\boldsymbol{k}} \frac{(h\boldsymbol{Lk}-\boldsymbol{x})^{\boldsymbol{n}}}{\boldsymbol{n}!} \cdot w(\frac{\boldsymbol{x}-h\boldsymbol{Lk}}{h}). \tag{4}$$

This equation forms the basis of our filter design and classification approach. As argued by Möller et al. [2], using the analytic derivative of the interpolation filter to estimate gradients may not always be superior to using a combination of a discrete derivative filter and a continuous interpolation filter. Therefore, we seek to use the above equation to design discrete derivative filters. This is tantamount to setting $\boldsymbol{x} = \boldsymbol{0}$, thus yielding

$$a_{\boldsymbol{n}}^{\Delta} = \frac{h^{\boldsymbol{n}}}{\boldsymbol{n}!} \sum\nolimits_{\boldsymbol{k}} (\boldsymbol{Lk})^{\boldsymbol{n}} \cdot \Delta\left(\boldsymbol{L}\left(-\boldsymbol{k}\right)\right), \tag{5}$$

where Δ denotes a discrete filter defined on the lattice sites. This equation forms a system of linear equations where we set values for $a_{\boldsymbol{n}}^{\Delta}$ a priori and seek to find the unknowns $\Delta\left(\boldsymbol{L}\left(-\boldsymbol{k}\right)\right)$. The number of equations is determined by how many $a_{\boldsymbol{n}}^{\Delta}$ are fixed, which in turn is governed by the approximation order of the filter. The number of unknowns is given by how many different $\boldsymbol{k}$ vectors, i.e. neighbourhood of the filter, we want to restrict the filter to. This parameter therefore decides the support of the discrete filter. For instance, for a fourth order discrete derivative filter along x on the BCC lattice, we need to impose the conditions,

$$a_{\boldsymbol{n}}^{\Delta} = 0, \forall \boldsymbol{n} \in \{\boldsymbol{m} \in \mathbb{N}^3 : \|\boldsymbol{m}\|_1 \leq 4 \text{ and } \boldsymbol{m} \neq [1,0,0]\}$$

$$a_{\boldsymbol{n}}^{\Delta} = 1, \boldsymbol{n} = [1,0,0]$$

and solve the resulting system of linear equations. The resulting filter can further be optimized by reducing the error in the 5th polynomial order. Likewise filters of varying orders for different lattice types can be similarly obtained. Some examples are provided in the accompanying poster.

2.2 Orthogonal Projections

A very useful gradient estimation strategy comes from the Hilbert-space formulation of signal processing [3]. We shall restrict attention to real-valued functions that reside in the Hilbert space $L_2(\mathbb{R}^3)$ where we denote the inner product between two functions f and g as $\langle f, g\rangle := \int_{\mathbb{R}^3} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}$.

In visualization, we are typically interested in the Hilbert space formed by the shifts, on a regular lattice, of a generating function (reconstruction filter). A function is approximated by projecting it onto this space. The quality of the approximation is governed by the type of projection. An orthogonal projection gives a minimum error solution but can be computationally more expensive. On the other hand, an oblique projection is computationally faster but is not as accurate. This formulation encompasses the traditional Shannon's view as well. In particular, when the generating function is the *sinus cardinalis*, an orthogonal projection is equivalent to applying an ideal prefilter before sampling.

In order to find derivative filters that can be combined with a continuous generating function, we first approximate a given sampled sequence in an intermediate approximation space and then orthogonally project the gradient of the approximation to the target approximation space, the space in which the generating function lies. Let $f[\boldsymbol{k}] = f(h\boldsymbol{L}\boldsymbol{k})$ denote the given sampled sequence. We can express the first-stage approximation of f as

$$f_1(\boldsymbol{x}) = \sum\nolimits_{\boldsymbol{k}} c_1[\boldsymbol{k}]\psi_{h,\boldsymbol{k}}(\boldsymbol{x}) = \sum\nolimits_{\boldsymbol{k}} (f * p_1)\,[\boldsymbol{k}]\,\psi_{h,\boldsymbol{k}}(\boldsymbol{x}), \quad (6)$$

where $*$, in this context, denotes the discrete convolution operation. As can be seen from the above equation, the first-stage approximation space is spanned by the dilated and shifted versions $\psi_{h,\boldsymbol{k}}(\boldsymbol{x})$ of the generator function $\psi(\boldsymbol{x})$ (in particular, $\psi_{h,\boldsymbol{k}}(\boldsymbol{x}) = \psi(\frac{\boldsymbol{x}}{h} - \boldsymbol{L}\boldsymbol{k})$ where $\boldsymbol{k} \in \mathbb{Z}^3$). The coefficient sequence c_1 is obtained from the sampled sequence f through the application of a suitable digital prefilter p_1. This prefilter ensures that the first stage approximation exactly interpolates the function values at the lattice sites. This prefiltering operation can be efficiently performed in the Fourier-domain [3].

In order to approximate the gradient of the original function $f(\boldsymbol{x})$, we take the gradient of the first-stage approximation f_1 and orthogonally project it onto the space spanned by the dilated and shifted versions of the target generating function $\varphi(\boldsymbol{x})$. This is tantamount to performing three orthogonal projections, one for each component of the gradient. Let $\partial_i f$ denote the partial derivative $\frac{\partial f}{\partial x_i}$, $i \in \{1, 2, 3\}$. The orthogonal projection is equivalent to computing the inner product of $\partial_i f_1$ with the bi-orthogonal dual of $\varphi_{h,\boldsymbol{k}}$ [3]. After expanding $\partial_i f_1$ in terms of its first-stage approximation, the resulting second-stage approximation can be written as

$$\begin{aligned} f_{2,i}(\boldsymbol{x}) &:= \sum\nolimits_{\boldsymbol{k}} \langle \partial_i f_1, \mathring{\varphi}_{h,\boldsymbol{k}}\rangle \varphi_{h,\boldsymbol{k}}(\boldsymbol{x}) \\ &= \sum\nolimits_{\boldsymbol{k},\boldsymbol{m}} c_1[\boldsymbol{m}]\langle \partial_i \psi_{h,\boldsymbol{m}}, \mathring{\varphi}_{h,\boldsymbol{k}}\rangle \varphi_{h,\boldsymbol{k}}(\boldsymbol{x}) \\ &= \sum\nolimits_{\boldsymbol{k},\boldsymbol{m}} c_1[\boldsymbol{m}]\langle \partial_i \psi_{h}, \mathring{\varphi}_{h,\boldsymbol{k}-\boldsymbol{m}}\rangle \varphi_{h,\boldsymbol{k}}(\boldsymbol{x}) \\ &= \sum\nolimits_{\boldsymbol{k}} (c_1 * \mathring{d}_i)[\boldsymbol{k}]\varphi_{h,\boldsymbol{k}}(\boldsymbol{x}), \end{aligned} \quad (7)$$

where $\mathring{d}_i$ is a digital derivative filter given by the inner product $\mathring{d}_i[\boldsymbol{n}] := \langle \partial_i \psi_h, \mathring{\varphi}_{h,\boldsymbol{n}}\rangle$. The basis function $\mathring{\varphi}$ represents the bi-orthogonal dual of φ and satisfies the relationship $\langle \mathring{\varphi}_{h,\boldsymbol{k}}, \varphi_{h,\boldsymbol{l}}\rangle = \delta_{\boldsymbol{k}-\boldsymbol{l}}$.

In order to have faithful approximations, ψ should be chosen so that it has a higher approximation order as compared to φ. On the other hand, when visual quality and efficiency are important, ψ can be chosen so that it has comparable smoothness properties. Once ψ and φ have been chosen, the remaining key step in the above scheme is the evaluation of the inner product (**??**) that yields the discrete derivative filter $\mathring{d}_i$. We have used the tensor-product B-splines on the CC lattice as well as four-directional box-splines on the BCC lattice to design derivative filters that implement the above two-stage approximation scheme. Examples of our filters are provided in the accompanying poster.

3 Results and Conclusion

We evaluated the performance of our filters by volume rendering shaded isosurface images of both synthetic and real volumetric data. We kept the underlying scalar data interpolation the same (tricubic B-spline on CC and quintic box-spline on BCC) and investigated how the quality of the images changes with different gradient filters.

The gradient filters derived according to the Taylor-series recipe have compact support. We implemented these filters with a straightforward discrete convolution in a pre-processing step. However, these filters are sufficiently compact so that, when storage is limited, on-the-fly computation is also feasible. On the other hand, the orthogonal projection fitlers have comparatively large support and need to be combined with an all-poles prefilter. To efficiently implement these filters, we employed the multi-dimensional discrete Fourier transform in a pre-processing step and stored the result in a gradient volume. Gradients were then estimated by simply interpolating the gradient volume.

Our tests on synthetic data indicate the superiority of the BCC lattice, the 2nd and 4th order BCC Taylor filters clearly outperform the corresponding CC Taylor filters (2nd and 4th order central differencing respectively) both qualitatively and quantitatively. We observed some rippling artifacts when estimating the derivative on the BCC lattice by locally computing the derivative of the interpolation kernel. These artifacts were not observed for CC. The higher order orthogonal projection filters showed the best results overall with significant quantitative and qualitative improvements over their Taylor counterparts. However, these improvements come at a significant storage overhead.

We also rendered isosurface images of the carp and bunny data sets obtained through CT scans. We observed that the 4th order Taylor filter on CC shows some slight detail enhancement as compared to the 2nd order Taylor filter, specially in high frequency regions. With the orthogonal projection filter, image quality is appreciably enhanced with significant improvements in image contrast and detail.

We have presented an overview of two gradient estimation methods. We believe that these methods have a broad range of applicability as they give the practitioner the flexibility to suit their needs. We extended the 1D Taylor series framework to multiple dimensions and used it to design compact filters that can be computed on the fly. We also considered the idea of prefilters and derived high quality filters using tensor-product B-splines on CC and box splines on BCC. Our results reveal that, when accuracy and quality are crucial, a filter based on the Hilbert space framework should be employed to appreciably improve image quality at a price of increased storage overhead. We plan to carry out more extensive tests to carefully assess the the storage and runtime requirements of these filters.

References

[1] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A comparison of normal estimation schemes. *Proceedings of the IEEE Conference on Visualization*, pages 19–26, Oct. 1997.

[2] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. Evaluation and design of filters using a Taylor series expansion. *Visualization and Computer Graphics, IEEE Transactions on*, 3(2):184–199, Apr-Jun 1997.

[3] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.