A Novel Approach and Comparison of Normal Estimation Techniques on Body Centric Cubic (BCC) Lattice

Zahid Hossain *MSc, SFU* and Torsten Möller *Associate Prof. SFU*

ABSTRACT

We compared the existing techniques for estimating normals at the lattice sites of a Body Centric Cubic (BCC) grid. We also investigated newer approaches for estimating the normals that takes the special geometric arrangement of the BCC lattice into consideration and compared the results with other existing techniques. A ray-caster engine has been developed in this project which can trace ray on any arbitrary grid and also on analytic functions and dump data that can be used by other softwares to do statistical calculations.

Keywords: BCC,normal,reconstruction

1 INTRODUCTION

Recent studies on Body Centric Cubic (BCC) lattices have shown significant improvement in reconstruction quality compared to Catersian Cubic (CC). Not only data is reconstructed better but the method proposed by Entezari and Torsten [1] is almost twice faster than the best known method for CC, i.e. Tri-Cubic BSpline. It has also been shown that only 70% of the samples required by that of CC are sufficient to produce a similar quality of reconstruction in the BCC method. This intuitively led us to beleive that normal estimation on BCC grid and continuous reconstruction of the normals using techniques proposed by [1] would yield better results. Also it is conjectured that normal estimation on the lattice points of BCC grid has the higher potential of being more accurate than CC due to the fact that eight first-order neighbours (Figure 1) are closer in euclidean sense compared to the six axis-aligned neighbours on CC. If the conjecture is proven true and a method is developed to estimate more accurate normals on BCC then it will have a significant role in solving PDEs apart from higher quality of surface shading in volume visualization. In this paper, we compared older methods for estimating normals on BCC grid with that of CC and also investigated a newer approach which attempted to incorporate the firt-order neighbours and finally compared the results of the newer technique with the older ones.

2 RELATED WORKS

In all of the previous works, including Entezari [1] and Usman [4] normals at the BCC lattice points were estimated using the six axis-alinged second-order neighbours. The other eight first-order neighbours, which are rather closer in euclidean sense, were never taken into account. Hence the full potential of the BCC structure was never exploited.

3 APPROACH

Since the investigation was mostly about comparing older techniques of normal estimation and finding out newer techinques, we developed a versatile ray caster engine that can render a volume with phong shading and shadows on any arbitrary data grids, e.g. CC and BCC. It can also render a 3D analytic function without any underlying grid and provides mechanism to extract various data. This data can then be fed into other softwares like Matlab to perform statistical measurements and produce error images. Figure 2 shows the basic system model of the Ray-Caster engine.

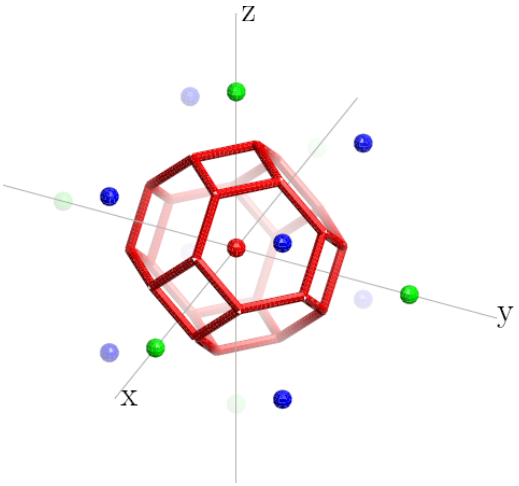


Figure 1: Red is the sample for which we want to estimate the normal for. Blue neighbours are the first-order neighbours while green neighbours are the second-order neighbours. Note that first-order neighbours have the bigger voronoi face which means they are closer to the red sample.

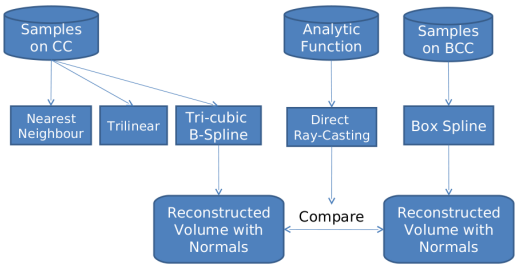


Figure 2: System model of the Ray-Caster engine

3.1 Data collection methodology

The Ray-Caster engine operates in two basic modes: Render mode and Compare mode. For this study we focused mainly on the "Compare" mode.

The main objective of the "Compare" mode is to compare the reconstruction quality of different grids with an analytic function for which we chose the widely used Marschner-Lobb function [2]. We chose $\alpha = 0.25$ and $f_M = 6.0$ for the Marschner-Lobb function and Figure 3 shows an iso-surface of 0.5 of the said unsampled function. In this mode, ray is casted into the analytic function and whenever an iso-surface is hit, all the other grid in the system are evaluated exactly at the same 3D point for both data value and normal. This design was adopted instead of ray-casting seperately on grids to make sure we are comparing between the analytic function and all the other grids exactly at the same 3D points. The actual function value and normal and all the other evluated data value with normals corresponding to the same 3D point were then dumped into a text file. Matlab was then used to read that text file, compute various

error measures and produce error images.

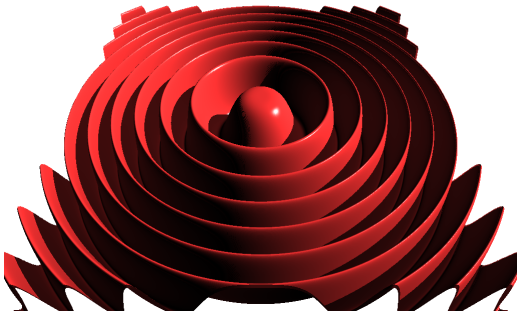


Figure 3: An iso-surface of 0.5 of the unsampled Marscher-Lobb function with $\alpha = 0.25$ and $f_M = 6.0$

Apart from full-blown ray-casting we were also interested to know how good the normals were estimated at only the lattice sites. This inquery is particularly important to investigate the potential of a certain normal estimation technique for solving PDEs. Therefore our Ray-Caster was also designed to dump normal data on the lattice sites only too.

4 Implementation

The software was implemented as a command line tool using C/C++ where one thread would be rendering the image and another thread would be showing a preview of the image "so far rendered." The previwer was implemented using OpenGL, however the basic image rendering takes places in the CPU in a non-realtime fashion. Iso-surfaces are shaded using Phong shading model [3] and the final images are output as BMP files along with various data in text files as mentioned in the Section 3.1. Figure 4 shows one screen-shot of the Ray-Caster. The software has be developed in a cross-platform complaint manner and so far it has been tested to have run well in both Windows and Linux.

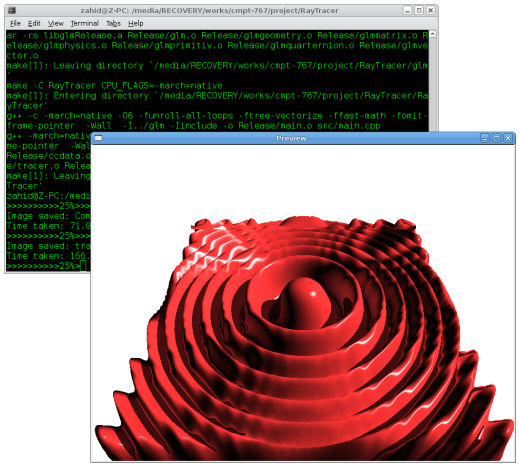


Figure 4: Screeen-shot of the Ray-Caster in action

5 Formulation

For CC grid we mainly considered TriCubic-BSpline for reconsctruction because that is the best known method so far. For BCC grid we considered the Quintic-BoxSpline method as proposed by [1]. Normals are estimated at the lattice sites once and then the above interpolation methods are applied on per-component basis (x,y and z seperately) for the corresponding grid to calculate non-lattice site normals. For CC grid, we estimated the normal at the lattice site using central differencing along three axes, i.e. using six samples whereas for BCC grid, we estimated normals using the methods listed below and then we compared the effectiveness of each with CC and the Marschner-Lobb analytic function as mentioned in Section 3.1.

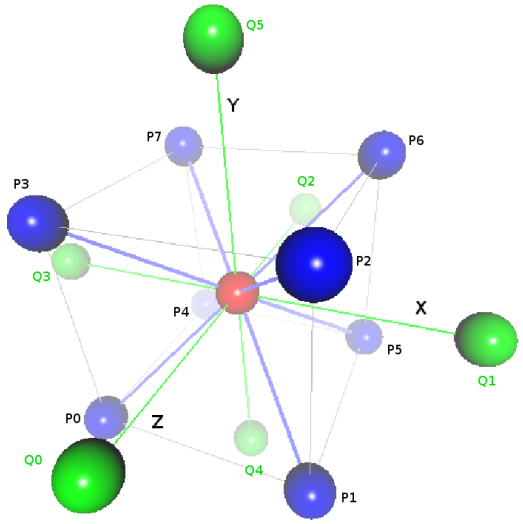


Figure 5: BCC Lattice. Red sample is the sample we want to compute normal for. Green samples are the second-order neighbours and Blue samples are the first-order neighbours

Refering to Figure 5, we define

$$\begin{aligned} P &= \{P_i | P_i = \\ &\quad \text{Sample value at the first-order neighbour sites}\} \\ Q &= \{Q_i | Q_i = \\ &\quad \text{Sample value at the second-order neighbour sites}\} \end{aligned} \tag{1}$$

$$\begin{aligned} h_x &= \text{Half of the length of the lattice along X-Axis} \\ h_y &= \text{Half of the length of the lattice along Y-Axis} \\ h_z &= \text{Half of the length of the lattice along Z-Axis} \end{aligned} \tag{2}$$

BCC lattice site normals were estimated by:

Second-Order Central Differencing (SOCD) This is similar to the axis aligned central differencing of the CC grid except that second-order neighbours (Figure 1) are taken into account because they are axis-aligned. SOCD estimation for a given sample point (Red sample in Figure 5) is given by

$$\begin{aligned} \frac{\partial f}{\partial x} &\approx \frac{Q_1 - Q_3}{4h_x} \\ \frac{\partial f}{\partial y} &\approx \frac{Q_5 - Q_4}{4h_y} \\ \frac{\partial f}{\partial z} &\approx \frac{Q_0 - Q_2}{4h_z} \end{aligned} \tag{3}$$

Box Central Differencing (BCD) In this method the eight first-order neighbours (Figure 1) were taken into consideration. However, first-order neighbours are not axis-aligned. This creates some complications which is resolved using Taylor series expansion. A detail discussion on this topic is made in Appendix A. BCD is given by the following:

$$\begin{aligned}\frac{\partial f}{\partial x} &\approx \frac{(P_2-P_3)+(P_6-P_7)+(P_1-P_0)+(P_5-P_4)}{8h_x}\\ \frac{\partial f}{\partial y} &\approx \frac{(P_3-P_0)+(P_2-P_1)+(P_6-P_5)+(P_7-P_4)}{8h_y}\\ \frac{\partial f}{\partial z} &\approx \frac{(P_2-P_6)+(P_1-P_5)+(P_0-P_4)+(P_3-P_7)}{8h_z}\end{aligned} \tag{4}$$

A simple interpretation of the above formula is: each of the differentials is merely an average of all the four differentials along a particular axis. For example:

$$\begin{aligned}\frac{\partial f}{\partial x} &\approx \frac{(P_2-P_3)+(P_6-P_7)+(P_1-P_0)+(P_5-P_4)}{8h_x}\\ &= \frac{\frac{(P_2-P_3)}{2h_x}+\frac{(P_6-P_7)}{2h_x}+\frac{(P_1-P_0)}{2h_x}+\frac{(P_5-P_4)}{2h_x}}{4}\end{aligned} \tag{5}$$

Interpolated Central Differencing (ICD) One point in the center of each of the six faces of the BCC lattice is computed using Quintic-BoxSpline. This basically is similar to computing the FCC lattice points on the six faces of the BCC lattice and then using them as pseudo-samples for doing central-differencing along each of the axes. This is similar to generating higher resolution cartesian grid within the BCC grid and then estimating the normal. Let

$$\begin{aligned}I = \{I_i | I_i = &\\ &\text{Interpolated value at the mid-point}\\ &\text{between the center, Red sample, and } Q_i\}\end{aligned} \tag{6}$$

It is easy to see that I_i are the points at the centre of each face of the lattice. Now ICD is given by the following:

$$\begin{aligned}\frac{\partial f}{\partial x} &\approx \frac{I_1-I_3}{2h_x}\\ \frac{\partial f}{\partial y} &\approx \frac{I_5-I_4}{2h_y}\\ \frac{\partial f}{\partial z} &\approx \frac{I_0-I_2}{2h_z}\end{aligned} \tag{7}$$

The main motivation of including ICD is to check if we could have achieved similar or maybe better result than BCD method by taking only 6 samples.

6 Results

All the experiments were performed using Marschner-Lobb function with $F_M = 6.0$ and $\alpha = 0.25$ with the iso-surface of 0.5. Quality metric of the normal reconstruction is defined as the angle (in degrees) between the actual analytic normal and the reconstructed normal. To have an overall picture of the total error made in normals, we defined an energy function as follows.

$$E = \frac{180}{\pi}\sum_k (cos^{-1}(\langle \tilde{N}, N \rangle))^2 \tag{8}$$

Where,

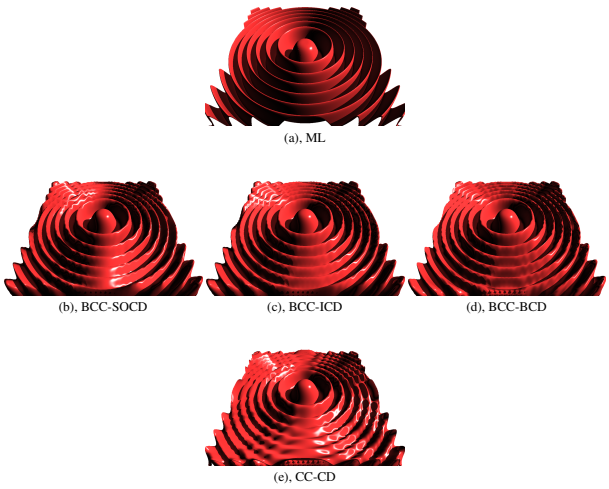
$\tilde{N}$ = Normalized reconstructed normal
N = Normalized analytic normal

Typically, the best method would produce the smallest E. We also calculated the RMS of the angles and again the best method should produce the smallest RMS. Table 1 summarizes our findings. Clearly BCC-BCD method approximates the normals better than other methods. It is quite interesting to note that BC-BCD method is significantly better than than BCC-ICD method which uses axis-aligned central differencing and with the neighbouring samples being exactly same distance away along the axes as BCC-BCD. In this experiment we rendered Marschner-Lobb isosurface with shading on an output image resolution of 1024x758.

	CC-CD	BCC-SOCD	BCC-ICD	BCC-BCD
E ($\times 10^8$)	11.8949	11.2586	8.16344	4.85875
RMS Angle Error (degrees)	54.2459	48.8836	41.6253	32.1132

Table 1: Comparison of normal estimation on BCC and CC using different methods

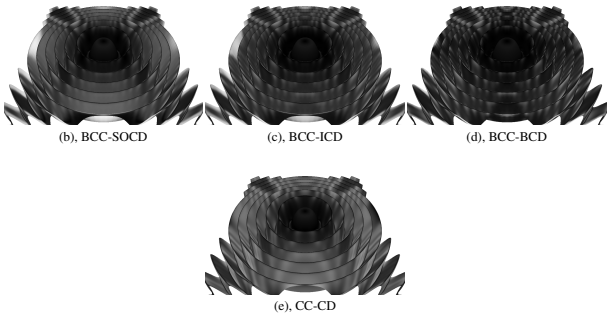
This fact is also noticible perceptually in the rendered images. Figure 6 shows the different images produced by different methods. If we look at the shading of the original Marschnar-Lobb, Figure 6(a), and specially note the distribution of the "specular highlights" we will immediately see that BCC-BCD method,Figure 6(d), produces the closest approximation.



First Row:	(a) Unsampled Marscher-Lobb (ML).
Second Row:	Normals estimated on BCC grid.
	(b) Second-Order Central Differencing (BCC-SOCD)
	(c) Interpolated Central Differencing (BCC-ICD)
	(d) Box Central Differencing (BCC-BCD)
Third Row:	(e) Normals estimated on CC grid using Central Differencing (CC-CD)

Figure 6: Comparison of data and normal reconstruction of various grids

Finally, we were also interested to know the extent of errors in the normal estimation each method had only at the grid points. For that, we ran a simple experiment to estimate and compare normals only at the grid points. Table 2 shows our findings.



First Row:	(a) Unsampled Marscher-Lobb (ML).
Second Row:	Normals estimated on BCC grid.
	(b) Second-Order Central Differencing (BCC-SOCD)
	(c) Interpolated Central Differencing (BCC-ICD)
	(d) Box Central Differencing (BCC-BCD)
Third Row:	(e) Normals estimated on CC grid using Central Differencing (CC-CD)

Figure 7: Comparison of data and normal reconstruction with error images of various grids

	CC-CD	BCC-SOCD	BCC-ICD	BCC-BCD
E ($\times 10^8$)	0.880199	1.11468	0.73573	0.517233
RMS Angle Error (degrees)	35.7367	40.216	32.6726	27.3947

Table 2: Comparison of normal estimation only at the grid points on BCC and CC using different methods

Clearly Table 2 is consistent with the fact that BCC-BCD approximates normals better than other methods.

7 Conclusion

In this project we have studied some existing techniques for estimating normals at the lattice point of CC and BCC grids. We have also proposed a novel approach towards estimating normals on a BCC grid which exploits the nearer eight first order neighbours. Mathematical analysis (see Appendix A) and our experiments verify that the newer Box Central Differencing (BCD) technique is indeed superior. Not only this method has great potential for shading volumes better but also has potential along the lines of solving PDEs.

Acknowledgements

The authors wish to thank Usman Alim, Steven Bergner, Bernhard Finkbeiner of GRUVi lab at Simon Fraser University for their relentless support.

A Appendix

A.1 BCC Box Central Differencing (BCC-BCD)

To derive the BCC-BCD method we assume a trivariate function $f(u,v,w)$. The Taylor series expansion upto quadratic order of this function about the point (x,y,z) is given by

$$\begin{aligned} f(u,v,w) = & f(x,y,z) + (u-x)\frac{\partial f}{\partial u}(x,y,z) + \\ & (v-y)\frac{\partial f}{\partial v}(x,y,z) + \\ & (w-z)\frac{\partial f}{\partial w}(x,y,z) + \\ & \frac{1}{2}(u-x)^2\frac{\partial^2 f}{\partial u^2}(x,y,z) + \\ & (u-x)(v-y)\frac{\partial}{\partial v}\frac{\partial f}{\partial u}(x,y,z) + \\ & (u-x)(w-z)\frac{\partial}{\partial w}\frac{\partial f}{\partial u}(x,y,z) + \\ & (v-y)(w-z)\frac{\partial}{\partial w}\frac{\partial f}{\partial v}(x,y,z) + \\ & \frac{1}{2}(v-y)^2\frac{\partial^2 f}{\partial v^2}(f)(x,y,z) + \\ & \frac{1}{2}(w-z)^2\frac{\partial^2 f}{\partial w^2}(x,y,z) + \ldots \end{aligned} \tag{9}$$

Now let us define the points $P = \{P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$ as follows:

$$\begin{aligned} P_0 &= f(x-h_x, y-h_y, z+h_z) \\ P_1 &= f(x+h_x, y-h_y, z+h_z) \\ P_2 &= f(x+h_x, y+h_y, z+h_z) \\ P_3 &= f(x-h_x, y+h_y, z+h_z) \\ P_4 &= f(x-h_x, y-h_y, z-h_z) \\ P_5 &= f(x+h_x, y-h_y, z-h_z) \\ P_6 &= f(x+h_x, y+h_y, z-h_z) \\ P_7 &= f(x-h_x, y+h_y, z-h_z) \end{aligned} \tag{10}$$

Note that the point set P is exactly the same P point set of Figure 5

Now let us write

$$S = \mathbf{A}P_0 + \mathbf{B}P_1 + \mathbf{C}P_2 + \mathbf{D}P_3 + \mathbf{E}P_4 + \mathbf{F}P_5 + \mathbf{G}P_6 + \mathbf{H}P_7 \tag{11}$$

Where $\{A, B, C, D, E, F, G, H\}$ are all constants

Writing Equation 9 up to cubic order and substituting Equation 10 into it we can rewrite S as

$$
\begin{aligned}
S =& (A+B+C+E+F+G+H+D)f(x,y,z)+ \\
& h_x(-A+B+C-E+F+G-H-D)\frac{\partial f}{\partial u}(x,y,z)+ \\
& h_y(-A-B+C-E-F+G+H+D)\frac{\partial f}{\partial v}(x,y,z)+ \\
& h_z(A+B+C-E-F-G-H+D)\frac{\partial f}{\partial w}(x,y,z)+ \\
& \frac{h_x^2}{2}(A+B+C+E+F+G+H+D)\frac{\partial^2 f}{\partial u^2}(x,y,z)+ \\
& \frac{h_y^2}{2}(A+B+C+E+F+G+H+D)\frac{\partial^2 f}{\partial y^2}(x,y,z)+ \\
& \frac{h_z^2}{2}(A+B+C+E+F+G+H+D)\frac{\partial^2 f}{\partial w^2}(x,y,z)+ \\
& \frac{h_x^3}{6}(-A+B+C-E+F+G-H-D)\frac{\partial^3 f}{\partial u^3}(x,y,z)+ \\
& \frac{d_y^3}{6}(-A-B+C-E-F+G+H+D)\frac{\partial^3 f}{\partial v^3}(x,y,z)+ \\
& \frac{h_z^3}{6}(A+B+C-E-F-G-H+D)\frac{\partial^3 f}{\partial w^3}(x,y,z) \\
& h_x h_y(A-B+C+E-F+G-H-D)\frac{\partial}{\partial v}\frac{\partial f}{\partial u}(x,y,z)+ \\
& h_x h_z(-A+B+C+E-F-G+H-D)\frac{\partial}{\partial w}\frac{\partial f}{\partial u}(x,y,z)+ \\
& h_y h_z(-A-B+C+E+F-G-H+D)\frac{\partial}{\partial w}\frac{\partial f}{\partial v}(x,y,z)+ \\
& \frac{h_x^2 h_y}{2}(-A-B+C-E-F+G+H+D)\frac{\partial}{\partial v}\frac{\partial^2 f}{\partial u^2}(x,y,z)+ \\
& \frac{h_x^2 h_z}{2}(A+B+C-E-F-G-H+D)\frac{\partial}{\partial w}\frac{\partial^2 f}{\partial u^2}(x,y,z)+ \\
& \frac{h_x h_y^2}{2}(-A+B+C-E+F+G-H-D)\frac{\partial^2}{\partial v^2}\frac{\partial f}{\partial u}(x,y,z)+ \\
& \frac{h_x h_z^2}{2}(-A+B+C-E+F+G-H-D)\frac{\partial^2}{\partial w^2}\frac{\partial f}{\partial u}(x,y,z)+ \\
& \frac{h_y^2 h_z}{2}(A+B+C-E-F-G-H+D)\frac{\partial}{\partial w}\frac{\partial^2 f}{\partial v^2}(x,y,z)+ \\
& \frac{h_y h_z^2}{2}(-A-B+C-E-F+G+H+D)\frac{\partial^2}{\partial w^2}\frac{\partial f}{\partial v}(x,y,z)+ \\
& h_x h_y h_z(A-B+C-E+F-G+H-D)\frac{\partial}{\partial w}\frac{\partial}{\partial v}\frac{\partial f}{\partial u}(x,y,z)+
\end{aligned} \tag{12}
$$

Now, let us try to isolate $\frac{\partial f}{\partial u}$ out of the above equation and for that let us take the following:

$$
\begin{aligned}
A+B+C+E+F+G+H+D &= 0, \quad \text{with } f(x,y,z) \\
-A+B+C-E+F+G-H-D &= 1, \quad \text{with } \frac{\partial f}{\partial u} \\
-A-B+C-E-F+G+H+D &= 0, \quad \text{with } \frac{\partial f}{\partial v} \\
A+B+C-E-F-G-H+D &= 0, \quad \text{with } \frac{\partial f}{\partial w} \\
A-B+C+E-F+G-H-D &= 0, \quad \text{with } \frac{\partial}{\partial v}\frac{\partial f}{\partial u} \\
-A+B+C+E-F-G+H-D &= 0, \quad \text{with } \frac{\partial}{\partial w}\frac{\partial f}{\partial u} \\
-A-B+C+E+F-G-H+D &= 0, \quad \text{with } \frac{\partial}{\partial w}\frac{\partial f}{\partial v} \\
A-B+C-E+F-G+H-D &= 0, \quad \text{with } \frac{\partial}{\partial w}\frac{\partial}{\partial v}\frac{\partial f}{\partial u}
\end{aligned} \tag{13}
$$

This forms a system of linear equations and the solution of the above system is

$$
\left\{A=-\frac{1}{8}, B=\frac{1}{8}, C=\frac{1}{8}, D=-\frac{1}{8}, \right. \\
\left. E=-\frac{1}{8}, F=\frac{1}{8}, G=\frac{1}{8}, H=-\frac{1}{8}\right\} \tag{14}
$$

If we plug in the above solution to S, we have

$$
\begin{aligned}
S = & \frac{1}{2}h_x h_y^2 \frac{\partial^2}{\partial v^2}\frac{\partial f}{\partial u}(x,y,z) + \frac{1}{2}h_x h_z^2 \frac{\partial^2}{\partial w^2}\frac{\partial f}{\partial u}(x,y,z)+ \\
& h_x \frac{\partial f}{\partial u}(x,y,z) + \frac{1}{6}h_x^3 \frac{\partial^3 f}{\partial u^3}(x,y,z)
\end{aligned} \tag{15}
$$

Dividing both sides by h_x gives

$$
\begin{aligned}
\tilde{N}_x = \frac{S}{h_x} = & \frac{\partial f}{\partial u}(x,y,z) + \frac{1}{2}h_y^2 \frac{\partial^2}{\partial v^2}\frac{\partial f}{\partial u}(x,y,z) + \frac{1}{2}h_z^2 \frac{\partial^2}{\partial w^2}\frac{\partial f}{\partial u}(x,y,z)+ \\
& \frac{1}{6}h_x^2 \frac{\partial^3 f}{\partial u^3}(x,y,z)
\end{aligned} \tag{16}
$$

If $f(u,v,w)$ has a maximum order of two, i.e. quadratic in nature which in turns also means if $f(u,v,w)$ can be characterized by the linear combination of the shifted verson of $(u+v+w)^2$ then $\tilde{N}_x = \frac{\partial f}{\partial u}$. This is because if

$$
f(u,v,w) = \sum_i \sum_j \sum_k W_{ijk}\{(u-a_i)+(v-b_j)+(w-c_k)\}^2 \tag{17}
$$

Where, W_{ijk}, a_i, b_j, c_k are all constants

then it follows

$$
\begin{aligned}
\frac{\partial^2}{\partial v^2}\frac{\partial f}{\partial u}(x,y,z) &= 0 \\
\frac{\partial^2}{\partial w^2}\frac{\partial f}{\partial u}(x,y,z) &= 0 \\
\frac{\partial^3 f}{\partial u^3}(x,y,z) &= 0
\end{aligned}
$$

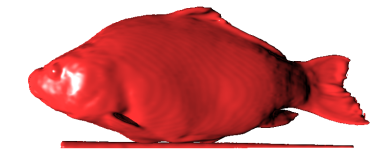
But for fairly small h_x we can always approximate

$$
\begin{aligned}
\tilde{N}_x &\approx \frac{\partial f}{\partial u}(x,y,z) \approx \frac{S}{h_x} \\
&= \frac{(P_2-P_3)+(P_6-P_7)+(P_1-P_0)+(P_5-P_4)}{8h_x}
\end{aligned} \tag{18}
$$

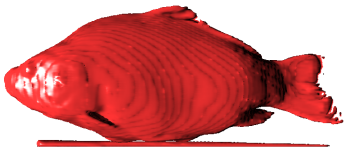
Similarly N_y, N_z can also be found using the same pattern.

B Appendix

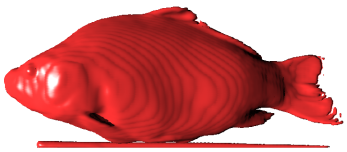
C Some Rendered Output of the Ray-Caster



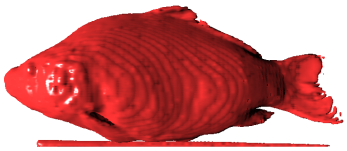
(b), CC



(d), BCC-SOCD



(c), BCC-ICD



(e), BCC-BCD

Figure 8: Comparison of data and normal reconstruction on Carp dataset

References

[1] A. Entezari, D. Van De Ville, and T. Möller. Practical box splines for volume rendering on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):313 – 328, 2008.

[2] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In R. D. Bergeron and A. E. Kaufman, editors, *Proceedings of the IEEE Conference on Visualization*, pages 100–107. IEEE Computer Society Press, Oct. 1994.

[3] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.

[4] A. E. Usman R. Alim and T. Möller. The lattice-boltzmann method on optimal sampling lattices. Technical report, 2008.