

# Exhibition and Evaluation of Two Schemes for Determining Hypersurface Curvature in Volumetric Data

Jacob D. Hauenstein

The University of Alabama in Huntsville  
301 Sparkman Drive  
Huntsville, AL 35899  
hauensj1@uah.edu

Timothy S. Newman

The University of Alabama in Huntsville  
301 Sparkman Drive  
Huntsville, AL 35899  
tnewman@cs.uah.edu

## ABSTRACT

Advancements in methodologies for determining 3-dimensional manifold (hypersurface) curvature in volumetric data are presented. Such determinations are requisite in certain shape-based visualization and analysis tasks. The methods explored here are convolution-based approaches. In addition to motivating and describing these methods, an evaluation of their (1) accuracy and (2) computational performance is also presented. That evaluation includes benchmarking on both noise-free and noisy volumetric data.

## Keywords

Curvature, Volumetric Data, 3D Manifold, Hypersurface, Imaging

## 1 INTRODUCTION

Curvature and related quantities have been found to be useful attributes for certain computer-based tasks utilizing range images (henceforth images), point clouds, or volumetric data (henceforth volumes or volume data). Application areas include computer graphics and visualization, healthcare, seismology, computational archaeology [Du18], etc. For example, curvature has been used in segmentation, object recognition, geometric modeling, and analysis of images and volumes [Bib16, Bes86, Bel12, Bag16, Lef18, Sou16], to perform reconstruction in images [Lef17], for biometrics [Sya17], for computer vision-based quality control in manufacturing [Kot18], etc. Other examples include emphasizing features in renderings of meshes [AR18] and images [Hau18], mesh parameterization [Vin17], highlighting shapes in volume renderings [Kin03], and visualization of medical data [Pre16]. In summary, curvature has been quite useful in computer graphics, visualization, and computer vision.

In fact, there is a rich literature that focuses on the use of curvature of 1-dimensional (1D) curves or 2-dimensional (2D) surfaces, including all of the works in the prior paragraph and [Lan07]. The works using curvature of conventional surfaces in 3-space are said,

more formally, to operate on 2D manifolds (commonly simply called *surfaces*). In such domains, the maximum and minimum curvatures (known as the *principal curvatures* and denoted  $\kappa_1$  and  $\kappa_2$ , respectively) can be valuable since they describe surface shape properties. These surfaces, despite existing in a 3D space, are 2D manifolds and thus have two principal curvatures.

Curvature of 3D manifolds (here called *hypersurfaces*, following the terminology used previously by Monga and Benayoun [Mon92]) also can be a useful descriptor for certain tasks, such as tasks operating on volumetric regions of data rather than on surface structures. For such tasks, the three principal curvatures,  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  (ordered such that  $\kappa_1 > \kappa_2 > \kappa_3$ ), can be of value. Some exploration of such curvatures in volume data has previously been described [Mon92, Ham94]. Recently, a number of works have reported use of these 3D manifold curvatures in medical assessment / segmentation, seismic data visualization, and surface reconstruction (e.g., [Ald14, Suz18, Pap07]). A previous work has also presented a framework capable of detecting crease surfaces in d-dimensional hypersurfaces [Yos12].

Another use for hypersurface curvatures is classification of volume data points based on the relative, relative absolute, and average values of these three curvature values [Hir01]. A table of all possible curvature classifications at each point has previously been presented (reproduced in part in Fig. 1), and has been used previously for visualization of volume shape [Hir01] and categorization of lung tumors [Hir18]. An example application of these curvature classifications and ensuing visualization (using the scheme from [Hir01]) on a CT scan of a phantom (CT scan obtained from <http://www.santec.lu/project/optimage/samples>)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

No.	Simple	Absolute	Average
1	$\kappa_1 > \kappa_2 > \kappa_3 > 0$	$ \kappa_1  >  \kappa_2  >  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
2	$\kappa_1 > \kappa_2 > \kappa_3 = 0$	$ \kappa_1  >  \kappa_2  >  \kappa_3  = 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
3	$\kappa_1 > \kappa_2 = \kappa_3 > 0$	$ \kappa_1  >  \kappa_2  =  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
4	$\kappa_1 > \kappa_2 = \kappa_3 = 0$	$ \kappa_1  >  \kappa_2  =  \kappa_3  = 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
5	$\kappa_1 = \kappa_2 > \kappa_3 > 0$	$ \kappa_1  =  \kappa_2  >  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
6	$\kappa_1 = \kappa_2 > \kappa_3 = 0$	$ \kappa_1  =  \kappa_2  >  \kappa_3  = 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
7	$\kappa_1 = \kappa_2 = \kappa_3 > 0$	$ \kappa_1  =  \kappa_2  =  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
8	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	$ \kappa_1  =  \kappa_2  =  \kappa_3  = 0$	$\kappa_1 + \kappa_2 + \kappa_3 = 0$
9	$\kappa_1 > \kappa_2 > 0 > \kappa_3$	$ \kappa_1  >  \kappa_2  >  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 > 0$
⋮	⋮	⋮	⋮
39	$0 > \kappa_1 > \kappa_2 > \kappa_3$	$ \kappa_3  >  \kappa_2  >  \kappa_1  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 < 0$
40	$0 > \kappa_1 > \kappa_2 = \kappa_3$	$ \kappa_2  =  \kappa_3  >  \kappa_1  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 < 0$
41	$0 > \kappa_1 = \kappa_2 > \kappa_3$	$ \kappa_3  >  \kappa_2  =  \kappa_1  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 < 0$
42	$0 > \kappa_1 = \kappa_2 = \kappa_3$	$ \kappa_1  =  \kappa_2  =  \kappa_3  > 0$	$\kappa_1 + \kappa_2 + \kappa_3 < 0$

Figure 1: A subset of the 42 possible classes of  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  (adapted from [Hir01]).

is shown in Fig. 2. In the figure, a grayscale encoding of the classes is used, with class 1 mapped to black and with brightness increasing linearly to class 42, which is mapped to white.

However, to use hypersurface curvatures on sensed data, there needs to be some way to determine them. Here, we describe two methods for their determination. Since no comparative evaluation of hypersurface curvature determination methods has previously been performed, here we also present an evaluation of these two methods. Our study considers both the accuracy and run times of the methods, thereby providing insight into inaccuracies from and relative limitations in 3D manifold curvature determinations.

This paper is organized as follows. Section 2 provides background information on hypersurface curvature and some related surface curvature details. Section 3 describes the two hypersurface curvature determination methods we propose for volume data. Section 4 details our experimental procedures and results. Section 5 presents visualizations of curvature values in sensed data using one of the methods described. Section 6 concludes the work.

## 2 BACKGROUND AND PREVIOUS WORK

Here, we first provide an overview of the mathematics for determining curvature of hypersurfaces within volumetric data. Readers looking for a more detailed presentation may wish to see [Mon92] or [Ham94]. Then, we describe two methods for determining conventional surface curvature in volumetric datasets. (The two hypersurface curvature determination methods we describe later in this paper are inspired by these methods.)

The notation we use is as follows.  $(u, v, w)$  denotes a grid (or sample) point within the volume, where

$0 \leq u < N_u$ ,  $0 \leq v < N_v$ ,  $0 \leq w < N_w$  for a volume of size  $N_u \times N_v \times N_w$ . The value at point  $(u, v, w)$  is denoted  $f(u, v, w)$ ;  $f$  represents the underlying function that generates the volume. Consequently,  $f_u$  represents the partial derivative of  $f$  in the  $u$  direction.

### 2.1 Hypersurface Curvature Mathematics

The hypersurface's three principal curvatures (i.e., of  $f$ ) are the eigenvalues of the matrix:

$$\frac{1}{l} \begin{bmatrix} f_{uu} & f_{uv} & f_{uw} \\ f_{uv} & f_{vv} & f_{vw} \\ f_{uw} & f_{vw} & f_{ww} \end{bmatrix} \begin{bmatrix} 1 + f_u^2 & f_u f_v & f_u f_w \\ f_u f_v & 1 + f_v^2 & f_v f_w \\ f_u f_w & f_v f_w & 1 + f_w^2 \end{bmatrix}^{-1}, \quad (1)$$

where

$$l = \sqrt{1 + f_u^2 + f_v^2 + f_w^2}. \quad (2)$$

Thus, computation of  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  requires knowledge of the first and second derivatives of  $f$ . For many curvature-based tasks on volumetric datasets, the continuous form of  $f$  is unknown because the data is sensed (i.e., acquired via a sensor). In such cases, the first and second derivatives must be estimated in order to determine  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  based on evaluation of Eqn. 1. The two hypersurface curvature determination methods discussed later in this work both operate by estimating the necessary derivatives and using them in Eqn. 1.

### 2.2 Related Comparisons of Curvature Determination Methods

While we are not aware of any studies comparatively evaluating hypersurface curvature determination methods, some studies exist for related domains, for example of methods to determine surface curvature in range data [Bes86, Hau18]. In range data, many curvature determination methods exist, but those studies [Bes86, Hau18] found that no one method is uniformly best across all types of input. Method accuracy was found to vary depending on data type (e.g., noise-free synthetic or noisy sensed data) and surface shape. A previous study of surface curvature determination methods in volume data [Hau14] similarly concluded that no one method is best for all input data types.

### 2.3 Related Methods for Surface Curvature in Volumes

The two methods for determining hypersurface curvature proposed in this paper, described later, are volumetric analogues of two methods for determining surface curvature in volume data. Here, we describe those existing methods. Those methods determine surface curvature by (1) estimating derivatives at every point

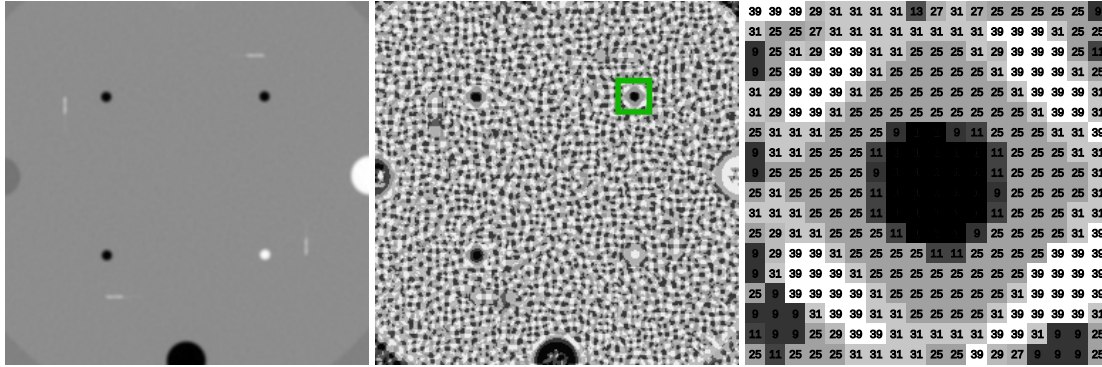


Figure 2: A slice from a CT scan of a phantom (left), the resulting data point-by-data point classifications of hypersurface curvatures according to Fig. 1 (middle), and zoomed-in detail (right). The classes are visualized (in middle and right) using a grayscale mapping.

in the volume and then (2) computing the curvature at every point in the volume using the estimated derivatives. We focus on these two existing methods because one of them was previously found to perform well on noise-free data, and one of them was previously found to perform well on noise-added and real data [Hau14], thus the volumetric analogues presented are based on two methods that, together, perform well across many types of input data.

### 2.3.1 Taylor Expansion

The first of those surface curvature methods was described by Kindlmann et al. [Kin03]. It uses separable convolution filters developed according to the framework of Möller et al. [Möl98]. Such filters are based on the Taylor Expansion and allow for a given accuracy and continuity.

The Kindlmann et al. method uses these first and second derivative filters at every data point in the volume to estimate the derivatives of  $f$  required for computation of surface curvature. Then, these estimates of  $f$ 's derivatives are used to determine gradients, Hessians, and, finally, curvatures. Previous studies [Kin03, Hau14] have found a Taylor Expansion-based strategy to be very fast and accurate for surface curvature determination on synthetic, noise-free data.

### 2.3.2 Orthogonal Polynomials

The second of those surface curvature methods for determining surface curvature in volumes was described by Hauenstein and Newman [Hau14]. It was motivated by a method for determining surface curvature in range images described by Besl and Jain [Bes86] based on a locally fit surface model [Fly89]. To estimate derivatives, it uses separable convolution filters sampled from orthogonal polynomials. Such filters implicitly perform a least squares fitting [Fly89].

Like the previously discussed Taylor Expansion-based method for determining surface curvature, the Hauenstein and Newman method uses convolution to estimate first and second derivatives and then uses these

derivative estimates to determine curvature. A previous study [Hau14] found this orthogonal polynomials-based method to be relatively accurate and fast for surface curvature determination on noise-added synthetic volumes as well as sensed volumes.

## 3 HYPERSURFACE CURVATURE DETERMINATION METHODS

Here, we describe in detail our proposed methods for determining hypersurface curvature within volume data.

### 3.1 Taylor Expansion Filters (TEF)

The first hypersurface curvature method we describe, denoted **TEF**, uses convolution filters derived from the Taylor Expansion. It is a hypersurface analogue of the Kindlmann et al. [Kin03] surface curvature determination approach. The method determines hypersurface curvatures by first performing a series of convolutions along each axis. These convolutions provide estimates for first and second directional derivatives. Once these derivatives are known, the three principal curvatures are computed as the eigenvalues of Eq. 1.

The method's filters are devised according to the Möller et al. framework [Möl98]. The filters have accuracy and continuity parameters, and the chosen accuracy and continuity parameters also impact the size of the resulting convolution filters. Here, we use filters with  $C^3$  continuity and fourth order accuracy, and they thus allow for exact reconstruction of functions of degree 3 or lower. The first derivative filter is:

$$\left\{ -\frac{1.0}{12.0}, \frac{2.0}{3.0}, 0.0, -\frac{2.0}{3.0}, \frac{1.0}{12.0} \right\}. \quad (3)$$

The second derivative filter is:

$$\left\{ -\frac{1.0}{24.0}, \frac{1.0}{6.0}, \frac{17.0}{24.0}, -\frac{5.0}{3.0}, \frac{17.0}{24.0}, \frac{1.0}{6.0}, -\frac{1.0}{24.0} \right\}. \quad (4)$$

Convolving the appropriate filters in the appropriate directions allows these filters to estimate all necessary

first, second, and mixed partial derivatives. Specifically,  $f_i$  is found by convolving the first derivative filter in the  $i$  direction. Other first derivatives are found analogously.  $f_{ij}$  is found by convolving the first derivative filter in both the  $i$  and  $j$  directions. Other mixed partials are found analogously.  $f_{ii}$  is found by convolving the second derivative filter in the  $i$  direction. Other second derivatives are found analogously.

These filters were previously found by Kindlmann et al. [Kin03] to perform well in determination of conventional surface curvature in volumes. Thus, we were motivated to extend them to hypersurface curvature.

### 3.2 Orthogonal Polynomials Filters (OPF)

The second hypersurface curvature method we describe, denoted **OPF**, also uses a series of 1D convolutions to determine directional derivatives. And, like **TEF**, **OPF** uses these estimated derivatives to compute the three principal curvatures as the eigenvalues of Eq. 1. Unlike **TEF**, the convolution filters for **OPF** are derived from orthogonal polynomials. Such filters implicitly perform a least squares fitting [Bes86, Fly89], and the derivatives estimated via this method are thus identical to those of a local surface that well-fits that neighborhood (i.e., via linear regression).

In **OPF**, derivative estimation filters of odd size  $N$  are used. These filters are generated by sampling orthogonal polynomials,  $b_0, b_1, b_2$ , at  $N$  locations (more details about these polynomials are in [Bes86]):

$$b_0(\theta) = \frac{1}{N}, \quad (5)$$

$$b_1(\theta) = \frac{3}{M(M+1)(2M+1)}\theta, \quad (6)$$

$$b_2(\theta) = \frac{1}{P(M)}\left(\theta^2 - \frac{M(M+1)}{3}\right), \quad (7)$$

where  $M = \frac{N-1}{2}$ ,  $P(M)$  is given by:

$$P(M) = \frac{8}{45}M^5 + \frac{4}{9}M^4 + \frac{2}{9}M^3 - \frac{1}{9}M^2 - \frac{1}{15}M, \quad (8)$$

and  $\theta$  denotes the locations at which each polynomial is sampled, with  $\theta \in \{-\frac{N-1}{2}, \dots, -1, 0, 1, \dots, \frac{N-1}{2}\}$ . Applying the  $b_0$  filter performs smoothing. Applying the  $b_1$  filter generates estimates of the first derivative. Applying the  $b_2$  filter generates estimates of the second derivative.

Using these kernels, estimated partial and mixed partial derivatives of  $f$  are found. Specifically, to find  $f_i$ ,  $i \in \{u, v, w\}$ , we (1) convolve in the  $i$  direction with the discrete filter resulting from sampling  $b_1$  and (2) convolve in each of the other two directions with the discrete filter resulting from sampling  $b_0$ . Moreover, to find  $f_{ij}$ ,  $i, j \in \{u, v, w\}$  and  $i \neq j$ , we (1) convolve in

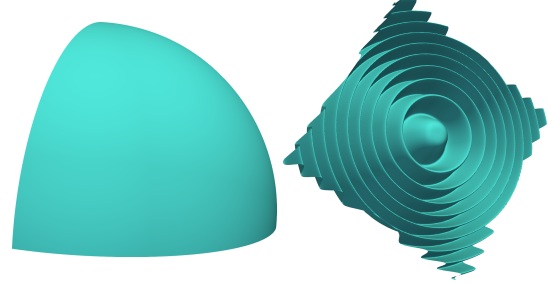


Figure 3: An isosurface at 10000 from the Spheres dataset (left), and an isosurface at 0 from the ML dataset (right).

the  $i$  and  $j$  directions with the discrete filter resulting from sampling  $b_1$  and (2) convolve in the other direction with the discrete filter resulting from sampling  $b_0$ . Lastly, to find  $f_{ii}$ ,  $i \in \{u, v, w\}$ , we (1) convolve in the  $i$  direction with the  $b_2$  filter sampling and (2) convolve in the other directions with the  $b_0$  filter sampling. Once all necessary derivatives are estimated through this process, the hypersurface curvatures are calculated as the eigenvalues of Eq. 1.

We have used a filter size,  $N$ , of 7 for our applications here, since a prior work [Hau14] found  $N = 7$  to be appropriate for surface curvature within volumes.

## 4 EXPERIMENTS AND RESULTS

In this section, we describe the experiments in and results of our comparative evaluation of the **TEF** and **OPF** hypersurface curvature methods. The experiments involve both accuracy and run time tests. Synthetic volume datasets are used in these tests. We next describe those datasets.

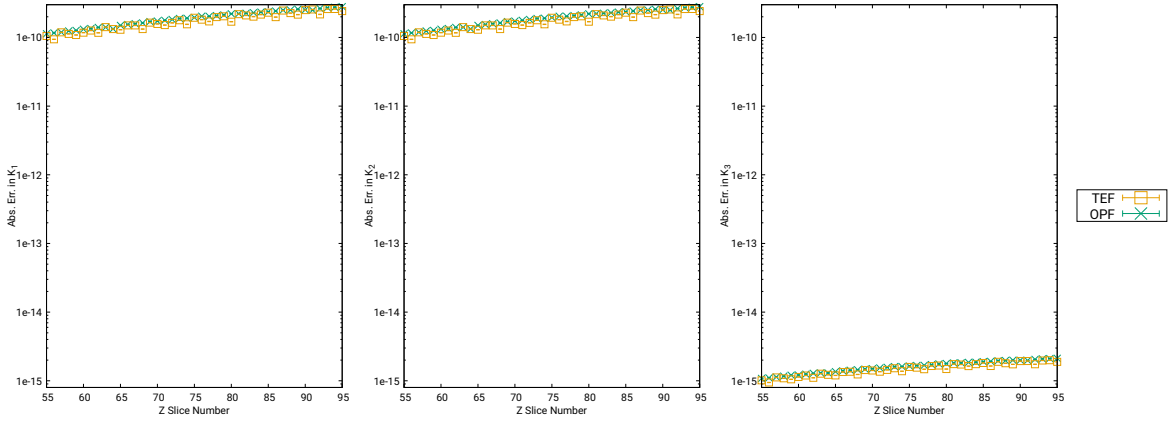
### 4.1 Synthetic Datasets

Noise-free and noise-added versions of two synthetic volumes were used. Each was sampled from a continuous volume. Since the continuous form of these volumes are known, the three principal curvatures of the continuous forms can be computed exactly. Our experiments evaluate accuracy by comparing these known curvatures to the curvatures determined by the methods.

The first dataset, called Spheres, is generated using the function:

$$f(u, v, w) = (u^2 + v^2 + w^2). \quad (9)$$

The formula was sampled on a  $256 \times 256 \times 256$  grid with each axis in the range  $[0, 255]$ , resulting in a volume containing values ranging from 0 to 195075. Our experiments consider this volume with and without added Gaussian noise ( $\mu = 0$ ,  $\sigma = 0.001$ ). A rendering of a level surface of this dataset is shown in Fig. 3 (left).

Figure 4: Per-slice average error in  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  for noise-free Spheres.

	$\kappa_1$ Avg. Abs. Err.	$\kappa_1$ Std. Err.	$\kappa_2$ Avg. Abs. Err.	$\kappa_2$ Std. Err.	$\kappa_3$ Avg. Abs. Err.	$\kappa_3$ Std. Err.
TEF	$3.74 \cdot 10^{-10}$	$2.21 \cdot 10^{-13}$	$3.74 \cdot 10^{-10}$	$2.21 \cdot 10^{-13}$	$2.01 \cdot 10^{-15}$	$8.46 \cdot 10^{-19}$
OPF	$4.09 \cdot 10^{-10}$	$2.32 \cdot 10^{-13}$	$4.09 \cdot 10^{-10}$	$2.32 \cdot 10^{-13}$	$2.18 \cdot 10^{-15}$	$8.73 \cdot 10^{-19}$

Figure 5: Global absolute average errors and standard errors for noise-free Spheres.

The second dataset, called ML, is generated from a function originally defined by Marschner and Lobb [Mar94]. This function is of particular interest because it is known to be band-limited but demanding on reconstruction tasks [Mar94]. The ML function has the form:

$$f(u, v, w) = \frac{(1 - \sin(\pi w/2) + \beta(1 + \rho(\sqrt{u^2 + v^2})))}{2(1 + \beta)} - 0.5, \quad (10)$$

where  $\rho$  is the function given by:

$$\rho(r) = \cos(2\pi f_M \cos(\frac{\pi r}{2})). \quad (11)$$

In the experiments we report here, we used  $\beta = 0.25$  with  $f_M = 6$ , which Marschner and Lobb chose because it placed a significant amount of the function's energy near the Nyquist frequency when sampled on a  $40 \times 40 \times 40$  grid in the range  $[-1.0, 1.0]$  [Mar94]. For our experiments, we sampled the function on a  $256 \times 256 \times 256$  grid in the range  $[-1.0, 1.0]$ , resulting in a volume containing values ranging from  $-0.5$  to  $0.5$ . Our experiments consider this volume with and without added Gaussian noise ( $\mu = 0$ ,  $\sigma = 0.001$ ). A rendering of a level surface from this volume is shown in Fig. 3 (right).

## 4.2 Accuracy Results

Accuracy experiment results (for the two methods) on the noise-free Spheres dataset are shown in Fig. 4. The figure shows the average absolute errors along a subset of Z slices. We note that since the methods exhibit very large error at edges of the dataset due to the convolution filters extending beyond the edges of the dataset, these plots exclude points within 10 units from the edges; the

plots are representative of the typical errors when the filters have support. Fig. 5 presents a table of the average global errors (again only in locations with support) and standard errors.

For this simple noise-free dataset, both **TEF** and **OPF** produce results with very low error, with similar error levels for  $\kappa_1$  and  $\kappa_2$  and smaller error levels for  $\kappa_3$ . This outcome is not unexpected since  $\kappa_1$  and  $\kappa_2$  are equal for this dataset and  $\kappa_3$  is much smaller. For all three curvatures, the **TEF** method exhibits slightly lower error.

Accuracy results for the noise-added Spheres dataset are shown in Fig. 6 and Fig. 7. In this case, Gaussian noise was added to the dataset prior to determining curvatures. In this noise-added case, despite the relatively small amount of noise, the **TEF** method exhibits more error than the **OPF** method. (A similar difference in error was noted for use of a Taylor Expansion-based strategy to determine conventional surface curvatures in volume data [Hau14].)

The average absolute errors for the noise-free ML dataset are shown in Fig. 8 and Fig. 9. Unlike Spheres, which can be exactly locally fit with a polynomial, this sinusoidal dataset cannot be exactly locally fit with a polynomial. Consequently, it is not surprising that **TEF** yielded more accurate curvatures since it employs higher degree polynomials (compared to **OPF**).

The average absolute errors for the noise-added ML are shown in Fig. 10 and Fig. 11. While in the noise-free case **TEF** exhibited much lower error compared to **OPF**, in this noise-added case **OPF** has much lower error. This is most likely due to the smoothing that implicitly occurs as a part of convolution with orthogonal polynomials.

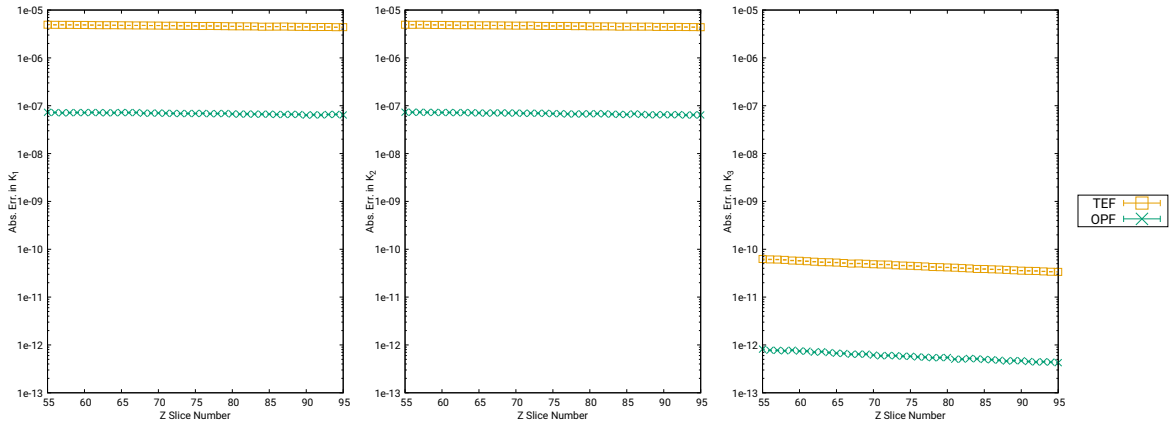


Figure 6: Per-slice average error in  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  for noise-added Spheres.

	$\kappa_1$ Avg. Abs. Err.	$\kappa_1$ Std. Err.	$\kappa_2$ Avg. Abs. Err.	$\kappa_2$ Std. Err.	$\kappa_3$ Avg. Abs. Err.	$\kappa_3$ Std. Err.
TEF	$4.04 \cdot 10^{-06}$	$1.04 \cdot 10^{-09}$	$4.04 \cdot 10^{-06}$	$1.04 \cdot 10^{-09}$	$3.88 \cdot 10^{-11}$	$7.23 \cdot 10^{-14}$
OPF	$5.97 \cdot 10^{-08}$	$1.49 \cdot 10^{-11}$	$5.97 \cdot 10^{-08}$	$1.49 \cdot 10^{-11}$	$5.00 \cdot 10^{-13}$	$8.76 \cdot 10^{-16}$

Figure 7: Global absolute average errors and standard errors for noise-added Spheres.

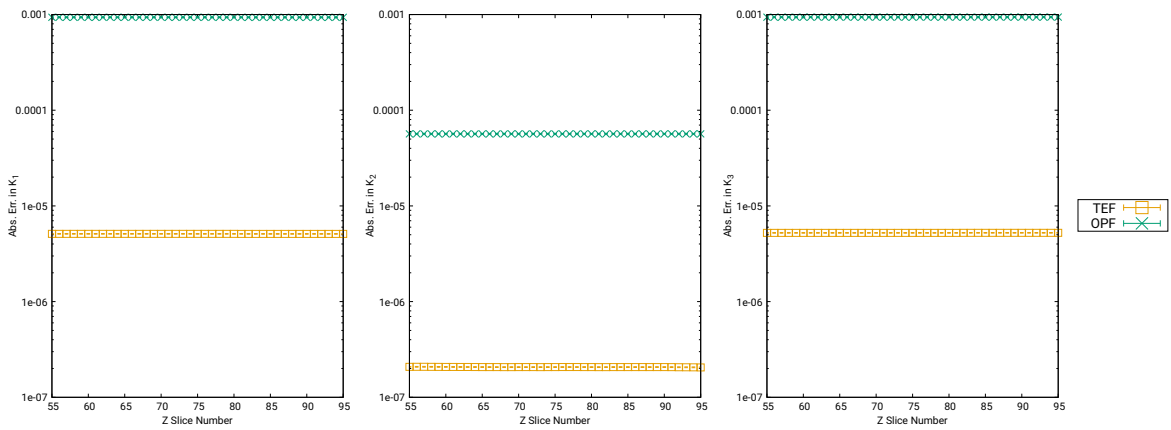


Figure 8: Per-slice average error in  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  for noise-free ML.

	$\kappa_1$ Avg. Abs. Err.	$\kappa_1$ Std. Err.	$\kappa_2$ Avg. Abs. Err.	$\kappa_2$ Std. Err.	$\kappa_3$ Avg. Abs. Err.	$\kappa_3$ Std. Err.
TEF	$5.09 \cdot 10^{-06}$	$2.31 \cdot 10^{-09}$	$2.09 \cdot 10^{-07}$	$1.23 \cdot 10^{-10}$	$5.24 \cdot 10^{-06}$	$2.39 \cdot 10^{-09}$
OPF	$9.34 \cdot 10^{-04}$	$3.74 \cdot 10^{-07}$	$5.66 \cdot 10^{-05}$	$2.08 \cdot 10^{-08}$	$9.42 \cdot 10^{-04}$	$3.75 \cdot 10^{-07}$

Figure 9: Global absolute average errors and standard errors for noise-free ML.

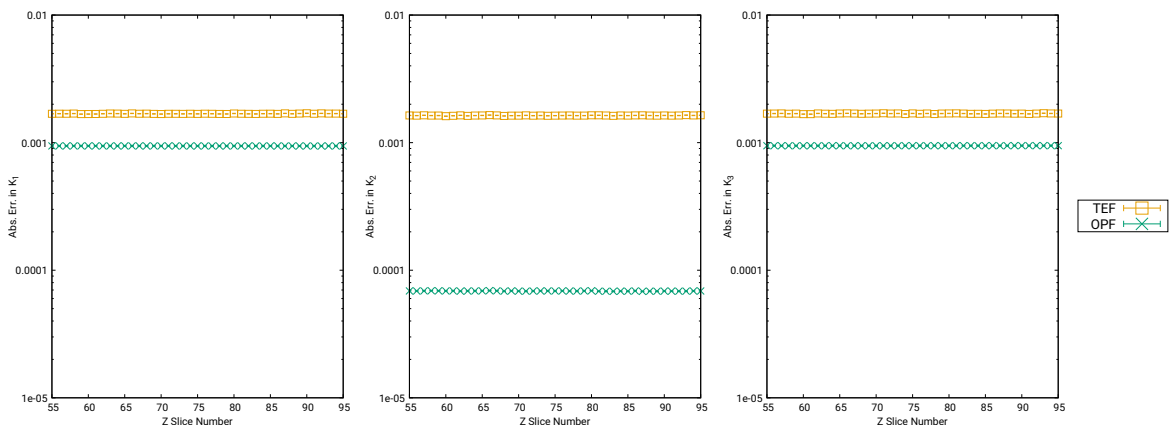


Figure 10: Per-slice average error in  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  for noise-added ML.

	$\kappa_1$ Avg. Abs. Err.	$\kappa_1$ Std. Err.	$\kappa_2$ Avg. Abs. Err.	$\kappa_2$ Std. Err.	$\kappa_3$ Avg. Abs. Err.	$\kappa_3$ Std. Err.
TEF	$1.68 \cdot 10^{-03}$	$3.50 \cdot 10^{-07}$	$1.63 \cdot 10^{-03}$	$3.40 \cdot 10^{-07}$	$1.69 \cdot 10^{-03}$	$3.50 \cdot 10^{-07}$
OPF	$9.40 \cdot 10^{-04}$	$3.73 \cdot 10^{-07}$	$6.87 \cdot 10^{-05}$	$1.89 \cdot 10^{-08}$	$9.48 \cdot 10^{-04}$	$3.74 \cdot 10^{-07}$

Figure 11: Global absolute average errors and standard errors for noise-added ML.

	Noise-Free Spheres	Noise-Added Spheres	Noise-Free ML	Noise-Added ML
TEF	58.30	64.32	57.60	64.98
OPF	103.30	106.84	103.10	106.72

Figure 12: Average run times (seconds).



Figure 13: An isosurface of a CT scan of the Stanford Bunny colored using a grayscale mapping of hypersurface curvature classes.

### 4.3 Run Time Results

Next, we report on computational performance of the methods on the four  $256 \times 256 \times 256$  datasets. Both methods were implemented in single-threaded C++ with the Armadillo library used for linear algebra operations. Testing was done on a machine running GNU/Linux equipped with 16GB of DDR3 RAM and an Intel i5-2310 processor. All code was compiled with g++. Run time (computational performance) results for each method on each dataset are reported in Fig. 12. These times represent the trimmed means of 10 runs with the fastest and slowest runs excluded.

**TEF** is substantially faster than **OPF**. Since both methods use the same linear algebra library, the time variation between the two is due only to the convolution step. While **OPF** convolves along all three axes at each point, even when measuring a derivative in only one direction, **TEF** only convolves along axes on which derivatives are being estimated. As a result, fewer convolutions

are performed in the **TEF** method, and it exhibits much faster run time.

## 5 VISUAL RESULTS

The superior resilience of **OPF** in the presence of noise motivates the selection of it when utilizing hypersurface curvatures on real data, and for that reason we have used its curvatures to produce the curvature classifications shown previously in Fig. 2, and we have additionally used it in related curvature classification visualization tasks utilizing sensed data. We describe those visualization tasks and their results next.

First, we consider a CT scan of the Stanford bunny (obtained from <http://graphics.stanford.edu/data/voldata/voldata.html>). In Fig. 13, we present a rendering of an isosurface of the bunny. This isosurface is colored using a grayscale mapping of hypersurface curvature classifications. The rendering thus presents a 2D

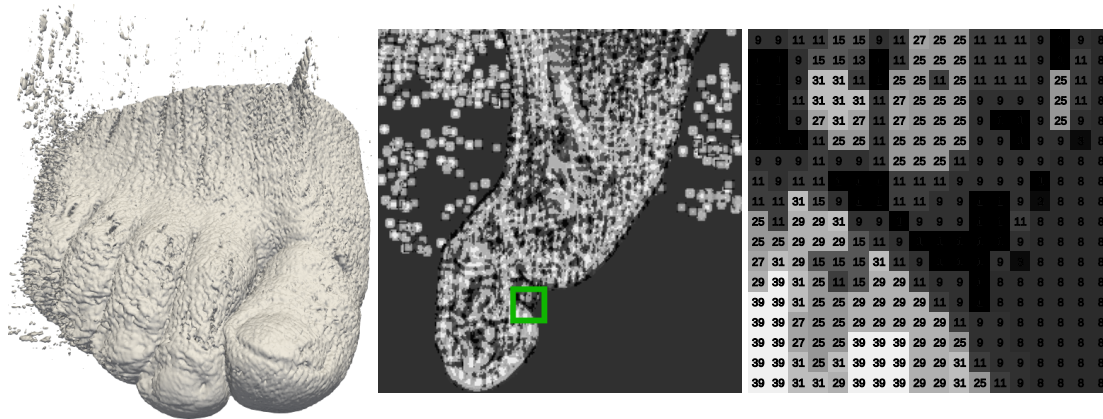


Figure 14: An isosurface rendering of a CT scan of a human foot (left), the resulting data point-by-data point classifications of hypersurface curvatures according to Fig. 1 (middle) for one slice, and zoomed-in detail (right) for that slice. The classes are visualized (in middle and right) using a grayscale mapping.

manifold (the isosurface) colored using the curvature values of the 3D manifold present at each isosurface point. The grayscale mapping used is identical to the one used in the middle and right of Fig. 2. As this is sensed data, some noise is clearly present in the classification results (e.g., a notable stair-stepping can be seen on the face and ears of the bunny), however, many features are also clearly indicated from changes in hypersurface curvature classification, including the bunny's whiskers, eye socket, and toes.

Second, we consider a CT scan of a human foot (obtained from <http://volvis.org>). An isosurface rendering of the foot is shown in the left part of Fig. 14. In the middle of that figure, grayscale mapped hypersurface curvature classifications of z-slice 127 (located on the index toe next to the big toe) of the dataset are shown (again using the same mapping as in Fig. 2). The green box shown in the middle of the figure represents the location of the zoomed-in region shown in the right of the figure. This CT scan exhibits much noise, as can be seen in both the isosurface rendering and the hypersurface curvature classifications. Consequently, it is difficult to visually locate fine features within the slice.

## 6 CONCLUSION

In this paper we have exhibited two methods for determining hypersurface curvature within volume datasets. Both of these methods use convolution to estimate derivatives and then these estimated derivatives are used to determine the three hypersurface curvatures. One method, **TEF**, uses convolution with filters based on the Taylor Expansion. The other, **OPF**, uses convolution with filters sampled from orthogonal polynomials. We have additionally comparatively evaluated both these methods in terms of (1) accuracy and (2) run time.

In summary, in our tests we found that **TEF** exhibits relatively low error levels when no noise is present. However, when (even relatively low level) noise is present,

**OPF** generally exhibits much lower error levels than **TEF**. However, **TEF** exhibits much faster run times than **OPF** in both noise-free and noise-added cases. Thus, if time is critical or no noise is present in the dataset, **TEF** is the preferable choice. If noise is present and accuracy is critical, **OPF** is a better choice.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for contributing their input and insight. These contributions helped to greatly improve the paper.

## 7 REFERENCES

- [Ald14] G. Aldrich, A. Gimenez, M. Oskin, R. Strelitz, J. Woodring, L. H. Kellogg, and B. Hamann. Curvature-based crease surfaces for wave visualization. *Vision, Modeling & Vis.*, pp. 39–46, 2014.
- [AR18] R. Al-Rousan, M. S. Sunar, and H. Kolivand. Geometry-based shading for shape depiction enhancement. *Multimedia Tools and Appl.*, 77(5):5737–5766, Mar 2018.
- [Bag16] P. Bagchi, D. Bhattacharjee, and M. Nasipuri. A robust analysis, detection and recognition of facial features in 2.5d images. *Multimedia Tools and Appl.*, 75(18):11059–11096, Sep 2016.
- [Bel12] A. Belyaev. Focal surfaces, skeletons, and ridges for shape analysis and processing. *Proceedings of the 2012 Joint International Conference on Human-Centered Computer Environments, HCCE '12*, pp. 58–58, 2012.
- [Bes86] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3d object recognition in range images. *Comput. Vision, Graphics, Image Processing*, 33(1):33 – 80, 1986.
- [Bib16] P. Bibiloni, M. González-Hidalgo, and S. Massanet. A survey on curvilinear ob-



- ject segmentation in multiple applications. *Pattern Recognition*, 60:949–970, Dec 2016.
- [Du18] G. Du, C. Yin, M. Zhou, Z. Wu, and F. Duan. Part-in-whole matching of rigid 3D shapes using geodesic disk spectrum. *Multimedia Tools and Appl.*, 77(15):18881–18901, Aug 2018.
- [Fly89] P. Flynn and A. Jain. On reliable curvature estimation. *Proc., IEEE Comput. Vision and Pat. Recog.*, pp. 110 – 116, 1989.
- [Ham94] B. Hamann. Curvature approximation of 3D manifolds in 4D space. *Comput. Aided Geometric Design*, 11(6):621 – 632, 1994.
- [Hau14] J. D. Hauenstein and T. S. Newman. On reliable estimation of curvatures of implicit surfaces. *Proc., 2nd Int'l Conf. 3D Vision (3DV)*, pp. 697 – 704, Dec 2014.
- [Hau18] J. D. Hauenstein and T. S. Newman. Curvature determination in range images: new methods and comparison study. *Multimedia Tools and Appl.*, Advance online publication, Aug 2018.
- [Hir01] Y. Hirano, A. Shimizu, J.-i. Hasegawa, and J.-i. Toriwaki. A tracking algorithm for extracting ridge lines in three-dimensional gray images using curvature of four-dimensional hypersurface. *Systems and Comput. in Japan*, 32(12):25–37, 2001.
- [Hir18] Y. Hirano. Categorization of lung tumors into benign/malignant, solid/GGO, and typical benign/others. K. Suzuki and Y. Chen, editors, *Artificial Intelligence in Decision Support Systems for Diagnosis in Medical Imaging*, pp. 193–208. 2018.
- [Kin03] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: methods and applications. *Proc., Vis. '03*, pp. 513 – 520, 2003.
- [Kot18] K. Kottari, K. Delibasis, and V. Plagianakos. Real time vision-based measurements for quality control of industrial rods on a moving conveyor. *Multimedia Tools and Appl.*, 77(8):9307 – 9324, Apr 2018.
- [Lan07] T. Langer, A. Belyaev, and H.-P. Seidel. Exact and interpolatory quadratures for curvature tensor estimation. *Computer Aided Geometric Design*, 24(8):443 – 463, 2007.
- [Lef17] D. Lefloch, M. Kluge, H. Sarbolandi, T. Weyrich, and A. Kolb. Comprehensive use of curvature for robust and accurate online surface reconstruction. *IEEE T-Pat. Anal. Machine Int.*, 39(12):2349–2365, Dec 2017.
- [Lef18] L. Lefkovits and S. Lefkovits. Two-phase MRI brain tumor segmentation using random forests and level set methods. *Short Papers Proc., Int'l Conf. in Central Europe on Comput. Graphics, Vis. and Comput. Vision (WSCG2018)*, pp. 152–159, 2018.
- [Mar94] S. Marschner and R. Lobb. An evaluation of reconstruction filters for volume rendering. *Proc., Vis. '94*, pp. 100 – 107, 1994.
- [Möl98] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of accurate and smooth filters for function and derivative reconstruction. *Proc., IEEE Symp. Volume Vis. 1998*, pp. 143 – 151, 1998.
- [Mon92] O. Monga and S. Benayoun. Using partial derivatives of 3D images to extract typical surface features. Research Report RR-1599, INRIA, 1992.
- [Pap07] L. Papaleo. An approach to surface reconstruction using uncertain data. *Int'l J. of Image and Graphics*, 07(01):177–194, 2007.
- [Pre16] B. Preim, A. Baer, D. Cunningham, T. Isenberg, and T. Ropinski. A survey of perceptually motivated 3D visualization of medical image data. *Comput. Graphics Forum*, 35(3):501–525, 2016.
- [Sou16] M. Soufi, H. Arimura, K. Nakamura, F. P. Lestari, F. Haryanto, T.-a. Hirose, Y. Umedu, Y. Shioyama, and F. Toyofuku. Feasibility of differential geometry-based features in detection of anatomical feature points on patient surfaces in range image-guided radiation therapy. *Int'l J. of Comput. Assisted Radiology and Surgery*, 11(11):1993–2006, 2016.
- [Suz18] H. Suzuki, Y. Kawata, N. Niki, T. Sugiura, N. Tanabe, M. Kusumoto, K. Eguchi, and M. Kaneko. Automated assessment of aortic and main pulmonary arterial diameters using model-based blood vessel segmentation for predicting chronic thromboembolic pulmonary hypertension in low-dose ct lung screening. *Medical Imaging 2018: Comput.-Aided Diagnosis*, volume 10575, 2018.
- [Sya17] M. A. Syarif, T. S. Ong, A. B. J. Teoh, and C. Tee. Enhanced maximum curvature descriptors for finger vein verification. *Multimedia Tools and Appl.*, 76(5):6859–6887, Mar 2017.
- [Vin17] A. M. Vintescu, F. Dupont, and G. Lavoué. Least squares affine transitions for global parameterization. *Journal of WSCG*, 25(01):21–30, 2017.
- [Yos12] S. Yoshizawa, A. Belyaev, and H. Yokota. Shape and image interrogation with curvature extremalities. *J. for Geometry and Graphics*, 16:81–95, Jan 2012.

