

# An Examination of the Effects of Noise Level on Methods to Determine Curvature in Range Images

Jacob D. Hauenstein, Timothy S. Newman; The University of Alabama in Huntsville; Huntsville, AL, USA

## Abstract

A study of the impact of image noise on well-known range image curvature determination methods is presented here. The study considers 12 methods, and each is analyzed based on its performance at varying levels of input noise. The performance analyses consider quality factors of (1) absolute error, (2) correlation with correct, expected curvature values, and (3) signal-to-noise ratio (SNR). Curvature-based renderings are also presented for some data to provide basic visualizations of the impact of noise on one curvature-based task. The work can benefit tasks using range data (e.g., from Kinect or commercial-grade sensors).

## Introduction

Curvature is a commonly used surface shape measure. It has previously been shown to be useful for segmentation [1], object recognition [2], face analysis [3], surface description [4] and reconstruction [5], and a multitude of other tasks. These tasks often utilize sensed data (for example, data from consumer- or commercial-grade range scanners) from which exact curvature calculation is impossible due to the presence of noise and measurement error. While exact computation is impossible in these cases, there exist a number of strategies capable of determining, with varying accuracy, curvature within such data [6].

The focus in this work is on the analysis of the impact of varying noise levels on twelve such curvature determination methods for range data (range images). The analysis here considers synthetic range images of two scenes, with each scene generated at varying levels of noise. At each noise level, the quality of curvature values produced by each method is assessed using average absolute error in determined curvatures and correlation of determined curvatures with actual curvature. The relationship to signal-to-noise ratio (SNR) is also considered. Prior studies have not considered relative sensitivity to a range of noise levels of a wide variety of curvature determination methods; this work provides new insights into method quality.

This paper is organized as follows. The following section provides a brief background on both surface curvature generally as well as each curvature determination method considered here. Some previous studies of curvature accuracy are considered there as well. The subsequent sections describe the range images tested, testing methodology, and results, including some curvature-based renderings.

## Background and Previous Work

In this section, we provide a short a background on surface curvature and existing methods for determining surface curvature in range images.

## Surface Curvature

The *principal curvatures* are two common descriptors for expressing surface curvature in range images. They are denoted by  $\kappa_1$  (for the larger of the two) and  $\kappa_2$  (for the smaller). At each point on the surface, their magnitudes describe the “bending” of the surface away from the surface tangent plane, and their signs describe the direction of the bend relative to that plane. Mathematically, calculation of surface curvature at a surface point requires knowledge of the first and second derivatives of the surface function at the point.

Two other common curvature measures, often defined as functions of  $\kappa_1$  and  $\kappa_2$ , are the mean curvature,  $H = \frac{1}{2}(\kappa_1 + \kappa_2)$ , and the Gaussian curvature,  $K = \kappa_1 \times \kappa_2$ . These curvatures can be computed directly from the surface derivatives, or they can be computed from  $\kappa_1$  and  $\kappa_2$ .

## Existing Accuracy Studies

Since, in practice, the derivatives necessary to exactly compute curvature may be unknown, as in the case of acquired data (e.g., from the Microsoft Kinect), a number of methods to determine curvature from such data have been devised. And a few studies of these methods in terms of accuracy and/or execution time have been reported previously (e.g., [7, 8, 6]). These previous studies have focused on a limited number of noise levels and/or determination methods. (E.g., in [6], for most methods we considered only one noise level versus no noise.) Our work here provides a study of method performance as noise level increases for all methods.

## Curvature Determination Methods

Here, we briefly describe each of the curvature determination methods considered. Broadly, there are three strategies used by these methods. One is computing curvature by first fitting a continuous surface to the range image, then computing derivatives from this continuous representation, and finally using those derivatives to estimate curvatures. The first four methods use such a strategy. The second strategy, used by the next four methods, is to use convolution with derivative-estimating kernels and compute curvatures using these estimated derivatives. The other strategy is a hybrid strategy that combines elements of the first and second strategies. The last four methods use such a strategy. Some of the methods considered here were designed originally to determine the principal curvatures. Others were designed to determine  $H$  and  $K$ . Since  $H$  and  $K$  can be computed from the principal curvature values, and vice-versa, here we take only  $\kappa_1$  and  $\kappa_2$  equivalences, allowing all 12 determination methods to be compared on a common basis. As space here is limited, we refer to our previous work [6] for further details (e.g., implementation details and parameter selection motivation) on each method. Im-

plementations and parameter choices here are identical to [6].

Flynn and Jain [7] described a B-Spline-based method that determines curvatures by first fitting a B-Spline using linear regression. This fitting produces a continuous representation of the range image from which derivatives and ultimately curvatures are computed at each point in the range image. We denote this method **BSF**.

The second method [6] determines curvatures at each point within a range image using B-Splines, but without the linear regression fitting step required by the **BSF** approach; it directly uses the range image data as the B-Spline coefficients, providing a continuous representation of the data from which curvatures are computed. We denote this method **SBS**.

The third method [6] determines curvatures by first fitting a Catmull-Rom spline to the range image. From this continuous representation of the data, derivatives (and thus curvatures) are determined. We denote this method **CRS**.

The fourth method [7] is linear regression-based. It first solves for quadratic surface coefficients using linear regression within a neighborhood around each image point. Curvatures are determined from this representation. We denote this method **LR**.

The fifth method, from Besl and Jain [9], uses convolution with orthogonal polynomials to directly estimate derivatives from the range image. Curvatures are then computed from these estimated derivatives. We denote this method **OP**.

The sixth method, from Fan et al. [10, 11], is convolution-based. It uses four directionally sensitive convolution kernels to obtain four directional derivatives at each point in the range image. Using these four directional derivatives, curvatures are computed. We denote this method **DE**.

The seventh method [6] first performs convolution with kernels based on the Taylor Expansion. Derivatives estimated from that are used in curvature determination. We denote this method **TE**.

The eighth method [6] uses convolution with 3D Deriche Filters to provide estimates of the first and second derivatives. From those estimates, the principal curvatures are computed. We denote this method **DF**.

The ninth method, from Hoffman and Jain [12], first estimates normals at each image point. Then, changes in the normal at each point are used to determine the curvatures. We denote this method **DN**.

The tenth method, from Martin [13], considers triplets of points formed within a neighborhood of each range image point. A circle is fit to each triplet of points, providing an ensemble of curvature estimates. The principal curvatures are then determined at each point using its ensemble. We denote this method **CF**.

The eleventh method, from Yang and Qian [14], was described for determining curvatures in point cloud data. It can also be applied to range images. In the approach, each point is projected to a moving least squares (MLS) surface, and curvatures are computed directly from this MLS surface. We denote this method **MLS**.

The twelfth method, from Zhang et al. [15], was also described for point cloud data. (It can also be applied to range images.) It first computes a number of normal curvatures at each point. These normal curvatures are then used to find the curvatures at each point. We denote it **CAN**.

## Experiments

As our goal with this work is to provide an analysis of determined curvature quality at a variety of noise levels, our experiments are limited to range images with exactly known curvatures, which allows comparison of each method's results versus the known curvatures. Specifically, we focus on two synthetic range image datasets. One consists of a sinusoidally varying surface. The other consists of a cylinder. Our experiments here focus on error levels in the principal curvatures.

The sinusoidally varying surface range images are generated using a function defined by Marschner and Lobb [16]. That function, which generates a volume rather than a range image, is:

$$g(u, v, w) = \frac{(1 - \sin(\pi w/2) + \beta(1 + \rho_s(\sqrt{u^2 + v^2})))}{2(1 + \beta)} - 0.5, (1)$$

where  $\rho_s$  is given by:

$$\rho(s) = \cos(2\pi f_M \cos(\frac{\pi s}{2})). (2)$$

We used  $\beta = 0.25$  and  $f_M = 6$ , following [16]. We generated a  $256 \times 256$  range image from this volume in which each image point records its vertical distance to the level 0 isosurface (i.e.,  $g(u, v, w) = 0$  surface). We denote this dataset **ML**. In the table below, we show some statistics about the resulting range image. The exact curvatures of the **ML** dataset were determined based on known derivatives (found using the Sage mathematics software [17]) calculated at each point.

Table 1: ML Dataset Statistics

Max	0.16
Min	-0.16
Mean	0.000079
Median	0.0016
Std. Dev.	0.11

The cylinder range image (henceforth denoted **Cylinder**) was generated at a size of  $128 \times 128$ . It consists of a single cylinder of radius 170 oriented along the x-axis. The cylinder's principal curvatures are therefore (by definition):

$$\kappa_1 = \frac{1}{170}; \kappa_2 = 0. (3)$$

In the table below, we show some statistics about the **Cylinder** dataset.

Table 2: Cylinder Dataset Statistics

Max	12.51
Min	0.000038
Mean	4.11
Median	3.04
Std. Dev.	3.71

To analyze the performance of each curvature method at a variety of noise levels, we considered noise-free and noise-added versions of each of these range images at a number of different levels of Gaussian noise. The mean ( $\mu$ ) of the Gaussian noise

was kept at zero, and the standard deviation ( $\sigma$ ) was varied. In all cases, the image noise was added prior to curvature determination, and no pre-process smoothing or any other pre-process noise mitigation efforts were attempted. Our error analysis focused on the average absolute error exhibited by each method. We explored the signal-to-noise ratio (SNR) exhibited by each dataset at each noise level.

We also did a subjective analysis of curvature-based renderings of the ML datasets, which illustrate the impact of noise on a common curvature-based task.

## Results

In this section we present the results of those experiments.

### ML

Fig. 1 shows the average absolute error of each curvature determination method when operating on the ML dataset at a variety of noise levels, with each solid line representing a curvature determination method and the left y-axis showing the average absolute error level. Also shown is the SNR at each noise level, represented by the dotted blue line corresponding with the right y-axis. The leftmost data points represent the noise-free version of the dataset (shown here as  $\sigma = 0$ ), and thus exhibit relatively low errors and an infinite SNR. The rightmost datapoint represents a noise-added version of the dataset with  $\sigma = 1.0$  and an SNR of about  $-18.9$ . The errors shown here are just for  $\kappa_1$  values; the  $\kappa_2$  chart looks very similar.

As noise increases, every method exhibits an approximately linear increase in average absolute error in curvature up to around  $\sigma = 0.25$  (SNR =  $-6.86$ ). However, the slope of the linear relationship differs for each method. **CRS** scales worst with noise, exhibiting a slope of 5.15. In contrast, **CAN**, one of the methods which scales best with noise, exhibits a slope of just 0.01. As a result, the error performance of the methods diverges as error increases. (**MLS**, **DF**, and **OP** also exhibit low slopes.)

It is worth noting that the linear increase in error in curvature continues even past the point of SNR becoming negative (which happens at around  $\sigma = 0.15$ ). In fact, as shown in Fig. 2, a simple curvature-based rendering (where the mean curvature at each point is mapped according to a colormap) of a selection of the best performing methods clearly shows that a substantial amount of curvature detail is preserved even at this noise level. This showing is surprising (1) due to negative SNR and (2) because for even the most accurate of these methods, the median relative error in curvature is 1.5, which implies that relative error is likely not particularly indicative of the utility of a curvature measure.

Fig. 3 shows each method's correlation with the expected curvatures for the same set of noise levels as presented in Fig. 1. **CAN**, **DF**, **OP**, and **MLS**, which exhibit among the lowest errors across the range of noise levels, also exhibit the strongest correlations with the correct curvatures.

### Cylinder

Fig. 4 shows the average absolute error of each curvature determination method when operating on the Cylinder dataset at a variety of noise levels. The chart style is identical to the ML case, with solid lines representing the average absolute error for each method, and a single dashed line representing SNR. Again, this chart shows only  $\kappa_1$  errors, but the  $\kappa_2$  chart is essentially identical.

Here, methods again exhibit linearly increasing errors as  $\sigma$  increases, although the error increases more slowly for  $\sigma$  above 0.5. This error behavior may be due to a wider range of values exhibited by the Cylinder function (compared to ML). While ML exhibits a standard deviation of 0.11, the standard deviation here is 3.71. As a result, even though the overall signal strength is still stronger than the noise (i.e., the SNR is positive) at  $\sigma = 0.5$ , this wider range of values implies that some of the smaller values within the Cylinder range image will have large relative noise levels. As a result, there is less uniformity in the relative noise levels across the image, and, consequently, even when the SNR is still relatively high, some data values exhibit extremely high relative noise. The result is that curvatures are gradually dominated by noise (as more and more points in the range image are gradually dominated by noise), and the error in curvature flattens out with increasing  $\sigma$ .

## Conclusion

While curvature is a useful quantity for a number of tasks, its estimation is often complicated by various levels of noise present in the range images considered. Here, we have applied a large number of existing curvature determination methods to two range images at a variety of noise levels. We observed that average absolute error in determined curvatures tends to vary linearly with input noise, up to a point. The primary difference in the behaviors of methods is the slope of the linear relationship with noise. From tests here, it appears the **CRS** approach tends to degrade more rapidly with noise than do other methods. The **CAN**, **MLS**, **DF**, and **OP** methods appear to degrade the most gracefully.

Neither SNR nor relative error seem to be useful measures for assessing the usability of curvature results. In the ML experiment, for example, some methods are able to produce mostly usable renderings even at data levels that result in a negative SNR and very high median relative error.

## References

- [1] T. Atta-Fosu, W. Guo, D. Jeter, C. M. Mizutani, N. Stopeczynski, and R. Sousa-Neves, "3D clumped cell segmentation using curvature based seeded watershed," *J. Imaging*, vol. 2, no. 4, 2016.
- [2] L.-C. Chen, H. H. Hai, X.-L. Nguyen, and H.-W. Wu, "Novel 3-D object recognition methodology employing a curvature-based histogram," *Int'l J. of Advanced Robotic Systems*, vol. 10, no. 7, p. 294, 2013.
- [3] S. Ganguly, D. Bhattacharjee, and M. Nasipuri, "Fuzzy matching of edge and curvature based features from range images for 3D face recognition," *Int. Automation & Soft Comput.*, vol. 23, no. 1, pp. 51–62, 2017.
- [4] A. J. Rodríguez-Sánchez, S. Szedmak, and J. Piater, "SCurV: A 3D descriptor for object classification," in *Proc., IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS) 2015*, pp. 1320–1327, 2015.
- [5] 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.*, vol. 39, pp. 2349–2365, Dec. 2017.
- [6] J. Hauenstein and T. Newman, "Curvature determination in range images: new methods and comparison study," *Multimedia Tools and Appl.*, 2018. Accepted for publication.

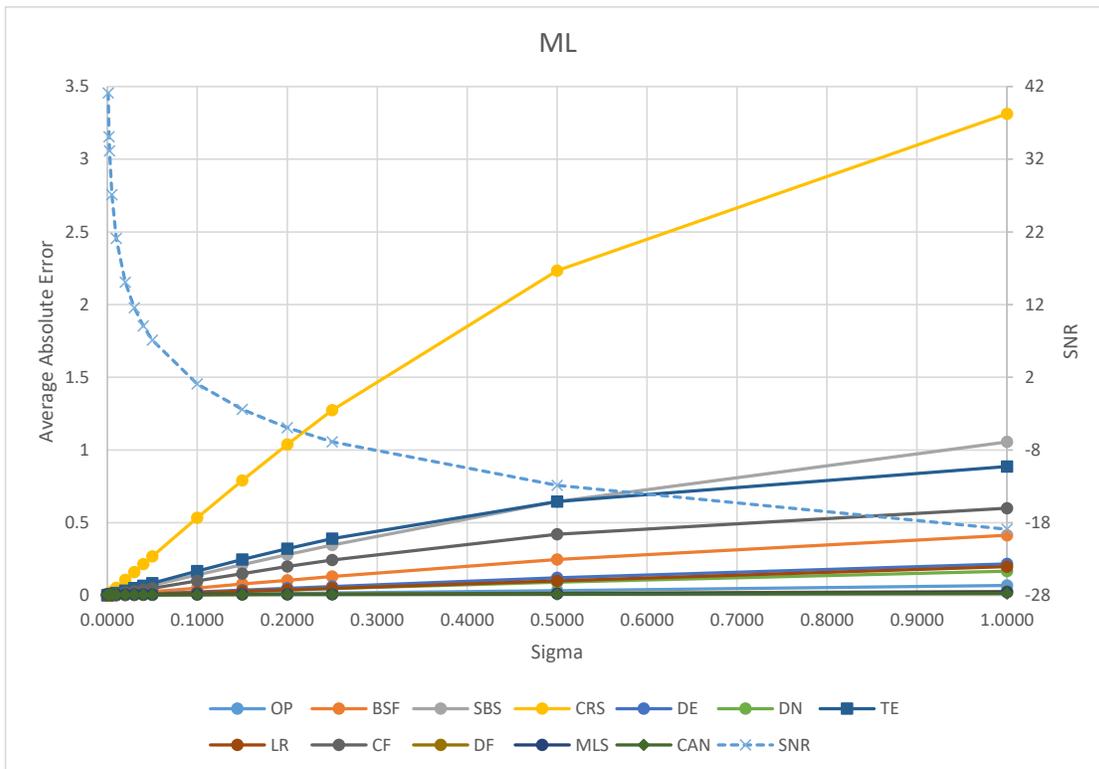


Figure 1: Average absolute error of each method at each noise level on the ML dataset. SNR is also shown as a dotted blue line. The increase in error with noise is approximately linear for each method up to about  $\sigma = 0.25$ , despite the SNR being negative already by  $\sigma = 0.15$ .

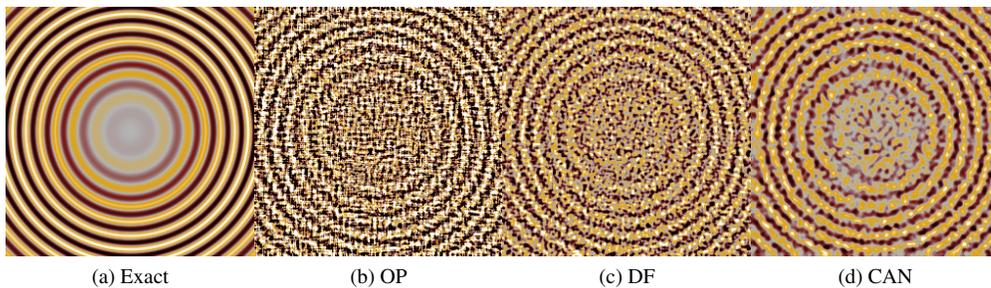


Figure 2: Curvature-based renderings of the ML dataset. (a) “correct” rendering achieved when using exact curvatures. (b,c,d) renderings achieved using curvatures from OP, DF, and CAN when  $\sigma = 0.25$ . Despite the extremely high level of noise and the high relative errors, these renderings still resemble the correct one.

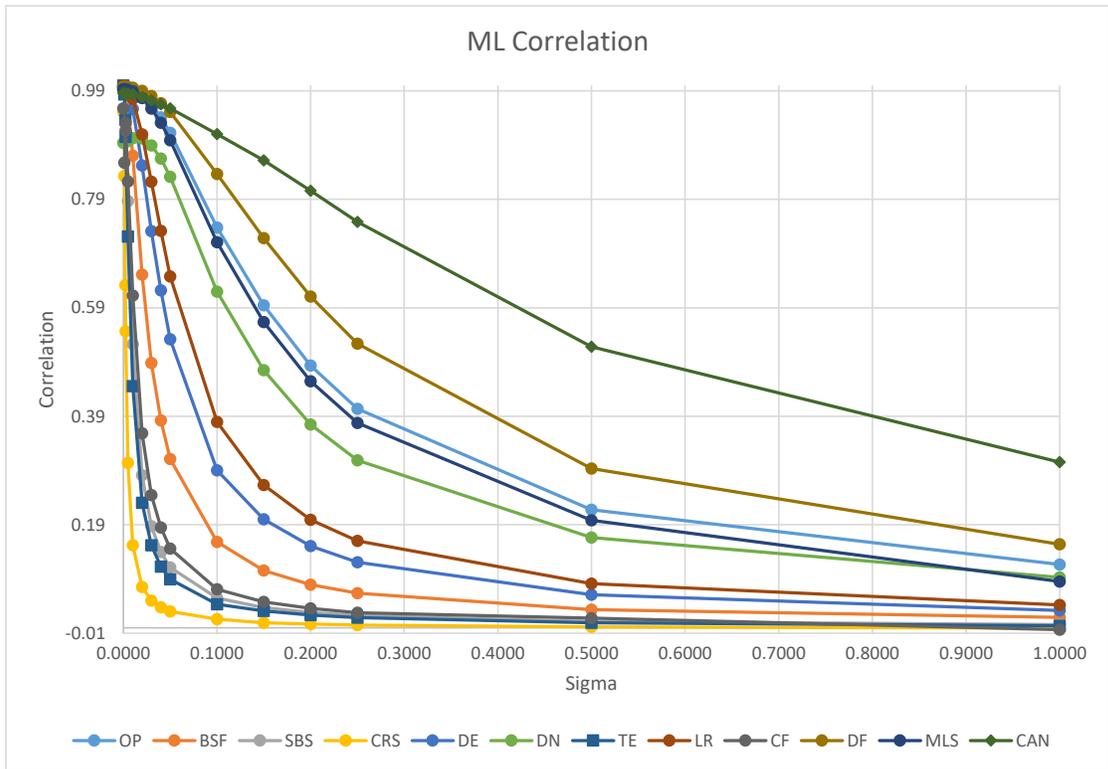


Figure 3: Correlation with correct curvatures of each method at each noise level on the ML dataset.

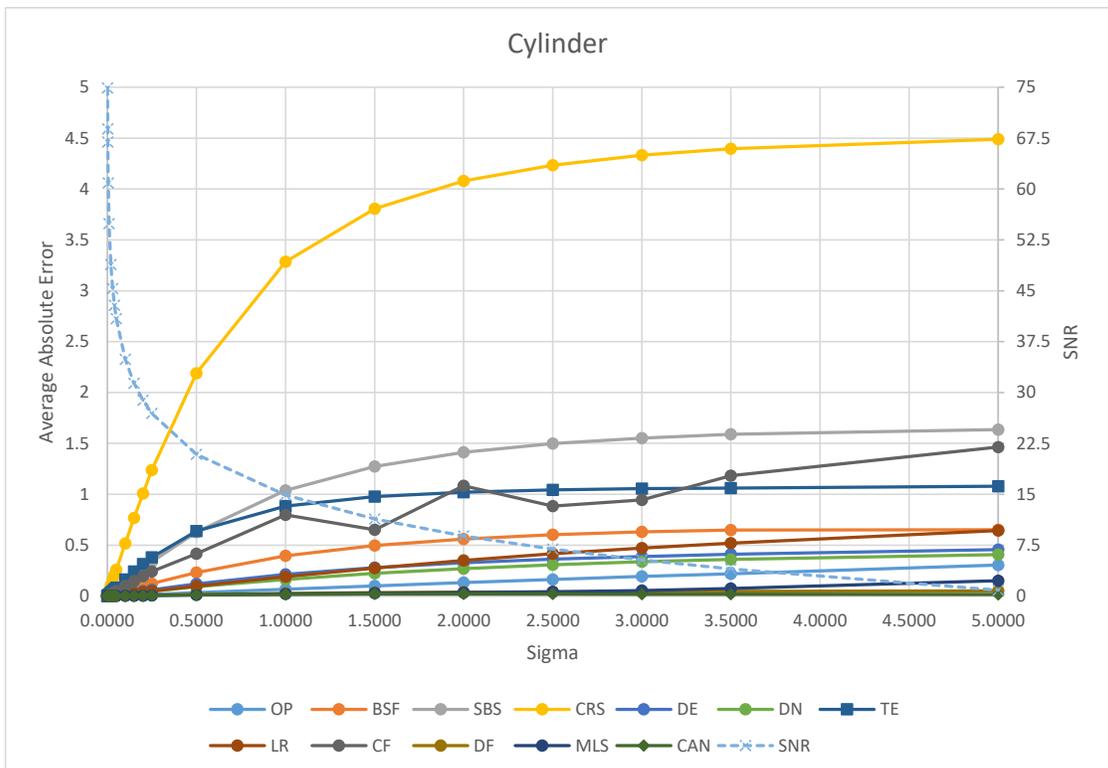


Figure 4: Average absolute error of each method at each noise level on the Cylinder dataset. SNR is also shown as a dotted blue line. Here, each method is linear only up to about  $\sigma = 0.5$ .

- [7] P. Flynn and A. Jain, "On reliable curvature estimation," in *Proc., IEEE Comput. Vision and Pat. Recog. 1989*, pp. 110 – 116, 1989.
- [8] P. Krsek, G. Lukcs, and R. R. Martin, "Algorithms for computing curvatures from range data," in *The Mathematics of Surfaces VIII, Information Geometers*, pp. 1–16, 1998.
- [9] P. J. Besl and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range images," *Comput. Vision, Graphics, Image Processing*, vol. 33, no. 1, pp. 33 – 80, 1986.
- [10] T. Fan, G. Medioni, and R. Nevatia, "Description of surfaces from range data using curvature properties," in *Proc., IEEE Comput. Vision and Pat. Recog. 1986*, pp. 86 – 91, 1986.
- [11] T. Fan, G. Medioni, and R. Nevatia, "Surface segmentation and description from curvature features," in *Proc., DARPA Image Understanding Work. 1987*, pp. 351 – 359, 1987.
- [12] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE T-Pat. Anal. Machine Int.*, vol. 9, no. 5, pp. 608 – 620, 1987.
- [13] R. R. Martin, "Estimation of principal curvatures from range data," *Int'l J. Shape Modeling*, vol. 04, no. 03n04, pp. 99–109, 1998.
- [14] P. Yang and X. Qian, "Direct computing of surface curvatures for point-set surfaces," in *Proc., IEEE/Eurographics Symp. on Point-based Graphics (SPBG) 2007*, pp. 29 – 36, 2007.
- [15] X. Zhang, H. Li, and Z. Cheng, "Curvature estimation of 3d point cloud surfaces through the fitting of normal section curvatures," in *Proc., ASIAGRAPH 2008*, pp. 72 – 79, Oct 2008.
- [16] S. Marschner and R. Lobb, "An evaluation of reconstruction filters for volume rendering," in *Proc., Vis. '94*, pp. 100 – 107, 1994.
- [17] W. A. Stein and The Sage Authors, *Sage Mathematics Software (Ver. 5.4.1)*. The Sage Development Team, 2012. <http://www.sagemath.org>.

## Author Biography

*Jacob D. Hauenstein is a Ph.D. student in the Computer Science Department at the University of Alabama in Huntsville (UAH). He received his M.S. from UAH. Among Jacob's research interests are curvature estimation and range imaging.*

*Timothy S. Newman is Professor of Computer Science at the University of Alabama in Huntsville (UAH). He received his Ph.D. from Michigan State University. He was also a National Research Council postdoctoral fellow at the National Institutes of Health. Range image processing and understanding is one of the several research interests of Dr. Newman.*

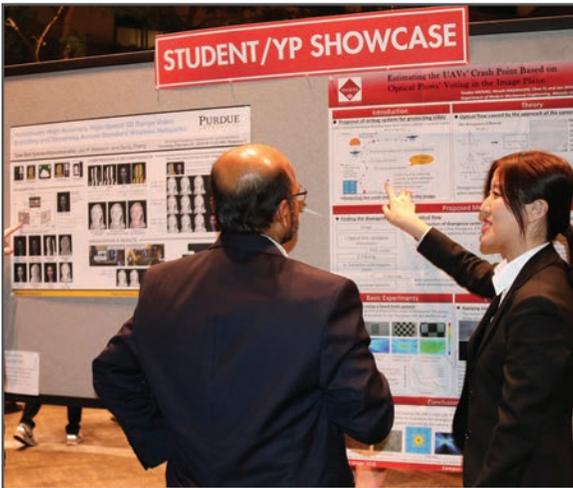
**JOIN US AT THE NEXT EI!**

IS&T International Symposium on

# Electronic Imaging

SCIENCE AND TECHNOLOGY

*Imaging across applications . . . Where industry and academia meet!*



- **SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
- **INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

[www.electronicimaging.org](http://www.electronicimaging.org)

