LINEAR APPROXIMATION OF MEAN CURVATURE

Yuanhao Gong

ETH Zurich, Switzerland

Yuan Xie

Chinese Academy of Sciences, China

ABSTRACT

Mean curvature has been shown a good regularization for many image processing tasks. Computing mean curvature, however, usually requires the image at least twice differentiable, which is an issue for discrete images, especially at edges. In this paper, we present several linear schemes to approximate the mean curvature of discrete images, based on Euler Theorem from differential geometry. We further compare these schemes with the traditional formula in terms of accuracy, computational efficiency, convexity, etc. The experiments confirm that these schemes are good approximations to the mean curvature of discrete images.

***Index Terms*—** mean curvature, curvature filter, linear approximation, weighted mean curvature, convex

1. INTRODUCTION

Various problems in image processing are ill-posed, such as denoising [1, 2, 3, 4, 5, 6, 7], scatter light removal [8, 9, 10], enhancement [11, 12, 13], etc. These tasks usually need to estimate the unknown ground truth $U(\vec{x})$ from the observed image data $I(\vec{x})$, where $\vec{x} \in \Omega$ is the coordinate in the imaging domain Ω. This process can be formulated in Bayesian framework, which maximizes the following probability

$$p(U|I) = \frac{p(I|U)p(U)}{p(I)} \propto p(I|U)p(U)\,, \tag{1}$$

where $p(\cdot)$ and $p(\cdot|\cdot)$ indicate probability and conditional probability, respectively. If the prior term $p(U)$ in Bayesian framework has a proper mathematical expression, it leads to a regularization term in the corresponding variational model.

There are many regularization for image processing. For example, the very popular one is Total Variation (TV) [1, 14], which assumes the gradient of the ground truth is small. This implicitly imposes the sparsity of the signal. Other regularizations include gradient distribution prior [10, 9], mean curvature [15, 16, 17], Gaussian curvature [2], difference curvature [18], spatial weighted regularization, adaptive norm regularization [19], symmetry [20], etc. Choosing a proper regularization term becomes the central stage, because it provides the prior knowledge about the ground truth.

Among these regularizations, mean curvature has been shown a good regularization for several reasons. First, it assumes the ground truth is a minimal surface, which is very common in our physical world [10]. The statistics of mean curvature from natural scene images confirms that mean curvature indeed should be minimized [10]. Second, mean curvature regularized models lead to better results in practice, compared with that from TV regularized models [21, 16]. This fact is theoretically explained in [10, 7, 4]. Third, as shown in this paper, the linear approximation of mean curvature leads to a convex regularization term, which indicates the convexity of the full model if the data fitting term is also convex. In all these models, computing mean curvature is an important step.

In this paper, we present several linear schemes to approximate mean curvature for discrete images. First, we show the Euler Theorem from differential geometry, which is the theoretical foundation of these linear schemes. Then, we show how to construct these schemes from Euler Theorem. We further analyze their accuracy, computational complexity, convexity, etc. We benchmark them on synthetic dataset and also on real images. The results confirm that these schemes are good approximation to the mean curvature.

1.1. Mean Curvature

We embed the two dimensional image into three dimensional space, forming a surface [15]. Specifically, we embed the data $U(\vec{x})$ into a higher dimension, forming a surface $\Psi(\vec{x}) = (\vec{x}, U(\vec{x}))$. It is worth pointing out that $U(\vec{x})$ and $(\vec{x}, U(\vec{x}))$ can be uniquely mapped between each other. Although we only show the case for two dimensional image here, this representation is also valid for higher dimensional images.

With this surface representation, the first and second fundamental forms of Ψ are

$$\boldsymbol{F} = \begin{pmatrix} 1+U_x^2, & U_xU_y \\ U_xU_y, & 1+U_y^2 \end{pmatrix} \tag{2}$$

and

$$\boldsymbol{D} = \begin{pmatrix} \Psi_{xx}\cdot\vec{n}, & \Psi_{xy}\cdot\vec{n} \\ \Psi_{yx}\cdot\vec{n}, & \Psi_{yy}\cdot\vec{n} \end{pmatrix}, \tag{3}$$

respectively, where subscripts denote differentiation with respect to the corresponding variable and the normal vector $\vec{n}$ of Ψ is given by

$$\vec{n} = \frac{(-U_x, -U_y, 1)}{\sqrt{1+U_x^2+U_y^2}}\,. \tag{4}$$

From differential geometry, mean curvature H is defined as

$$H(U) = H(\Psi) = \frac{1}{2}\text{Trace}(\frac{\boldsymbol{D}}{\boldsymbol{F}}) = \frac{(1+U_x^2)U_{yy} - 2U_xU_yU_{xy} + (1+U_y^2)U_{xx}}{2(1+U_x^2+U_y^2)^{\frac{3}{2}}}. \quad (5)$$

In the rest of this paper, we always use H_0 denote the standard mean curvature computed by Eq. 5.

1.2. Mean Curvature Flow

In general, imposing mean curvature can be carried out by the standard mean curvature flow. Taking $I(\vec{x})$ as initial condition, the image evolves according to its mean curvature

$$\frac{\partial U(\vec{x},t)}{\partial t} = \lambda H(U)\|\nabla U\|_2\,, \;\; U(\vec{x},0) = I(\vec{x}), \quad (6)$$

where λ is a diffusion coefficient. Computing the mean curvature plays an important role in such geometric flows. A proper computing scheme does not only accurately estimate the geometry, but also reduces the computation cost.

1.3. Traditional Computing Schemes

Traditionally, there are two ways of computing the mean curvature [22]. One is to directly discretize Eq. 5, where central finite difference is adopted to approximate the derivatives. This method is considered as a classical scheme for computing mean curvature in image processing.

The other is to fit local samples by a quadratic surface and then compute the mean curvature from this surface. We could fit the image data in a local window by a quadratic function

$$f(x,y) = C_5x^2 + C_4y^2 + C_3xy + C_2x + C_1y + C_0\,, \quad (7)$$

where C_i are parameters and they can be obtained by the least square method. After computing the C_i and taking the derivatives, we plug the derivatives into Eq. 5 and get

$$H = \frac{(1+C_2^2)C_4 - C_2C_1C_3 + (1+C_1^2)C_5}{(1+C_1^2+C_2^2)^{\frac{3}{2}}}. \quad (8)$$

This scheme is usually used in the unstructured data, such as point cloud [13], not for the structured data like images, because it does not use the regularity of sampling in the image.

2. LINEAR APPROXIMATION SCHEMES

In this section, we show how to use Euler theorem to approximate mean curvature. Before stating the Euler theorem, we explain several basic concepts in differential geometry. As shown in Fig. 1, for a given surface S, there are planes that intersect with it. The intersection forms many curves C_θ, which defines the directional curvature κ_θ. Among these curvatures, there is a maximum one and a minimum one, which are called principal curvatures κ_1 and κ_2, respectively. From the classical differential geometry, it is known that $H = \frac{\kappa_1+\kappa_2}{2}$.

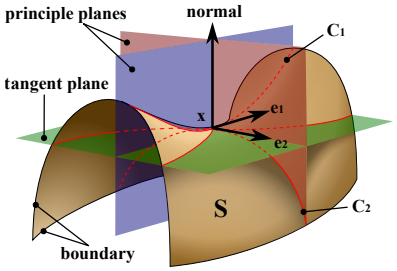


Fig. 1. Curvature on a surface.

2.1. Euler Theorem

The famous Euler theorem gives the relationship between the directional curvature κ_θ and the principal curvatures $\kappa_{1,2}$.

Theorem 1. [Euler Theorem, 1760] $\kappa_\theta = \kappa_1\cos^2\theta + \kappa_2\sin^2\theta$, *where θ is the angle to the principle plane.*

Based on this theorem, we have

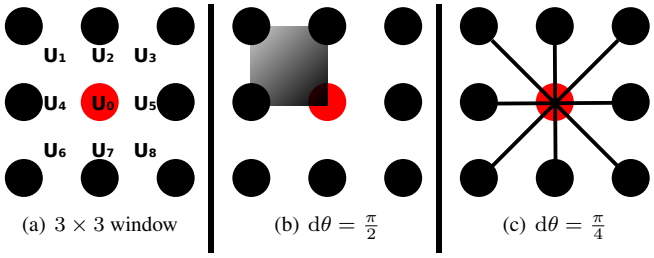
$$\begin{aligned}\int_{-\pi}^{\pi}\kappa_\theta \mathrm{d}\theta =& \kappa_1\int_{-\pi}^{\pi}\cos^2\theta\mathrm{d}\theta + \kappa_2\int_{-\pi}^{\pi}\sin^2\theta\mathrm{d}\theta \\ =& \kappa_1\pi + \kappa_2\pi = \pi(\kappa_1+\kappa_2) = 2\pi H\end{aligned}. \quad (9)$$

Therefore, we have following analytic expression

$$H = \frac{1}{2\pi}\int_{-\pi}^{\pi}\kappa_\theta\mathrm{d}\theta\,, \quad (10)$$

which shows the physical meaning of mean curvature: it is the average of all directional curvatures κ_θ. Equation 10 is the theoretical foundation of our schemes since it provides a novel way to estimate mean curvature.

Now, we need to disretize Eq. 10 by choosing proper $\mathrm{d}\theta$. In a local 3×3 window as shown in Fig. 2(a), we use U_0 denote the central pixel and U_i for its eight neighbor pixels. In this 3×3 window, we could choose $\mathrm{d}\theta = \frac{\pi}{2}$ (Fig. 2(b)) or $\mathrm{d}\theta = \frac{\pi}{4}$ (Fig. 2(c)). We first show the linear schemes for these two cases and then the schemes for their hybrid cases.



(a) 3×3 window (b) $\mathrm{d}\theta = \frac{\pi}{2}$ (c) $\mathrm{d}\theta = \frac{\pi}{4}$

Fig. 2. Discrete pixels and directional curvatures

2.2. Scheme 1: $\mathrm{d}\theta = \frac{\pi}{2}$

In Fig. 2(b), the directional curvatures can be computed by

$$\kappa^1 \approx \quad U_2 + U_4 - U_1 - U_0 \tag{11}$$
$$\kappa^2 \approx \quad U_2 + U_5 - U_3 - U_0 \tag{12}$$
$$\kappa^3 \approx \quad U_7 + U_4 - U_6 - U_0 \tag{13}$$
$$\kappa^4 \approx \quad U_7 + U_5 - U_8 - U_0 \tag{14}$$

According to Eq. 10, we have

$$H = \frac{1}{2\pi}\int_{-\pi}^{\pi} \kappa_\theta \mathrm{d}\theta \approx \frac{1}{2\pi}\sum \kappa^i \frac{\pi}{2} = \frac{\sum \kappa^i}{4}\,. \tag{15}$$

Then, the mean curvature can be approximated by the H_1

$$H \approx H_1 \equiv \begin{pmatrix} \frac{-1}{4} & \frac{1}{2} & \frac{-1}{4} \\ \frac{1}{2} & -1 & \frac{1}{2} \\ \frac{-1}{4} & \frac{1}{2} & \frac{-1}{4} \end{pmatrix} * U\,, \tag{16}$$

where $*$ is the convolution operator.

2.3. Scheme 2: $\mathrm{d}\theta = \frac{\pi}{4}$

As shown in Fig. 2(c), the four directional curvatures can be approximately computed by

$$\tilde{\kappa}^1 \approx \quad \tfrac{1}{2}(U_1 + U_8) - U_0 \tag{17}$$
$$\tilde{\kappa}^2 \approx \quad \tfrac{1}{2}(U_2 + U_7) - U_0 \tag{18}$$
$$\tilde{\kappa}^3 \approx \quad \tfrac{1}{2}(U_3 + U_6) - U_0 \tag{19}$$
$$\tilde{\kappa}^4 \approx \quad \tfrac{1}{2}(U_5 + U_4) - U_0 \tag{20}$$

Therefore, we have

$$H = \frac{1}{2\pi}\int_{-\pi}^{\pi} \kappa_\theta \mathrm{d}\theta \approx \frac{1}{2\pi}(2\sum \tilde{\kappa}^i)\frac{\pi}{4} = \frac{\sum \tilde{\kappa}^i}{4}\,. \tag{21}$$

Thus, the mean curvature can be approximated by the H_2

$$H \approx H_2 \equiv \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & -1 & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{pmatrix} * U\,. \tag{22}$$

2.4. Scheme 3

Since Eq. 16 and Eq. 22 are estimations of H, we can take their average as a new estimation. Simply averaging the two convolution kernels, we have following new estimation

$$H \approx H_3 \equiv \begin{pmatrix} \frac{-1}{16} & \frac{5}{16} & \frac{-1}{16} \\ \frac{5}{16} & -1 & \frac{5}{16} \\ \frac{-1}{16} & \frac{5}{16} & \frac{-1}{16} \end{pmatrix} * U\,, \tag{23}$$

which is exactly the Eq.6.12 in work [10]. From the analysis in this paper, it becomes clear that Eq.6.12 in work [10] takes the average of all possible directional curvatures with the same weight. This is exactly the definition of mean curvature (Eq. 10).

2.5. Scheme 4

Because Eq. 22 has higher sampling density ($\mathrm{d}\theta = \frac{\pi}{4}$) than Eq. 16 ($\mathrm{d}\theta = \frac{\pi}{2}$), their accuracy is different. Noticing $\frac{\pi}{2} = 2\frac{\pi}{4}$, we can set the weight coefficient to be $\frac{1}{3}$ and $\frac{2}{3}$ for Eq. 16 and Eq. 22, respectively. This leads to another scheme

$$H \approx H_4 \equiv \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} * U\,, \tag{24}$$

where the kernel is the Laplacian operator multiplied by $\frac{1}{4}$. Thus, Eq. 24 links the mean curvature with Laplace operator.

3. LINEAR SCHEMES ANALYSIS

In this section, we analyze these linear schemes and compare them with the classical formula H_0 (Eq. 5) in terms of accuracy, computational complexity and convexity.

3.1. Accuracy

Let H_0 denote the result from Eq. 5 with central difference scheme for derivatives. To get the accuracy of H_1, H_2, H_3 and H_4, we randomly generate 100 images that are piecewise linear. For each image, the boundary of each region is ignored when we compute mean curvature. Then, we compute the mean curvature energy for each image

$$\mathcal{E}_i = \iint |H_i| \mathrm{d}x\mathrm{d}y, \;\; i = 0, \cdots, 4\,. \tag{25}$$

We take the $\mathcal{E}_0$ as the baseline to define the energy ratio

$$R_i = \frac{\mathcal{E}_i}{\mathcal{E}_0}, \;\; i = 1, \cdots, 4\,. \tag{26}$$

The mean and variance of all R_i are shown in Table 1.

	R_1	R_2	R_3	R_4
mean	1.36	1.35	1.20	1.15
variance	0.0038	0.0003	0.0004	0.0002

Table 1. The mean and variance of R_i.

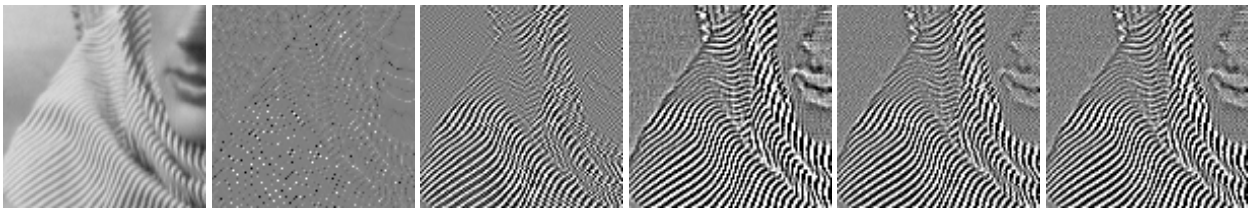
Three conclusions can be made from this benchmark. First, all these linear schemes are not far away from the classical formula. Second, scheme 4 has the best accuracy while scheme 1 has the worst. Scheme 4 adopts the weighted average which should be more accurate. Third, all these schemes lead to larger values than the classical formula.

scheme	H_1	H_2	H_3	H_4
complexity	9N	9N	9N	5N

Table 2. Computational complexity for our linear schemes.

(a) original (b) H_0 (c) H_1 (d) H_2 (e) H_3 (f) H_4

Fig. 3. Mean curvature estimation on camaeraman image. The mean curvature H_i is multiplied by 10 for better visualization.



(a) original (b) H_0 (c) H_1 (d) H_2 (e) H_3 (f) H_4

Fig. 4. Mean curvature estimation on Barbara image. The mean curvature H_i is multiplied by 10 for better visualization.

3.2. Linear Computational Complexity

The presented four schemes have linear computational complexity. Scheme 1 and 3 can be computed by six addition, two multiply and one minus operations. Thus, $9N$ operations are needed for the whole image, where N is the total number of pixels. The computation cost is summarized in Table 2. In contrast, the classical formula Eq. 5 is much more complex.

3.3. Convexity

It is clear that the convolution kernels in $H_i, i = 1, \cdots, 4$ are linear. Thanks to the linearity, we can have following result

Theorem 2. $\mathcal{E}^{H_i}(U) = \iint |H_i(U)|^q \mathrm{d}\vec{x}, \;\; q \geq 1$ *is convex.*

Proof. Let $0 \leq \alpha \leq 1$ and U_1 and U_2 be two discrete images.

$$
\begin{aligned}
&\mathcal{E}^{H_i}(\alpha U_1 + (1-\alpha)U_2) \\
&= \int |H_i((\alpha U_1 + (1-\alpha)U_2))|^q \mathrm{d}\vec{x} \\
&= \int |\alpha H_i(U_1) + (1-\alpha) H_i(U_2)|^q \mathrm{d}\vec{x} \\
&\leq \int |\alpha H_i(U_1)|^q \mathrm{d}\vec{x} + \int |(1-\alpha) H_i(U_2)|^q \mathrm{d}\vec{x} \qquad (27) \\
&= \alpha^q \int |H_i(U_1)|^q \mathrm{d}\vec{x} + (1-\alpha)^q \int |H_i(U_2)|^q \mathrm{d}\vec{x} \\
&\leq \alpha \int |H_i(U_1)|^q \mathrm{d}\vec{x} + (1-\alpha) \int |H_i(U_2)|^q \mathrm{d}\vec{x} \\
&= \alpha \mathcal{E}^{H_i}(U_1) + (1-\alpha)\mathcal{E}^{H_i}(U_2)\,.
\end{aligned}
$$

The second equality comes from the linearity of convolution kernels in H_i. □

3.4. Test On Real Images

We run these schemes on two standard test images, as shown in Fig. 3 and Fig. 4. The presented linear schemes can capture the local geometry better. For example, in both images, the detailed structure is captured by H_2, H_3 and H_4.

As mentioned, computing H_0 is more time consuming. The running time on these two images is given in Table 3, where each scheme was performed 200 times on a laptop with 2.6GHz CPU. The two tested images are 512×512 in gray scale. All code is implemented in MATLAB. The presented schemes are about 20 times faster than the classical Eq. 5.

	H_0	H_1	H_2	H_3	H_4
cameraman	7.8	0.52	0.42	0.50	0.53
Barbara	7.8	0.42	0.42	0.44	0.40

Table 3. Running time in seconds for different schemes.

4. CONCLUSION

In this paper, we presented several linear schemes that approximate the mean curvature of discrete images. We show how to construct these schemes from Euler Theorem. We benchmark them on synthetic data and real images. The experiments confirm that they are good approximations to the mean curvature and can be computed efficiently. Moreover, they provide convex regularization terms for image processing problems. Thanks to the convexity, linear complexity and good approximation accuracy, we believe these schemes will popularize the mean curvature regularization for many variational models from image processing problems.

5. REFERENCES

[1] Leonid I. Rudin, Stanley Osher, and Emad Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1, pp. 259–268, 1992.

[2] Yuanhao Gong and Ivo F. Sbalzarini, "Local weighted Gaussian curvature for image processing," *Intl. Conf. Image Proc. (ICIP)*, pp. 534–538, September 2013.

[3] Guy Gilboa and Stanley Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.

[4] Yuanhao Gong and Ivo F. Sbalzarini, "Curvature filters efficiently reduce certain variational energies," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1786–1798, April 2017.

[5] Miao Jin, Junho Kim, Feng Luo, and Xianfeng Gu, "Discrete surface Ricci flow," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 5, pp. 1030–1043, Sep-Oct 2008.

[6] Bibo Lu, Hui Wang, and Zhonghua Lin, "High order Gaussian curvature flow for image smoothing," in *Multimedia Technology (ICMT), 2011 International Conference on*, july 2011, pp. 5888–5891.

[7] Yuanhao Gong, "Bernstein filter: A new solver for mean curvature regularized models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1701–1705.

[8] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *ECCV 2010*, pp. 1–14, 2010.

[9] Y. Gong and I.F. Sbalzarini, "A natural-scene gradient distribution prior and its application in light-microscopy image processing," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 10, no. 1, pp. 99–114, Feb 2016.

[10] Yuanhao Gong, *Spectrally regularized surfaces*, Ph.D. thesis, ETH Zurich, Nr. 22616, 2015, http://dx.doi.org/10.3929/ethz-a-010438292.

[11] Hongteng Xu, Guangtao Zhai, Xiaolin Wu, and Xiaokang Yang, "Generalized equalization model for image enhancement," *Multimedia, IEEE Transactions on*, vol. 16, no. 1, pp. 68–82, Jan 2014.

[12] Yuanhao Gong and Ivo F. Sbalzarini, "Image enhancement by gradient distribution specification," in *In Proc. Workshop "Emerging Topics in Image Enhancement and Restoration", 12th Asian Conference on Computer Vision*, Singapore, Nov 2014, pp. w7–p3.

[13] Y. Gong, G. Paul, and I. F. Sbalzarini, "Coupled signed-distance functions for implicit surface reconstruction," in *IEEE Intl. Symp. Biomed. Imaging (ISBI)*, May 2012, pp. 1000–1003.

[14] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Proc.*, vol. 10, no. 2, pp. 266–277, feb 2001.

[15] A. Yezzi, "Modified curvature motion for image smoothing and enhancement," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 345–352, Mar 1998.

[16] Wei Zhu, Xue-Cheng Tai, and Tony Chan, "Augmented Lagrangian method for a mean curvature based image denoising model," *Inverse Probl. Imaging*, vol. 7, no. 4, pp. 1409–1432, November 2013.

[17] Marcelo Bertalmio and Stacey Levine, "Denoising an image by denoising its curvature image," *SIAM J. Imaging Sci.*, vol. 7, no. 1, pp. 187–211, 2014.

[18] Qiang Chen, Philippe Montesinos, Quan Sen Sun, Peng Ann Heng, and De Shen Xia, "Adaptive total variation denoising based on difference curvature," *Image & Vision Comput.*, vol. 28, no. 3, pp. 298–306, 2010.

[19] Haoying Fu, Michael K. Ng, Mila Nikolova, and Jesse L. Barlow, "Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration," *SIAM Journal on Scientific Computing*, vol. 27, no. 6, pp. 1881–1902, 2006.

[20] Yuanhao Gong, Qicong Wang, Chenhui Yang, Yahui Gao, and Cuihua Li, "Symmetry detection for multi-object using local polar coordinate," *Lecture Notes in Computer Science*, vol. 5702, pp. 277, 2009.

[21] H. Zhu, H. Shu, J. Zhou, X. Bao, and L. Luo, "Bayesian algorithms for PET image reconstruction with mean curvature and Gauss curvature diffusion regularizations," *Computers in Biology and Medicine*, vol. 37, no. 6, pp. 793–804, 2007.

[22] Evgeni Magid, Octavian Soldea, and Ehud Rivlin, "A comparison of gaussian and mean curvature estimation methods on triangular meshes of range image data," *Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 139–159, 2007.