

Shape and Image Interrogation with Curvature Extremalities

Shin Yoshizawa¹, Alexander Belyaev², and Hideo Yokota¹

¹ *Bio-Research Infrastructure Construction Team, RIKEN
2-1, Hirosawa, Wako, Saitama, 351-0198, Japan
email: {shin,hyokota}@riken.jp*

² *Vision, Image & Signal Processing, ERPem, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
email: a.belyaev@hw.ac.uk*

Abstract. Reliable detection of surface creases defined via loci of the principal curvatures along their corresponding curvature lines is important for many geometrical and graphical applications. Multivariate analogues of such creases have received a considerable attention in recent studies on multidimensional image visualization and analysis. In this paper, we propose a numerically efficient and reliable approach for estimating multidimensional curvature extremalities and detecting ridge-like structures in multidimensional images. The approach is based on local fitting of hypercubic polynomials and calculating their extremalities by using newly derived formulas. We also propose a new thresholding scheme for removing spurious and unessential extremalities. We test our approach by detecting crease structures on 2D and 3D real-world images and demonstrating their ability to capture salient geometric image features.

Key Words: Curvature extremalities, multidimensional ridges and creases, multivariate image analysis.

1. Introduction

Accurate and robust detection of surface and creases and associated skeletal structures is important for many of computer graphics, geometric modeling, and medical imaging applications [9, 4, 7, 13]. Recent progress in multidimensional and multivariate data acquisition calls for developing advanced shape interrogation methods. In particular, extracting crease structures turns out to be useful for computational fluid dynamics and medical imaging purposes [18, 17, 12] (see also references therein).

In contrast to the so-called height ridges [8] and their multidimensional generalizations [4, 18, 17], the principal direction ridges are rarely used for multivariate data analysis. The principal direction ridges possess beautiful mathematical properties [10, 14], have numerous 2D surface-based analysis and modeling applications [9, 7, 13]. However their multivariate extensions may seem too difficult to deal with. In this paper ¹, we develop a computational theory of multivariate principal direction ridges.

Our Approach. In this work, we introduce a multivariate analogue of the principal direction ridges and propose an efficient numerical procedure for estimating multidimensional curvature extremalities. Our ridge detection method can be considered as an extension of our cubic-polynomial fitting scheme developed in [20] for tracing salient curvature extrema on 2D surfaces approximated by dense triangle meshes, see Figure 1. We first derive a simple (and novel) formula for extremalities of hyper-polynomial surfaces, and then the extremalities of images are estimated by local fitting hyper-polynomial patches using our formula. We also introduce a differential invariant measure based on the curvature extremalities (we call it *cyclidity*), and use it for detecting salient ridge-like image structures.

In our study, we deal with more difficult problems than those considered in [4, 18, 17] and study pure geometric surface ridges and their multidimensional analogues. Such ridges are invariant w.r.t. the Euclidean motions, scale changes, and generalized inversions (conformal invariant), and, therefore, being extracted from a multidimensional image, convey intrinsic information about image geometry.

Paper Organization. The rest of paper is organized as follows. Sections 2 and 3 define curvature extremalities and cyclidity measures for multidimensional surfaces, respectively. We derive a simple formula of the extremalities in Section 4. Our algorithm of calculating the extremalities and its numerical experiments are given in Sections 5 and 6, respectively. We conclude the paper in Section 7.

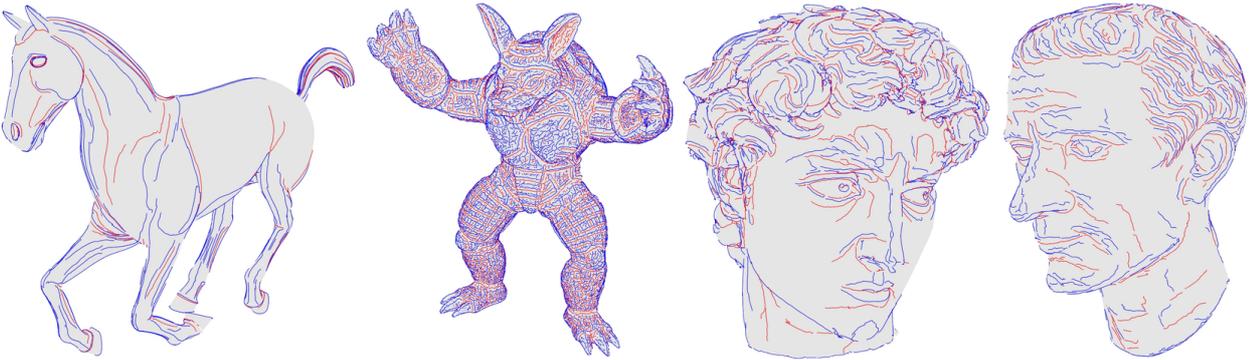


Figure 1: Creases extracted on surface meshes by using the method of [20]. Red and blue lines correspond to the salient extrema of principal curvatures along their curvature lines (a subset of principal direction ridges called crest lines), respectively.

¹It is an extension of our previous work [19]. The main difference from [19] is our thresholding scheme consisting of the conformal invariant differential quantity (1) and its discrete implementation and numerical experiments.

2. Curvature Extremalities

Consider a smooth hyper-surface (d -dimensional manifold) \mathcal{S} in \mathbb{R}^{d+1} . Let us assume that the hyper-surface is defined parametrically, $\mathcal{S} = \mathcal{S}(\mathbf{x})$ where the vector of parameters

$$\mathbf{x} = (x_1, x_2, \dots, x_d)$$

lives in \mathbb{R}^d . The basic tangent vectors of \mathcal{S} at \mathbf{x} are given by $\mathcal{S}_i = \frac{\partial \mathcal{S}}{\partial x_i}$, $i = 1, \dots, d$. The unit normal vector of \mathcal{S} is defined by the so-called wedge product of $\mathcal{S}_1 \dots, \mathcal{S}_d$

$$\mathbf{n} = \frac{\bigwedge (\mathcal{S}_1, \dots, \mathcal{S}_d)}{|\bigwedge (\mathcal{S}_1, \dots, \mathcal{S}_d)|}, \quad \bigwedge (\mathcal{S}_1, \dots, \mathcal{S}_d) = \begin{vmatrix} \mathcal{S}_1^{(1)} & \dots & \mathcal{S}_1^{(d+1)} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_d^{(1)} & \dots & \mathcal{S}_d^{(d+1)} \\ \mathbf{e}_1 & \dots & \mathbf{e}_{d+1} \end{vmatrix},$$

where $\{\mathbf{e}_1, \dots, \mathbf{e}_{d+1}\}$ is the coordinate basis and $[\mathcal{S}_i^{(1)}, \dots, \mathcal{S}_i^{(d+1)}]$ are components of \mathcal{S}_i w.r.t. the coordinate basis, $i = 1, \dots, d$.

Let us define the curvature tensor (generalization of the Waingarten map) of \mathcal{S} by the $d \times d$ matrix $\mathbf{W} = \mathbf{IIg}^{-1}$, where the Riemannian covariant metric tensor \mathbf{g} is given by the i -th row and j -th column element $\mathbf{g}_{ij} = \frac{\partial \mathcal{S}}{\partial x_i} \cdot \frac{\partial \mathcal{S}}{\partial x_j}$, and the i -th row and j -th column element of the matrix \mathbf{II} is defined by $\mathbf{II}_{ij} = \frac{\partial^2 \mathcal{S}}{\partial x_i \partial x_j} \cdot \mathbf{n}$. Here $\mathbf{a} \cdot \mathbf{b}$ is the inner product between \mathbf{a} and \mathbf{b} . The quadratic forms \mathbf{g} and \mathbf{II} are natural analogues of the two-dimensional first and second fundamental forms.

The d eigenvalues $k_1 \leq k_2 \leq \dots \leq k_d$ and their corresponding eigenvectors $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_d\}$ of \mathbf{W} give us the principal curvatures and directions of \mathcal{S} .

We introduce generalized curvature extremalities of \mathcal{S} as $e_i = \partial k_i / \partial \mathbf{t}_i = \nabla k_i \cdot \mathbf{t}_i$, where $\nabla = \{\partial / \partial x_1, \partial / \partial x_2, \dots, \partial / \partial x_d\}$ is the standard gradient operator. Zero-crossings of e_i , $i = 1, 2, \dots, d$ describe principal direction ridges on \mathcal{S} .

3. Dupin's Cyclides

The extremalities are closely related to a special family of surfaces called Dupin's cyclides which have been intensively studied in connection with various shape modeling tasks [3, 5]. The classical 2D Dupin's cyclides live in three-dimensional space and can be characterized by the conditions $e_1 = 0 = e_2$. A straightforward generalization leads us to the notion of Dupin hyper-surfaces defined by a system of d equations $e_i = 0$, $i = 1, 2, \dots, d$ and, therefore, constituting hyper-surfaces in \mathbb{R}^{d+1} .

Let us introduce a cyclidity by

$$C = \sum_i^d |e_i|.$$

The cyclidity measures how a point on \mathcal{S} is close to the Dupin hyper-surfaces. Recall the zero-crossings of extremalities correspond to the surface creases. On the other hand, the cyclides do not contain surface creases in general. Thus, the 2D analogue of cyclidity was employed in [20] to

filter out the phantom surface creases around the surface parts close to the cyclides:

$$T_c = \int \sqrt{|e_1| + |e_2|} ds,$$

where the integral is calculated along each extracted creases, ds is the arclength element of the crease, and T_c is invariant w.r.t. the conformal transformations [20]. Figure 2 demonstrates how well our threshold T_c work to filter our phantom surface creases around the surface parts close to the Dupin's cyclides.

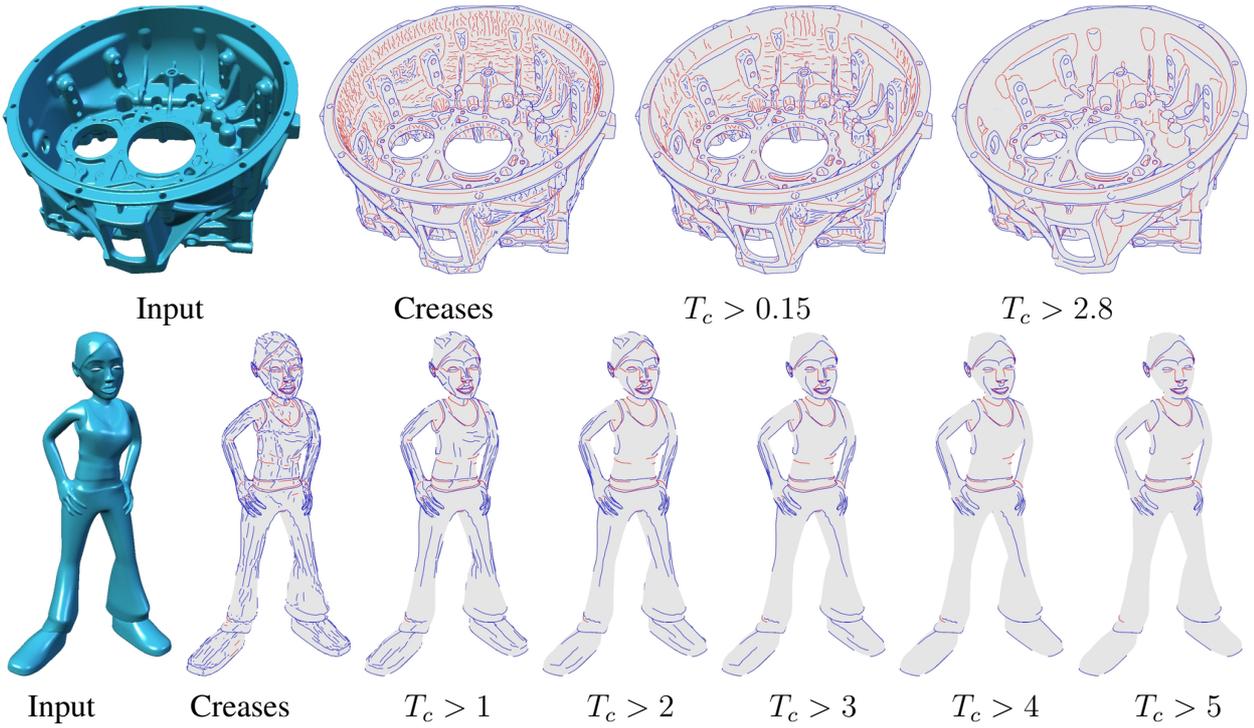


Figure 2: Filtering phantom surface creases (crest lines) via the conformal invariant differential quantity T_c [20]. The images show the filtered surface creases whose T_c is greater than the given thresholds.

In this paper, we use the cyclidity together with $\{e_i\}$ for extracting salient ridge-like image structures by employing the following integral (1).

$$T_{\mathbb{R}^d} = \int \left(\sum_{i=1}^d e_i^2 \right)^{\frac{d}{4}} dA, \quad (1)$$

where $dA = \sqrt{1 + \sum_{i=1}^d \mathcal{S}_i^2} dx_1 dx_2 \dots dx_d$ is the area element of \mathcal{S} . We model that (1) is also the conformal invariant differential quantity, since the ridges are invariant under the conformal transformations.

4. Formula of Extremalities

Consider a multidimensional image $x_{d+1} = I(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_d)$. For the sake of simplicity, we deal with single-modality images, i.e., the image intensity I is a scalar function (our results can be easily extended to multichannel images). Note that the equation $x_{d+1} = I(\mathbf{x})$ defines a hyper-surface in the extended spatial-tonal image space $\mathbb{R}^{d+1} = \{(x_1, \dots, x_d, x_{d+1})\}$. Our main task now is to achieve fast and robust estimation of the extremalities $\{e_i\}$. Note that the curvature extremalities are shift and rotation invariant. The following formula is very useful because it significantly reduces computational time needed for the curvature extremalities.

Main Formula. Assume that the coordinates in \mathbb{R}^{d+1} are chosen such that the origin of coordinates $\mathbf{0}$ is situated on $\mathcal{S} = \{x_{d+1} = I(\mathbf{x})\}$ and the hyperplane formed by the coordinate axes x_1, \dots, x_d is orthogonal to \mathcal{S} at $\mathbf{0}$. Then the extremality $e(\mathbf{x})$ of \mathcal{S} at $\mathbf{x} = \mathbf{0}$ is given by

$$e = \sum_{i,j,l=1}^d I_{ijl} t_i t_j t_l, \quad (2)$$

where $I_{ijl} = \frac{\partial^3 I(\mathbf{x})}{\partial x_i \partial x_j \partial x_l}$ are third-order partial derivatives of $I(\mathbf{x})$ and $\mathbf{t} = (t_1, t_2, \dots, t_d)$ is the principal direction corresponding to the curvature extremality e .

Derivation of Main Formula. Let

$$\mathbf{t} = (t_1, t_2, \dots, t_{d+1}) \text{ and } \mathbf{n} = (n_1, n_2, \dots, n_{d+1})$$

be a principal direction and the unit normal of \mathcal{S} at $(\mathbf{x}, x_{d+1}) \in \mathbb{R}^{d+1}$, respectively. Let us represent \mathcal{S} in implicit form

$$0 = I(\mathbf{x}) - x_{d+1} \equiv F(\mathbf{x}, x_{d+1}).$$

Similar to the 3D case (see, for example, [11, 1]) the extremality e of

$$F(\mathbf{x}, x_{d+1}) = 0$$

is given by

$$e = \frac{\partial k}{\partial \mathbf{t}} = \frac{\sum_{i,j,l=1}^{d+1} F_{ijl} t_i t_j t_l + 3k \sum_{i,j=1}^{d+1} F_{ij} t_i n_j}{|\nabla F|}, \quad (3)$$

where $F_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}$ and $F_{ijl} = \frac{\partial^3 F}{\partial x_i \partial x_j \partial x_l}$.

Since $F(\mathbf{x}, x_{d+1}) \equiv I(\mathbf{x}) - x_{d+1}$, we have $F_{i,d+1} = 0$. Further, $\nabla F \equiv (\nabla_{\mathbf{x}} I, -1)$ and $\nabla_{\mathbf{x}} I(\mathbf{0}) = 0$. Thus $|\nabla F(\mathbf{0})| = 1$. Note that the sum of $F_{ij} t_i n_j$ in (3) the indices vary from 1 to d and, therefore, the sum vanishes because only the first d components of $\mathbf{n}(\mathbf{0})$ are zeros. Formula (2) is proved.

For the sake of completeness, we present here a derivation of (3). Assume that the components of the surface orientation normal are given by $n_i = -F_i/g$, where $g = |\nabla F|$ is the absolute value of the gradient. Let k , \mathbf{t} , and s stand for a principal curvature, the associated principal direction,

and the arclength parameter along the curvature line corresponding to k and \mathbf{t} . According to the Frenet formulas we have at a point on \mathcal{S}

$$\frac{d\mathbf{t}}{ds} = k\mathbf{n}, \quad \frac{d\mathbf{n}}{ds} = -k\mathbf{t}.$$

Differentiating $\sum_{i=1}^{d+1} F_i t_i = 0$ w.r.t. s yields

$$\frac{d}{ds} \left(\sum_{i=1}^{d+1} F_i t_i \right) = \sum_{i,j=1}^{d+1} F_{ij} t_i t_j + k \sum_{i=1}^{d+1} F_i n_i = \sum_{i,j=1}^{d+1} F_{ij} t_i t_j - kg.$$

Thus the principal curvatures k is given by

$$k = \frac{1}{g} \sum_{i,j=1}^{d+1} F_{ij} t_i t_j \quad (4)$$

Note that

$$\frac{dg}{ds} = \sum_{i=1}^{d+1} g_i t_i = \frac{1}{g} \sum_{i,j=1}^{d+1} F_{ij} t_i F_j = - \sum_{i,j=1}^{d+1} F_{ij} t_i n_j.$$

Finally differentiating (4) w.r.t. s yields

$$e = \frac{dk}{ds} = \frac{d}{ds} \left(\frac{1}{g} \sum_{i,j=1}^{d+1} F_{ij} t_i t_j \right) = \frac{1}{g} \left(\sum_{i,j,l=1}^{d+1} F_{ijl} t_i t_j t_l + 3k \sum_{i,j=1}^{d+1} F_{ij} t_i n_j \right)$$

which is desired formula (3).

Relation to Derivative of Curvature Tensor. Our formula (2) can be also obtained by extending the derivative-of-curvature tensor derived by Rusinkiewicz [16] (see formula (8) there) to d -manifolds. He used a $2 \times 2 \times 2$ tensor composed by $\{\partial\mathbf{W}/\partial x_1, \partial\mathbf{W}/\partial x_2\}$ for 2-manifolds. The curvature derivative w.r.t. \mathbf{t} was given by multiplying \mathbf{t} three times to the tensor. According to this observation, the extremality of \mathcal{S} is given by the gradient of the Waingarten map multiplying by \mathbf{t} three times: $e = \nabla\mathbf{W}(\mathbf{t}, \mathbf{t}, \mathbf{t})$ where $\nabla\mathbf{W}$ is a $d \times d \times d$ tensor. Since the Waingarten map \mathbf{W} is equivalent to the Hessian of $I(\mathbf{x})$ in our case because \mathbf{II} becomes an identity matrix at $\mathbf{x} = \mathbf{0}$, we have

$$e = \nabla\text{Hess}(I(\mathbf{x}))(\mathbf{t}, \mathbf{t}, \mathbf{t}) = \sum_{i,j,l=1}^d I_{ijl} t_i t_j t_l.$$

Implementation to Cubic-Polynomials. By using our formula (2), the following simple forms are derived for the 2D and 3D polynomials.

For a 2D cubic-polynomial

$$f(x_1, x_2) = \frac{1}{2}(b_0 x_1^2 + 2b_1 x_1 x_2 + b_2 x_2^2) + \frac{1}{6}(c_0 x_1^3 + 3c_1 x_1^2 x_2 + 3c_2 x_1 x_2^2 + c_3 x_2^3),$$

we have

$$\mathbf{W} = \begin{pmatrix} b_0 & b_1 \\ b_1 & b_2 \end{pmatrix} \quad (5)$$

and

$$e = \begin{pmatrix} t_1^2 \\ t_2^2 \end{pmatrix}^T \begin{pmatrix} c_0 & 3c_1 \\ 3c_2 & c_3 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \quad (6)$$

at $(x_1, x_2) = (0, 0)$.

For a 3D cubic-polynomial

$$\begin{aligned} f(x_1, x_2, x_3) = & \frac{1}{2}(b_0x_1^2 + 2b_1x_1x_2 + 2b_2x_1x_3 + b_3x_2^2 + 2b_4x_2x_3 + b_5x_3^2) + \\ & + \frac{1}{6}(c_0x_1^3 + 3c_1x_1^2x_2 + 3c_2x_1^2x_3 + 3c_3x_2^2x_1 + c_4x_2^3 + 3c_5x_2^2x_3 + \\ & + 3c_6x_1x_3^2 + 3c_7x_2x_3^2 + c_8x_3^3), \end{aligned}$$

we have

$$\mathbf{W} = \begin{pmatrix} b_0 & b_1 & b_2 \\ b_1 & b_3 & b_4 \\ b_2 & b_4 & b_5 \end{pmatrix} \quad (7)$$

and

$$e = \begin{pmatrix} t_1^2 \\ t_2^2 \\ t_3^2 \end{pmatrix}^T \begin{pmatrix} c_0 & 3c_1 & 3c_2 \\ 3c_3 & c_4 & 3c_5 \\ 3c_6 & 3c_7 & c_8 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (8)$$

at $(x_1, x_2, x_3) = (0, 0, 0)$.

5. Interrogation Algorithm

Let r and r_f be the user-specified fitting and thresholding radii, respectively. We perform the following local polynomial fitting procedure to each image element (pixels/voxels) in order to obtain $\{e_i\}$ and C .

Step 1: For an image element at \mathbf{x}_i , consider its $(2r + 1)^d$ neighboring elements centered at \mathbf{x}_i .

Fit the polynomial $f(\mathbf{x})$ to $I(\mathbf{x}) - I(\mathbf{x}_i)$ in the least-squares sense by using the neighboring elements.

Step 2: Calculate the Waingarten map \mathbf{W} via the equation (5) or (7), and compute the eigenvalues (principal curvatures $\{k_i\}$) and eigenvectors (principal directions $\{\mathbf{t}_i\}$) of \mathbf{W} .

Step 3: Obtain the curvature extremalities $\{e_i\}$ via the equation (6) or (8).

Step 4: Apply thresholding (if necessary) to the obtained extremalities. Consider the $(2r_f + 1)^d$ neighboring elements centered at \mathbf{x}_i . In our implementation, we approximate the threshold (1) by $T_{\mathbb{R}^d} \approx \sum_{j=1}^{(2r_f+1)^d} a_j (\sum_{i=1}^d (\mathbf{t}_i \cdot \mathbf{t}_{i,j}) e_{i,j}^2)^{\frac{d}{4}}$, where a_j , $\mathbf{t}_{i,j}$, and $e_{i,j}$ are the area element, principal direction, and extremality of neighboring element centered at \mathbf{x}_i , respectively. We estimate how the neighboring element is close to be located on the ridge-like structure at \mathbf{x}_i by the inner-product of their principal directions $(\mathbf{t}_i \cdot \mathbf{t}_{i,j})$. The second order central finite

difference scheme is employed for approximating partial derivatives of $\sqrt{1 + \sum_i^d \left| \frac{\partial I(\mathbf{x})}{\partial x_i} \right|^2} \approx a_j$.

Discussion. In our current implementation of polynomial fitting, we did not use the normals of \mathcal{S} , and we omitted a constant and linear terms. Although our fitting scheme provides smoother results than the case of incorporating these terms, adapting the normal-based [6] or/and osculating jets [2] fitting methods to our approach is promising to improve accuracy of polynomial fitting.

We employed same fitting radius for all image elements. Thus, the fitting process is time consuming when the radius is large (especially for $d \geq 3$). Adaptive radius fitting with multiresolution or/and scale-space strategy may accelerate our approach, and also gain flexible control over extracting multi-scale features.

Future work includes developing a robust and efficient method to extract zero-crossings of the computed extremalities.

6. Numerical Experiments

In our implementation, the singular value decomposition [15] for least-squares fitting and Jacobi method [15] for eigenanalysis (because \mathbf{W} is a real symmetric matrix in our case) were employed. All our numerical experiments reported in this paper were performed on a Core2 Extreme X9770 (3.2 GHz quad core, no parallelization was used) PC with 16GB RAM and 64 bit OS.

We tested our approach on the real-world 2D and 3D images and find it useful for extracting meaningful image structures, see Figures 3, 6, 7, 9, and 10. Since the extremalities vary with the High Dynamic Range (HDR), we applied the following normalization technique for visualization purposes. To map HDR values onto more intuitively visible domain, $|e_i|$ and C are normalized into $[0, 1]$ range and white (black) color corresponds to high (low) intensity value, while clipping 1% on the low (black) end and 2% on the on the high (white) end of the histogram (16 bit bins). We also used RGB color channels to visualize $|e_i|$ and C simultaneously. For example, the 2D extremalities are visualized by $\mathbf{e}_{rgb} = (\mathbf{R}, \mathbf{G}, \mathbf{B}) = (1 - |e_1|, 1 - C, 1 - |e_2|)$ in Figures 6 and 7. The convex and concave edge regions are well distinguished similar to true ridges in meshes (as in Figure 1). Similar coloring technique was also employed in the 3D images, see caption of Figures 9 and 10.

Our approach has a user-specified parameter for polynomial fitting: fitting radius (r). The influence of changing the radius is demonstrated in Figure 6. As we can see the multi-scale features as shadows in the left image of Figure 5, increasing the radius provides extraction of large size ridge-like structures.

Figures 8 and 11 demonstrate our thresholding scheme applied to the extremalities of Lena and MRI-Head images (Figures 7 and 10), respectively. We can see that the extremalities at the image regions close to the Dupin's cyclides (no creases) are adaptively removed according to increasing the threshold (1) while the salient ridge-like structures are preserved well.

The computational complexity of our approach is linear ($O((2r + 1)^{2d})$) w.r.t. image size (fitting radius, since the singular value decomposition possess quadratic complexity). We examined computational time of our approach on varying image size (Figure 4) and radius (center and right images of Figure 5). The slope of resulting timing lines are reasonably close to the complexity. Our approach approximately processes 66K pixels and 8K voxels per second in average for small fitting radius ($r \leq 5$ and $r \leq 3$), respectively.



Figure 3: Input images: Lena (512^2), Trui (256^2), Mandrill (512×506), and Zebra (507×379).

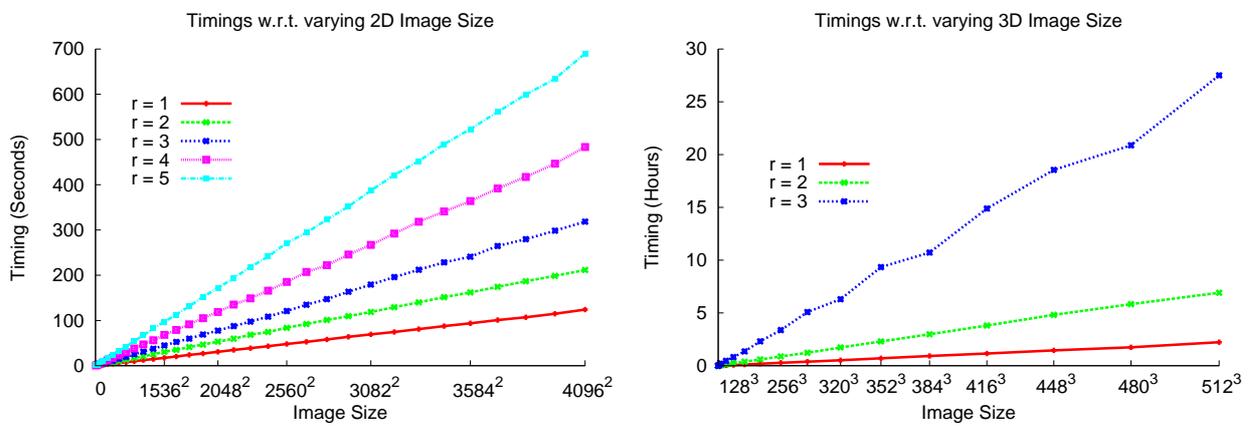


Figure 4: Computational time required for all necessary computations on 2D (left) and 3D (right) images.

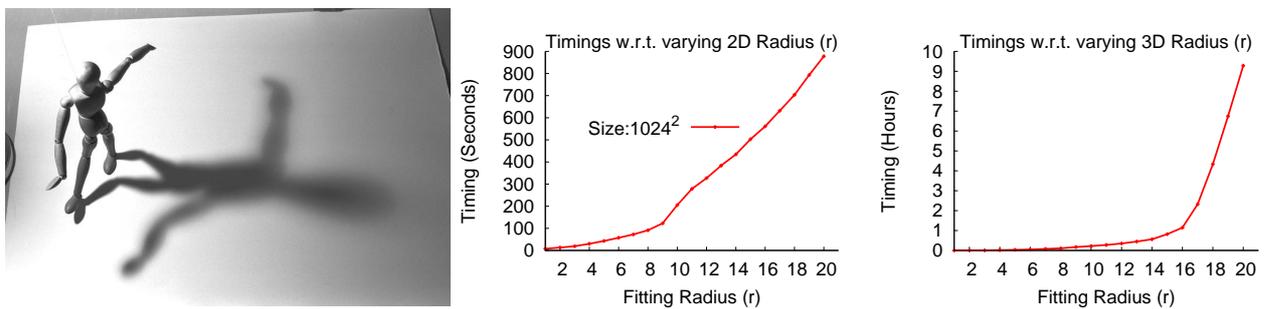


Figure 5: Left: image consists of multi-scale features. Center and right: computational time on 2D (center) and 3D (right) images w.r.t. varying fitting radius (r).

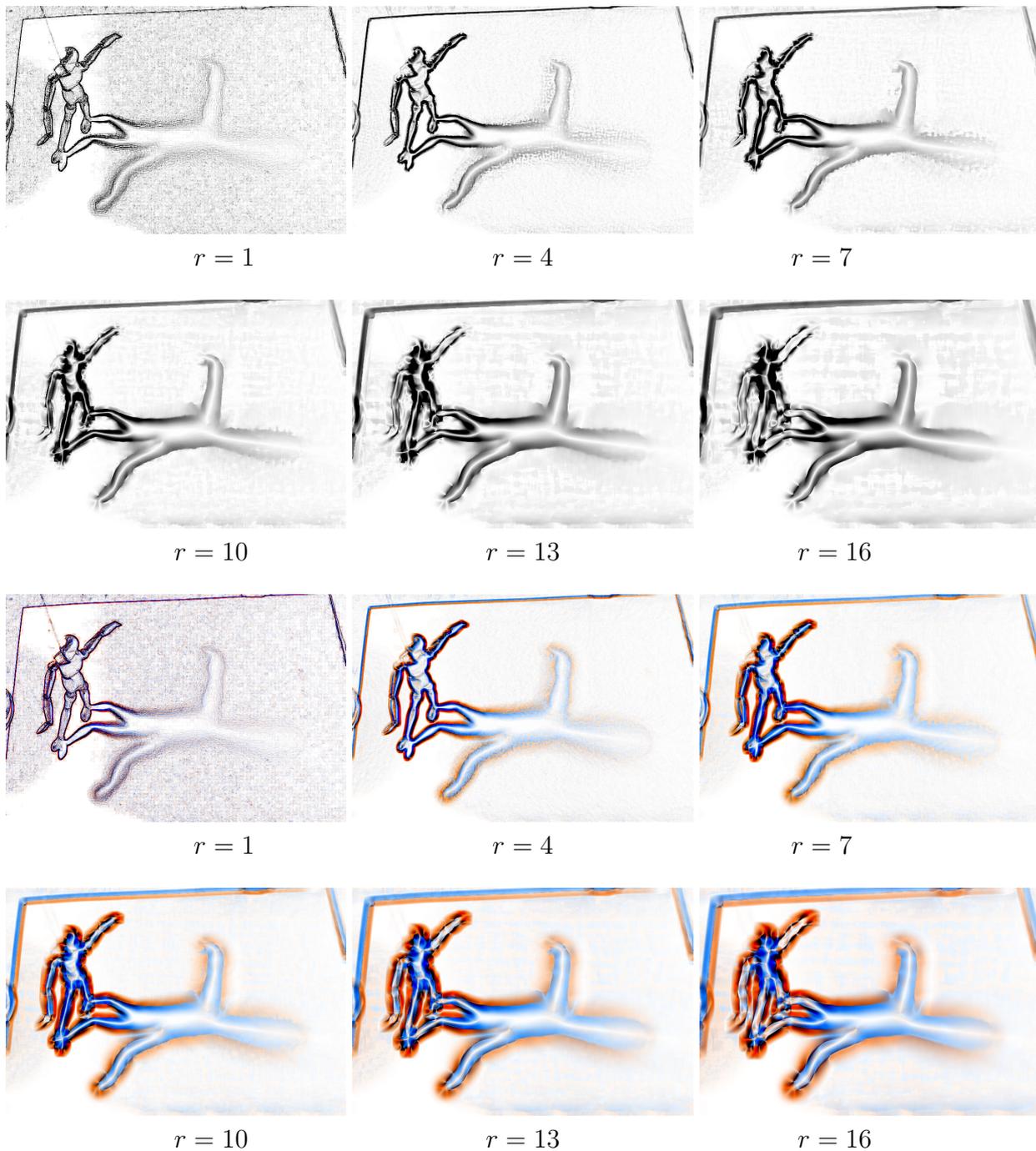
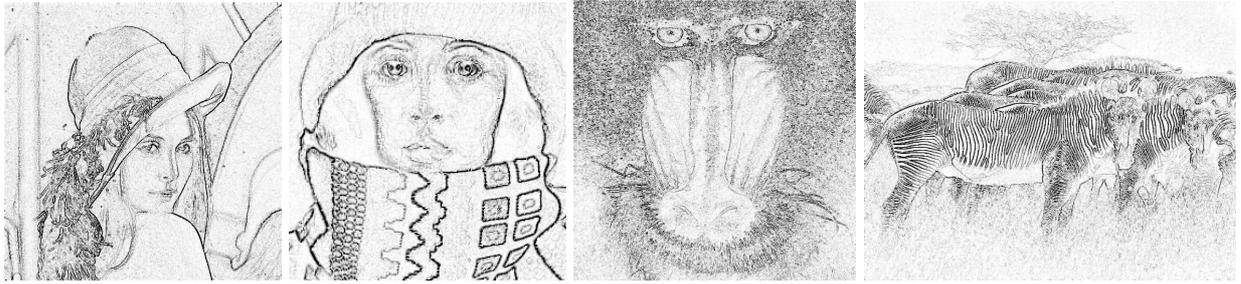
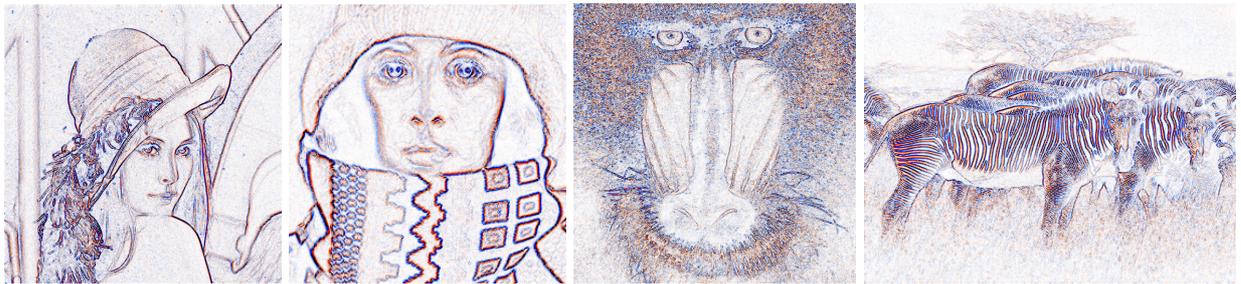


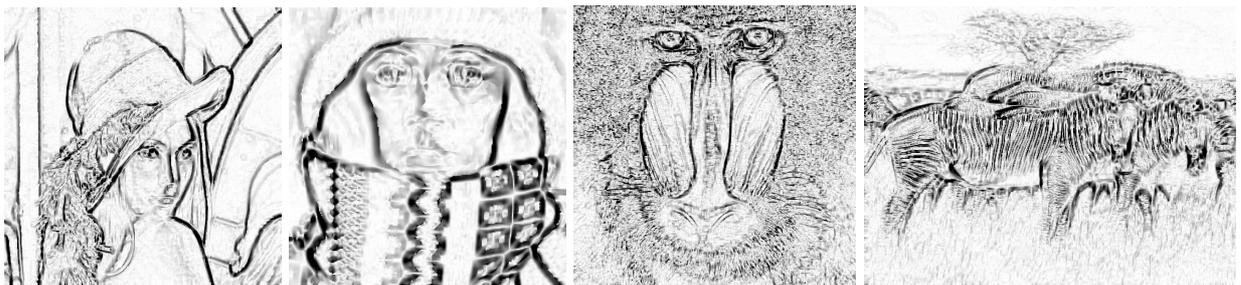
Figure 6: Cyclidity (C , top two rows) and extremalities (e_{rgb} , bottom two rows) with varying the fitting radius r . The image consisting of 768×512 pixels shown in the left image of Figure 5 took 2.9 ($r = 1$), 11 ($r = 4$), 27.2 ($r = 7$), 88.7 ($r = 10$), 172 ($r = 13$), and 251 ($r = 16$) seconds, respectively for all necessary computations. Increasing the radius for our approach extracts the large size ridge-like structures. See the shadow regions in the left image of Figure 5 and corresponding parts in the above images.



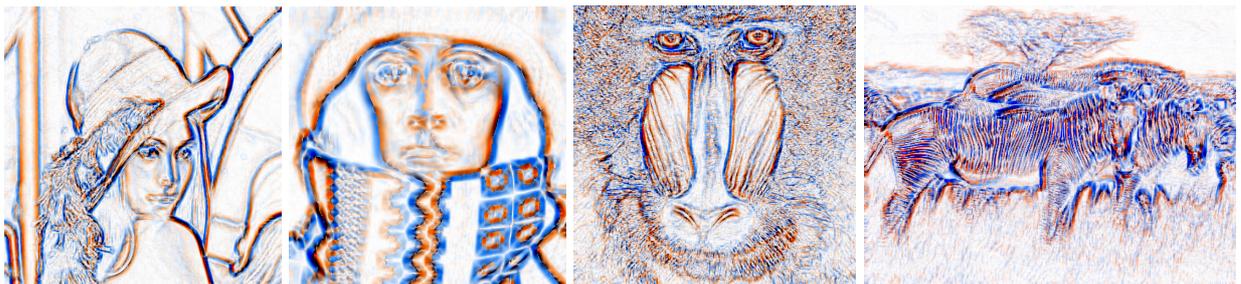
Cyclidity ($r = 1$): C



Extremalities ($r = 1$): e_{rgb} , timing: 1.96, 0.48, 1.93, and 1.44 seconds.



Cyclidity ($r = 5$): C



Extremalities ($r = 5$): e_{rgb} , timing: 10.66, 2.71, 10.46, and 8.05 seconds.

Figure 7: The images show both cyclidity C and extremalities e_{rgb} by using 3^2 and 11^2 pixel neighborhoods for polynomial fitting, where the input images are given in Figure 3. Timings listed above present computational times required for all necessary computations. Here red and blue regions correspond to low value (close to zero) of $|e_1|$ and $|e_2|$, respectively.

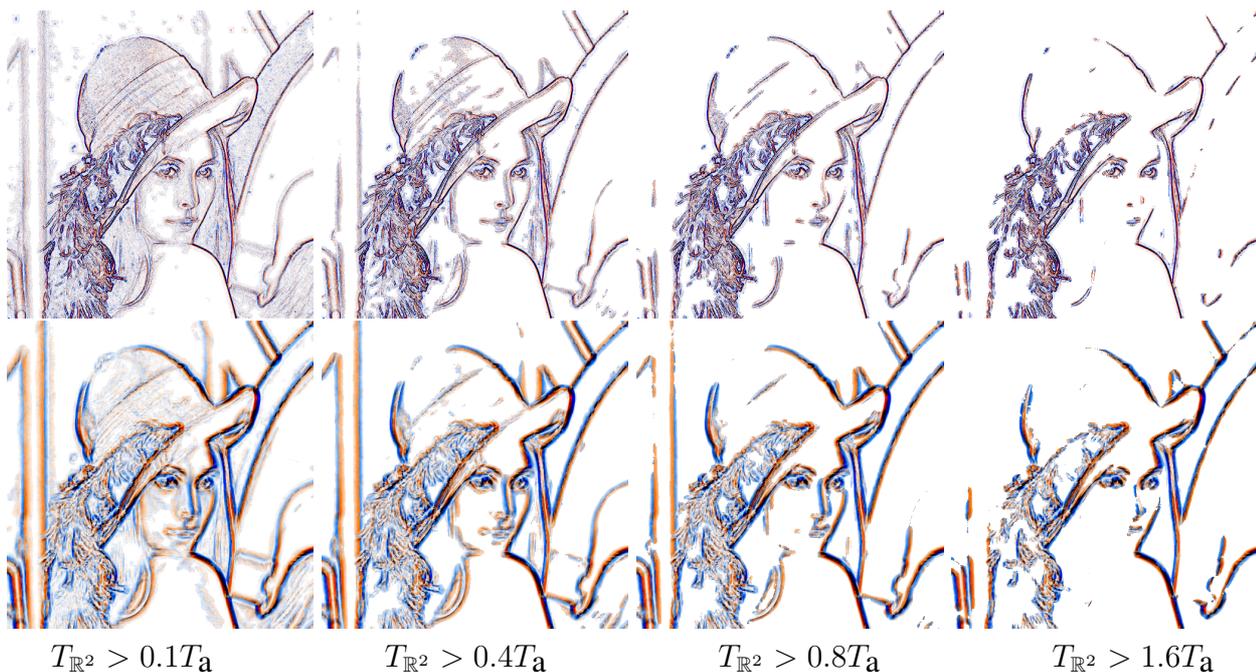


Figure 8: Filtering phantom ridge-like structures via the conformal invariant differential quantity (1), where the input image is shown in Figure 3, the extremalities (e_{rgb}) corresponding to $T_{\mathbb{R}^2} \geq 0$ are illustrated in Figure 7 (upper: $r = 1$ and bottom: $r = 5$), we use $r_f = 5$, and T_a is an average of $T_{\mathbb{R}^2}$ within the input image.

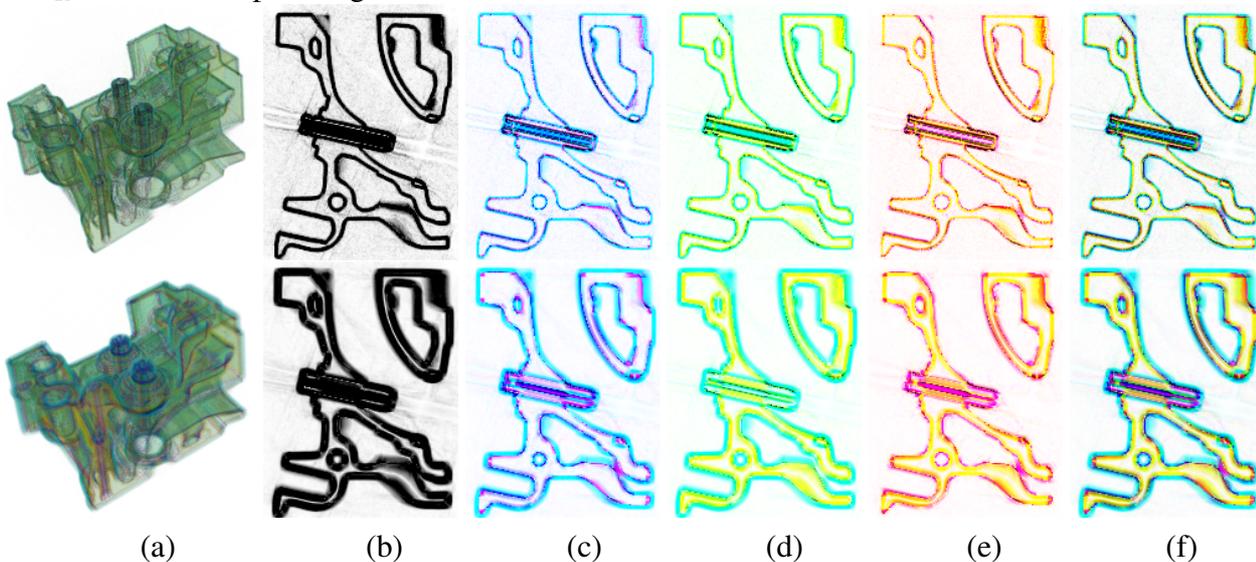


Figure 9: 3D extremalities of CT-Engine ($256^2 \times 110$) took 6.4 and 85 minutes respectively for $r = 1$ and $r = 3$. (a): volume rendering of $\{|e_1|, |e_2|, |e_3|\}$ where RGB color channels are given by $(R, G, B) = (1 - |e_1|, 1 - |e_2|, 1 - |e_3|)$. (f): sectional images of (a). (b, c, d, e): sectional images where their locations correspond to (f), and their RGB color channels consist of $(1 - C, 1 - C, 1 - C)$, $(1 - |e_1|, 1 - |e_2|, 1 - C)$, $(1 - |e_1|, 1 - C, 1 - |e_3|)$, and $(1 - C, 1 - |e_2|, 1 - |e_3|)$, respectively. The upper and bottom images are obtained by using 3^3 ($r = 1$) and 7^3 ($r = 3$) voxel neighborhoods to the fitting process, respectively.

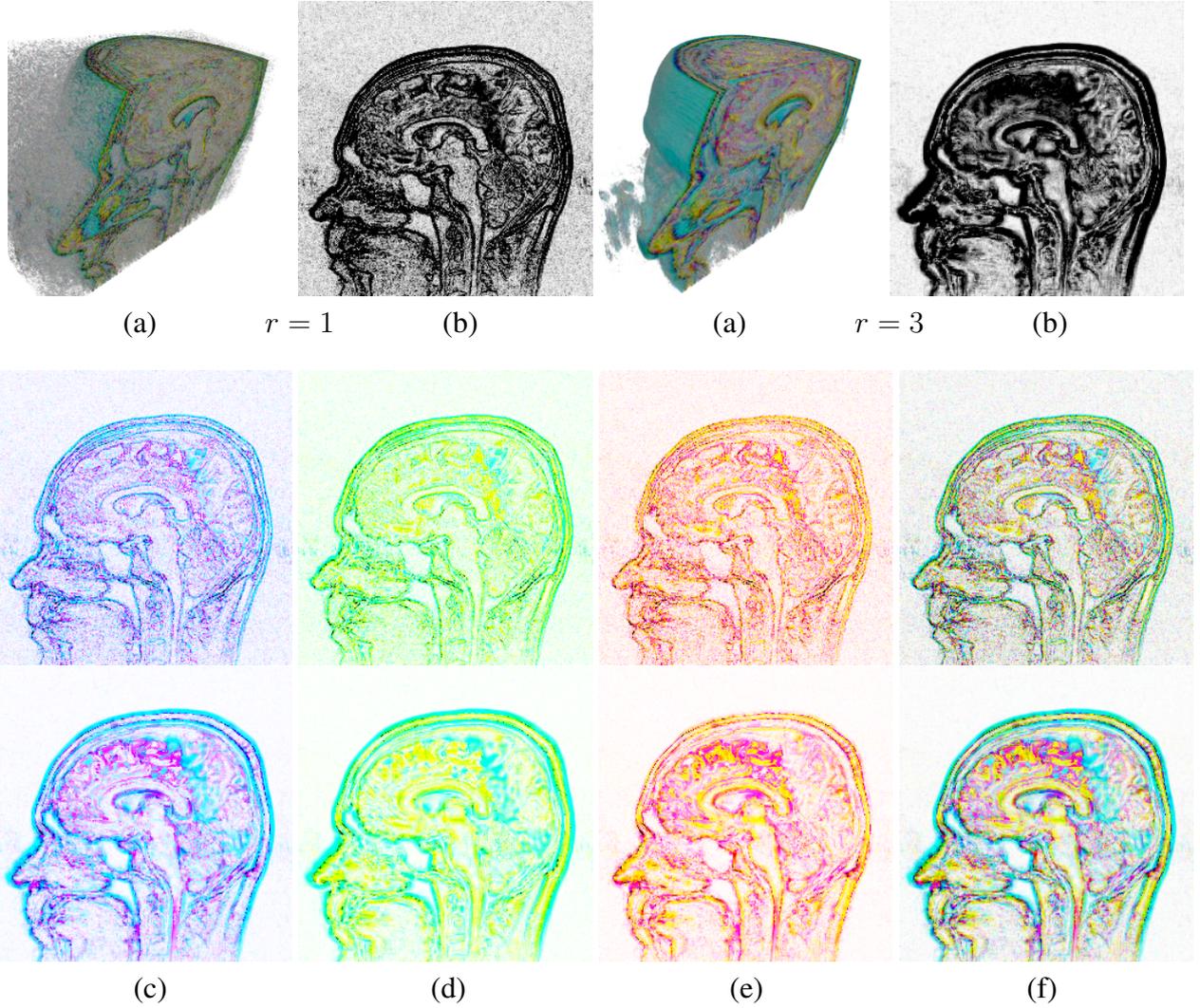


Figure 10: 3D extremalities of MRI-Head (256^3) took 17 and 195 minutes respectively for $r = 1$ and $r = 3$. (a): volume rendering of $\{|e_1|, |e_2|, |e_3|\}$ where RGB color channels are given by $(R,G,B) = (1 - |e_1|, 1 - |e_2|, 1 - |e_3|)$. (f): sectional images of (a). (b,c,d,e): sectional images where their locations correspond to (f), and their RGB color channels consist of $(1 - C, 1 - C, 1 - C)$, $(1 - |e_1|, 1 - |e_2|, 1 - C)$, $(1 - |e_1|, 1 - C, 1 - |e_3|)$, and $(1 - C, 1 - |e_2|, 1 - |e_3|)$, respectively. The upper and bottom images are obtained by using 3^3 ($r = 1$) and 7^3 ($r = 3$) voxel neighborhoods to the fitting process, respectively.

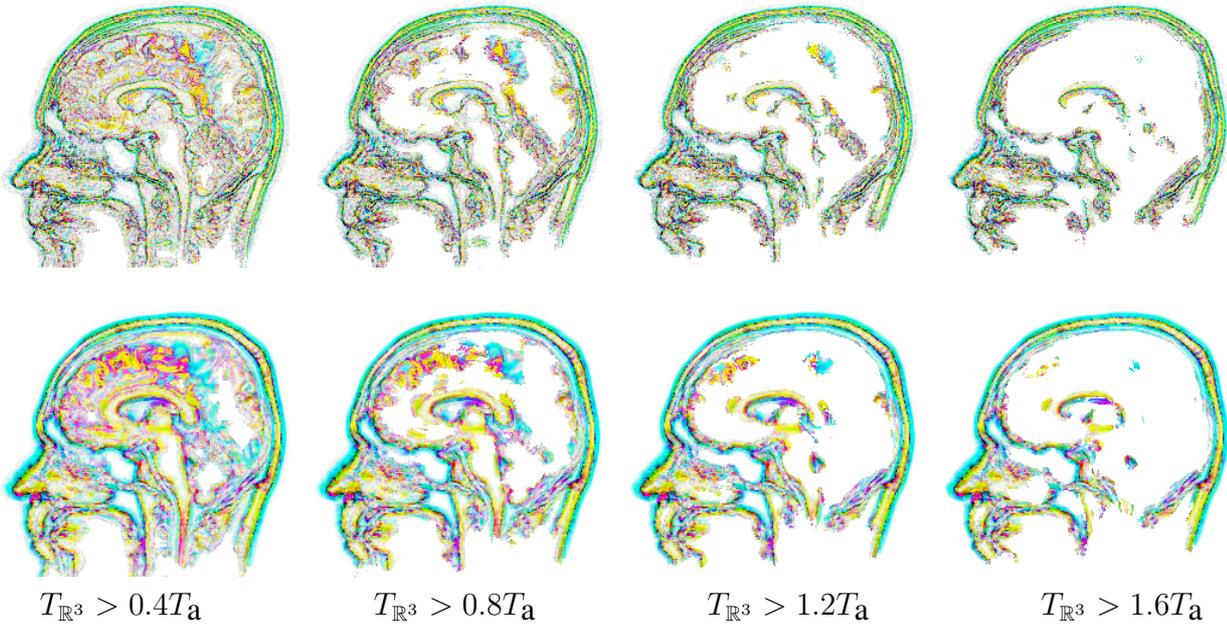


Figure 11: Filtering phantom ridge-like structures via the conformal invariant differential quantity (1), where the input image and its extremalities $(R,G,B) = (1-|e_1|, 1-|e_2|, 1-|e_3|)$ corresponding to $T_{\mathbb{R}^3} \geq 0$ are illustrated in Figure 10 (upper: $r = 1$ and bottom: $r = 3$), we use $r_f = 5$, and T_a is an average of $T_{\mathbb{R}^3}$ within the input image.

7. Conclusion

In this paper, we have developed a numerically efficient approach to extracting the curvature extremalities from multidimensional images. We have also introduced the formula and differential invariant measure for extremalities. Our numerical experiments demonstrate that the proposed approach is capable of detecting salient ridge-like image structures on real-world images. We believe that our study in a multidimensional analogue of the surface creases as geometric shape descriptors will be beneficial to obtain new aspects of feature analysis in early vision tasks.

Acknowledgements. This work was supported in part by Grants-in-Aid for Scientific Research of Japan (20700175 and 20113007).

References

- [1] A. G. Belyaev, A. A. Pasko, and T. L. Kunii. Ridges and ravines on implicit surfaces. In *Proc. of Computer Graphics International*, pages 530–535, 1998.
- [2] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. of Eurographics Symposium on Geometry Processing*, pages 177–187, 2003.
- [3] V. Chandru, D. Dutta, and C. M. Hoffmann. On the geometry of Dupin cyclides. *The Visual Computer*, 5(5):277–290, 1989.

- [4] D. Eberly. *Ridges in Image and Data Analysis*. Kluwer, 1996.
- [5] S. Fofou and L. Garnier. Dupin cyclide blends between quadric surfaces for shape modeling. *Computer Graphics Forum*, 23(3):321–330, 2004. Proc. of Eurographics.
- [6] J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. on Graphics*, 23(1):45–63, 2004.
- [7] P. L. Hallinan, G. G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Tree-Dimensional Patterns of the Face*. A K Peters, 1999.
- [8] R.M. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22(1):28–38, 1983.
- [9] M. Hosaka. *Modeling of Curves and Surfaces in CAD/CAM*. Springer, 1992.
- [10] J. J. Koenderink. *Solid Shape*. MIT Press, 1990.
- [11] O. Monga, S. Benayoun, and O.D. Faugeras. From partial derivatives of 3-D density images to ridge lines. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 354–359. IEEE, 1992.
- [12] S. Musuvathy, T. Martin, and E. Cohen. Ridge extraction from isosurfaces of volumetric data using implicit B-splines. In *Shape Modeling International*, pages 163–174, 2010.
- [13] X. Pennec, N. Ayache, and J. P. Thirion. Landmark-based registration using features identified through differential geometry. In I. N. Bankman, editor, *Handbook of Medical Imaging*, pages 499–513. Academic Press, 2000.
- [14] I. R. Porteous. *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge University Press, 1994.
- [15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [16] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Proc. of International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 486–493, 2004.
- [17] T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Trans. on Visualization and Computer Graphics*, 16:109–119, 2010.
- [18] X. Tricoche, G. Kindlmann, and C.-F. Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.
- [19] S. Yoshizawa, A. Belyaev, and H. Yokota. Curvature extremalities for intelligent shape and image interrogation. In *International Conference on Geometry and Graphics*, pages 242:1–242:10. International Society for Geometry and Graphics, 2010.
- [20] S. Yoshizawa, A. Belyaev, H. Yokota, and H.-P. Seidel. Fast, robust, and faithful methods for detecting crest lines on meshes. *Computer Aided Geometric Design*, 25(8):545–560, 2008.

Received