© Artificial Intelligence in Vision and Ophthalmology 2026; 2:1-27
Original article
https://doi.org/10.35119/aivo.v2i1.155

Tubular curvature filter: pointwise curvature calculation for tubular objects in images

Elifnur Sunger*[1], Beyza Kalkanli*[1], Veysi Yildiz[2], Tales Imbiriba[3], Giovanna Guidoboni[4], Peter Campbell[5], Deniz Erdogmus[1]

[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA; *[2]Reinforce Labs, Palo Alto, CA, USA*; *[3]Department of Computer Science, University of Massachusetts Boston, Boston, MA, USA*; *[4]Department of Electrical and Computer Engineering, University of Maine, Orono, ME, USA*; *[5]Casey Eye Institute, Oregon Health and Sciences University, Portland, OR, USA*

Abstract

Purpose: Accurate estimation of blood vessel tortuosity from medical images is an extremely important and challenging task. It is particularly relevant in the context of retinopathy of prematurity (ROP), where the staging of disease severity and consequent therapeutic approaches are heavily informed by the presence and prominence of vessel tortuosity. Existing methods based on centerline or skeleton curvature fail to capture curvature gradients across a rotating tubular structure, thereby limiting their effectiveness in the case of ROP.
Design: A methodological study with validation on artificial and real images.
Methods: This paper defines local tubular curvature and presents the Tubular Curvature Filter (TCF) method, which locally calculates the acceleration of curve bundles traversing a tubular object parallel to its centerline. These curves are parallel to the centerline in the sense that they are integral curves following the direction of the ridge. This is achieved by examining the directional rate of change in the eigenvectors of the Hessian matrix of a tubular intensity function in space. TCF implicitly calculates the local tubular curvature without the need to explicitly segment or extracting the centerline of the tubular object.
Results: Experimental results demonstrate that TCF provides accurate estimates of

Correspondence: Deniz Erdogmus, Department of Electrical and Computer Engineering, Northeastern University, 360 Huntington Avenue, Boston, MA, USA.
E-mail: d.erdogmus@northeastern.edu

local curvature at any point inside tubular structures. Results on two- and three-dimensional images show that TCF discerns curvature differences between the inner and outer sides of curved tubular objects, while centerline-based approaches cannot. *Conclusion*: Our findings highlight that TCF's ability to discern between the inner and outer sides of curved tubular objects is particularly useful in medical fields that require vasculature curvature analysis from images, especially where vascular structures often have non-uniform diameters, such as in ROP.

Keywords: curvature, retinopathy of prematurity, tortuosity, tubular structure, vascular structure

1. Introduction

Curvature calculations in images have utility in many application domains, such as object recognition,[1,2] corner detection,[3–5] face recognition,[6] image registration,[7] image reconstruction,[8], image denoising,[9–11] and segmentation.[12–16] Curvature calculations are particularly important in the context of retinopathy of prematurity (ROP), a vasoproliferative disease that poses high risks for retinal detachment and irreversible vision loss in infants who are born prematurely.[17] ROP treatment options and outcomes strongly depend on the severity of the disease stage, which is classified based on the appearance of blood vessels in the retina.[18] In particular, the presence and prominence of vessel tortuosity is a major indicator of disease severity,[18] hence the importance of accurately computing curvature.

The image processing literature explores multiple forms of curvature, such as the Gauss curvature of an intensity function obtained from the image of a tubular object, *e.g.*, a vessel, viewed as a two-dimensional (2-D) surface. Gauss curvature for such a surface is defined as the product of the principal curvatures and is an intrinsic property of the surface.[19] Gauss curvature is commonly employed in image processing to assess local smoothness and feature extraction.[20] Another curvature definition that is encountered and employed in image processing is mean curvature. Mean curvature is the average of the principal curvatures of a surface and is an extrinsic property of the surface.[21] However, the problem set of interest for this paper cannot be addressed with existing definitions. Motivated by ROP vessel analysis, this paper introduces a method for calculating curvature locally along a bundle of curves that extends through a tubular object in the image, without extracting the centerline.

In medical applications, centerline-based curvature methods are commonly used to design medical devices;[22] assess disease severity;[23,24] segment vascular structures;[25,26] model and visualize vascular or bony structures;[27–32] analyze fluid-structure interactions in arterial flow[33] and blood flow in the retina;[34] and understand how bone curvature influences the response to mechanical stimuli.[35] There are also deep learning methods to estimate the centerline curvature of bony or vascular structures.[36–38] Note that centerline-based curvature assessments do not accurately capture the varying thickness of tubular structures or the differences in curvature

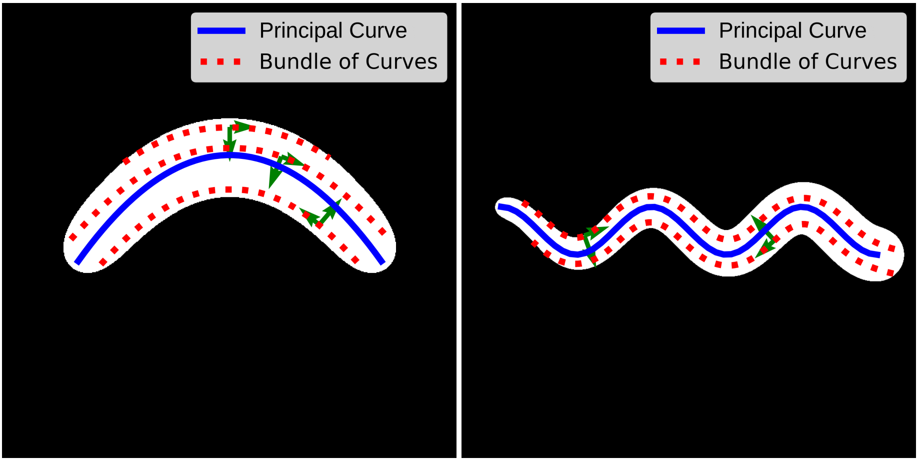


Fig. 1. Example functions: *(left)* a quadratic function and *(right)* a sine wave with varying width. The principal curve (PC)[40] is shown in blue, an example bundle of curves in red, and the Hessian eigenvectors at sample points are indicated by green arrows. The PC follows the ridge of the function, while the bundle of curves is parallel to the centerline in the sense that they are integral curves following the direction of the ridge.

between the inner and outer sides of tubular objects during turns. Since biological vascular structures often vary in diameter, this difference is crucial for modeling purposes, such as the relationship between curvature and local wall shear stress, with local wall shear stress being known to differ between the inner and outer bends of curved vessels.[39]

This paper focuses on local tubular curvature and proposes the Tubular Curvature Filter (TCF) method, which is based on the local acceleration of tangent vectors to a bundle of curves in the tubular object. These curves are parallel to the centerline in the sense that they are integral curves following the direction of the ridge, which are traced by the major eigenvectors corresponding to the large eigenvalue of an at least thrice-differentiable intensity function that highlights the tubular object. In prior work, these tangent vectors are identified as specific eigenvectors of the Hessian of the intensity function.[40] Figure 1 is an illustration of two functions and their principal curves, along with bundle of curves and the eigenvectors of the Hessian at sample points. The method assumes that the image is appropriately preprocessed to yield an intensity image that highlights the tube-shaped objects and is based on interpolating this intensity image to obtain an intensity function over spatial coordinates.

To the best of our knowledge, TCF is the first pointwise curvature calculation method for tubular objects. TCF can be applied pointwise to any thrice continuously differentiable interpolated vessel/tube intensity image, thereby eliminating the need to segment or extract centerlines of tubular objects, unlike centerline-based methods. Experiments with 2-D and 3-D images show that TCF can discern curvature differences between the inner and outer sides of a curved tubular object, achieving a level of spatial precision that centerline-based approaches cannot. With appropriate local image interpolation techniques, TCF is highly suitable for parallel processing.

2. Methods

This section describes TCF, develops a method to compute local tubular curvature at any point for a given thrice continuously differentiable function of Cartesian spatial coordinates in an Euclidean space, and discusses application to intensity images of tubular shapes.

2.1 Tubular curvature

To compute the curvature of tubular shapes over space we exploit the geometrical interpretation of the eigenvalues of the Hessian provided by Ozertem *et al.*,[40] who redefined principal curves and surfaces in terms of first and second derivatives of probability density functions. This definition of the principal curves corresponds to the ridges of said probability density function. We motivate our discussion using the example in Figure 1, which presents a discretized image of a function that forms a tubular shape over 2-D space; function values are pixel intensities. The *principal curve* that forms the ridge of this function and bundle of curves traced by select eigenvectors of the Hessian matrix, as defined by Ozertem *et al.*,[40] are represented by the solid blue and red-dotted lines, respectively. The eigenvectors of the function's Hessian evaluated at a given point are represented by green arrows. At any point over the 2-D space, such eigenvectors point toward the principal and to a direction tangential to the bundle of curves. The following definition is adapted from *Definitions 1-4* in Ozertem *et al.*[40]

2.1.1 Definition 0

A point $\mathbf{x}$ is on the major ridge/valley curve of an at least three times continuously differentiable function $p : \mathbb{R}^n \to \mathbb{R}$ if the $(n-1)$ eigenvectors of its Hessian matrix corresponding to the $(n-1)$ smallest/largest eigenvalues have zero inner product with its gradient, and all of these eigenvalues are negative/positive.

Note that if these $(n-1)$ eigenvalues have mixed signs, the curve is a cliffside path. Also note that, as demonstrated by Ozertem *et al.*,[40] the major ridge/valley curves remain unchanged if a monotonically increasing function is applied to the value of p. This will be used in the proposed method derivation in Appendix A.

The bundle of curves that are of interest for a given ridge/valley at any point in space is traced by major Hessian matrix eigenvectors, similar to the ridge/valley, and their curvatures can be defined based on the rate of change of eigenvectors tangent to the bundle of curves; for unit-length eigenvectors, this rate of change becomes the acceleration vector. Curvature of these curves at any point is the norm of the local acceleration vector. When computing the curvature of tubular structures in an image, an interpolation method provides a thrice-continuously differentiable intensity function, from which the necessary gradient and Hessian values are computed for any given point.

Without loss of generality, in this paper, we consider the case where the centerline of the tubular object is a ridge curve. Let $\mathbf{x}(s) \in \mathbb{R}^2$ be a point on a ridge curve with arc length parameter s[41]; $p(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}_+^* = \{p(\mathbf{x}(s)) \in \mathbb{R} \,|\, p(\mathbf{x}(s)) > 0\}$ be a positive, at least thrice differentiable function, for example, an interpolated image as explained in Section 2.2. $\mathbf{H}(\mathbf{x}(s))$ denotes the Hessian of the logarithm of the function (see Appendix A), with $\{((\lambda_1(\mathbf{x}(s)), \mathbf{q}_1(\mathbf{x}(s)))\,, ((\lambda_2(\mathbf{x}(s)), \mathbf{q}_2(\mathbf{x}(s)))\}$ eigenvalue-vector pairs of $\mathbf{H}(\mathbf{x}(s))$, where $\lambda_1(\mathbf{x}(s)) > \lambda_2(\mathbf{x}(s))$.

2.1.2 Definition 1

Acceleration $\mathbf{a}(\mathbf{x}(s)) = \partial \mathbf{v}(\mathbf{x}(s))/\partial s$, at any point $\mathbf{x}(s)$, is the rate of change of velocity $\mathbf{v} = \mathbf{q}_1$ with respect to the arc length parameter s, which is the unit-length tangent to the curve. Using chain rule:

$$\frac{\partial \mathbf{v}(\mathbf{x}(s))}{\partial s} = \frac{\partial \mathbf{q}_1(\mathbf{x}(s))}{\partial s} = \frac{\partial \mathbf{q}_1(\mathbf{x}(s))}{\partial \mathbf{x}(s)} \frac{\partial \mathbf{x}(s)}{\partial s}. \tag{1}$$

Since $\mathbf{q}_1(\mathbf{x}(s))$ is the unit length velocity vector, we write $\partial \mathbf{x}(s)/\partial s = \mathbf{q}_1(\mathbf{x}(s))$ as $\partial \mathbf{x}(s) = \mathbf{q}_1(\mathbf{x}(s))\partial s$.

2.1.3 Definition 2

Tubular curvature of the intensity function $p(\mathbf{x}(s))$ at $\mathbf{x}(s)$ is the Euclidean norm of its acceleration vector: curvature $c(\mathbf{x}(s)) = ||\mathbf{a}(\mathbf{x}(s))||$.

In the remainder of this paper, we will not explicitly indicate dependency on s for notation simplicity.

2.1.4 Derivation 1

To calculate $\mathbf{a}$ at $\mathbf{x}$, the directional derivatives of $\mathbf{q}_1$, i.e., $\partial \mathbf{q}_1/\partial x_1$ and $\partial \mathbf{q}_1/\partial x_2$, are required, which can be computed using the eigendecomposition of $\mathbf{H}(\mathbf{x}) = \mathbf{Q}(\mathbf{x})\mathbf{\Lambda}(\mathbf{x})\mathbf{Q}(\mathbf{x})^{\mathbf{T}}$ as:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2\end{bmatrix} \begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix} \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2\end{bmatrix}^T, \tag{2}$$

where $\mathbf{Q}$ is the 2×2 orthonormal matrix whose i^{th} column is the eigenvector $\mathbf{q}_i$, and $\mathbf{\Lambda}$ is diagonal with elements as corresponding eigenvalues. In Equation (2), $\mathbf{H}(\mathbf{x})$ is assumed to be diagonalizable with two linearly independent eigenvectors. If this assumption is invalid, ambiguity in eigenvectors leads to inability to uniquely trace a curve passing through point $\mathbf{x}$, resulting in local curvature ambiguity.

Calculating $\partial \mathbf{q}_1(\mathbf{x})/\partial \mathbf{x}$ requires taking derivatives of this Hessian decomposition. Let $\mathbf{T}_i(\mathbf{x})$, $\mathsf{V}_i(\mathbf{x})$ and $\mathbf{L}_i(\mathbf{x})$ be the i^{th} directional derivatives of $\mathbf{H}(\mathbf{x})$, $\mathbf{Q}(\mathbf{x})$ and $\mathbf{\Lambda}(\mathbf{x})$ respectively:

$$\mathbf{T}_i(\mathbf{x}) = \frac{\partial \mathbf{H}(\mathbf{x})}{\partial x_i} = \begin{bmatrix} \mathsf{t}_{i11} & \mathsf{t}_{i12} \\ \mathsf{t}_{i21} & \mathsf{t}_{i22} \end{bmatrix}, \tag{3}$$

$$\mathsf{V}_i(\mathbf{x}) = \frac{\partial \mathbf{Q}(\mathbf{x})}{\partial x_i} = \begin{bmatrix} \frac{\partial \mathbf{q}_1}{\partial x_i} & \frac{\partial \mathbf{q}_2}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \mathsf{v}_{i11} & \mathsf{v}_{i12} \\ \mathsf{v}_{i21} & \mathsf{v}_{i22} \end{bmatrix}, \tag{4}$$

$$\mathbf{L}_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial \lambda_1}{\partial x_i} & 0 \\ 0 & \frac{\partial \lambda_2}{\partial x_i} \end{bmatrix} = \begin{bmatrix} l_{i1} & 0 \\ 0 & l_{i2} \end{bmatrix}. \tag{5}$$

Omitting dependencies on $(\mathbf{x})$ for brevity, $\mathbf{T}_i(\mathbf{x})$ is:

$$\mathbf{T}_i = \mathsf{V}_i \mathbf{\Lambda} \mathbf{Q}^{\mathbf{T}} + \mathbf{Q}\mathbf{L}_{\mathbf{i}}\mathbf{Q}^{\mathbf{T}} + \mathbf{Q}\mathbf{\Lambda}\mathsf{V}_{\mathbf{i}}^{\mathbf{T}}. \tag{6}$$

Here, the derivatives of eigenvectors and eigenvalues are unknown. Since $\mathbf{T}_i$ matrices are symmetric, we get 3 equations from $\mathbf{T}_i$. Using two properties of eigenvectors, we obtain additional equations: 1) eigenvectors are unit length; hence, derivatives of their magnitude are zero, and 2) they are perpendicular to each other. With these, we have:

$$\frac{\partial ||\mathbf{q}_1(\mathbf{x})||^2}{\partial x_i} = 2\mathbf{q}_1^T \frac{\partial \mathbf{q}_1}{\partial x_i} = 0, \tag{7}$$

$$\frac{\partial ||\mathbf{q}_2(\mathbf{x})||^2}{\partial x_i} = 2\mathbf{q}_2^T \frac{\partial \mathbf{q}_2}{\partial x_i} = 0, \tag{8}$$

$$\frac{\partial \mathbf{q}_1^T \mathbf{q}_2}{\partial x_i} = \mathbf{q}_1^T \frac{\partial \mathbf{q}_2}{\partial x_i} + \frac{\partial \mathbf{q}_1^T}{\partial x_i}\mathbf{q}_2 = 0. \tag{9}$$

From $\mathbf{T}_i$ and the properties of eigenvectors, we obtain six equations with 12 unknowns. These yield the following sets of linear equations from Equations (6)-(9), where $\mathbf{q}_i = \begin{bmatrix} \mathsf{q}_{i1} & \mathsf{q}_{i2} \end{bmatrix}^T$:

$$\begin{bmatrix} \mathsf{t}_{111} \\ \mathsf{t}_{112} \\ \mathsf{t}_{122} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathsf{M} \begin{bmatrix} \mathsf{v}_{111} \\ \mathsf{v}_{112} \\ l_{11} \\ l_{12} \\ \mathsf{v}_{121} \\ \mathsf{v}_{122} \end{bmatrix} \text{and} \begin{bmatrix} \mathsf{t}_{211} \\ \mathsf{t}_{212} \\ \mathsf{t}_{222} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathsf{M} \begin{bmatrix} \mathsf{v}_{211} \\ \mathsf{v}_{212} \\ l_{21} \\ l_{22} \\ \mathsf{v}_{221} \\ \mathsf{v}_{222} \end{bmatrix}, \tag{10}$$

$$\mathsf{M} = \begin{bmatrix} 2\lambda_1 \mathsf{q}_{11} & 2\lambda_2 \mathsf{q}_{21} & \mathsf{q}_{11}^2 & \mathsf{q}_{21}^2 & 0 & 0 \\ \lambda_1 \mathsf{q}_{12} & \lambda_2 \mathsf{q}_{22} & \mathsf{q}_{11}\mathsf{q}_{12} & \mathsf{q}_{21}\mathsf{q}_{22} & \lambda_1 \mathsf{q}_{11} & \lambda_2 \mathsf{q}_{21} \\ 0 & 0 & \mathsf{q}_{12}^2 & \mathsf{q}_{22}^2 & 2\lambda_1 \mathsf{q}_{12} & 2\lambda_2 \mathsf{q}_{22} \\ \mathsf{q}_{11} & & 0 & 0 & \mathsf{q}_{12} & 0 \\ 0 & \mathsf{q}_{21} & 0 & 0 & 0 & \mathsf{q}_{22} \\ \mathsf{q}_{21} & \mathsf{q}_{11} & 0 & 0 & \mathsf{q}_{22} & \mathsf{q}_{12} \end{bmatrix}. \tag{11}$$

We provide the complete derivation of linear equations that leads to **M** in Appendix A. Please see Appendix B for derivations of linear equations solved in order to obtain the acceleration vector for functions with n-dimensional arguments, including the 3-D special case (for volumetric images).

By solving the linear equations in Appendix A, we find $\mathbf{V}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$. Then, we calculate $\partial \mathbf{q}_1(\mathbf{x})/\partial \mathbf{x}$ from $\mathbf{V}(\mathbf{x})$. Finally, we obtain the acceleration vector at $\mathbf{x}$ following Equation (1).

2.2 Tubular Curvature Filter on images

The methodology discussed above is designed to operate over continuous thrice differentiable functions defined over $\mathbb{R}^2$ (and in general $\mathbb{R}^n$). To apply the TCF to images, we need to smoothly interpolate intensity images to obtain intensity functions with necessary continuity and differentiability properties. This paper does not advocate a particular interpolation methodology. For illustration, we consider a kernel regression-based approach. Note that to operate on continuous thrice-differentiable functions defined over $\mathbb{R}^2/\mathbb{R}^3$, tubular shapes should have a width of no less than 3 or 4 pixels/voxel lengths.

We use point location to calculate curvature; thus, a pixel can be converted to physical units as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (p_1 - c_1 + 0.5) \times \nu_1 \\ (p_2 - c_2 + 0.5) \times \nu_2 \end{bmatrix}, \tag{12}$$

where $\mathbf{p} \in \mathbb{Z}_+^2$ corresponds to a pixel in the image, $\mathbf{x} \in \mathbb{R}^2$ represents the physical location in physical units of pixel $\mathbf{p}$, $\mathbf{c} \in \mathbb{Z}_+^2$ is the pixel corresponding to the physical origin in pixel coordinates, and $\nu \in \mathbb{R}^2$ represents the physical lengths per pixel in the horizontal and vertical directions. A similar conversion can be done for three-dimensional images (or n-dimensional). $\tilde{\mathbf{x}}_j \in \mathbb{R}^2$ and $w_j \in \mathbb{R}_+$, $j \in \{1, \ldots, n\}$ are the location and intensity of the j-th pixel on a image, and $K_{S_j}(\cdot) : \mathbb{R}^2 \to \mathbb{R}$ is the interpolation kernel with scale matrix $\mathbf{S}_j$. The kernel-based interpolation is:

$$p(\mathbf{x}) = \sum_{j=1}^{n} w_j K_{\mathbf{S}_j}(\mathbf{x} - \tilde{\mathbf{x}}_j). \tag{13}$$

Variable-scale (width) kernels may be employed in interpolation to achieve robustness against outlier pixel intensities that may arise due to noise or other reasons. The approach outlined in Ozertem *et al.*[42] proposes to select k-nearest pixels to $\mathbf{x}$. For each $\mathbf{x}$, a sequence $\{\tilde{\mathbf{x}}_{\pi(1)}, \tilde{\mathbf{x}}_{\pi(2)}, \ldots, \tilde{\mathbf{x}}_{\pi(n)}\}$ is defined, where $\pi(i) \in \{1, 2, \ldots, n\}$ represents the sorted set of pixel indexes such that $\|\mathbf{x} - \tilde{\mathbf{x}}_{\pi(i)}\|_2 \leq \|\mathbf{x} - \tilde{\mathbf{x}}_{\pi(i+1)}\|_2$, and the k-nearest pixels are selected based on this sequence, denoted as $\{\tilde{\mathbf{x}}_{\pi(1)}, \tilde{\mathbf{x}}_{\pi(2)}, ..., \tilde{\mathbf{x}}_{\pi(k)}\}$, with $k \leq n$. These k-nearest pixels can then be used for kernel-based interpolation:

$$p_\pi(\mathbf{x}) = \sum_{j=1}^{k} w_{\pi(j)} K_{\mathbf{s}_j}(\mathbf{x} - \tilde{\mathbf{x}}_{\pi(j)}). \tag{14}$$

Given $p(\mathbf{x})$, considering that the tubular structure of interest is represented with large function values (high-intensity pixels over a dark background), the application of the TCF is straightforward. The methodology presented in Section 2.1 was derived, without loss of generality, for images with tubular shapes represented by light pixels over dark backgrounds and used $\mathbf{q}_1$ for curvature calculations. If tubular objects appear as dark pixels over a light background, the intensity function is negated to achieve tubular objects with high intensity values.

In terms of computational complexity, pixel-wise curvature calculation itself is constant time since both Hessian eigendecomposition and solving linear equations are constant time O(1). However, interpolating intensity images to obtain intensity functions with third-order differentiability is linear in time $O(N \times k)$, where *N* is the number of pixels and k is the number of neighborhood pixels used in interpolation. Source code is available at: https://github.com/neu-spiral/TCF.

3. Results

TCF is applied to artificial and medical images in both 2-D and 3-D. Within this section, we have categorized the results based on the number of dimensions and image type (artificial or medical). We also included additional analyses to identify the limits and capabilities of TCF. If not specified otherwise, Gaussian kernels are used to obtain these results, but other unimodal kernels can be used as well. For medical images, one should carefully consider structures in the image and choose interpolation parameters accordingly, as selecting appropriate Gaussian scales is important when adjacent tubular structures are present in the image[43,44] or in noisy images.[45]

3.1 Two-dimensional results

This subsection has three parts: (1) curvature results for artificial images with various curved structures, (2) analyses of medical images, and (3) additional analyses on TCF and its comparison with centerline-based curvature.

3.1.1 Artificial images

This section presents 2-D artificial images, including a blurry ring, two sine waves, and a sine wave with changing amplitude and frequency. Figure 2 shows these images with their TCF curvature results, where brighter pixels indicate higher curvature values.

To generate images of the two sine waves, and a sine wave with changing amplitude and frequency, we employ Gaussian kernels as described in Equation (13),

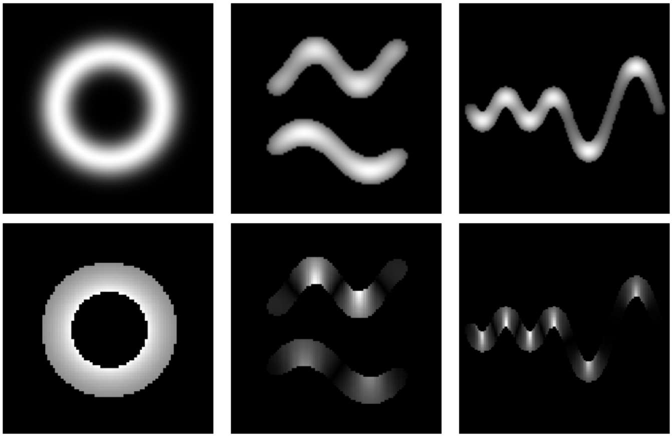


Fig. 2. (First row) Blurry ring function $p_{\text{ring}}(\mathbf{x})$, two sine functions with different frequencies $p_{sin,f}(\mathbf{x})$, and a sine function with changing amplitude and frequency $p_{sin,A,f}(\mathbf{x})$ are presented, respectively. *(Second row)* Curvature values with TCF, where brighter pixels indicate higher curvature values.

with the scale matrix $\mathbf{S}_j = 2\mathbf{I}_2$, where $\mathbf{I}_2$ represents the two-dimensional identity matrix, and $w_j = 1$ for $j \in \{1, \ldots, n\}$. All functions are defined by placing evenly spaced isotropic Gaussian kernels, followed by the creation of images using uniform grids $\mathcal{G}(l, a, b)$ where $\mathcal{G}$ represents a uniform grid with l samples in the interval $[a, b]$.

Image of a blurry ring

$p_{\text{blurry\_ring}}(\mathbf{x}) = e^{-(\sqrt{\mathbf{x}^T\mathbf{x}}-r)^2/m}$ is a blurry ring function where $r = 1$ is the radius of the ring and $m = 0.1$ is the thickness parameter. The first column in Figure 2 displays $p_{\text{blurry\_ring}}(\mathbf{x})$ along with its curvature results. The curvature is reciprocal to the radius of the ring at every point in the ring function. Consequently, points closer to the center of the ring are expected to exhibit higher curvature values, as demonstrated by TCF.

Image of two sine waves

We present the difference in curvature values for two sine functions with different frequencies. We place equally spaced isotropic Gaussian kernels on $y = \sin(f\mathbf{x})$ function. The sine function is defined as $p_{sine,f}(\mathbf{x}) = \sum_{j=1}^{n} K_I(\mathbf{x} - \mathbf{x}_{j,f})$, where $\mathbf{x}_{j,f} \in \{\mathbf{x} = [m, \sin(fm)]^T \mid m \in \mathcal{G}(50, -\pi, \pi)\}$. f is the frequency of the sine wave. We created an image using a higher-frequency sine function $p_{\text{sin,hf}}(\mathbf{x})$ with $f = 1.5$ and a lower-frequency sine function $p_{\text{sin,lf}}(\mathbf{x})$ with $f = 1.0$.

The third column in Figure 2 presents the two sine waves with their TCF results. As expected, TCF values at local minima and maxima in $p_{\text{sin,hf}}(\mathbf{x})$ are higher than those in $p_{\text{sin,lf}}(\mathbf{x})$.

Image of a sine wave with changing amplitude and frequency

We show the change of the curvature values along a single sine function by modifying its amplitude and frequency. Equally spaced isotropic Gaussian kernels are placed on the $y = A\sin(f\mathbf{x})$ function, defined as $p_{sine,A,f}(\mathbf{x}) = \sum_{j=1}^{n} K_I(\mathbf{x} - \mathbf{x}_{j,A,f})$, where $\mathbf{x}_{j,A,f} \in \{\mathbf{x} = [m, A\sin(fm)]^T \mid m \in \mathcal{G}(50, -2\pi, 2\pi)\}$. f is the frequency and A is the amplitude of the sine wave. We generated an image by varying the values of A and f along the function. Specifically, for the first 25 samples (in the interval $[-2\pi, 0]$), we set $A = 1$ and $f = 2$, while for the remaining samples (in the interval $[0, 2\pi]$), we used $A = 3$ and $f = 1$.

The last column in Figure 2 presents $p_{sin,A,f}(\mathbf{x})$ with its curvature results. Along the function, the amplitude of the sine wave increases while the frequency decreases. TCF gives higher values at the local minima and maxima in $p_{sin,A,f}(\mathbf{x})$ at the first half of the function since the frequency in the first half is higher than the second half. Also, the amplitude of the sine wave increases along the function, and this creates flat function parts where the curvature values are almost zero.

3.1.2 *Medical images*

We examine 2-D medical images from the Retina[46] and X-ray coronary angiogram[47] datasets. Figure 3 presents these examples with their TCF curvature results. In the curvature images, intensity values indicate curvature, with brighter pixels representing higher values. Also, contrast enhancement is applied for visualization. Gaussian kernels, as explained in Equation (14), are applied to the medical examples using the scale matrix $\mathbf{S}_j = 0.5 \times \mathbf{I}_2$, where $\mathbf{I}_2$ is the 2-D identity matrix, and $k = 440$.

Retinal images

The two panels on the left of Figure 3 show a sample zone II retinal vessel segmentation image and its curvature results. The segmented retinal image includes branching points and vessel segments with different degrees of tortuosity. Branching points occur where a vessel splits into smaller vessels or where multiple vessel segments overlap. Branching points are naturally high-curvature regions because vessels create a cross-shaped structure at these points. We observe this in TCF. Since retinal vessels

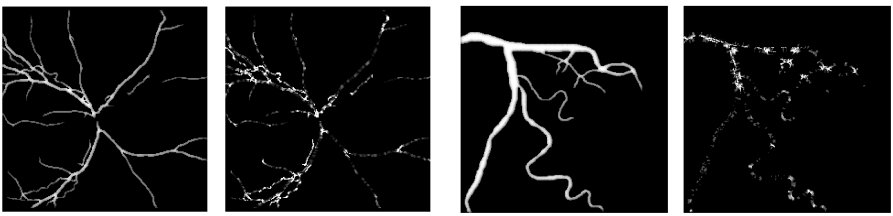


Fig. 3. *(Left)* An interpolated retinal image and its curvature calculated using TCF. *(Right)* An interpolated angiography image and its curvature calculated using TCF. Contrast enhancement is applied to the curvature values for visualization. In the curvature plots, intensity values represent curvature, with brighter pixels indicating higher values.

originate around the optic disc, many branching points appear in this region, where we observe high curvature values. The vessel segments on the left of the image have high tortuosity, whereas the segments on the right are almost flat. The curvature image shows higher values for tortuous vessel segments compared to other segments, demonstrating that TCF captures the tortuosity of retinal blood vessels.

See Section 3.1.3 for a detailed analysis of retinal images with ROP categories and an explanation of why centerline methods fail to capture vessel structures.

Coronary angiography images

Figure 3 shows the interpolation function of a segmented coronary angiography and its TCF curvature values on the right. Similar to the retina example in Section 3.1.2, high curvature values appear at branching points and in highly tortuous regions. The vessel segment at the bottom left is nearly flat, while the right segment is highly tortuous. TCF captures this difference, assigning low curvature values to the flat segment and high values to the tortuous regions.

3.1.3 Additional analyses

This section analyzes TCF to assess centerline limitations, kernel choices, intensity variations, and image resolution. In all cases where we define a ring function, we use an inner radius of 0.5 and an outer radius of 1.0, except where noted in “Limitations of the centerline” section. We also include an analysis of the numerical stability of TCF on a retinal image. Additionally, we analyze retinal images to qualitatively highlight regions where the method fails for different stages of ROP.

We define the ground-truth curvature for ring images as $1/r$, where r is the distance of pixels from the ring’s center. For ring examples with an inner radius of 0.5 and the outer radius is 1.0, the curvature values range from 1.0 to 2.0. We evaluate performance using the mean absolute difference between the ground truth and TCF.

Limitations of the centerline

In this section, we focus on why centerline methods cannot capture curvature changes across the image. We present two examples: one compares two different ring examples with the same centerline, and the other examines a sine wave with increasing thickness.

We generate an artificial ring example by defining two circles to generate a ring image $p_{\text{ring}}(\mathbf{x}) = \sum_{j=1}^{n} K(\mathbf{x} - \mathbf{x}_j)$, where $\mathbf{x}_j \in \{\mathbf{x}_j = [r_j \cos(\theta)), r_j \sin(\theta)]^T \mid r_j \in [0.5, 1], \mid \theta \in [0, 2\pi) \mid \mathbf{x}_j \in \mathcal{G}(125, -1.25, 1.25)\}$. K is either the Gaussian kernel. The pixel intensity is set to 1 for pixels between the rings ($w_j = 1$ for $j \in 1, \ldots, n$) and the scale matrix is set to $\mathbf{S}_j = 0.1 \times \mathbf{I}_2$.

Figure 4 shows two ring images with the same centerline: one with radii 0.5 and 1.0 (first row) and another with radii 0.4 and 1.1 (second row). From left to right, it displays the ring function, ground-truth curvature, and TCF results. The ring functions differ but their centerline remains the same. Centerline-based approaches compute curvature along this line but cannot capture curvature variations across the ring's radii.

Figure 5 presents a tube example where the centerline is defined as $y = sin(x)$ and the thickness varies along the tube. From left to right, it shows the tube with its centerline and the TCF results. Similar to the ring example, the centerline alone cannot capture the thickness variations across the tube.

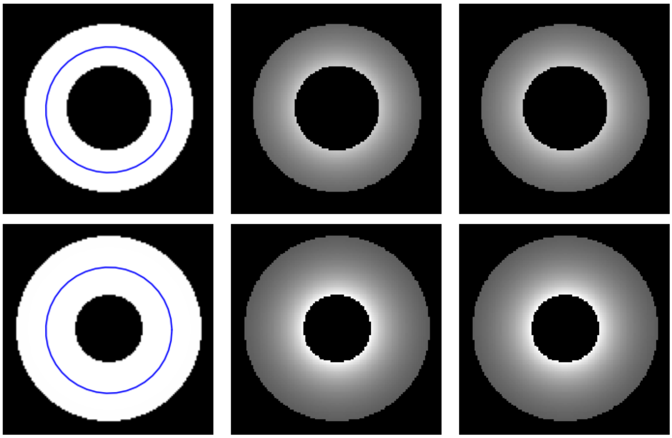


Fig. 4. (First row, left to right) The thin ring function (inner radius 0.5, outer radius 1.0), ground-truth curvature, and TCF result. *(Second row, left to right)* The thick ring function (inner radius 0.4, outer radius 1.1), ground-truth curvature, and TCF result. Intensity values represent curvature, with brighter pixels indicating higher values. Both rings have the same centerline (blue), illustrating that centerline-based methods fail to capture the shape.

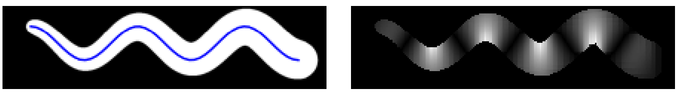


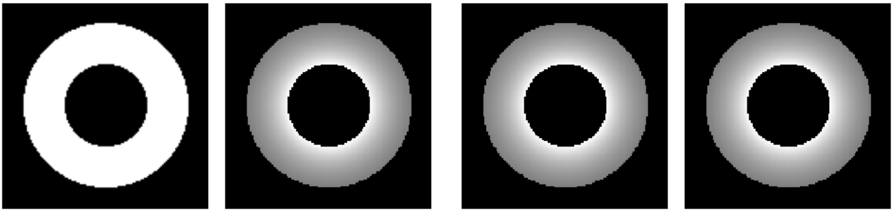
Fig. 5. (Left to right) A tube with changing thickness and its centerline $y = sin(x)$, followed by the curvature result with TCF, where intensity values represent curvature, with brighter pixels indicating higher values. The centerline alone cannot capture the varying thickness, highlighting that methods relying solely on the centerline fail to represent the tube's shape.

Results with different kernels

In this paper, we primarily use the Gaussian kernel; however, in this section, we also employ the Epanechnikov kernel, defined as, $K_e(\mathbf{x}) = 1 - \mathbf{x}^T\mathbf{x}$ for $|\mathbf{x}| \leq 1$, for a given point $\mathbf{x} \in \mathbb{R}^2$.

We generate a ring example as in "Limitations of the centerline" section. Here, K is either the Gaussian kernel or the Epanechnikov kernel.

Figure 6a shows the image of the ring function $p_{\text{ring}}(\mathbf{x})$ and the ground-truth curvature. Fig. 6b shows TCF results from left to right with Gaussian and Epanechnikov kernels. Both kernels perform well in calculating the pixel-wise curvature, with mean absolute differences between the ground-truth and TCF of 1.09×10^{-8} and 5.98×10^{-15}, respectively.



(a) *(b)*

Fig. 6. (a) Ring function $p_{\text{ring}}(\mathbf{x})$ and its ground-truth curvature. *(b)* Curvature values with TCF, from left to right, using the Gaussian and Epanechnikov kernels, with intensity values representing curvature, where brighter pixels indicate higher values. We calculated the mean absolute difference between the ground-truth and TCF. For the Gaussian kernel, it is 1.09×10^{-8}; for the Epanechnikov kernel, it is 5.98×10^{-15}, respectively.

Intensity profiles

We use the ring function $p_{\text{ring}}(\mathbf{x})$ as described in "Limitations of the centerline" section. We evaluate the sensitivity of TCF to different intensity profiles, such as "uniform", where all of the pixels in the region have a value of 1; "higher inside", where $w_j = 1.5 - r_j$; "higher outside", where $w_j = r_j$, with r_j being the distance of the j-th pixel from the center.

Figure 7 demonstrates the ring functions with different intensity values in the first row and their respective TCF results with a Gaussian kernel in the second row. All of these cases perform well in estimating the curvature, achieving mean absolute differences from the ground-truth results of 1.09×10^{-8}, 1.56×10^{-8}, and 0.94×10^{-8}, respectively.

Resolution

We use ring images with pixel widths and heights of $0.02, 0.04$, and 0.08, maintaining uniform intensity as in "Limitations of the centerline" section. These correspond to resolutions of 125×125, 62.5×62.5, and 31.25×31.25. Figure 8 displays these resolutions from left to right in the first row and their TCF results with a Gaussian kernel in the second row. As the resolution decreases, the estimation quality worsens. The mean absolute differences from the ground-truth are 1.09×10^{-8}, 6.75×10^{-8}, and 1.27×10^{-3}, respectively.

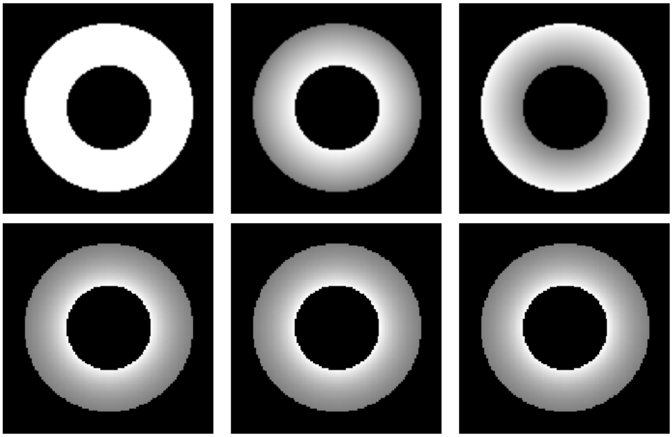


Fig. 7. (First row) Ring function $p_{\text{ring}}(\mathbf{x})$ with different intensity profiles, uniform inside the inner and outer rings, higher near the inner ring, and lower near the outer ring. *(Second row)* Curvature values with TCF: intensity represents curvature, with brighter pixels indicating higher values. We computed the mean absolute difference between the ground truth and TCF. From left to right, the values are: 1.09×10^{-8}, 1.56×10^{-8}, and 0.94×10^{-8}.

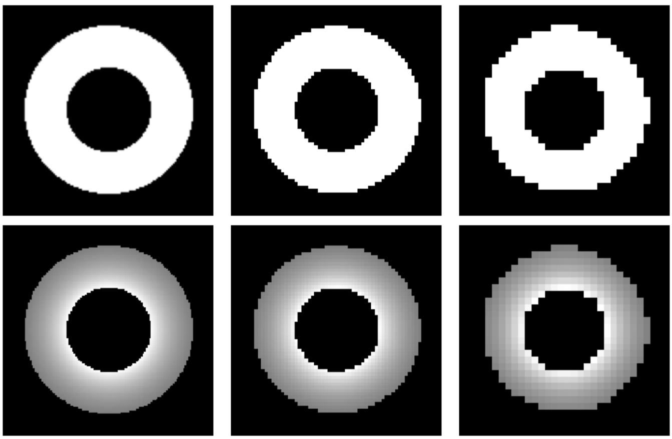


Fig. 8. (First row) Ring function $p_{\text{ring}}(\mathbf{x})$ with resolutions of $125 \times 125, 62.5 \times 62.5$, and 31.25×31.25. *(Second row)* Curvature values with TCF: intensity represents curvature, with brighter pixels indicating higher values. We computed the mean absolute difference between the ground truth and TCF. From left to right, the values are $1.09 \times 10^{-8}, 6.75 \times 10^{-8}$, and 1.27×10^{-3}.

Numerical stability

We focus on the numerical stability of the inversion of the linear system given in Equation (10) for medical images, such as the retinal image. To assess stability, we calculate the condition number of $\mathbf{M}$. A large condition number indicates the matrix is ill-conditioned, meaning that small changes in the input can lead to large changes in the solution, while a smaller condition number suggests greater stability. We calculate the condition number as $\kappa(\mathbf{M}) = \|\mathbf{M}\|_2 \cdot \|\mathbf{M}^{-1}\|_2$.

Figure 9 presents the retinal image and its corresponding curvature image, where pixels with a condition number larger than 100 are marked in orange. The condition number of $\mathbf{M}$ ranges from 5.1 to 234.1. Equation (2) assumes the Hessian matrix is diagonalizable. Therefore, we cannot calculate curvature values with TCF when this occurs, which is expected in the middle of vessel branching regions. Additionally, as shown in Figure 9, even if the Hessian is diagonalizable in the middle of these regions, the condition number of $\mathbf{M}$ is large, indicating that TCF calculations are not numerically stable in these areas.

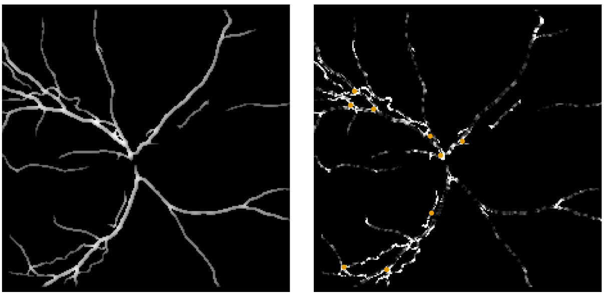


Fig. 9. (Left to right) A retinal image and its curvature with TCF, where pixels with a condition number for matrix M greater than 100 are marked in orange. The condition number of $\mathbf{M}$ ranges from 5.1 to 234.1 for this image.

Retinopathy of prematurity

We focus on the Retina[46] dataset with ROP images labeled as “Normal,” “Pre-plus”, or “Plus”, indicating no ROP, intermediate, or severe cases, respectively. Generally, the more dilated (increased vessel thickness) and tortuous (higher curvature) the vessels, the more severe the ROP.[18] Figure 10 presents Pre-Plus and Plus zone II images from top to bottom, with TCF results in the second column and the centerline in the last. The red box regions highlight examples of dilated vessels, where, as discussed in “Limitations of the centerline” section, the centerline cannot capture the shape in these regions.

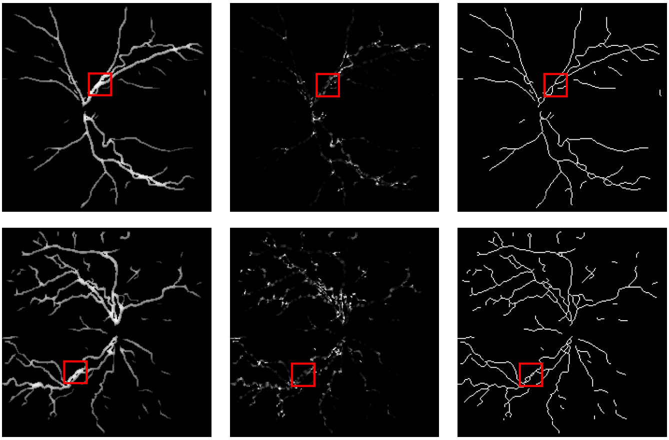


Fig. 10. (Left to right) Images from the Retina dataset (with Pre-Plus and Plus ROP from top to bottom), the corresponding curvature with TCF, and curvature with the centerline. The centerline is calculated using the "thin" function from the morphological operations in scikit-image.[48] We compute the centerline curvature using the implicit planar curvature calculation.[19] Contrast enhancement is applied to the curvature values. In the curvature plots, intensity represents curvature, with brighter pixels indicating higher values. The red squares highlight examples where vessels are dilated and the centerline curvature cannot capture the shape.

3.2 Three-dimensional results

Here, we include results from (1) 3-D artificial images with various curved structures, (2) 3-D medical images, and (3) an additional analysis on voxel isotropy.

3.2.1 *Artificial Images*

We present 3-D artificial images: a blurry ring and a sine wave. Figure 11 displays these images with their corresponding curvature images using TCF, and the corresponding colormap. We employ Gaussian kernels given in Equation (13), with the scale matrix $\mathbf{S}_j = \mathbf{I}_3$, where $\mathbf{I}_3$ is the three-dimensional identity matrix, and $w_j = 1$ for $j \in \{1, \ldots, n\}$. Functions are defined by positioning evenly spaced isotropic Gaussian kernels, followed by the creation of images using uniform grids as explained in Section 3.1.1.

Image of a three-dimensional blurry ring

$p_{\text{blurry\_ring\_3d}}(\mathbf{x}) = \sum_{j=1}^{n} K_I(\mathbf{x} - \mathbf{x}_j)$ is the three-dimensional blurry ring function, which is defined by placing equally spaced isotropic Gaussian kernels. Here, $\mathbf{x}_j \in \{\mathbf{x} = [r \sin(\theta), r\cos(\theta), \sin(r)]^T \mid \theta \in \mathcal{G}(40, 0, 2\pi)\}$, and r is 2. Figure 11 presents $p_{\text{blurry\_ring\_3d}}(\mathbf{x})$ and its curvature results on the left. Similar to Section 3.1.1, the curvature is reciprocal to the radius of the ring and TCF gives higher values to the points closer to the center.

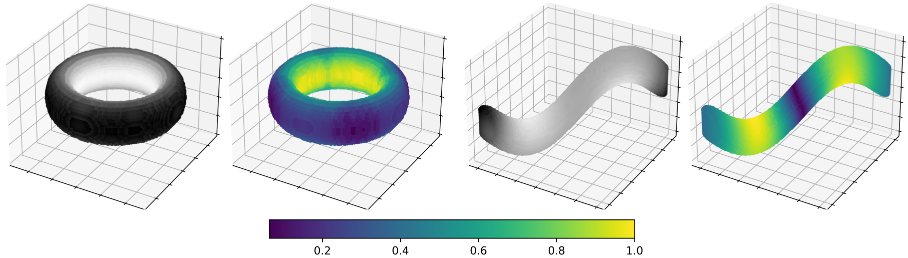


Fig. 11. (Left) The 3-D blurry-ring function image and its TCF result. *(Right)* The 3-D sine function image and its TCF result. *(Bottom)* The Viridis colormap for the curvature plots.

Image of a three-dimensional sine wave

We define a sine function and place equally spaced isotropic Gaussian kernels as $p_{\text{sine\_3d}}(\mathbf{x}) = \sum_{j=1}^{n} K_I(\mathbf{x} - \mathbf{x}_j)$, where $\mathbf{x}_j \in \{\mathbf{x} = [m, m, \sin(m)]^T \mid m \in \mathcal{G}(40, -\pi, \pi)\}$. Figure 11 shows $p_{\text{sine\_3d}}(\mathbf{x})$ and its TCF result on the right. Pixels on the inner side of the curves have higher curvature than those on the outer side, and pixels near the function's maximum and minimum exhibit higher curvature values than those farther away.

3.2.2 Medical images

This section examines examples from the Computed Tomography Angiography[49] and the Coronary Artery Segmentations[50] datasets. We employed Gaussian kernels as the interpolation method, as defined in Equation (14). In three dimensions, $\tilde{\mathbf{x}}_j \in \mathbb{R}^3_+$ and $K_{S_j}(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ is the interpolation kernel and we used a scale matrix $\mathbf{S}_j = \mathbf{I}_3$. For the Computed Tomography Angiography dataset,[49] $k = 20$, and for the Coronary Artery Segmentations dataset,[50] $k = 40$. Curvature results were visualized with contrast enhancement using histogram equalization.[51]

Computed tomography angiography

We calculated the curvature of a segmented coronary artery image from a 3-D computed tomography angiography scan, downsampling the image by a factor of two. Figure 12 shows the interpolated image on the first row, and the curvature results with TCF on the second row. The second, third, and fourth columns display the views in the XY, XZ, and YZ planes, respectively. Viridis colormap is used for curvature results. Similar to Section 3.1.2, high curvature values at the branching points and twisted vessels are observed in this example, as expected.

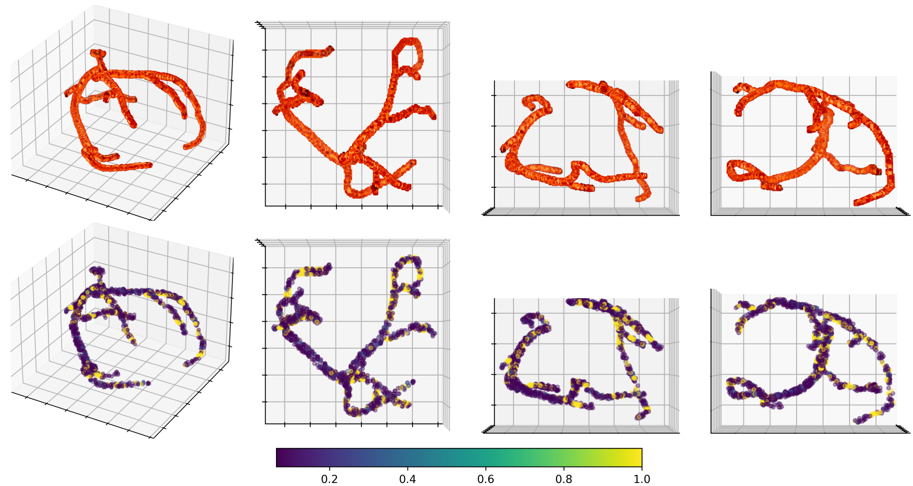


Fig. 12. (First row) The coronary angiography with views from XY, XZ and YZ planes, respectively. *(Second row)* Their TCF results with contrast-enhanced using histogram equalization.[51] *(Bottom)* Viridis colormap applied to curvature plots.

Coronary artery

The Coronary Artery Segmentations dataset[50] includes both segmented coronary arteries and their reference centerlines. We selected a sample coronary vessel segment from the dataset, downsampled it by a factor of two, and then applied interpolation to calculate the TCF.

Figure 13 shows the interpolated vessel segment on the first row, the TCF results on the second row, and the centerline curvatures computed using the implicit planar curvature method[19] on the third row. The second, third, and fourth columns show views from the XY, XZ, and YZ planes. Curvature is visualized using the Viridis colormap. This dataset includes arteries with varying vessel widths, which centerlines alone cannot capture. For example, in Figure 13, high curvature values are expected in the third row, last column view from the centerline results. However, when we examine the image itself on the first row, this region appears to be an artery segment with a large width and no sharp curvature changes. TCF results show that this region has low curvature values, which the centerline cannot detect. Similar to Section 3.2.2, high curvature values are detected at the twisted vessel (in the bottom right region of the second column in Figure 13), as expected.

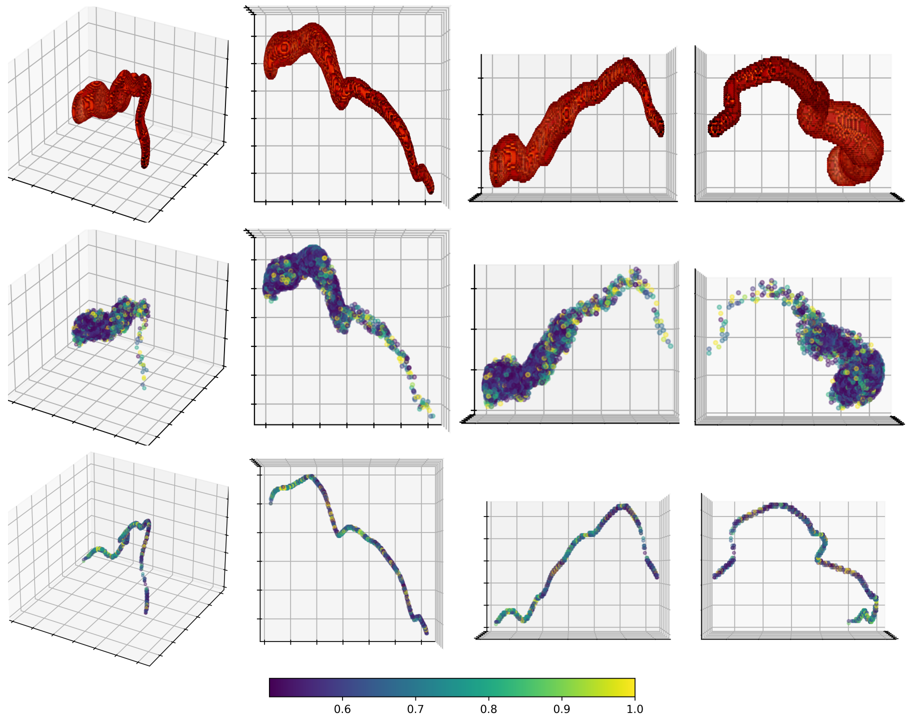


Fig. 13. (First row) The coronary artery segment with views from XY, XZ and YZ planes, respectively. *(Second row)* Their TCF results. *(Third row)* The centerline curvatures using the implicit planar curvature method.[19] Curvature contrast is enhanced with histogram equalization[51] with the Viridis colormap applied to curvature plots.

3.2.3 Additional analyses

We analyze TCF on images with isotropic and anisotropic voxel sizes. The ground-truth curvature is calculated as $1/r$, where r is the distance of the points from the ring's center. We compute the mean absolute difference between the ground-truth results and TCF.

Voxel anisotropy

We use $p_{\text{blurry\_ringring\_3d}}(\mathbf{x})$ to show how voxel anisotropy affects TCF results. For the isotropic case, the voxel sizes are $0.6 \times 0.6 \times 0.6$. For the anisotropic case, they are $0.6 \times 0.8 \times 0.4$. Figure 14 shows the image of $p_{\text{blurry\_ringring\_3d}}(\mathbf{x})$ for isotropic and anisotropic voxels on the left, with their TCF results on the right. Both cases perform similarly, with mean absolute differences of 2.493×10^{-3} and 2.495×10^{-3}, respectively. The ground truth curvature values range between 0.37 and 0.59 at the points in the image.

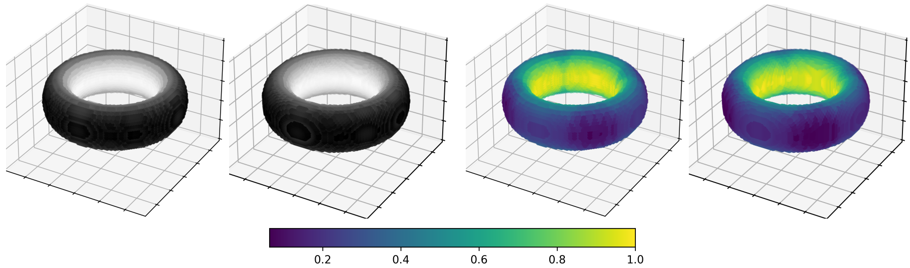


*Fig. 14. (Left)*The 3-D images of the blurry-ring function with isotropic and anisotropic voxels, respectively. *(Right)* Their TCF results. The Viridis colormap at the bottom is used for curvature plots. The mean absolute difference between the ground truth and TCF is 2.493×10^{-3} for the isotropic and 2.495×10^{-3} for the anisotropic case.

4. Conclusions and future perspectives

This paper defines local tubular curvature and presents a methodology, TCF, to calculate local curvature of tubular objects using interpolated intensity images. TCF implicitly calculates the tubular curvature locally for a bundle of curves that extend along the tube. TCF does not require the extraction of centerlines or curves. To our knowledge, TCF is the first of its kind, in the sense that it focuses on local curvature calculations for curves that traverse space along the tubular object, and achieves implicit calculations by determining the acceleration vectors for these curves using the eigenvectors of Hessian matrices of the intensity function. TCF exhibits sensitivity to varying curvature levels in a tube along the cross-section. The results on artificial and medical images demonstrate that this method can highlight tortuous regions in vessels and detect curvature differences between points on the inner and outer sides of curved structures, unlike centerline-based methods.

TCF is particularly useful in medical fields that require vasculature curvature analysis from images. Biological vascular structures often have non-uniform diameters and may exhibit various conformations in both normal physiology and disease states. Measuring the curvature of anatomic structures is a common task for clinicians and is increasingly performed by automated techniques. In diseases such as ROP, a leading cause of childhood blindness, the dilation and tortuosity of the retinal blood vessels are key components of disease classification: the more dilated and tortuous the vessels, the more severe the ROP. This has led the quantification of "vascular severity" to become a key biomarker target for image analysis techniques. One application of this technique would be to apply TCF to better understand the relationship between local retinal vessel tortuosity and disease progression. TCF has an advantage over deep learning techniques in that it is interpretable and can be applied both locally and globally within medical images and across image types, whereas many convolutional neural networks may face generalizability challenges.

Appendix

A. Derivation of Definition 1

Given a point $\mathbf{x} \in \mathbb{R}^2$, let $f(\mathbf{x})$ be the logarithm of the interpolation function $p(\mathbf{x})$. Note that $\mathbf{x}$ is a point on the curve s, *i.e.*, $\mathbf{x}(s)$. As explained in Section 2.1, we drop the (s) notation for simplicity. Let us define $p(\mathbf{x})$ and $f(\mathbf{x})$ as:

$$p(\mathbf{x}) = \sum_{j=1}^{n} w_j K_{\mathbf{S}_j}(\mathbf{x} - \mathbf{x}_j) \text{ and } f(\mathbf{x}) = \log(p(\mathbf{x})). \tag{15}$$

$K_{S_j}(\cdot) : \mathbb{R}^2 \to \mathbb{R}$ is the interpolation kernel with scale matrix $\mathbf{S}_j$, $\mathbf{x}$ and $w_j \in \mathbb{R}_+$, $j \in \{1, \ldots, n\}$, are the location and intensity of a point indexed by $j \in \{1, \ldots, n\}$ with n pixels.

Consider ∇ to be a column derivative operator. Let $\mathbf{g}(\mathbf{x})$ be the gradient of $f(\mathbf{x})$, $\mathbf{g}_p(\mathbf{x})$ be the gradient of $p(\mathbf{x})$, $\mathbf{H}(\mathbf{x})$ be the Hessian of $f(\mathbf{x})$, and $\mathbf{H}_p(\mathbf{x})$ be the Hessian of $p(\mathbf{x})$. Dropping $(\mathbf{x})$ for brevity, $\mathbf{g} = \nabla f = \nabla p/p = \mathbf{g}_p/p$ and:

$$\mathbf{H} = p^{-1}\nabla\nabla^T p - p^{-2}\nabla p \nabla^T p = p^{-1}\mathbf{H}_p - \mathbf{g}\mathbf{g}^T. \tag{16}$$

If $\mathbf{H}_p$ has eigenvectors that are equal or close to each other for a point under a ridge with a constant value, then the second gradient term in Equation (16) aligns one of the eigenvectors of $\mathbf{H}$ to be orthogonal to the ridge and the other to be parallel. We will use $\partial\mathbf{H}/\partial x_i$ for (18) where x_i is the i-th dimension of $\mathbf{x}$.

$$\begin{aligned}
\frac{\partial \mathbf{H}}{\partial x_i} =& \frac{\frac{\partial \nabla\nabla^T p}{\partial x_i} p - \nabla\nabla^T p \frac{\partial p}{\partial x_i}}{p^2} \\
&- \frac{(\frac{\partial \nabla p}{\partial x_i}\nabla^T p + \nabla p \frac{\partial \nabla^T p}{\partial x_i})p^2}{p^4} + \frac{2p\nabla\nabla^T p \frac{\partial p}{\partial x_i}}{p^4} \\
=& \frac{1}{p}\frac{\partial \nabla\nabla^T p}{\partial x_i} - \frac{1}{p^2}(\nabla\nabla^T p \frac{\partial p}{\partial x_i} + \frac{\partial \nabla p}{\partial x_i}\nabla^T p \\
&+ \nabla p \frac{\partial \nabla^T p}{\partial x_i}) + \frac{2}{p^3}\nabla p \nabla^T p \frac{\partial p}{\partial x_i}.
\end{aligned} \tag{17}$$

Following the steps from Equations (2)-(11), Hessian and its derivatives are obtained to calculate the acceleration vector at $\mathbf{x}$, where $\{(\lambda_1(\mathbf{x}), \mathbf{q}_1(\mathbf{x})), (\lambda_2(\mathbf{x}), \mathbf{q}_2(\mathbf{x})))\}$ are the eigenvalue-eigenvector pairs of $\mathbf{H}(\mathbf{x})$, and $\lambda_1(\mathbf{x}) > \lambda_2(\mathbf{x})$. We define $\mathbf{q}_i$ as $\mathbf{q}_i = \begin{bmatrix} \mathsf{q}_{i1} & \mathsf{q}_{i2} \end{bmatrix}^T$ for 2-D. $\mathbf{H}(\mathbf{x})$ is decomposed into its eigenvectors in Equation (2), where $\mathbf{Q}$ is the 2×2 matrix whose ith column is the eigenvector $\mathbf{q}_i$, and $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. To obtain $\partial\mathbf{q}_1(\mathbf{x})/\partial\mathbf{x}$ differentiating the Hessian eigendecomposition, and introducing $\mathbf{T}_i(\mathbf{x})$,

$\mathbf{V}_i(\mathbf{x})$ and $\mathbf{L}_i(\mathbf{x})$ as in Equations (3)-(5), $\mathbf{T}_i$ for $i=1$ is derived as:

$$\mathbf{T}_1 = \sum_{j=1}^{2} \lambda_j \frac{\partial \mathbf{q}_j}{\partial x_1}\mathbf{q}_j^T + l_{1j}\mathbf{q}_j\mathbf{q}_j^T + \lambda_j \mathbf{q}_j \frac{\partial \mathbf{q}_j}{\partial x_1}^T. \tag{18}$$

Linear equations for $i=1$ from Equation (10) are derived in Equations (19)-(20). From Equation (18), we have three equations which are defined for $i=1$:

$$\begin{aligned} \mathsf{t}_{111} &= \lambda_1 \mathsf{v}_{111}\mathsf{q}_{11} + \lambda_2 \mathsf{v}_{112}\mathsf{q}_{21} + l_{11}\mathsf{q}_{11}\mathsf{q}_{11} \\ &\quad + l_{12}\mathsf{q}_{21}\mathsf{q}_{21} + \lambda_1 \mathsf{q}_{11}\mathsf{v}_{111} + \lambda_2 \mathsf{q}_{21}\mathsf{v}_{112}, \\ \mathsf{t}_{112} &= \lambda_1 \mathsf{v}_{111}\mathsf{q}_{12} + \lambda_2 \mathsf{v}_{112}\mathsf{q}_{22} + l_{11}\mathsf{q}_{11}\mathsf{q}_{12} \\ &\quad + l_{12}\mathsf{q}_{21}\mathsf{q}_{22} + \lambda_1 \mathsf{q}_{11}\mathsf{v}_{121} + \lambda_2 \mathsf{q}_{21}\mathsf{v}_{122}, \\ \mathsf{t}_{122} &= \lambda_1 \mathsf{v}_{121}\mathsf{q}_{12} + \lambda_2 \mathsf{v}_{122}\mathsf{q}_{22} + l_{11}\mathsf{q}_{12}\mathsf{q}_{12} \\ &\quad + l_{12}\mathsf{q}_{22}\mathsf{q}_{22} + \lambda_1 \mathsf{q}_{12}\mathsf{v}_{121} + \lambda_2 \mathsf{q}_{22}\mathsf{v}_{122}. \end{aligned} \tag{19}$$

Two properties of eigenvectors are: they are unit vectors and perpendicular to each other. Using these properties, we derived three linear equations in Equation (20).

$$\begin{aligned} 0 &= \mathsf{q}_{11}\mathsf{v}_{111} + \mathsf{q}_{12}\mathsf{v}_{121}, \\ 0 &= \mathsf{q}_{21}\mathsf{v}_{112} + \mathsf{q}_{22}\mathsf{v}_{122}, \\ 0 &= \mathsf{q}_{21}\mathsf{v}_{111} + \mathsf{q}_{22}\mathsf{v}_{121} + \mathsf{q}_{11}\mathsf{v}_{112} + \mathsf{q}_{12}\mathsf{v}_{122}. \end{aligned} \tag{20}$$

We have six equations and 12 unknowns. Writing the set of linear equations, we obtain Equations (10)-(11). The elements of $\mathsf{V}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$ are found via a linear solver. $\partial \mathbf{q}_1(\mathbf{x})/\partial \mathbf{x}$ are calculated from $\mathsf{V}(\mathbf{x})$ to obtain the acceleration vector at $\mathbf{x}$.

B. Generalization to functions of N-dimensional input

Similar to the 2-D case, the curvature in N-dimensional space can be obtained from the acceleration of the first eigen vector of the $N \times N$ Hessian matrix. Let $\mathbf{x} \in \mathbb{R}^N$ and $f(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$ as before with w_j, $\mathbf{x}_j$ indicating image intensity and location for hypervoxel $j \in \{1, \ldots, n\}$. Interpolate using a suitable kernel $K_{S_j}(\cdot) : \mathbb{R}^N \to \mathbb{R}$ to obtain $f(\mathbf{x}) = ln(p(\mathbf{x}))$ as in Equation (15). At $\mathbf{x}$, $\mathbf{q}_1$ of the Hessian is the unit velocity vector of the curve s, which is the integral curve to the principle curve of the function. With this as the unit velocity vector, acceleration is obtained as its directional derivative yielding the local curvature value. Following **Definition 1**, direction derivatives $\mathbf{q}_1$, i.e., $\left[\partial \mathbf{q}_1/\partial x_1 \cdots \partial \mathbf{q}_1/\partial x_N\right]$ must be determined. Following Equations (2)-(9), Hessian and its derivatives are identified for the N-dimensional case.

B.1 Three-dimensional input

For the 3-D case, following the steps in Equations (2)-(11), Hessian and its derivatives are identified to determine the acceleration vector at $\mathbf{x} \in \mathbb{R}^3$, where

$\{(\lambda_1(\mathbf{x}), \mathbf{q}_1(\mathbf{x})), (\lambda_2(\mathbf{x}), \mathbf{q}_2(\mathbf{x})), (\lambda_3(\mathbf{x}), \mathbf{q}_3(\mathbf{x}))\}$ are the eigenpairs of $\mathbf{H}(\mathbf{x})$, and $\lambda_1(\mathbf{x}) > \lambda_2(\mathbf{x}) > \lambda_3(\mathbf{x})$. Define $\mathbf{q}_i$ as $\mathbf{q}_i = \begin{bmatrix} \mathsf{q}_{i1} & \mathsf{q}_{i2} & \mathsf{q}_{i3} \end{bmatrix}^T$. Decompose $\mathbf{H}(\mathbf{x})$ as in (2), where $\mathbf{Q}$ is now a 3×3 matrix whose ith column is the eigenvector $\mathbf{q}_i$, and $\mathbf{\Lambda}$ is the diagonal matrix that contains the corresponding eigenvalues:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \end{bmatrix}^T, \tag{21}$$

Calculate $\partial \mathbf{q}_1(\mathbf{x})/\partial \mathbf{x}$ by differentiating the eigendecomposition of $\mathbf{H}(\mathbf{x})$ and define $\mathbf{T}_i(\mathbf{x})$, $\mathsf{V}_i(\mathbf{x})$ and $\mathbf{L}_i(\mathbf{x})$ as:

$$\begin{aligned} \mathbf{T}_i(\mathbf{x}) &= \frac{\partial \mathbf{H}(\mathbf{x})}{\partial x_i} = \begin{bmatrix} \mathsf{t}_{i11} & \mathsf{t}_{i12} & \mathsf{t}_{i13} \\ \mathsf{t}_{i21} & \mathsf{t}_{i22} & \mathsf{t}_{i23} \\ \mathsf{t}_{i31} & \mathsf{t}_{i32} & \mathsf{t}_{i33} \end{bmatrix}, \\ \mathsf{V}_i(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \mathbf{q}_1}{\partial x_i}, \frac{\partial \mathbf{q}_2}{\partial x_i}, \frac{\partial \mathbf{q}_3}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \mathsf{v}_{i11} & \mathsf{v}_{i12} & \mathsf{v}_{i13} \\ \mathsf{v}_{i21} & \mathsf{v}_{i22} & \mathsf{v}_{i23} \\ \mathsf{v}_{i31} & \mathsf{v}_{i32} & \mathsf{v}_{i33} \end{bmatrix}, \\ \mathbf{L}_i(\mathbf{x}) &= \begin{bmatrix} \frac{\partial \lambda_1}{\partial x_i} & 0 & 0 \\ 0 & \frac{\partial \lambda_2}{\partial x_i} & 0 \\ 0 & 0 & \frac{\partial \lambda_3}{\partial x_i} \end{bmatrix} = \begin{bmatrix} l_{\mathsf{i}1} & 0 & 0 \\ 0 & l_{\mathsf{i}2} & 0 \\ 0 & 0 & l_{\mathsf{i}3} \end{bmatrix}. \end{aligned} \tag{22}$$

$\mathbf{T}_i$ is symmetric due to thrice-differentiable interpolation, and calculated as in Equation (18) with $j = 1, 2, 3$. As in Appendix A, one can derive linear equations to obtain Equation (23). By utilizing two properties of the eigenvectors are explained in Appendix A; M is defined in Equation (24). The elements of $\mathsf{V}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$ are determined by solving the linear equations to obtain $\partial \mathbf{q}_1(\mathbf{x})/\partial \mathbf{x}$ from $\mathsf{V}(\mathbf{x})$. The acceleration at $\mathbf{x}$ is then computed using Equation (1).

$$\begin{bmatrix} \mathsf{t}_{111} \\ \mathsf{t}_{112} \\ \mathsf{t}_{113} \\ \mathsf{t}_{122} \\ \mathsf{t}_{123} \\ \mathsf{t}_{133} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathsf{M} \begin{bmatrix} \mathsf{v}_{111} \\ \mathsf{v}_{112} \\ \mathsf{v}_{113} \\ l_{11} \\ l_{12} \\ l_{13} \\ \mathsf{v}_{121} \\ \mathsf{v}_{122} \\ \mathsf{v}_{123} \\ \mathsf{v}_{131} \\ \mathsf{v}_{132} \\ \mathsf{v}_{133} \end{bmatrix}, \begin{bmatrix} \mathsf{t}_{211} \\ \mathsf{t}_{212} \\ \mathsf{t}_{213} \\ \mathsf{t}_{222} \\ \mathsf{t}_{223} \\ \mathsf{t}_{233} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathsf{M} \begin{bmatrix} \mathsf{v}_{211} \\ \mathsf{v}_{212} \\ \mathsf{v}_{213} \\ l_{21} \\ l_{22} \\ l_{23} \\ \mathsf{v}_{221} \\ \mathsf{v}_{222} \\ \mathsf{v}_{223} \\ \mathsf{v}_{231} \\ \mathsf{v}_{232} \\ \mathsf{v}_{233} \end{bmatrix}, \begin{bmatrix} \mathsf{t}_{311} \\ \mathsf{t}_{312} \\ \mathsf{t}_{313} \\ \mathsf{t}_{322} \\ \mathsf{t}_{323} \\ \mathsf{t}_{333} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathsf{M} \begin{bmatrix} \mathsf{v}_{311} \\ \mathsf{v}_{312} \\ \mathsf{v}_{313} \\ l_{31} \\ l_{32} \\ l_{33} \\ \mathsf{v}_{321} \\ \mathsf{v}_{322} \\ \mathsf{v}_{323} \\ \mathsf{v}_{331} \\ \mathsf{v}_{332} \\ \mathsf{v}_{333} \end{bmatrix}, \tag{23}$$

$$\mathbf{M} = \begin{bmatrix} 2\lambda_1 q_{11} & 2\lambda_2 q_{21} & 2\lambda_3 q_{31} & q_{11}^2 & q_{21}^2 & q_{31}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_1 q_{12} & \lambda_2 q_{22} & \lambda_3 q_{32} & q_{11}q_{12} & q_{21}q_{22} & q_{31}q_{32} & \lambda_1 q_{11} & \lambda_2 q_{21} & \lambda_3 q_{31} & 0 & 0 & 0 \\ \lambda_1 q_{13} & \lambda_2 q_{23} & \lambda_3 q_{33} & q_{11}q_{13} & q_{21}q_{23} & q_{31}q_{33} & 0 & 0 & 0 & \lambda_1 q_{11} & \lambda_2 q_{21} & \lambda_3 q_{31} \\ 0 & 0 & 0 & q_{12}^2 & q_{22}^2 & q_{32}^2 & 2\lambda_1 q_{12} & 2\lambda_2 q_{22} & 2\lambda_3 q_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{12}q_{13} & q_{22}q_{23} & q_{32}q_{33} & \lambda_1 q_{13} & \lambda_2 q_{23} & \lambda_3 q_{33} & \lambda_1 q_{12} & \lambda_2 q_{22} & \lambda_3 q_{32} \\ 0 & 0 & 0 & q_{13}^2 & q_{23}^2 & q_{33}^2 & 0 & 0 & 0 & 2\lambda_1 q_{13} & 2\lambda_2 q_{23} & 2\lambda_3 q_{33} \\ q_{11} & 0 & 0 & 0 & 0 & 0 & q_{12} & 0 & 0 & q_{13} & 0 & 0 \\ 0 & q_{21} & 0 & 0 & 0 & 0 & 0 & q_{22} & 0 & 0 & q_{23} & 0 \\ 0 & 0 & q_{31} & 0 & 0 & 0 & 0 & 0 & q_{32} & 0 & 0 & q_{33} \\ q_{21} & q_{11} & 0 & 0 & 0 & 0 & q_{22} & q_{12} & 0 & q_{23} & q_{13} & 0 \\ q_{31} & 0 & q_{11} & 0 & 0 & 0 & q_{32} & 0 & q_{12} & q_{33} & 0 & q_{13} \\ 0 & q_{31} & q_{21} & 0 & 0 & 0 & 0 & q_{32} & q_{22} & 0 & q_{33} & q_{23} \end{bmatrix}. \quad (24)$$

Declarations

Ethics approval and informed consent

None to declare, as this study did not involve human subjects.

Competing interests

Giovanna Guidoboni serves as Chief and Managing Editor of AIVO. She was not involved in an editorial capacity at any point of the editorial process prior to acceptance. AIVO conducts single-blind peer review. Veysi Yildiz was affiliated with Northeastern University during the conduct of this research and is currently employed by Reinforce Labs. The remaining authors declare no competing interests.

Funding

This work was supported by grants R01 EY019474, R01 EY034718, and P30 EY10572 from the National Institutes of Health, by unrestricted departmental funding from Research to Prevent Blindness, and by the Malcolm Marquis Innovation Fund and National Science Foundation grant DMS 2108711/2327640.

Author contributions

Elifnur Sunger and Beyza Kalkanli share first coauthorship. Elifnur Sunger, Beyza Kalkanli, and Veysi Yildiz led the derivation and coding of the methods, and carried out the analyses and interpretation of the results in 2-D and 3-D experiments. Tales Imbiriba and Deniz Erdogmus drove the derivations of the method and offered insightful guidance in the theoretical and practical analyses of factors that impact its behavior. Peter Campbell and Giovanna Guidoboni provided valuable guidance, particularly in addressing the needs within the medical domain and suggesting practical applications and interpretations.

Artificial intelligence

AI-assisted tools were used exclusively for grammar checking purposes. The authors take full responsibility for the integrity of the scientific content.

Acknowledgements

None to declare.

References

1. Deng H, Zhang W, Mortensen E, et al. Principal Curvature-Based Region Detector for Object Recognition. 2007 IEEE Conference on Computer Vision and Pattern Recognition. 2007; 1–8. doi: https://doi.org/10.1109/CVPR.2007.382972.
2. Thanh DN, Prasath VS, Hien NN, et al. Melanoma skin cancer detection method based on adaptive principal curvature, colour normalisation and feature extraction with the ABCD rule. J. Digit. Imaging, 2020;33(3): 574. doi: https://doi.org/10.1007/s10278-019-00316-x.
3. Mokhtarian F, Suomela R. Robust image corner detection through curvature scale space. IEEE Trans. Pattern Anal. Mach. Intell. 1998;20(12): 1376–1381. doi: https://doi.org/10.1109/34.735812.
4. Zhang W, Sun C, Breckon T, et al. Discrete curvature representations for noise robust image corner detection. IEEE Trans. Image Process. 2019;28(9): 4444–4459. doi: https://doi.org/10.1109/TIP.2019.2910655.
5. He X, Yung NHC. Corner detector based on global and local curvature properties. Opt. Eng. 2008;47(5): 057008. doi: https://doi.org/10.1117/1.2931681.
6. Tang Y, Li H, Sun X, et al. Principal curvature measures estimation and application to 3D face recognition. J. Math. Imaging Vis. Oct. 2017;59(2): 211–233. ISSN: 0924-9907. doi: https://doi.org/10.1007/s10851-017-0728-2.
7. Fischer P, Brox T. Image descriptors based on curvature histograms. German Conference on Pattern Recognition. Sept. 2014; 239–249. ISBN: 978-3-319-11751-5. doi: https://doi.org/10.1007/978-3-319-11752-2_19.
8. Qiuxiang Z, Yin K, Duan Y. Image reconstruction by minimizing curvatures on image surface. J. Math. Imaging Vis. Jan. 2021;63. doi: https://doi.org/10.1007/s10851-020-00992-3.
9. Zhong Q, Li Y, Yang Y, et al. Minimizing discrete total curvature for image processing. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020; 9474–9482. doi: https://doi.org/10.1109/CVPR42600.2020.00949.
10. Lee SH, Seo JK. Noise removal with Gauss curvature-driven diffusion. IEEE Trans. Image Process. 2005;14(7): 904–909. doi: https://doi.org/10.1109/TIP.2005.849294.
11. Kim S. PDE-based image restoration: A hybrid model and color image denoising. IEEE Trans. Image Process. 2006;15(5): 1163–1170. doi: https://doi.org/10.1109/TIP.2005.864184.
12. Nc SK, Radhika Y. Optimized maximum principal curvatures based segmentation of blood vessels from retinal images. Biomed. Res. (Tokyo), 2019;30, 308–318. doi: https://doi.org/10.35841/biomedicalresearch.30-19-068.
13. Lutton JE, Collier S, Bretschneider T. A curvature-enhanced random walker segmentation method for detailed capture of 3D cell surface membranes. IEEE Trans. Med. Imaging, 2021;40(2): 514–526. doi: https://doi.org/10.1109/TMI.2020.3031029.
14. Goldluecke B, Cremers D. Introducing total curvature for image processing. 2011 International Conference on Computer Vision. IEEE. 2011; 1267–1274. doi: https://doi.org/10.1109/ICCV.2011.6126378.
15. Chen D, Mirebeau JM, Cohen LD. A new finsler minimal path model with curvature penalization for image segmentation and closed contour detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; 355–363. doi: https://doi.org/10.1109/CVPR.2016.45.

16. El-Zehiry NY, Grady L. Contrast driven elastica for image segmentation. IEEE Trans. Image Process. 2016;25(6): 2508–2518. doi: https://doi.org/10.1109/TIP.2016.2545244.
17. Gilbert C, Foster A. Childhood blindness in the context of VISION 2020: the right to sight. Bull. World Health Organ. 2001;79(3): 227–232. Available from: https://pubmed.ncbi.nlm.nih.gov/11285667/.
18. Chiang MF, Quinn GE, Fielder AR, et al. International classification of retinopathy of prematurity. Ophthalmology, 2021;128(10): e51–e68. Available from: https://doi.org/10.1016/j.ophtha.2021.05.031.
19. Goldman R. Curvature formulas for implicit curves and surfaces. Comput. Aided Geom. Des. 2005;22(7): Geometric Modelling and Differential Geometry, 632–658. ISSN: 0167-8396. doi: https://doi.org/10.1016/j.cagd.2005.06.005.
20. Gong, Yuanhao and Tang, Wenming and Zhou, Lebin and Yu, Lantao and Qiu, Guoping. A discrete scheme for computing image's weighted gaussian curvature. 2021 IEEE International Conference on Image Processing (ICIP). 2021; 1919–1923. doi: https://doi.org/10.1109/ICIP42928.2021.9506611.
21. Taylor JE. II—mean curvature and weighted mean curvature. Acta Metall. Mater. 1992;40(7): 1475–1485. doi: https://doi.org/10.1016/0956-7151(92)90091-R.
22. Choi G, Cheng C, Wilson N, et al. Methods for quantifying three-dimensional deformation of arteries due to pulsatile and nonpulsatile forces: Implications for the design of stents and stent grafts. Ann. Biomed. Eng. Dec. 2008;37, 14–33. doi: https://doi.org/10.1007/s10439-008-9590-0.
23. Atiskova Y, Wildner J, Spitzer M, et al. Retinal vessel tortuosity as a prognostic marker for disease severity in Fabry disease. Orphanet J. Rare Dis. Dec. 2021;16. doi: https://doi.org/10.1186/s13023-021-02080-0.
24. Bullitt E, Gerig G, Pizer SM, et al. Measuring tortuosity of the intracerebral vasculature from MRA images. IEEE Trans. Med. Imaging, 2003;22(9): 1163–1171. doi: https://doi.org/10.1109/TMI.2003.816964.
25. Lorigo LM, Faugeras OD, Grimson WEL, et al. Curves: Curve evolution for vessel segmentation. Med. Image Anal. 2001;5(3): 195–206. doi: https://doi.org/10.1016/S1361-8415(01)00040-8.
26. Law MW, Chung AC. Efficient implementation for spherical flux computation and its application to vascular segmentation. IEEE Trans. Image Process. 2009;18(3): 596–612. doi: https://doi.org/10.1109/TIP.2008.2010073.
27. Han HC. Twisted blood vessels: Symptoms, etiology and biomechanical mechanisms. J. Vasc. Res. Mar. 2012;49, 185–97. doi: https://doi.org/10.1159/000335123.
28. Hathout L, Do HM. Vascular tortuosity: A mathematical modeling perspective. J. Physiol. Sci. 2012;62, 133–145. doi: https://doi.org/10.1007/s12576-011-0191-6.
29. Gerig G, Gouttard S, Corouge I. Analysis of brain white matter via fiber tract modeling. The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2. IEEE. 2004; 4421–4424. doi: https://doi.org/{10.1109/IEMBS.2004.1404229}.
30. Ge S, Tian H, Zhang W, et al. Automatic bony structure segmentation and curvature estimation on ultrasound cervical spine images-a feasibility study. Journal of Physics: Conference Series. 2822. (1): IOP Publishing. 2024; 012023. doi: https://doi.org/10.1088/1742-6596/2822/1/012023.
31. Zhou GQ, Zheng YP. Assessment of scoliosis using 3-D ultrasound volume projection imaging with automatic spine curvature detection. 2015 IEEE International Ultrasonics Symposium (IUS). IEEE. 2015; 1–4. doi: https://doi.org/10.1109/ULTSYM.2015.0485.
32. Vogel P, Kampf T, Guggenberger K, et al. Robust centerline prediction for accurate vessel wall visualization of intracranial vessels in multi-contrast 3D MRI data. arXiv preprint arXiv:2209.01422, 2022; doi: https://doi.org/10.48550/arXiv.2209.01422.
33. Aletti M, Gerbeau JF, Lombardi D. A simplified fluid–structure model for arterial flow. Application to retinal hemodynamics. Comput. Methods Appl. Mech. Eng. 2016;306, 77–94. ISSN: 0045-7825. doi: https://doi.org/10.1016/j.cma.2016.03.044.
34. Koogler B, Young B, Campbell JP, et al. Analysis of vessel tortuosity and its impact on hemodynamics in retinopathy of prematurity. Investig. Ophthalmol. Vis. Sci. 2023;64(8): 1237–1237. Available from: https://iovs.arvojournals.org/article.aspx?articleid=2790163.

35. Javaheri B, Razi H, Gohin S, et al. Lasting organ-level bone mechanoadaptation is unrelated to local strain. Sci. Adv. 2020;6(10): eaax8301. doi: https://doi.org/10.1126/sciadv.aax8301.
36. Ernst P, Hille G, Hansen C, et al. A CNN-based framework for statistical assessment of spinal shape and curvature in whole-body MRI images of large populations. Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22. Springer. 2019; 3–11. doi: https://doi.org/10.1007/978-3-030-32251-9_1.
37. Liu T, Yang Y, Wang Y, et al. Spinal curve assessment of idiopathic scoliosis with a small dataset via a multi-scale keypoint estimation approach. Adjunct proceedings of the 2020 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2020 aCM international symposium on wearable computers. 2020; 665–670. doi: https://doi.org/10.1145/3410530.3414317.
38. Wang H, Song Q, Yin R, et al. B-spine: Learning B-spline curve representation for robust and interpretable spinal curvature estimation. Proceedings of the AAAI Conference on Artificial Intelligence. 38. (6): 2024; 5381–5389. doi: https://doi.org/10.1609/aaai.v38i6.28346.
39. Santamarina A, Weydahl E, Siegel JM, et al. Computational analysis of flow in a curved tube model of the coronary arteries: effects of time-varying curvature. Ann. Biomed. Eng. 1998;26, 944–954. doi: https://doi.org/10.1114/1.113.
40. Ozertem U, Erdogmus D. Locally Defined Principal Curves and Surfaces. J. Mach. Learn. Res. 2011;12(34): 1249–1286. Available from: http://jmlr.org/papers/v12/ozertem11a.html.
41. Montiel S, Ros A. Curves and surfaces. 69. American Mathematical Soc., 2009; Available from: https://bookstore.ams.org/gsm-69-r/.
42. Ozertem U, Erdogmus D. Nonparametric snakes. IEEE Trans. Image Process. 2007;16(9): 2361–2368. doi: https://doi.org/10.1109/TIP.2007.902335.
43. Law MW, Chung AC. Three dimensional curvilinear structure detection using optimally oriented flux. ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10. Springer. 2008; 368–382. doi: https://doi.org/10.1007/978-3-540-88693-8_27.
44. Law MW, Chung AC. An oriented flux symmetry based active contour model for three dimensional vessel segmentation. ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11. Springer. 2010; 720–734. doi: https://doi.org/10.1007/978-3-642-15558-1_52.
45. Law MW, Garvin GJ, Tummala S, et al. Gradient competition anisotropy for centerline extraction and segmentation of spinal cords. Information Processing in Medical Imaging: 23rd International Conference, IPMI 2013, Asilomar, CA, USA, June 28–July 3, 2013. Proceedings 23. Springer. 2013; 49–61. doi: https://doi.org/10.1007/978-3-642-38868-2_5.
46. Brown JM, Campbell JP, Beers A, et al. Automated diagnosis of plus disease in retinopathy of prematurity using deep convolutional neural networks. JAMA Ophthalmol. 2018;136(7): 803–810. doi: https://doi.org/10.1001/jamaophthalmol.2018.1934.
47. Cervantes-Sanchez F, Cruz-Aceves I, Hernandez-Aguirre A, et al. Automatic segmentation of coronary arteries in X-ray angiograms using multiscale analysis and artificial neural networks. Appl. Sci. 2019;9(24): ISSN: 2076-3417. doi: https://doi.org/10.3390/app9245507.
48. Van der Walt S, Schönberger JL, Nunez-Iglesias J, et al. scikit-image: image processing in Python. PeerJ, 2014;2, e453. doi: https://doi.org/10.7717/peerj.453.
49. Zeng A, Wu C, Xie W, et al. ImageCAS: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images. Comput. Med. Imaging Graph. 2023; 102287. ISSN: 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2023.102287.
50. Dalvit Carvalho da Silva R, Soltanzadeh R, Figley CR. Automated Coronary Artery Tracking with a Voronoi-Based 3D Centerline Extraction Algorithm. J. Imaging, 2023;9(12): 268. doi: https://doi.org/10.3390/jimaging9120268.
51. Jain AK. Fundamentals of digital image processing. Prentice-Hall, Inc., 1989; Available from: https://dl.acm.org/doi/10.5555/59921.