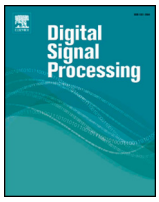
Digital Signal Processing 128 (2022) 103623

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp



A novel family of edge preserving anisotropic filters

Haydar Kilic [a,*], Salim Ceyhan [a], Omer Nezih Gerek [b]

[a] Computer Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey
[b] Electrical and Electronics Engineering, Eskisehir Technical University, Eskisehir, Turkey

ARTICLE INFO

Article history:
Available online 17 June 2022

Keywords:
Noise reduction
Edge preservation
Polyakov action
Finsler
Beltrami flow
Anisotropic filter

ABSTRACT

In this article, new anisotropic image filters are introduced and their performances are compared with existing isotropic and anisotropic filters. The developed filters were examined according to their image noise removing performances in terms of standard image quality metrics, as well as the edge preserving properties of the filtered images. Mathematical inferences of anisotropic filters are made based on the minimization of the Polyakov action energy integral. New anisotropic metrics are found by means of Finsler metrics that minimize the corresponding integral. The new metrics perform filtering by updating the image with anisotropic Laplace-Beltrami flow. After filtering, it was observed that the introduced metrics perform well against other anisotropic metrics. It is also observed that the developed New Randers, New Normalized Miron, and New Metric filters preserve edges better than other filters, making it a plausible noise removal tool prior to edge detection in image processing. The source codes of proposed filters are publicly available at https://github.com/HAYDARKILIC.

Noise reduction is an important pre-processing step in image processing applications, yet utilization of partial differential equations (PDEs) is a rarely utilized technique for this purpose. To achieve a related PDE, a Beltrami frame is created by embedding a digital image with a map into a 3-dimensional manifold as a 2-dimensional manifold. Next, the Polyakov action is defined, which allows measuring between an embedded manifold (image surface) and the embedding space in the Beltrami framework. The defined Polyakov action is a functional that depends on the metrics of the image manifold and embedding space. This functional brings together scale space methods in image processing and allows smoothing and denoising of images [1].

The Polyakov action yields two main results from the minimization of the image manifold according to the Riemann metric and the embedding map. First, the image manifold metric is induced. The second output is the Beltrami flow, which gives the evolution of the image manifold according to a time scale parameter. The Beltrami flow, created by taking the image manifold metric, i.e. the induced metric, is a minimal area flow. The Beltrami flow can be isotropic (point-based) or anisotropic (point and direction-based) depending on the image manifold metric selection [2].

Since the image surfaces are uneven structures, measurements made by choosing non-Euclidean metrics tend to provide more accurate results. Non-Euclidean metrics can be considered in two classes; Riemannian or Finslerian. Measurements made on the image surface by choosing a Riemannian metric can be made by considering only the point on the surface, while measurements made using the Finslerian metric can be made by considering point and direction. Special cases related to these metrics have been studied in the literature under the categorization of isotropic and anisotropic cases [2], [3].

Partial differential equation-based anisotropic filtering techniques are widely used in computer vision for image enhancement and noise reduction. Anisotropic filters have become popular in many research fields because they overcome the undesirable effects of other well-known filters and improve the quality of images. In the classical filtering approach, pixel values are replaced by linear combinations of pixels around a predefined pixel neighborhood. In many applications, this class of methods, such as median and convolutional (Gaussian, uniform blur, etc.) filtering are more widely used because of their simple and standardized implementations. The major disadvantage of these filters is that, except for median filtering under heavily tailed additive noise distributions, they tend to destroy important details and blur prominent geometric edges. The reason for this tendency is their application invariance to the class of the pixel; they perform the filtering regardless of whether the pixel is noisy, or not. Consequently, a filtered and noise-free location may spoil important characteristics of the image, such as edges. In such cases, anisotropic filtering could be the remedy. Anisotropic filters reduce noise without edge blur, thus preserving edges. The details regarding edge preserving filtering through anisotropic filtering can be found in [4] and [5]. First

* Corresponding author.
E-mail address: haydar.kilic@bilecik.edu.tr (H. Kilic).

https://doi.org/10.1016/j.dsp.2022.103623
1051-2004/

mentioning of anisotropic flows in image processing dates back to 1987 and its first use of noise reduction with the aim ofedge and detail preservation was proposed by Perona and Malik [6]. Later, many researchers worked on this method and certain improvements were made for particularly altering attention to higher or lower gradient edges [7], [8]. A detailed comparison of these methods can be reached from [9]. As an example to specific applications of noise reduction in images, certain anisotropic filters were applied to medical ultrasound image processing in [10], [11], [12]. Another case of astrophysical image processing can be found in [13] and a study on satellite images can be found in [14]. In another recent work, artificial neural networks were used together with anisotropic filtering [15].

The main purpose of this work is to preserve important geometric edges and details during noise removal in the image, regardless of the gradient strength. The results are in accordance with a recent work that proves that anisotropic filtering preserves edges and details better in an image [16]. In this study, new variations of metrics are proposed, and non-Euclidean and newly defined metrics are discussed. The responses of these new metrics to image de-noising applications are examined. In addition, the contribution of isotropic and anisotropic Beltrami flows to noise reduction are examined by comparing them with various image quality measures. In the next section, creation of the Beltrami framework based on the Polyakov Action is examined. In addition, a framework is deduced for a Riemannian manifold and the Beltrami flow and its operation algorithm is given. In Sec 3, the details and algorithms of the anisotropic metrics used in this study are presented. Next, image quality metrics that are used to evaluate the image enhancement application are provided. In the fifth and last section, the responses of the metrics examined in this study for noise reduction and edge preservation are discussed.

1. Beltrami framework and Polyakov action

In this section, the Beltrami framework will be explained and then isotropic and anisotropic Beltrami flows will be examined.

1.1. Beltrami framework

A 2-D digital image I can be modeled as a Σ surface by using the following X embedding:

$$X : \left(x^1, x^2\right) \rightarrow \left(x^1, x^2, I\left(x^1, x^2\right)\right). \tag{1}$$

A positively defined, symmetrical $g_{\mu\nu}(x^1, x^2)$ Riemannian metric at x^1, x^2 local coordinates is defined as the square of a distance on image manifold Σ as follows:

$$ds^2 = \left\langle g_\mu dx^\mu, g_\nu dx^\nu \right\rangle = g_{\mu\nu} dx^\mu dx^\nu. \tag{2}$$

Using the $X : \Sigma \rightarrow M \subset \mathbb{R}^3$ transform, the metrics $g_{\mu\nu}$ and h_{ij} for the Σ and M manifolds define a $\left(X, \left(\Sigma, g_{\mu\nu}\right), \left(M, h_{ij}\right)\right)$ Beltrami framework. Here, $\mu, \nu \in 1, 2$ and $i, j \in 1, 2, 3$. According to the Einstein summation rule, tensors with repeated indexes are summed over.

In this study, the embedding space $M \equiv \mathbb{R}^3$ and its metric $h_{ij} = diag(1, 1, \beta^2)$ are taken, where β is a positive parameter indicating the relative scale (aspect ratio) between spatial coordinates and image density. Since the $M \equiv \mathbb{R}^3$ a scalable structure using β, the image can be examined at desired scales [17]. In addition, the metric of embedded image manifold Σ, $g_{\mu\nu} = h_{ij} X^i_\mu X^j_\nu$ is chosen and in the following form [2]:

$$\left(g_{\mu\nu}\right) = \begin{pmatrix} 1 + \beta^2 I^2_{x^1} & \beta^2 I_{x^1} I_{x^2} \\ \beta^2 I_{x^1} I_{x^2} & 1 + \beta^2 I^2_{x^2} \end{pmatrix}, \tag{3}$$

where $I_{x^\mu} = \dfrac{\partial \mathrm{I}}{\partial \mathrm{x}^\mu}$ is used for the sake of brevity.

1.2. Riemannian (isotropic) Beltrami flow

A Polyakov action on the $\left(X, \left(\Sigma, g_{\mu\nu}\right), \left(M, h_{ij}\right)\right)$ Beltrami framework is defined as follows [18]:

$$S\left[X^i, g_{\mu\nu}, h_{ij}\right] = \int_\Sigma \sqrt{g} g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(X) dx^1 dx^2, \tag{4}$$

where the determinant g of the $g_{\mu\nu}$ is

$$g = \det\left(g_{\mu\nu}\right) = 1 + \beta^2 \left(I^2_{x^1} + I^2_{x^2}\right), \tag{5}$$

and inverse of the $g_{\mu\nu}$ is the following:

$$\left(g^{\mu\nu}\right) = \frac{1}{g} \begin{pmatrix} 1 + \beta^2 I^2_{x^2} & -\beta^2 I_{x^1} I_{x^2} \\ -\beta^2 I_{x^1} I_{x^2} & 1 + \beta^2 I^2_{x^1} \end{pmatrix}. \tag{6}$$

To minimize the Polyakov Action above, the metric $g_{\mu\nu}$ must be induced and in the following form [2]:

$$g_{\mu\nu} = \partial_\mu X^i \partial_\nu X^j h_{ij}(X). \tag{7}$$

Substituting this induced metric in the Polyakov action and taking into account that $g^{\mu\nu} g_{\mu\nu} = 2(\dim(\Sigma))$, a Polyakov action of the following form is obtained:

$$S = 2 \int_\Sigma \sqrt{g} dx^1 dx^2. \tag{8}$$

This integral measures the surface area of an uneven surface. Therefore the functional in Eq. (8) says that if an induced metric is chosen on the image surface, the resulting flow will evolve the surface towards a surface with minimal area.

The minimization of the Polyakov action according to the embedding in Eq. (1) is given by the following:

$$\frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu X^i\right) = 0. \tag{9}$$

This minimization is independent of the parametrization [2]. A solution of the family of $X = X(x_1, x_2, t)$ transformations with t parameters that satisfy this equation gives the Beltrami flow for $X(x_1, x_2, 0) = \Sigma$ initial condition [3]. Accordingly, the isotropic Beltrami flow that evolves the image surface until it is minimal is given as follows:

$$\frac{\partial X^i}{\partial t} = \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu X^i\right), X^i(0) = \Sigma. \tag{10}$$

Beltrami flow can be written as follows, with the Δ_g operator acting on X^i embedding:

$$\frac{\partial X^i}{\partial t} = \Delta_g X^i = \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu X^i\right). \tag{11}$$

The operator denoted by Δ_g is called the Beltrami operator or the Laplace-Beltrami operator. This operator is an extension of the Laplacian operator from Euclidean spaces to curvature manifolds [2].

Considering the X embedding in Eq. (1), since the variation of the x^1 and x^2 local coordinates with respect to the t parameter is constant, we can write:

$$\frac{\partial X^i}{\partial t} = \left(0, 0, \frac{\partial I\left(x^1, x^2, t\right)}{\partial t}\right). \tag{12}$$

Thus, the Beltrami flow becomes a flow that only affects the I feature of the image. If Eq. (12) is substituted in Eq. (11), the isotropic Beltrami flow through the image feature becomes:

$$\frac{\partial I}{\partial t} = \Delta_g I \equiv \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu I \right) \tag{13}$$

When the derivative on the right side of Eq. (13) is calculated explicitly, it is seen that $\Delta_g I = \frac{H^3}{\sqrt{g}}$, where H^3 is as follows [19]:

$$H^3 = \frac{1}{g^{3/2}} \left(g_{22} I_{x^1 x^1} - 2 g_{12} I_{x^1 x^2} + g_{11} I_{x^2 x^2} \right). \tag{14}$$

The Isotropic Beltrami flow given by Eq. (13) attains a significantly useful property for image processing applications: It makes a smoothing operation in the homogeneous regions of the image surface and evolves the image surface towards a surface with minimal area, enabling it to have *edge preservation property* [2]. The isotropic Beltrami flow algorithm is provided in Algorithm 1.1.

Algorithm 1.1: Isotropic Beltrami Flows.

1 Load the gray-level image $I(x^1, x^2)$
2 Change the image data type to the floating points data type
3 Enter aspect ratio β
4 Enter time increment dt
5 **for** *each iteration* **do**
6 Calculate the first and second order central difference of $I(x^1, x^2)$
7 At each point, calculate the components of the Riemannian metric tensor $g^{\mu\nu}$ and determinant g
8 Calculate the H^3 as following:
9
$$H^3 = \frac{1}{g^{\frac{3}{2}}} \left(\left(1 + \beta^2 I_{x^2 x^2} \right) I_{x^1 x^1} - 2\beta^2 I_{x^1} I_{x^2} I_{x^1 x^2} + \left(1 + \beta^2 I_{x^1 x^1} \right) I_{x^2 x^2} \right)$$
10 Calculate the Laplace-Beltrami operator $\Delta_g I = \frac{H^3}{\sqrt{g}}$
11 $I \leftarrow I + dt \cdot \Delta_g I$
12 **end**

1.3. Finslerian (anisotropic) Beltrami flow

In order to give the Polyakov action an anisotropic structure in the Beltrami framework, the Riemannian metric defined on the image surface is deformed by a Finsler extension $\varphi(x, v)$ as follows [20]:

$$\gamma_{\mu\nu}(x, v) = g_{\mu\nu}(x) + \varphi(x, v), \tag{15}$$

where $x = (x^1, x^2) \in \Sigma$ is a point on the image surface, $v = (v^1, v^2) \in T_x\Sigma$ is a vector of the tangent vector space of the surface at that point, and $\varphi(x, v)$ is a positive homogeneous tensor. The anisotropic metric $\gamma_{\mu\nu}$ generated on the image surface is a Randers type Finsler metric. According to this new Finsler metric, the form of the Polyakov action can be rewritten as follows, similar to the Riemannian case:

$$S\left[X^i, \gamma_{\mu\nu}, h_{ij} \right] = \int_\Sigma \sqrt{\gamma} \gamma^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(X) dx^1 dx^2, \tag{16}$$

where γ is the determinant of the Finsler metric and $\gamma_{\mu\nu}$ is its inverse metric tensor. Thus, the Polyakov action takes on the anisotropic form, by means of the anisotropic $\gamma_{\mu\nu}(x, v)$ metric. The Beltrami flow, which minimizes the anisotropic Polyakov action, can be written according to the Finsler metric $\gamma_{\mu\nu}$ similar to the Riemannian case (Eq. (13)), as follows [20]:

$$\frac{\partial I}{\partial t} = \frac{1}{\sqrt{\gamma}} \partial_\mu \left(\sqrt{\gamma} \gamma^{\mu\nu} \partial_\nu I \right). \tag{17}$$

This flow is called the anisotropic Beltami flow, which gives the variation of the I feature values of the image using the Randers type Finsler metric. When the right-hand side of Eq. (17) is calculated openly, the extension of the Laplace-Beltrami operator to Finsler spaces is obtained. This extension calculation is presented in the following proposition and proof.

Proposition 1. *The extension of the Laplace-Beltrami operator to Finsler spaces is given by the following:*

$$\Delta_\gamma I = \gamma^{\mu\nu} \left(\frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \Gamma^\alpha_{\mu\nu} \frac{\partial I}{\partial x^\alpha} \right), \tag{18}$$

where $\Gamma^\alpha_{\mu\nu}$, Σ are the components of the Christoffel symbol defined on the image manifold and is defined as follows [21]:

$$\Gamma^\alpha_{\mu\nu} = \frac{1}{2} \gamma^{\alpha\theta} \left(\gamma_{\theta\nu;\mu} + \gamma_{\mu\theta;\nu} - \gamma_{\mu\nu;\theta} \right), \quad \gamma_{\mu\alpha;\theta} = \frac{\partial \gamma_{\mu\nu}}{\partial x^\theta}. \tag{19}$$

Proof. If the derivative on the right side of Eq. (18) is calculated explicitiy and the equation $\frac{\partial \gamma^\nu}{\partial x^\mu} = -\Gamma^\nu_{\mu\theta} \gamma^\theta$ is used [22], the following equations are obtained:

$$\begin{aligned}
\frac{\partial I}{\partial t} &= \frac{1}{\sqrt{\gamma}} \frac{\partial}{\partial x^\mu} \left(\sqrt{\gamma} \gamma^{\mu\nu} \frac{\partial I}{\partial x^\nu} \right) \\
&= \frac{1}{\sqrt{\gamma}} \left(\frac{\partial \sqrt{\gamma}}{\partial x^\mu} \right) \gamma^{\mu\nu} \frac{\partial I}{\partial x^\nu} + \frac{\partial}{\partial x^\mu} \left(\gamma^\mu \cdot \gamma^\nu \frac{\partial I}{\partial x^\nu} \right) \\
&= \frac{1}{2} \left(\frac{\partial \ln \gamma}{\partial x^\mu} \right) \gamma^{\mu\nu} \frac{\partial I}{\partial x^\nu} + \frac{\partial \gamma^\mu}{\partial x^\mu} \cdot \left(\gamma^\nu \frac{\partial I}{\partial x^\nu} \right) \\
&\quad + \gamma^\mu \cdot \left(\frac{\partial \gamma^\nu}{\partial x^\mu} \right) \frac{\partial I}{\partial x^\nu} + \gamma^\mu \cdot \gamma^\nu \frac{\partial^2 I}{\partial x^\mu \partial x^\nu} \\
&= \frac{1}{2} \left(\frac{\partial \ln \gamma}{\partial x^\mu} \right) \gamma^{\mu\nu} \frac{\partial I}{\partial x^\nu} + \frac{\partial \gamma^\mu}{\partial x^\mu} \cdot \left(\gamma^\nu \frac{\partial I}{\partial x^\nu} \right) \\
&\quad + \gamma^\mu \cdot \left(-\Gamma^\nu_{\mu\theta} \gamma^\theta \right) \frac{\partial I}{\partial x^\nu} + \gamma^{\mu\nu} \frac{\partial^2 I}{\partial x^\mu \partial x^\nu} \\
\frac{\partial I}{\partial t} &= \frac{1}{2} \left(\frac{\partial \ln \gamma}{\partial x^\mu} \right) \gamma^{\mu\nu} \frac{\partial I}{\partial x^\nu} + \frac{\partial \gamma^\mu}{\partial x^\mu} \cdot \left(\gamma^\nu \frac{\partial I}{\partial x^\nu} \right) \\
&\quad - \gamma^{\mu\theta} \Gamma^\nu_{\mu\theta} \frac{\partial I}{\partial x^\nu} + \gamma^{\mu\nu} \frac{\partial^2 I}{\partial x^\mu \partial x^\nu}
\end{aligned} \tag{20}$$

where, γ^μ are the contravariant vectors on the image surface. The last two terms to the right of Eq. (20) are called the extension of the Laplace-Beltrami operator to Finsler spaces, which can be compactly written as follows [3]:

$$\begin{aligned}
\Delta_\gamma I &= \gamma^{\mu\nu} \frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \gamma^{\mu\theta} \Gamma^\nu_{\mu\theta} \frac{\partial I}{\partial x^\nu} \\
&= \gamma^{\mu\nu} \left(\frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \Gamma^\alpha_{\mu\nu} \frac{\partial I}{\partial x^\alpha} \right) \quad \square
\end{aligned} \tag{21}$$

In this study, only Δ_γ, which is the expansion of the Laplace-Beltrami operator to Finsler spaces, is considered to avoid complex computations in the anisotropic Beltrami flow. Thus, the approximate anisotropic Beltrami flow is obtained as follows:

$$\frac{\partial I}{\partial t} \cong \Delta_\gamma I = \gamma^{\mu\nu} \left(\frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \Gamma^\alpha_{\mu\nu} \frac{\partial I}{\partial x^\alpha} \right). \tag{22}$$

2. Finsler metrics used in the experimental study

Finsler spaces contain anisotropic structures because they depend on a point on the manifold and vectors in the tangent vector space. A Finsler fundamental function is defined as follows [20]:

Definition 1 *(Finsler Fundamental Function).* The function $F : TM \to [0, \infty)$ on an n-dimensional differentiable manifold M is a Finsler fundamental function if it satisfies the following conditions:

- Regularity: F, C^∞ differentiable on $TM\backslash 0$
- Positive homogenity: $F(x, y)$ is first-order positive homogeneous with respect to tangent vectors $v \in TM$

$$F(x, \lambda v) = \lambda F(x, v), \forall \lambda > 0 \tag{23}$$

- Strong convexity: For $\forall v \in TM\backslash\{0\}$, a positive definite fundamental tensor on TM is defined as follows:

$$g_{\mu\nu}(x, v) = \left(\frac{\partial^2}{\partial v^\mu \partial v^\nu} \left[\frac{1}{2} F^2(x, v) \right] \right), \tag{24}$$

where $\mu, \nu = 1, 2, ..., dim(M)$.

This Finsler fundamental function will be used in the metrics defined below. Consequently, the Finsler structure will be of importance in establishing these metrics.

2.1. Randers type metrics

The fundamental form of the Randers type metric, which is one of the simplest examples of Finsler metric structures, is as follows [23]:

$$F(x, v) = \sqrt{g_{\mu\nu}(x) v^\mu v^\nu} + b_\mu(x) v^\mu, \tag{25}$$

where $g = g_{\mu\nu}(x) dv^\mu dv^\nu$ is a Riemannian metric on manifold M, and $b = b_\mu(x) dv^\mu$ is an 1-form. For the fundamental function F to be positive definite, an 1-form must be determined on M such that the following condition is met [20]:

$$\|b\|_g = \sqrt{g^{\mu\nu}(x) b_\mu b_\nu} < 1. \tag{26}$$

The norm of the $b_\mu = \dfrac{\beta^2 g}{1 - g} I_{x^\mu}$ 1-form defined in [23] satisfies the following:

$$\|b\|_g = \sqrt{\frac{\beta^2 g}{1 - g}} > 1, \tag{27}$$

and it is seen that it does not meet the $\|b\|_g < 1$ condition. Therefore, instead of the Randers metric suggestion given in [10], a new Randers metric is defined by taking the components of the 1-form as $b_\mu = \frac{1}{V} I_{x^\mu}$, yielding the following two definitions:

- **New Randers Metric:** The fundamental function of the new Randers-type metric is as follows:

$$F(x, v) = \sqrt{g_{\mu\nu}(x) v^\mu v^\nu} + \left(\frac{1}{V} I_{x^\mu} \right) v^\mu, \tag{28}$$

where $V = g_{\mu\nu}(x) v^\mu v^\nu = \|v\|_g^2$ is the square of the norm of a tangent vector $v \in T_x\Sigma$ on the surface. An anisotropic Beltrami flow algorithm with Randers metric is given in Algorithm 2.1.

Algorithm 2.1: Anisotropic Beltrami Flow Algorithm with Randers Metric.

1 Load gray-level $I(x^1, x^2)$ image
2 Change the image data type to the floating points data type
3 Enter β aspect ratio
4 Enter the time step dt
5 **for** *each iteration* **do**
6 Calculate the first and second order central difference derivatives of the $I(x^1, x^2)$
7 Determine the $\underset{v \in T_x\Sigma}{\operatorname{argmax}} \left(\left| I(i, j) - I\left(i + v^1, j + v^2\right) \right| \right)$ the contravariant components of the tangent vector v which gives the neighborhood maximizing by using the 3×3 windowing
8 Calculate the Riemannian metric tensor $g_{\mu\nu}$, determinant g, and the inverse metric $g^{\mu\nu}$ at each point
9 Calculate the derivatives $\dfrac{\partial g_{\mu\nu}}{\partial x^\alpha} = \beta^2 (I_{x^\mu x^\alpha} I_{x^\nu} + I_{x^\nu x^\alpha} I_{x^\mu})$
10 Calculate the quadratic norm $V = g_{\mu\nu} v^\mu v^\nu$
11 Calculate the derivatives $\dfrac{\partial V}{\partial x^\alpha} = \dfrac{\partial g_{\mu\nu}}{\partial x^\alpha} v^\mu v^\nu$
12 Calculate the components of 1-form $b_\mu = \frac{1}{V} I_{x^\mu}$, $\|b\|_g$ norm, and the derivatives $\dfrac{\partial b_\mu}{\partial x^\alpha}$
13 Calculate the scalar linear function $\Omega = b_\mu v^\mu$, and $G = \sqrt{V}$ g-norm
14 Calculate the Finsler fundamental function $F = G + \Omega$
15 Calculate the complementary metric tensor $\varphi_{\mu\nu} = \frac{\Omega}{G} g_{\mu\nu} - \frac{\Omega}{G^3} g_{\mu\rho} g_{\nu\theta} v^\rho v^\theta + \frac{1}{G} \left(g_{\mu\theta} b_\nu + g_{\nu\theta} b_\mu \right) v^\theta + b_\mu b_\nu$, and its inverse $\varphi^{\mu\nu} = -\frac{\Omega}{F} g^{\mu\nu} + \frac{\Omega + G\|b\|_g^2}{F^3} v^\mu v^\nu - \frac{G}{F^2} \left(g^{\mu\alpha} b_\alpha v^\nu + g^{\nu\alpha} b_\alpha v^\mu \right)$
16 Calculate the Randers metric components $\gamma_{\mu\nu} = g_{\mu\nu} + \varphi_{\mu\nu}$, and its derivatives $\frac{\partial \gamma_{\mu\nu}}{\partial x^\alpha} = \frac{\partial g_{\mu\nu}}{\partial x^\alpha} + \frac{\partial \varphi_{\mu\nu}}{\partial x^\alpha}$
17 Calculate the inverse Randers metric components $\gamma^{\mu\nu} = g^{\mu\nu} + \varphi^{\mu\nu}$
18 Calculate the Christoffel symbol components $\Gamma^\alpha_{\mu\nu}$
19 Calculate the Laplace-Beltrami operator $\Delta_\gamma I = \gamma^{\mu\nu} \left(\frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \Gamma^\alpha_{\mu\nu} \frac{\partial I}{\partial x^\alpha} \right)$
20 $I \leftarrow I + dt \cdot \Delta_\gamma I$
21 **end**

- **Ingarden Metric:** By taking the components of the 1-form $b_\mu = I_{x^\mu}$, Ingarden-type fundamental function is defined as follows:

$$F(x, v) = \sqrt{g_{\mu\nu}(x) v^\mu v^\nu} + I_{x^\mu} v^\mu. \tag{29}$$

An anisotropic Beltrami flow with the Ingarden metric can also be obtained with the same algorithm as in Algorithm 2.1, only by taking $b_\mu = I_{x^\mu}$ in step 12.

2.2. Synge-Beil type metrics

Synge-Beil type metrics over a Σ manifold have the following form [23].

$$\gamma_{\mu\nu} = g_{\mu\nu} + c \cdot v_\mu v_\nu, \tag{30}$$

where $v_\mu = g_{\mu\nu} v^\nu$ is the covariant component of a tangent vector of the manifold, and $c = c(I, I_{x^\alpha})$ is a smooth scalar field on the image manifold that can depend on both the image properties and their derivatives. Synge-Beil type metrics defined according to the choices of c are given below.

- **Normalized Miron Metric:** By choosing the $c = \dfrac{1}{V}$ scalar function, the normalized Miron metric is defined [23]. An anisotropic Beltrami flow with a normalized Miron metric is in the given by Algorithm 2.2.
- **New Normalized Miron Metric:** A new normalized Miron metric is defined by choosing the scalar function $c = \left(1 - \dfrac{1}{g}\right) \dfrac{1}{V}$, where g (an edge detector) is equal to 1 in homogeneous (or flat) regions and nearly zero around inhomogeneous regions (such as edges). The anisotropic Beltrami flow with the

(a) (b) (c) (d) (e)

Fig. 1. All test images: (a) Clear Lena, (b) Gaussian noisy Lena, (c) Salt-Pepper noisy Lena, (d) Vein, (e) Camouflaged Field.

Algorithm 2.2: Normalized Miron Algorithm.

1 Load the gray-level $I(x^1, x^2)$ image
2 Change image data type to floating points data type
3 Enter the aspect ratio β
4 Enter the time step dt
5 **for** *each iteration* **do**
6 Calculate the first and second order central difference of the $I(x^1, x^2)$ image values
7 Determine the contravariant components of the v^μ tangent vector, which gives the neighborhood maximizing the $\underset{v \in T_x \Sigma}{\operatorname{argmax}} \left(\left| I(i,j) - I\left(i + v^1, j + v^2\right) \right| \right)$ difference by using 3×3 windowing
8 Calculate Riemannian metric tensor $g_{\mu\nu}$, and its g determinant, and its inverse $g^{\mu\nu}$
9 Determine the $v_\mu = g_{\mu\nu} v^\nu$ contravariant tangent components
10 Calculate the quadratic norm $V = g_{\mu\nu} v^\mu v^\nu$
11 Calculate the scalar function $c = \frac{1}{V}$ on image
12 Calculate the Normalized Miron metric $\gamma_{\mu\nu} = g_{\mu\nu} + c \cdot v_\mu v_\nu$
13 Calculate the Christoffel symbol components $\Gamma^\alpha_{\mu\nu}$
14 Calculate the characteristics of Normalized Miron metric $K = 1 + c \cdot V, S = -\frac{c}{K}$
15 Calculate the inverse Normalized Miron metric $\gamma^{\mu\nu} = g^{\mu\nu} + S \cdot v^\mu v^\nu$
16 Calculate the Laplace-Beltrami operator $\Delta_\gamma I = \gamma^{\mu\nu} \left(\frac{\partial^2 I}{\partial x^\mu \partial x^\nu} - \Gamma^\alpha_{\mu\nu} \frac{\partial I}{\partial x^\alpha} \right)$
17 $I \leftarrow I + dt \cdot \Delta_\gamma I$
18 **end**

newly defined normalized Miron metric is obtained by taking $c = \left(1 - \frac{1}{g}\right) \frac{1}{V}$ in step 11 of Algorithm 2.2.

The expression $1 - \frac{1}{g}$ performs the effect of the anisotropic part of the fundamental function, by making it almost zero in homogeneous (isotropic) regions of the image surface and by increasing it in inhomogeneous (anisotropic) regions. Therefore, $1 - \frac{1}{g}$ has the opposite effect of g, which can be defined as a homogeneous (isotropic) region detector.

- **New Metric:** A new metric is defined by choosing the $c = 1 - \frac{1}{g}$ scalar function in step 11 of Algorithm 2.2.

3. Image quality metrics

A non-orthodox application of the proposed metrics is noise filtering in images. Classical noise filtering methods in image processing applications aim to provide smoothing in noisy portions while keeping the edge portions visible and available for further applications such as edge detection. The results of image smoothing can be evaluated either by human eye, or objectively. In this work, we provide objective evaluations through smoothing output fidelity and through their suitability measurements for further edge detection performances. In this section, we briefly review the utilized fidelity measures, which are also known as image quality metrics. In calculations, an image X with size $m \times n$ will be taken as the original image and Y is the processed image.

Table 1
PSNR Results of the New Filter Family according to different values of β and dt.

Trial No	β	dt	New Metric	New N.Miron	New Randers
1	0.01	0.05	29.0727	29.0727	28.8833
2	0.01	0.1	27.8555	27.8555	27.6958
3	0.01	0.2	26.3092	26.3091	26.1258
4	0.01	0.01	25.5499	25.5499	25.5325
5	0.2	0.05	29.1010	29.1003	28.8873
6	0.2	0.1	27.9131	27.9126	27.7428
7	0.2	0.2	26.3108	26.3104	26.1277
8	0.2	0.01	25.5679	25.5683	25.5520
9	1	0.05	**29.1297**	**29.1131**	**28.9234**
10	1	0.1	27.9224	27.9091	27.7470
11	1	0.2	26.3454	26.3381	26.1766
12	1	0.01	25.4409	25.4521	25.4616

- **Mean Square Error (MSE):** It is calculated by averaging the squares of pixel value differences at the same coordinates of the two images [24][25]:

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(X_{i,j} - Y_{i,j} \right)^2 \tag{31}$$

- **Peak Signal to Noise Ratio (PSNR):** PSNR is the ratio of the peak power of a signal to the noise in the signal. Since it is inversely proportional to MSE, as the MSE approaches zero, the PSNR grows larger and approaches infinity, so a picture with a high reconstruction will have a higher PSNR [26][25].

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{MAX(Y)^2}{MSE} \right) \tag{32}$$

- **Image Fidelity (IF):** Image fidelity represents how well the resource is allocated between two images. For example, an IF value of 0.99 represents 1% information loss [27][25].

$$\mathrm{IF} = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (X(i,j) - Y(i,j))^2}{\sum_{i=1}^{n} \sum_{j=1}^{m} (X(i,j))^2} \tag{33}$$

- **Structural Similarity Index Measure (SSIM):** SSIM is a metric that measures the image quality degradation of digital pictures caused by information loss. The SSIM in the predicted image is based on the structural sources of the original, i.e. reference image. SSIM offers a similarity approach by using structural information such as contrast and light in the image. For example, SSIM of 0.9 gives the information that the estimated image is 90% structurally similar to the original image [28][25].

Table 2
Noise Reduction Results for Additive Gaussian Noise.

Filters	MSE	PEAKSNR	SSIM	IF	MI	Rank
Ingarden	0.001309705	28.82826672	0.794069177	0.99983037	2.242687698	5
Normalized Miron	0.001547515	28.10365142	0.657805686	1.00000000	1.999225724	6
New Randers	0.001281317	28.92343564	0.794718001	0.999876761	2.243635749	4
New N. Miron	0.001226552	29.11314059	0.795900885	1.000000000	2.248251732	2
New Metric	**0.001221863**	**29.12977532**	**0.79602522**	**1.000000000**	**2.249151847**	1
Beltrami	0.001232584	29.09183492	0.795772993	1.000000000	2.246884165	3
Gauss	0.00412837	23.8422139	0.410997405	1.000000000	1.46191881	9
Median	0.002095173	26.78780212	0.586454863	0.999931211	1.835653449	8
Mean	0.001701198	27.69245207	0.658238678	0.999601954	1.990697371	7

Table 3
Noise Reduction Results for Additive Salt Pepper Noise.

Filters	MSE	PEAKSNR	SSIM	IF	MI	Rank
Ingarden	0.001689799	27.72165069	0.760615578	0.999774805	2.162890067	5
Normalized Miron	0.002086916	26.80495066	0.642954189	1.000008887	2.220145374	6
New Randers	0.001611749	27.92702488	0.764977526	0.999864619	2.176803028	4
New N. Miron	0.001467996	28.33275233	0.777920753	1.000007435	2.224865613	3
New Metric	0.001465404	28.34042568	0.777905368	1.000007441	2.225185258	2
Beltrami	0.001492078	28.26208368	0.775628844	1.000007859	2.214994318	4
Gauss	0.006029128	22.19745483	0.45323612	1.000011062	3.516718791	8
Median	**0.000311034**	**35.07192355**	**0.947368091**	**0.999996349**	**3.60860497**	**1**
Mean	0.002307967	26.36770319	0.630016574	0.999628218	2.476146994	7

$$SSIM = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY}^2 + c_1)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \tag{34}$$

where μ corresponds to the mean, σ^2 corresponds to the (co)variance, and c_1 and c_2 are small numbers that stabilize the weak denominator problem.

- **Mutual Information (MI):** MI of two pictures actually means how much information one picture contains about the other. A high MI means a large reduction in uncertainty, a low MI means a small reduction, if the MI value is zero, the two images are independent of each other [29][25]:

$$MI = H(X) + H(Y) - H(X, Y), \tag{35}$$

where $H(X) = E\{-log_2(p(x))\}$ is the entropy of the source, X, and $H(X, Y)$ corresponds to the joint entropy of X and Y.

4. Experiments

The proposed metrics induce noise removal filters, which are compared with commonly available filters on the test images (Fig. 1). During the design of the proposed filters, the algorithm steps were run for 20 iterations, the scale parameter and the step size parameter were experimentally adjusted (see Sec. 4.2). The experimental tests consist of two stages:

1. Evaluation of the filtering result.
2. Evaluation of edge preservation habits.

In the filtering step, Gaussian noise was first added to a standard test image. Then comparisons were made with anisotropic and other classical filters. The filtered images and the noise-free images were compared based on the image quality metrics described in Sec. 3. After testing the filters with Gaussian noise, the filters were re-evaluated by adding salt and pepper noise to the noiseless image, since salt-pepper noise corresponds to a heavily tailed noise distribution, which is characteristically unique.

A noteworthy feature of anisotropic filters is that, despite the resulting smoothing effect, they preserve the edges. This property becomes critical in several applications, such as biomedical image processing (a vein image is provided as an example here). Notice that the vein image is natively quite blurred and noisy, therefore it would be a benchmark to see that the proposed filters preserve the edges of this image. In addition, a camouflaged field image was tested. This is because the edge details of this image are dense, making noise-detail separation a challenging problem.

4.1. Noise reduction

In the noise reduction tests, an anonymous image with Gaussian noise and salt and pepper noise is tested with the proposed anisotropic filters as well as previously available filters, according to the image quality metrics. In each column of tables, the value written in bold-face indicates the best numerical result.

Anisotropic filters are New Randers, Ingarden, Normalized Miron, New Normalized Miron, and New Metric. Other filters, which can be considered isotropic, are Beltrami, Gaussian, Mean, and Median filters. In the figures, the original image and the noisy images are on the top row. The other rows contain the results of conventional isotropic filters and the results of the anisotropic filters respectively.

The proposed anisotropic filters have two free parameters, that can be adjusted for the application; the scale:β, and the step size:dt. Using our test images, PSNR values were evaluated at various value settings of these parameters, and best results were obtained when $\beta = 1$ and $dt = 0.05$ (see Table 1). For the rest of the experiments, these parametric settings were used. Consequently, the filtering performances are evaluated and compared in Tables 2 and 3. In Table 2, the additive noise is Gaussian, in Table 3, the additive noise is a salt and pepper type noise. The performance rankings are given in the last columns. For the additive Gaussian noise, the proposed New Metric, together with the new proposed other metrics all perform superior to the classical and popular isotropic filters. For the additive salt and pepper noise, due to its fine tuned characteristic, the median filter jumps to the first rank, but it is followed by the proposed filters. Sample image outputs of the filter performances for both noise added cases are presented in Fig. 2.

Fig. 2. Gauss noise (left) and Salt-Pepper (right) reduction results: (a) Original, (b) Gaussian noisy image, (c) Salt-Pepper noisy image; Isotropic filters: (d) Beltrami, (e) Gauss, (f) Mean, (g) Median; Anisotropic filters: (h) New Randers, (i) Ingarden, (j) N. Miron, (k) New N. Miron, (l) New Metric.

4.2. Edge preservation

Noise filtering is usually not considered as a stand-alone process in image processing applications. Normally, the reason for noise filtering is to use it as a pre-processing for further semantic processing of the image, such as segmentation. Typically, an edge detection step follows noise filtering. One of the features that distinguishes anisotropic filters from others is their edge preserving property, despite their strong noise removal performance in smooth regions. In this section, the edge preservation properties of the same set of filters are examined.

In Fig. 3, the test image with artificially added noise was filtered and illustrated, whereas in Figs. 4 and 5, the intrinsically noisy image samples were processed. In all these figures, results of the popular Canny edge detector were presented as the edge detector output of the filtered images.

Looking at Fig. 3, we see that all isotropic filters, except Beltrami, failed to preserve the edges of the image. Notice that the classical filters in (d), (e) and (f) totally fail to eliminate the noise effect for consistent edge extraction. Considering the anisotropic filters in the third row of Fig. 3, the Normalized Miron in option (i) seems to perform visually worse than other new anisotropic filters.

Similar results were obtained for the intrinsically noisy "vein image" and "camouflaged field image". It was observed that isotropic filters, such as Gauss, mean or median filters fail to provide fair results in edge detection, with an exception of Beltrami flow, which, again provides good edge consistency.

Results for the vein image in Fig. 4 shows that the proposed anisotropic filters perform admirably, except certain ambiguous edges from the normalized Miron case (Fig. 4(e)). Similarly, the field image case in Fig. 5 shows almost excellent edge detection results for the anisotropic filters, except the normalized Miron case, where edges are inconsistent. It must be noted that the edge detection performances from Fig. 5(c),(d),(f),(g) are all on par, or beyond, state-of-the-art.

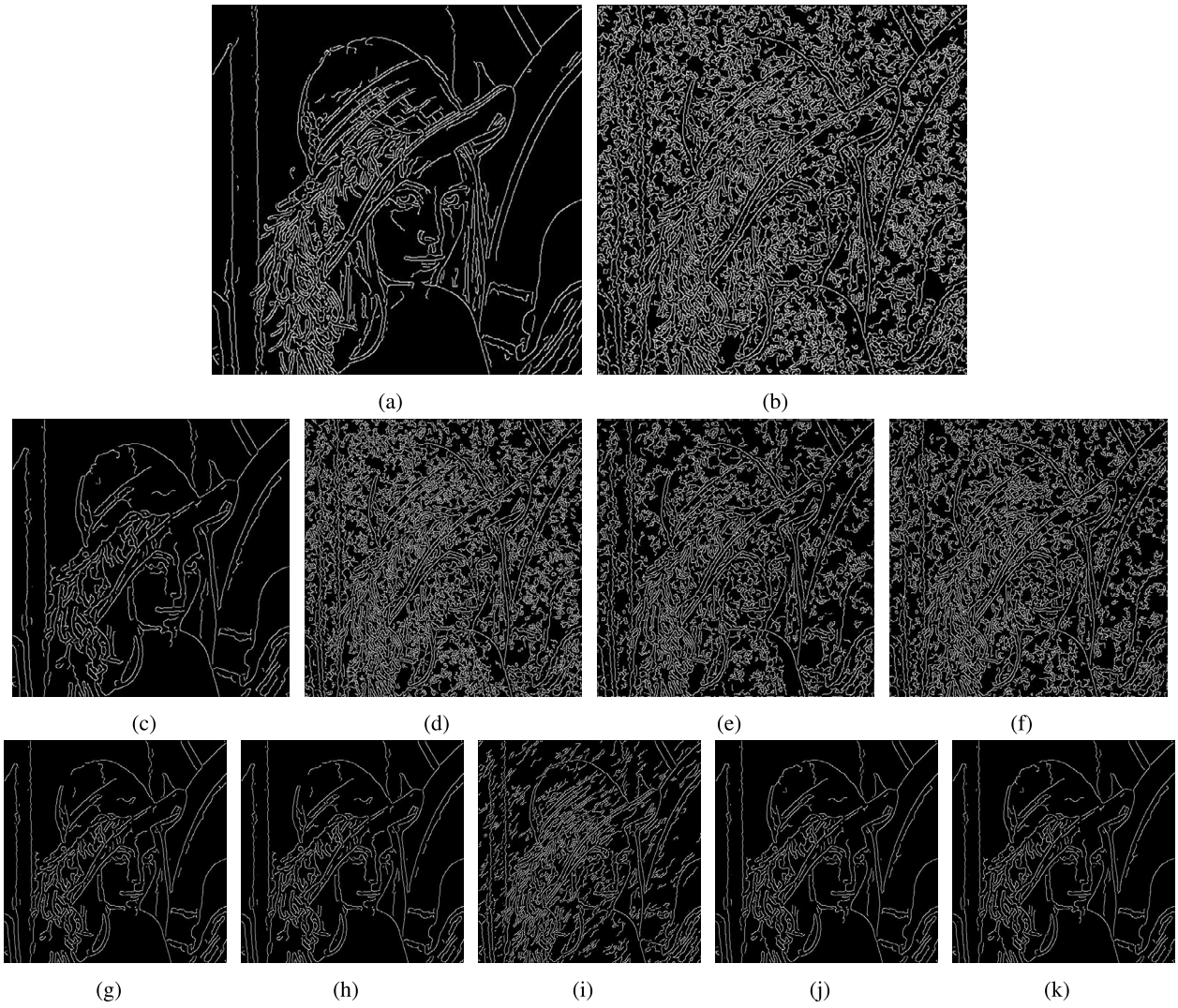


Fig. 3. Edge states of filtered images after Gaussian noise according to Canny edge detector [30]: (a) Original, (b) Noisy; Isotropic filters: (c) Beltrami, (d) Gauss, (e) Mean, (f) Median; Anisotropic filters: (g) New Randers, (h) Ingarden, (i) N. Miron, (j) New N. Miron, (k) New Metric.

4.3. Complexity analysis

In this section, computational complexity analysis of the developed algorithms is presented. As the computational times of the algorithms strongly depend on the parallelization and algorithm optimizations, multiplication-wise complexity orders make a more fair comparison of the algorithms. Furthermore, the source codes of these algorithms are made available at https://github.com/HAYDARKILIC. The anisotropic filters introduced in this study filter a noisy image by preserving its edges. Therefore, an $n = m \times m$ size image is taken as input and manipulated with filters to reduce noise. In addition, the number of iterations is taken as $k = m/25 = \sqrt{n}/25$.

The mathematical formulation of Gaussian, Median, and Mean filters can be found in [31]. From these formulations, the complexity of Gaussian, Median and Mean filters used in this study are all $O(n)$. Metric tensor, inverse metric tensor and quadratic norm calculation are common for all anisotropic algorithms, and their pseudo code is given in Algorithm 4.1 According to Algorithm 4.1, each element is executed n times. So the complexity of all derivatives is $O(n)$. There are quadratic derivatives and derivative products when calculating each element of the $g_{\mu\nu}$ metric tensor, so each element runs $n \times n$ times, making the complexity of the metric tensor $g_{\mu\nu}$ is $O(n^2)$. Similarly, the inverse of the metric tensor and the quadratic norm have the same complexity. The complexities of these common components are given in Table 4. The complexity of all anisotropic algorithms can be calculated using the these common components. According to Algorithm 2.1 of Randers type metrics and Algorithm 2.2 of Synge-Beil type metrics, the complexity of both algorithms becomes $O(n^2)$. The isotropic Beltrami flow in Algorithm 1.1 contains $O(n^2)$ complexity for the metric tensor and $O(n^2)$ complexity for H^3, ultimately making the complexity of the Beltrami flow, $O(n^2)$. The complexities of all algorithms are summarized in Table 5. It must be noted that, when

Algorithm 4.1: Common Component Calculation.

for *each image pixel position* **do**
  Set β and v
  Calculate first derivatives I_x, I_y
  Calculate second derivatives I_{xx}, I_{yy}, I_{xy}
  $g_{11} = 1 + \beta^2 I_x^2$
  $g_{12} = g_{21} = \beta^2 I_x I_y$
  $g_{22} = 1 + \beta^2 I_y^2$
  $detg = 1 + \beta^2 (I_x^2 + I_y^2)$
  $g_{\mu\nu} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$
  $V = v^T g_{\mu\nu}$
end

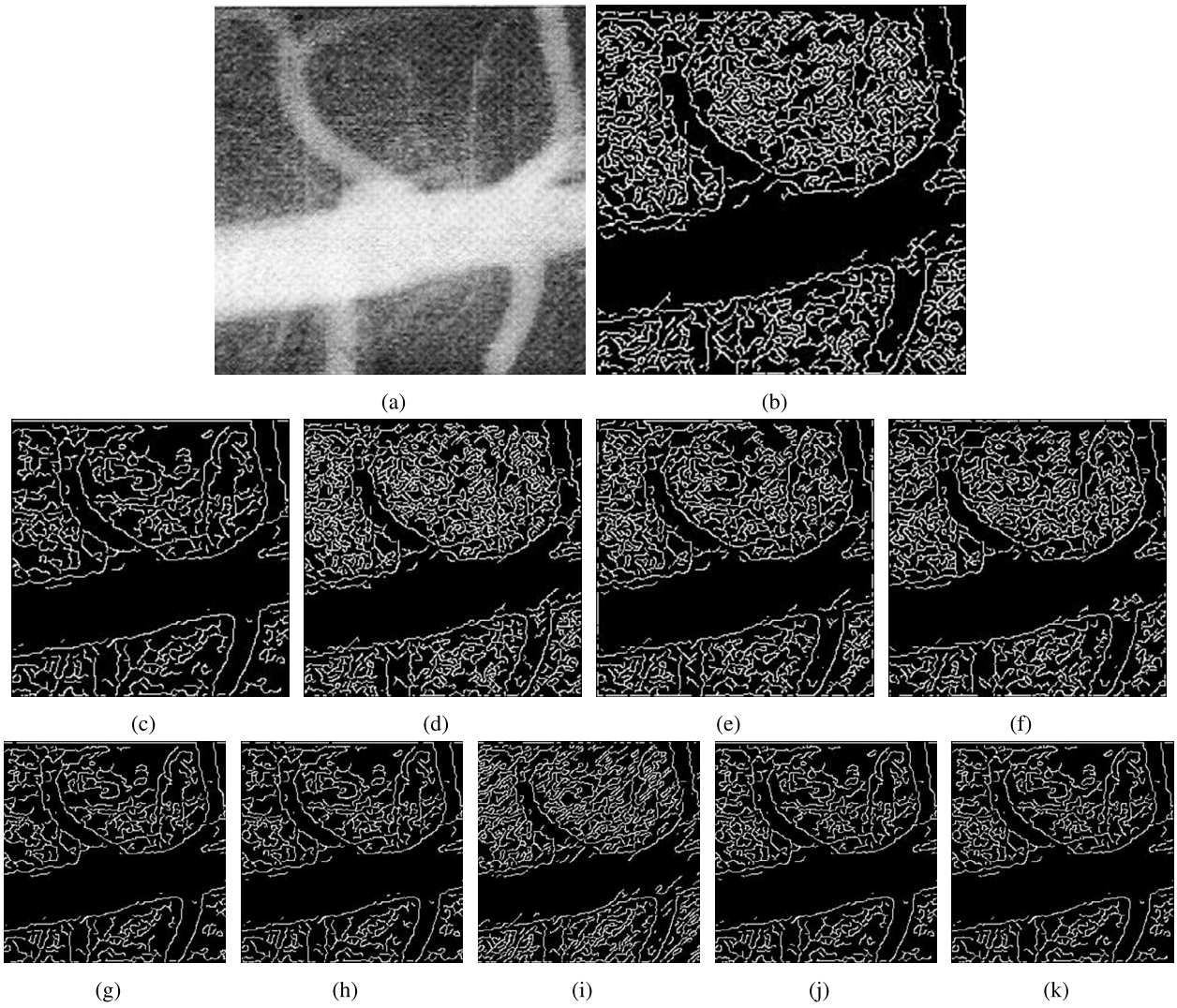


Fig. 4. Edge states after filtering of a self-noise vein image according to Canny edge detector [30]: (a) Original, (b) Edges; Isotropic filters: (c) Beltrami, (d) Gauss, (e) Mean, (f) Median; Anisotropic filters: (g) New Randers, (h) Ingarden, (i) N. Miron, (j) New N. Miron, (k) New Metric.

Table 4
Complexity of Common Algorithm Components.

Algorithm Component	Complexity
First derivatives: I_x, I_y	$O(n)$
Second derivatives: I_{xx}, I_{yy}, I_{xy}	$O(n)$
Metric tensor: $g_{\mu v}$	$O(n^2)$
Inverse Metric tensor: $g^{\mu v}$	$O(n^2)$
Quadratic norm: V	$O(n^2)$

Table 5
Complexity of Algorithms.

Algorithms	Complexity
New Metric	$O(n^2)$
New N. Miron	$O(n^2)$
New Randers	$O(n^2)$
Ingarden	$O(n^2)$
N. Miron	$O(n^2)$
Beltrami	$O(n^2)$
Gauss	$O(n)$
Median	$O(n)$
Mean	$O(n)$

the actual implementations are considered, the computation times of anisotropic filters vary according to the characteristics of the image. Therefore, the complexity figures provide only an order of the multiplication complexity, and the readers are advised to test the shared sources for their actual run-time speeds under the user's computer platform. As a case example, when the software provided in the github link is executed on an Intel i7 PC with Nvidia 940M GPU and 16 GB of RAM, the processing of a 256×256 with Gauss or mean linear filters take 0.11 ms on the average, whereas the process times for Median, Beltrami, N. Miron, Ingarden, New N. Miron, New Randers, and New Metric take 0.24, 0.27, 0.20, 0.32, 0.28, 0.22 and 0.21, respectively when averaged over the test images.

5. Conclusions

In this study, image filtering results and performance comparisons for a new class of anisotropic filters are presented. The new filters were achieved through a special selection of edge-preserving anisotropic Finsler metrics that produce anisotropic Laplace-Beltrami flows to minimize the Polyakov action energy integral. These novel anisotropic filters were applied to noisy images and compared according to their noise removal properties. It is experimentally observed that the new anisotropic filters give superior noise elimination performances as compared to the classical

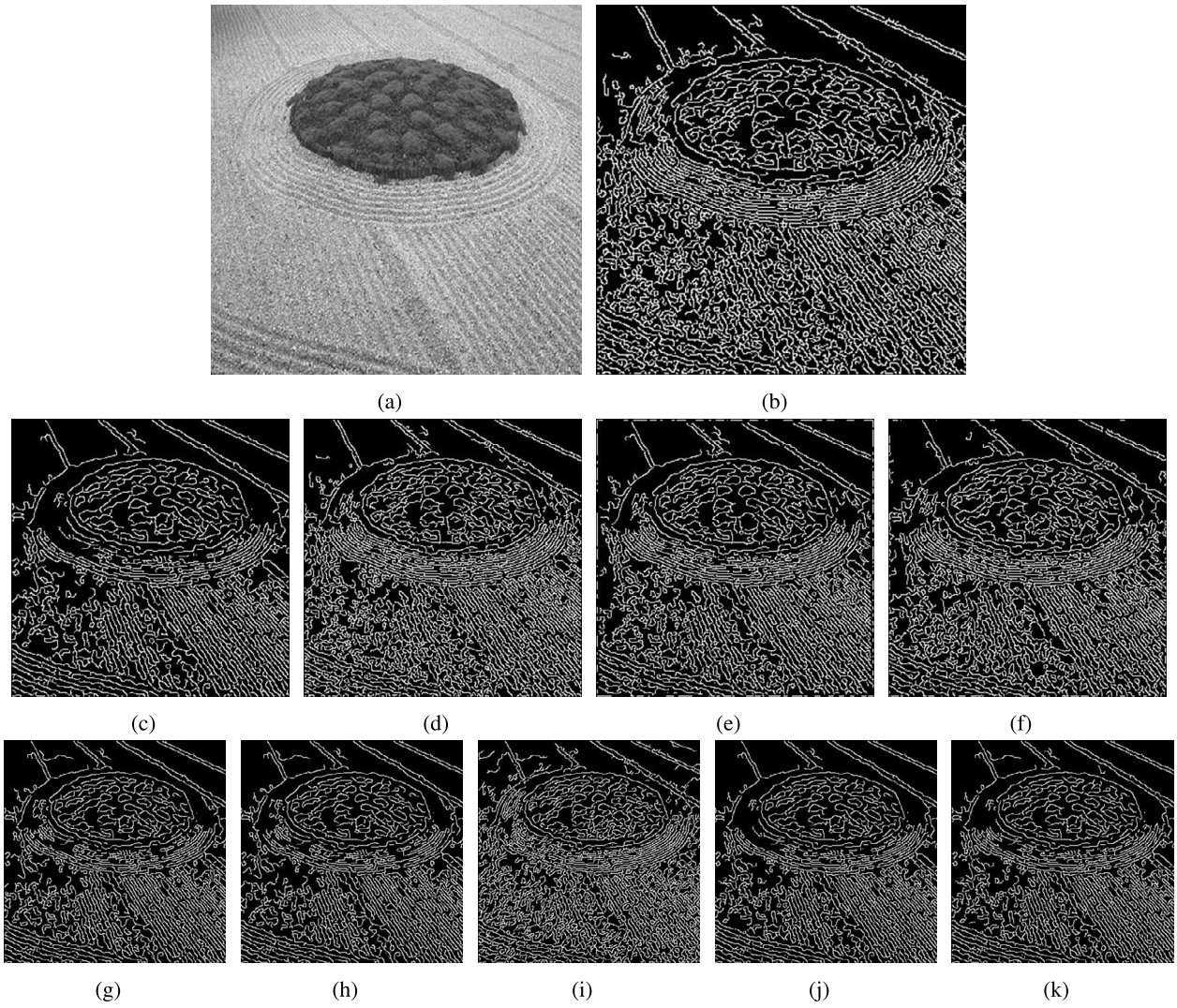


Fig. 5. Edge states of a camouflaged field image after filtering according to Canny edge detector [30]: (a) Original, (b) Edges; Isotropic filters: (c) Beltrami, (d) Gauss, (e) Mean, (f) Median; Anisotropic filters: (g) New Randers, (h) Ingarden, (i) N. Miron, (j) New N. Miron, (k) New Metric.

isotropic filters according to several image quality metrics, such as MSE, PSNR, SSIM, etc. However, the real advantage of the novel anisotropic filters and the Beltrami flow becomes even more eminent when a further processing step of edge detection is applied. It is critical to note that the anisotropic filters successfully eliminate noise from smooth regions while perfectly retaining edge information, thanks to their space-varying nature. Therefore, despite the slight increase in the computational cost, this strong property of the anisotropic filters is expected to be of critical significance in applications such as biomedical image processing and imaging systems that require accurate resolution, such as satellite imaging. Consequently, it is concluded that the new anisotropic filters obtained from Finsler geometry may provide a plausible alternative to traditional isotropic filters in image processing.

CRediT authorship contribution statement

Haydar Kilic: Conceptualization, Investigation, Methodology, Visualization, Writing – review & editing. **Salim Ceyhan:** Conceptualization, Methodology. **Omer Nezih Gerek:** Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Nir Sochen, Ron Kimmel, Ravi Malladi, A general framework for low level vision, IEEE Trans. Image Process. 7 (3) (1998) 310–318.
[2] Ron Kimmel, Nir Sochen, Ravi Malladi, From high energy physics to low level vision, in: International Conference on Scale-Space Theories in Computer Vision, Springer, 1997, pp. 236–247.
[3] Vladimir Balan, Jelena Stojanov, Finslerian extensions of geodesic active fields for digital image registration, PAMM 13 (1) (2013) 493–494.
[4] Youngjung Kim, Dongbo Min, Bumsub Ham, Kwanghoon Sohn, Fast domain decomposition for global image smoothing, IEEE Trans. Image Process. 26 (8) (2017) 4079–4091.
[5] Carlo Noel Ochotorena, Yukihiko Yamashita, Anisotropic guided filtering, IEEE Trans. Image Process. 29 (2019) 1397–1412.
[6] Pietro Perona, Jitendra Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639.
[7] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, Tomeu Coll, Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal. 29 (1) (1992) 182–193.
[8] Yunmei Chen, Célia A. Zorzo Barcelos, Bernard Anthony Mair, Smoothing and edge detection by time-varying coupled nonlinear diffusion equations, Comput. Vis. Image Underst. 82 (2) (2001) 85–100.

[9] Niveditta Thakur, Nafis Uddin Khan, Sunil Datt Sharma, A review on performance analysis of PDE based anisotropic diffusion approaches for image enhancement, Informatica 45 (6) (2021).
[10] Sumit Kushwaha, Rabindra Kumar Singh, Performance comparison of different despeckled filters for ultrasound images, Biomed. Pharmacol. J. 10 (2) (2017) 837–845.
[11] Sumit Kushwaha, Rabindra Kumar Singh, Optimization of the proposed hybrid denoising technique to overcome over-filtering issue, Biomed. Eng./Biomed. Tech. 64 (5) (2019) 601–618.
[12] Gabriel Ramos-Llordén, Gonzalo Vegas-Sánchez-Ferrero, Marcos Martin-Fernandez, Carlos Alberola-López, Santiago Aja-Fernández, Anisotropic diffusion filter with memory based on speckle statistics for ultrasound images, IEEE Trans. Image Process. 24 (1) (2014) 345–358.
[13] L. Blanc-Féraud, M. Barlaud, Edge preserving restoration of astrophysical images, Vistas Astron. 40 (4) (1996) 531–538.
[14] Christophe Samson, Laure Blanc-Féraud, Gilles Aubert, Josiane Zerubia, A variational model for image classification and restoration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (5) (2000) 460–472.
[15] Prabhishek Singh, Achyut Shankar, A novel optical image denoising technique using convolutional neural network and anisotropic diffusion for real-time surveillance applications, J. Real-Time Image Process. 18 (5) (2021) 1711–1728.
[16] Bhupendra Gupta, Subir Singh Lamba, et al., An efficient anisotropic diffusion model for image denoising with edge preservation, Comput. Math. Appl. 93 (2021) 106–119.
[17] Dominique Zosso, Xavier Bresson, Jean-Philippe Thiran, Geodesic active fields–a geometric framework for image registration, IEEE Trans. Image Process. 20 (5) (2010) 1300–1312.
[18] Alexander M. Polyakov, Quantum geometry of bosonic strings, Phys. Lett. B 103 (3) (1981) 207–210.
[19] Ravi Malladi, James A. Sethian, Image processing: flows under min/max curvature and mean curvature, Graph. Models Image Process. 58 (2) (1996) 127–141.
[20] Zhongmin Shen, Differential Geometry of Spray and Finsler Spaces, Springer Science & Business Media, 2013.
[21] Erwin Kreyzig, Differential Geometry, Dover Publications, 1991.
[22] Hung Nguyen-Schäfer, Jan-Philip Schmidt, Tensor Analysis and Elementary Differential Geometry for Physicists and Engineers, Springer, 2014.
[23] Jelena Stojanov, Anisotropic Frameworks for Dynamical Systems and Image Processing, PhD thesis, University of Novi Sad (Serbia), 2015.
[24] David M. Allen, Mean square error of prediction as a criterion for selecting variables, Technometrics 13 (3) (1971) 469–475.
[25] Mahdi Khosravy, Nilesh Patel, Neeraj Gupta, Ishwar K. Sethi, Image quality assessment: a review to full reference indexes, in: Recent Trends in Communication, Computing, and Electronics, 2019, pp. 279–288.
[26] D. Poobathy, R. Manicka Chezian, Edge detection operators: peak signal to noise ratio based comparison, in: IJ Image, Graphics and Signal Processing, vol. 10, 2014, pp. 55–61.
[27] D. Amnon Silverstein, Joyce E. Farrell, The Relationship Between Image Fidelity and Image Quality, Proceedings of 3rd IEEE International Conference on Image Processing, vol. 1, IEEE, 1996, pp. 881–884.
[28] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[29] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, Bart Preneel, Mutual information analysis, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2008, pp. 426–442.
[30] John Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1986) 679–698.
[31] Subrato Bharati, Tanvir Zaman Khan, Prajoy Podder, Nguyen Quoc Hung, A Comparative Analysis of Image Denoising Problem: Noise Models, Denoising Filters and Applications, Springer International Publishing, Cham, ISBN 978-3-030-55833-8, 2021, pp. 49–66.

Haydar Kilic is a Ph.D. student in electronics and computer engineering from Bilecik Seyh Edebali University, Bilecik, Turkey. He received M.Sc. degree in computer engineering Case Western Reserve University, Cleveland, Ohio, in 2013. He is currently teaching assistant with the Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey. His research interests include signal and image processing, and analysis.

Salim Ceyhan received the B.Sc., M.Sc., and Ph.D. degrees in Mathematical Engineering at Istanbul Technical University (ITU) in 1987, 1991, and 1994, respectively. Since 2011, he has been assistant professor in the Bilecik Seyh Edebali University, Bilecik, Turkey. His research areas include applied mathematics, Finsler geometry, metric spaces, and image processing.

Omer Nezih Gerek received the Ph.D. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 1998., During his Ph.D., he spent a semester at the University of Minnesota as a Researcher. From 1998 to 1999, he was a Technical Researcher with the Swiss Federal Institute of Technology, Lausanne, Switzerland. He is currently a Full Professor with the Department of Electrical and Electronics Engineering, Eskisehir Technical University, Eskisehir, Turkey. He has served in the editorial board of several journals, and is currently an area editor in Digital Signal Processing. His research interests include signal and image processing, and analysis.