Fourth-Order Anisotropic Diffusion for Inpainting and Image Compression

Ikram Jumakulyyev and Thomas Schultz

Abstract Edge-enhancing diffusion (EED) can reconstruct a close approximation of an original image from a small subset of its pixels. This makes it an attractive foundation for PDE based image compression. In this work, we generalize second-order EED to a fourth-order counterpart. It involves a fourth-order diffusion tensor that is constructed from the regularized image gradient in a similar way as in traditional second-order EED, permitting diffusion along edges, while applying a non-linear diffusivity function across them. We show that our fourth-order diffusion tensor formalism provides a unifying framework for all previous anisotropic fourth-order diffusion based methods, and that it provides additional flexibility. We achieve an efficient implementation using a fast semi-iterative scheme. Experimental results on natural and medical images suggest that our novel fourth-order method produces more accurate reconstructions compared to the existing second-order EED.

1 Introduction

The increased availability and resolution of imaging technology, including digital cameras and medical imaging devices, along with advances in storage capacity and transfer bandwidths, have led to a proliferation of large image data. This makes image compression an important area of research. Image compression techniques can be divided into two main groups: Lossy and lossless compression. Lossless compression techniques permit restoration of the full, unmodified image data, which however limits the achievable compression rates. Our work is concerned with lossy compression, which achieves much higher compression rates by replacing the original image with an approximation that can be stored more efficiently.

I. Jumakulyyev · T. Schultz (✉)
B-IT and Department of Computer Science II, University of Bonn,
Friedrich-Hirzebruch-Allee 5, 53115 Bonn, Germany
e-mail: schultz@cs.uni-bonn.de

I. Jumakulyyev
e-mail: ijumakulyyev@cs.uni-bonn.de

We continue a line of research that has explored the use of Partial Differential Equations (PDEs) for lossy image compression [12, 13, 23, 28]. This approach is based on storing only a small subset of all pixels, and interpolating between them in order to restore the remaining ones. There is a strong similarity between that interpolation process and image inpainting, whose goal it is to reconstruct missing or corrupted parts of an image. PDE-based methods for image inpainting and compression are inspired by the physical phenomenon of heat transport. It is described by the heat diffusion equation

$$\partial_t u = \mathrm{div}(D \cdot \nabla u) \ , \tag{1}$$

which relates temporal changes in a heat concentration $\partial_t u$ to the divergence of its spatial gradient ∇u. When diffusion takes place in an isotropic medium, the diffusivity D is a scalar that determines the rate of heat transfer. In an anisotropic medium, heat spreads out more rapidly in some directions than in others. In those cases, D is a diffusion tensor, i.e., a symmetric matrix that encodes this directional dependence.

When applied to image processing, the gray value at a certain location is interpreted as the heat concentration u. In diffusion-based image inpainting, Eq. (1) is used to propagate information from the known pixels, whose intensity is fixed, to the unknown pixels which will ultimately reach a steady state in which their intensity is determined by their surrounding known pixels. In this sense, Eq. (1) has a filling-in effect that can be exploited for image compression.

Different choices of the diffusivity function D lead to different kinds of diffusion. Linear diffusion [20] and nonlinear diffusion [24] were widely used for image smoothing and image enhancement. Edge structures in images can be enhanced by employing a diffusion tensor which allows diffusion in the direction perpendicular to the local gradient, while applying a nonlinear diffusivity function along the gradient direction. This idea has led to the development of anisotropic nonlinear edge-enhancing diffusion (EED) [34]. Among the six variants that were evaluated for image compression by Galić et al. [13], EED led to the most accurate reconstructions. Subsequently, this idea was applied to three-dimensional data compression [25], and combined with motion compensation in order to obtain a framework for video compression [2]. When combined with a suitable scheme for selecting and storing the preserved pixels, a few additional optimizations, and at sufficiently high compression rates, anisotropic diffusion has been shown to beat the quality even of JPEG2000 [27].

In this paper, we introduce a novel fourth-order PDE that generalizes second-order EED, and achieves even more accurate reconstructions. We build on prior works that proposed fourth-order analogs of the diffusion equation, and used them for image processing [10, 19, 21, 22, 26, 35]. In particular, we extend a work by Gorgi Zadeh et al. [15], who introduced the idea of steering anisotropic fourth-order diffusion with a fourth-order diffusion tensor. However, their method focuses on the curvature enhancement property of nonlinear fourth-order diffusion [10] in order to better localize ridge and valley structures. Deriving a suitable PDE for image inpainting requires a different definition of the diffusion tensor, more similar to the one in edge-

enhancing diffusion [34]. Two anisotropic fourth-order PDEs for inpainting were previously introduced by Li et al. [21]. However, they only apply them to image restoration tasks in which small parts of an image are missing (such as in Fig. 8), not to the reconstruction from a small subset of pixels. Moreover, we demonstrate that the fourth-order diffusion tensor based framework is more general in the sense that it can be used to express anisotropic fourth-order diffusion as it was described by Hajiaboli [19] or by Li et al. [21], while providing additional flexibility.

2 Background and Related Work

We will now formalize the above-mentioned idea of diffusion-based inpainting (Sect. 2.1), and review two concepts that play a central role in our method: Anisotropic nonlinear diffusion (Sect. 2.2) and fourth-order diffusion (Sect. 2.3). Further details can be found in works by Galić et al. [13] and Weickert [34], respectively. Finally, we provide additional context with a brief discussion of alternative approaches to image compression (Sect. 2.4).

2.1 Diffusion-Based Inpainting

In order to apply Eq. (1) to image smoothing, we have to restrict it to the image domain Ω, and specify the behavior along its boundary $\partial\Omega$. It is common to assume that no heat is transferred through that boundary (homogeneous Neumann boundary condition). Moreover, the positive real line $(0, \infty)$ is typically taken as the time domain. The resulting PDE can be written as

$$\partial_t u = \operatorname{div}(D \cdot \nabla u), \quad \Omega \times (0, \infty) \,,$$
$$\partial_n u = 0, \quad \partial\Omega \times (0, \infty) \,, \tag{2}$$

where n is the normal vector to the boundary $\partial\Omega$. The original image $f : \Omega \to \mathbb{R}$ is used to specify an initial condition $u = f$ at $t = 0$. For increasing diffusion time t, u will correspond to an increasingly smoothed version of the image.

In image inpainting, we know the pixel values on a subset $K \subset \Omega$ of the image, and aim to reconstruct plausible values in the unknown regions. A diffusion-based model for inpainting can be derived from the one for smoothing, by modeling the set of locations at which the pixel values are known with Dirichlet boundary conditions. In this case, $f : K \to \mathbb{R}$ will be used to model the known values. In inpainting-based image compression, K will consist of a small fraction of the pixels in the original image. With this, we obtain the following model for inpainting:

$$\partial_t u = \text{div}(D \cdot \nabla u), \quad \Omega \backslash K \times (0, \infty) ,$$
$$\partial_n u = 0, \quad \partial\Omega \times (0, \infty) , \tag{3}$$
$$u = f, \quad K \times [0, \infty)$$

In this case, the diffusion process spreads out the information from the known pixels to their spatial neighborhood. For time $t \to \infty$, image smoothing and inpainting both converge to a steady state, i.e., $\lim_{t \to \infty} \partial_t u = 0$. However, the steady-state of smoothing is trivial (u approaches a constant image with average gray value), while the Dirichlet boundary conditions in the inpainting case ensure a non-trivial steady-state, which is taken as the final inpainting result: $u_{\text{inpainted}} = \lim_{t \to \infty} u$.

2.2   From Linear to Anisotropic Nonlinear Diffusion

So far, we assumed that the diffusion coefficient D is a scalar and constant, independent from the location within the image. This results in an inpainting model based on linear homogeneous diffusion [20]. With $D = 1$, it can be written as

$$\partial_t u = \Delta u, \quad \Omega \backslash K \times (0, \infty) . \tag{4}$$

In this and all remaining equations in this section, the same boundary conditions are assumed as specified in Eq. (3). Despite its simplicity, it has been demonstrated that using this inpainting model for image compression can already beat the JPEG standard when applied to cartoon-like images, and selecting the retained pixels to be close to image edges [23].

When the diffusion coefficient is a scalar but depends on u, i.e., $D = g(u)$, then we call the model inpainting based on nonlinear isotropic diffusion [24]. A common variant is to make D depend on the local gradient magnitude, i.e.,

$$\partial_t u = \text{div}(g(||\nabla u_\sigma||^2)\nabla u), \quad \Omega \backslash K \times (0, \infty) , \tag{5}$$

where g is a decreasing nonnegative diffusivity function, e.g., the Charbonnier diffusivity

$$g(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\lambda^2}}} , \tag{6}$$

and λ is a contrast parameter separating low from high diffusion areas [8]. In order to localize edges better and to make the problem well-posed, the image is pre-smoothed with a Gaussian before taking its gradient, i.e., $g(||\nabla u_\sigma||^2)$ is used instead of $g(||\nabla u||^2)$ [6].

In the above-discussed models, the diffusion occurs only in the gradient direction. This can be changed by replacing the scalar diffusivity with a second-order diffusion tensor, i.e., a symmetric positive definite matrix. This is the basis of anisotropic nonlinear diffusion [34],

$$\partial_t u = \mathrm{div}(\mathbf{D} \cdot \nabla u), \quad \Omega \backslash K \times (0, \infty) \ . \tag{7}$$

In edge-enhancing diffusion (EED), the diffusion tensor $\mathbf{D}$ is defined as

$$\mathbf{D} = g(||\nabla u_\sigma||^2) \cdot \mathbf{v}_1 \mathbf{v}_1^{\mathrm{T}} + 1 \cdot \mathbf{v}_2 \mathbf{v}_2^{\mathrm{T}}, \tag{8}$$

where $\mathbf{v}_1 = \frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ and $\mathbf{v}_2 = \frac{\nabla u_\sigma^\perp}{||\nabla u_\sigma||_2}$. This means that diffusion across the edge ($\mathbf{v}_1$) is decreased depending on the gradient magnitude, while diffusion along the edge ($\mathbf{v}_2$) is allowed. Examples of EED based inpainting are included in our experimental results. In general, EED based inpainting results in better interpolated images than linear homogeneous or nonlinear isotropic PDEs. This makes it a current state-of-the-art choice for PDE-based image compression.

2.3 *From Second to Fourth Order Diffusion*

All models discussed above, as well as several others that have been proposed for inpainting [36], share a common property: They rely on second order PDEs. In image denoising, higher-order PDEs have a long history, going back to work by Scherzer [26]. You and Kaveh [35] propose fourth-order PDEs as a solution to the so-called staircasing problem that arises in edge-enhancing second-order PDEs, such as the filter proposed by Perona and Malik [24]: While the second-order Perona-Malik equation creates visually unpleasant step edges from continuous variations of intensity, corresponding fourth-order methods move these discontinuities into the gradients, where they are less noticeable to the human eye [16]. Subsequently, other fourth-order PDE-based models have been introduced, and have mostly been applied for denoising [18, 19, 22].

For a specific family of higher-order diffusion filters, Didas et al. [10] have shown that, in addition to preserving average gray value, they also preserve higher moments of the initial image. Moreover, depending on the diffusivity function, they can lead to adaptive forward and backward diffusion, and therefore to the enhancement of image features such as curvature. Gorgi Zadeh et al. [15] made use of this property in order to enhance ridges and valleys, by steering fourth-order diffusion with a fourth-order diffusion tensor. Our work adapts their method in order to achieve accurate inpainting and reconstruction from a small subset of pixels.

2.4   Alternative Approaches to Image Compression

The dominant lossy image compression techniques today are JPEG and JPEG2000. They are based on the discrete cosine transform (DCT) and wavelet transform, respectively. However, they are not sensitive to the geometry of an image, i.e. those standards are not tailored to their geometrical behavior [5]. Especially, the JPEG standard involves dividing the image into small square blocks. This can cause a degradation called "blocking effect" [30], and can result in unsatisfactory reconstructions especially at high compression rates.

It is an ongoing research trend to apply machine learning methods to image compression, such as convolutional and recurrent neural networks [3, 31, 32]. Learning based approaches tend to perform very well on the specific class of images on which they were trained, but require a huge amount of data. For example, Toderici et al. [33] used for training a dataset of 6 million 1280×720 images taken from the web.

3   Method

We will now introduce our novel PDE (Sect. 3.1), investigate its relationship to previously proposed anisotropic fourth-order diffusion (Sect. 3.2), and comment on our chosen discretization, as well as numerical stability (Sect. 3.3).

3.1   Anisotropic Edge-Enhancing Fourth Order PDE

Our fourth-order PDE builds on a model that was proposed by Gorgi Zadeh et al. [15] for ridge and valley enhancement. It can be stated concisely using Einstein notation, where summation is implied for indices appearing twice in the same expression:

$$\partial_t u = -\partial_{ji} \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) : \mathbf{H}(u) \right]_{ij} \tag{9}$$

In this equation, $\mathbf{H}(u)$ denotes the Hessian matrix of image u. The "double dot product" $\mathbf{T} = \mathcal{D} : \mathbf{H}$ indicates that matrix $\mathbf{T}$ is obtained by applying a linear map $\mathcal{D}$ to $\mathbf{H}$, and the square bracket notation $[\mathbf{T}]_{ij}$ indicates taking the (i, j)th component:

$$[\mathbf{T}]_{ij} = \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) : \mathbf{H}(u) \right]_{ij} = \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) \right]_{ijkl} [\mathbf{H}(u)]_{kl} \tag{10}$$

Since $\mathcal{D}$ maps matrices to matrices, it is a fourth-order tensor. Since its role is analogous to the second-order diffusion tensor in Eq. (7), it is referred to as a fourth-order diffusion tensor.

The diffusion tensor $\mathcal{D}$ in Eq. (9) is a function of the local normalized Hessian $\mathbf{H}_\rho(u_\sigma)$ which contains the information that is relevant to achieve curvature enhancement. For image inpainting, we propose to instead steer the fourth-order diffusion in analogy to edge-enhancing diffusion, i.e., as a function of the structure tensor $\mathbf{J}(u_\sigma)$, which is obtained from image u after Gaussian pre-smoothing with bandwidth σ. We construct our fourth order diffusion tensor $\mathcal{D}$ from its eigenvalues μ_i and eigentensors $\mathbf{E}_i$ via the spectral decomposition:

$$\mathcal{D}\left(\mathbf{J}(u_\sigma)\right) = \mu_1 \mathbf{E}_1 \otimes \mathbf{E}_1 + \mu_2 \mathbf{E}_2 \otimes \mathbf{E}_2 + \mu_3 \mathbf{E}_3 \otimes \mathbf{E}_3 + \mu_4 \mathbf{E}_4 \otimes \mathbf{E}_4 \qquad (11)$$

The eigenvalues and eigentensors are defined as

$$
\begin{aligned}
\mu_1 &= g(\lambda_1), \ \mathbf{E}_1 = \mathbf{v}_1 \otimes \mathbf{v}_1 \ , \\
\mu_2 &= 1, \ \mathbf{E}_2 = \mathbf{v}_2 \otimes \mathbf{v}_2 \ , \\
\mu_3 &= \sqrt{g(\lambda_1)}, \ \mathbf{E}_3 = \frac{1}{\sqrt{2}}(\mathbf{v}_1 \otimes \mathbf{v}_2 + \mathbf{v}_2 \otimes \mathbf{v}_1) \ , \\
\mu_4 &= 0, \ \mathbf{E}_4 = \frac{1}{\sqrt{2}}(\mathbf{v}_1 \otimes \mathbf{v}_2 - \mathbf{v}_2 \otimes \mathbf{v}_1) \ ,
\end{aligned}
\qquad (12)
$$

where g is a nonnegative decreasing diffusivity function, λ_i and $\mathbf{v}_i$ are eigenvalues and eigenvectors of the structure tensor $\mathbf{J}(u_\sigma) = \nabla u_\sigma \nabla u_\sigma^{\mathrm{T}}$, i.e., $\lambda_1 = ||\nabla u_\sigma||_2^2$, $\mathbf{v}_1 = \frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ and $\lambda_2 = 0$, $\mathbf{v}_2 = \frac{\nabla u_\sigma^\perp}{||\nabla u_\sigma||_2}$. The above-defined eigentensors are orthonormal with respect to the dot product $\mathbf{A} : \mathbf{B} = \mathrm{trace}(\mathbf{B}^{\mathrm{T}}\mathbf{A})$ [15].

Combining this new definition of the fourth-order diffusion tensor with Dirichlet boundary conditions as in Eq. (3) results in our proposed model:

$$
\begin{aligned}
\partial_t u &= -\partial_{ji}[\mathcal{D}(\mathbf{J}(u_\sigma)) : \mathbf{H}(u)]_{ij}, \quad \Omega \backslash K \times (0, \infty) \ , \\
u &= f, \quad K \times [0, \infty)
\end{aligned}
\qquad (13)
$$

As it is customary in PDE-based inpainting, we allow Eq. (13) to evolve until a steady state has been reached, i.e., the time derivative becomes negligible. In our numerical implementation, we use a Fast Semi-Iterative Scheme (FSI) [17] to greatly accelerate convergence to a large stopping time.

In the definition of our fourth-order diffusion tensor $\mathcal{D}$, the choice of μ_1 and μ_2 is analogous to anisotropic edge enhancing diffusion [34]. However, two additional terms occur in the fourth-order case, μ_3 and μ_4. As noted in [15], μ_4 is irrelevant, since the corresponding eigentensor $\mathbf{E}_4$ is anti-symmetric, and the dot product $\mathbf{E}_4 : \mathbf{H}$ with the Hessian of any sufficiently smooth image will be zero due to its symmetry. To better understand the role of μ_3, we observe that

$$\mathbf{E}_3 : \mathbf{H} = \frac{1}{\sqrt{2}} \left(\mathbf{v}_1^{\mathrm{T}} \mathbf{H} \mathbf{v}_2 + \mathbf{v}_2^{\mathrm{T}} \mathbf{H} \mathbf{v}_1 \right)$$

$$= \frac{1}{\sqrt{2}} \left(u_{\left(\frac{\mathbf{v}_1 + \mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1 + \mathbf{v}_2}{\sqrt{2}}\right)} - u_{\left(\frac{\mathbf{v}_1 - \mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1 - \mathbf{v}_2}{\sqrt{2}}\right)} \right), \tag{14}$$

which amounts to a mixed second derivative of u, in directions along and orthogonal to the regularized image gradient $\frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ or, equivalently, to the difference of second derivatives in the directions that are exactly in between the two. This term vanishes if the Hessian is isotropic, or if the gradient is parallel to one of the Hessian eigenvectors. Therefore, the role of μ_3 can be seen as steering the amount of diffusion in cases of a Hessian anisotropy that goes along with a misalignment between gradient and Hessian eigenvectors.

Gorgi Zadeh et al. [15] simply set μ_3 to the arithmetic mean of μ_1 and μ_2. In our work, we empirically evaluated several alternative options for μ_3 by reconstructing the test image shown in Fig. 1a, which contains one rectangle, one circle, and two stars, from a randomly selected subset of 5% of its pixels. In this experiment, we compare EED based inpainting with our novel fourth-order edge enhancing diffusion (FOEED) with different settings of μ_3: Specifically, $\mu_3 = 1$ corresponds to the max-
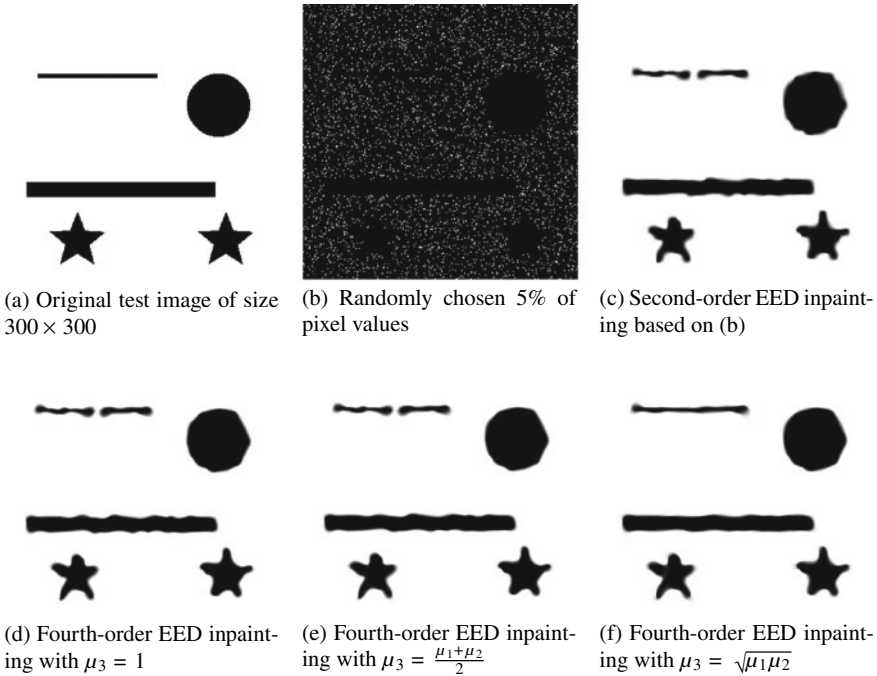

(a) Original test image of size 300×300

(b) Randomly chosen 5% of pixel values

(c) Second-order EED inpainting based on (b)

(d) Fourth-order EED inpainting with $\mu_3 = 1$

(e) Fourth-order EED inpainting with $\mu_3 = \frac{\mu_1 + \mu_2}{2}$

(f) Fourth-order EED inpainting with $\mu_3 = \sqrt{\mu_1 \mu_2}$

Fig. 1 Reconstruction of a synthetic test image (**a**) from 5% of its pixels (**b**) based on second-order diffusion (**c**) and fourth-order diffusion with different coefficients for the mixed term μ_3 (**d–f**). Visually, the reconstruction in (**f**) is most similar to the original image

Table 1 Numerical reconstruction errors on the test image (Fig. 1)

Errors	EED	FOEED ($\mu_3 = 1$)	FOEED ($\mu_3 = \frac{\mu_1+\mu_2}{2}$)	FOEED ($\mu_3 = \sqrt{\mu_1\mu_2}$)
MSE	647.183	660.588	634.321	533.987
AAE	5.043	4.581	4.505	4.140

imum of μ_1 and μ_2 (Fig. 1d), $\mu_3 = (1 + g(\lambda_1))/2$ corresponds to their arithmetic mean (Fig. 1e), and $\mu_3 = \sqrt{g(\lambda_1)}$ corresponds to their geometric mean (Fig. 1f). In all cases, we used the Charbonnier diffusivity (Eq. (6)), which is popular for image compression [13], the same contrast parameter ($\lambda = 0.1$) and smoothing parameter ($\sigma = 1$). The only difference is time step size, where second-order EED permitted a stable step size of 0.25, while a smaller step size of 0.05 was chosen for FOEED. A more detailed theoretical and empirical analysis of stability will be given in Sect. 3.3.

A numerical comparison of the results is given in Table 1. For evaluation, we used the well-known mean squared error (MSE) and average absolute error (AAE) between original and reconstructed images. For two-dimensional gray-valued images u and v with the same dimensions $m \times n$, the MSE and AAE are defined as

$$\text{MSE}(u, v) = \frac{1}{mn} \sum_{i,j} (u_{i,j} - v_{i,j})^2 ,$$

$$\text{AAE}(u, v) = \frac{1}{mn} \sum_{i,j} |u_{i,j} - v_{i,j}| . \tag{15}$$

According to Table 1, the most accurate results are achieved by setting μ_3 to the geometric mean of μ_1 and μ_2. Fourth-order EED with this setting produces higher accuracy than second-order EED. Visually, Fig. 1 supports this conclusion. Specifically, fourth-order EED with $\mu_3 = \sqrt{\mu_1\mu_2}$ is the only variant that correctly connects the thin bar at the top of the test image, and it leads to a straighter shape of the thicker bar below, which is more similar to its original rectangular shape. In all subsequent experiments, we set $\mu_3 = \sqrt{\mu_1\mu_2}$.

3.2 A Unifying Framework for Fourth-Order Diffusion

Several fourth-order diffusion PDEs have been used for image processing previously. We can better understand how they relate to our newly proposed PDE by observing that the fourth-order diffusion tensor $\mathcal{D}$ introduces a unifying framework for fourth-order diffusion filters. In particular, given its coefficients d_{ijkl}, we can expand Eq. (9) by using Einstein notation as

$$\partial_t u = -\partial_{ji}[d_{ijkl}u_{kl}] \tag{16}$$

Effectively, the fourth-order diffusion tensor allows us to separately set the diffusivities for all $2^4 = 16$ fourth-order derivatives of the two-dimensional image u. We will now demonstrate how several well-known fourth-order PDEs can be expressed in this framework, starting with the You-Kaveh PDE [35]

$$\partial_t u = -\Delta(g(|\Delta u|)\Delta u) , \tag{17}$$

which can be rewritten as

$$
\begin{aligned}
\partial_t u = -\partial_{xx}[g(|\Delta u|)u_{xx} + 0 \cdot u_{xy} + 0 \cdot u_{yx} + g(|\Delta u|)u_{yy}] \\
-\partial_{yx}[0 \cdot u_{xx} + 0 \cdot u_{xy} + 0 \cdot u_{yx} + 0 \cdot u_{yy}] \\
-\partial_{xy}[0 \cdot u_{xx} + 0 \cdot u_{xy} + 0 \cdot u_{yx} + 0 \cdot u_{yy}] \\
-\partial_{yy}[g(|\Delta u|)u_{xx} + 0 \cdot u_{xy} + 0 \cdot u_{yx} + g(|\Delta u|)u_{yy}] .
\end{aligned}
\tag{18}
$$

Here and in all subsequent examples, many terms have zero coefficients. For brevity, we will omit them from now on.

Hajiaboli's anisotropic fourth-order PDE [19] is

$$\partial_t u = -\Delta\left(g(||\nabla u||)^2 u_{NN} + g(||\nabla u||)u_{TT}\right) , \tag{19}$$

where N and T are unit vectors parallel and orthogonal to the gradient, respectively. It can be rewritten as

$$
\begin{aligned}
\partial_t u = -\partial_{xx}\Bigg[& \left(\frac{g(||\nabla u||)^2 u_x^2 + g(||\nabla u||)u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2 + u_y^2}\right)u_{xy} \\
& + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)^2 u_y^2 + g(||\nabla u||)u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\Bigg] \\
-\partial_{yy}\Bigg[& \left(\frac{g(||\nabla u||)^2 u_x^2 + g(||\nabla u||)u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2 + u_y^2}\right)u_{xy} \\
& + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)^2 u_y^2 + g(||\nabla u||)u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\Bigg]
\end{aligned}
\tag{20}
$$

From this method, Li et al. [21] derived two anisotropic fourth-order PDEs that, to our knowledge, are the only anisotropic fourth-order models that have been applied to inpainting previously. We will refer to them as Li 1

$$\partial_t u = -\Delta(g(||\nabla u||)u_{NN} + u_{TT}) \tag{21}$$

and Li 2

$$\partial_t u = -\Delta(u_{TT}) . \tag{22}$$

Li 1 can be re-written as

$$\partial_t u = -\partial_{xx}\left[\left(\frac{g(||\nabla u||)u_x^2 + u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)u_x u_y - u_x u_y}{u_x^2 + u_y^2}\right)u_{xy}\right.$$
$$\left. + \left(\frac{g(||\nabla u||)u_x u_y - u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)u_y^2 + u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\right]$$
$$-\partial_{yy}\left[\left(\frac{g(||\nabla u||)u_x^2 + u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)u_x u_y - u_x u_y}{u_x^2 + u_y^2}\right)u_{xy}\right.$$
$$\left. + \left(\frac{g(||\nabla u||)u_x u_y - u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)u_y^2 + u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\right], \tag{23}$$

while Li 2 becomes

$$\partial_t u = -\partial_{xx}\left[\left(\frac{u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{-u_x u_y}{u_x^2 + u_y^2}\right)u_{xy} + \left(\frac{-u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\right]$$
$$-\partial_{yy}\left[\left(\frac{u_y^2}{u_x^2 + u_y^2}\right)u_{xx} + \left(\frac{-u_x u_y}{u_x^2 + u_y^2}\right)u_{xy} + \left(\frac{-u_x u_y}{u_x^2 + u_y^2}\right)u_{yx} + \left(\frac{u_x^2}{u_x^2 + u_y^2}\right)u_{yy}\right]. \tag{24}$$

We observe that Li 1 is based on a similar idea as our proposed PDE: It permits fourth-order diffusion along the edge, while applying a nonlinear diffusivity function across the edge. However, expressing Li et al.'s models in terms of fourth-order diffusion tensors $\mathcal{D}_1$ and $\mathcal{D}_2$ reveals that our approach is more general. In particular, we can observe that

$$\mathcal{D}_1 : \mathbf{H} = g(||\nabla u||)u_{NN}\mathbf{I} + u_{TT}\mathbf{I}\,,$$
$$\mathcal{D}_2 : \mathbf{H} = u_{TT}\mathbf{I}\,, \tag{25}$$

where $\mathbf{I}$ is the 2×2 identity matrix. In our model, $\mathcal{D} : \mathbf{H}$ can yield arbitrary anisotropic tensors. In this sense, our model more fully accounts for anisotropy compared to the ones by Hajiaboli and Li et al.

The fourth-order Eq. (16) involves inner second derivatives of the image, which then get scaled by diffusivities, before outer second derivatives are taken. We observe that, in both cases, our model accounts for mixed derivatives that are ignored by previous approaches to anisotropic fourth-order diffusion: In the outer derivatives, this can be seen from the fact that Eq. (9) involves mixed derivatives, while Hajiaboli and Li et al. only consider the Laplacian.

Similarly, our definition of a fourth-order diffusion tensor $\mathcal{D}$ accounts for mixed derivatives also in the inner derivatives. Following Eq. (14), we obtain

$$\mathcal{D} : \mathbf{H} = \mu_1(\mathbf{E}_1 \otimes \mathbf{E}_1) : \mathbf{H} + \mu_2(\mathbf{E}_2 \otimes \mathbf{E}_2) : \mathbf{H} + \mu_3(\mathbf{E}_3 \otimes \mathbf{E}_3) : \mathbf{H}$$
$$= \mu_1 u_{\mathbf{v}_1 \mathbf{v}_1}\mathbf{E}_1 + \mu_2 u_{\mathbf{v}_2 \mathbf{v}_2}\mathbf{E}_2 + \frac{\mu_3}{\sqrt{2}}\left(u_{\left(\frac{\mathbf{v}_1 + \mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1 + \mathbf{v}_2}{\sqrt{2}}\right)} - u_{\left(\frac{\mathbf{v}_1 - \mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1 - \mathbf{v}_2}{\sqrt{2}}\right)}\right)\mathbf{E}_3\,. \tag{26}$$

Comparing Eqs. (25) and (26) reveals differences in the considered directions: First, N and T are derived from the unregularized gradient, while the corresponding directions $\mathbf{v}_1$ and $\mathbf{v}_2$ in our model include a Gaussian pre-smoothing. A second difference is that our model involves an additional term, which is steered by μ_3, and accounts for the directions in between the regularized gradient and its orthogonal vectors, i.e., $\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)$ and $\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)$. As it was demonstrated in the previous section, this term can have a noticeable effect on the outcome. Overall, we conclude that our newly proposed model is more general than the previously published ones.

3.3 Discretization and Stability

When discretizing Eq. (13) with standard finite differences

$$
\begin{aligned}
u_{xx} &\approx \frac{(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})}{(\Delta x)^2} \,, \\
u_{yy} &\approx \frac{(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})}{(\Delta y)^2} \,, \\
u_{xy} &\approx \frac{(u_{i-1,j-1} + u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1})}{4(\Delta x)(\Delta y)} \,, \\
u_{yx} &= u_{xy} \,,
\end{aligned}
\tag{27}
$$

we can write it down in matrix-vector form as in [15],

$$
\mathbf{u}^{k+1} = \mathbf{u}^k (\mathbf{I} - \tau \, \mathbf{P}_k) \,,
\tag{28}
$$

where u^k is an mn dimensional image vector at iteration k. m, n are image width and height respectively; Δx and Δy are the corresponding pixel edge lengths. $\mathbf{P}_k$ is a positive semi-definite matrix that, with step size τ, leads to the system matrix $(\mathbf{I} - \tau \, \mathbf{P}_k)$. The notation $\mathbf{P}_k$ indicates that it is iteration dependent, i.e., $\mathbf{P}_k = \mathbf{P}(\mathbf{u}_k)$.

Stability of fourth-order PDEs for image processing is typically formalized in an L_2 sense, i.e., a time step is chosen such that

$$
||\mathbf{u}^{k+1}||_2 \leq ||\mathbf{u}^k||_2 \,.
\tag{29}
$$

In an inpainting scenario, it depends on our initialization of the unknown pixels whether we can expect Eq. (29) to hold. Therefore, we rely on a stability analysis of the smoothing variant of our proposed PDE. This variant is obtained by removing the

Dirichlet boundary conditions and instead solving a standard initial value problem. In this case, the stability analysis presented by Gorgi Zadeh et al. [15] carries over. It ensures that time step sizes

$$\tau \leq \frac{2}{16(\Delta x)^2 + 16(\Delta y)^2 + 2(\Delta x \Delta y)} \tag{30}$$

are stable in the L_2 sense. For a spatial discretization $\Delta x = \Delta y = 1$, this yields $\tau \leq 1/17 \approx 0.0588$. In inpainting, we empirically obtained a useful steady state with a time step size $\tau \leq 0.066$, independent of the initialization. The fact that this slightly exceeds the theoretical step size reflects the fact that Eq. (30) results from deriving a sufficient, not a necessary condition for stability.

Stability of fourth-order schemes generally requires a quite small time step τ. This makes it computationally expensive to reach the steady state by evaluating Eq. (28). Hafner et al. [17] propose a remedy to this problem, the so-called Fast Semi-Iterative Scheme (FSI). It extrapolates the basic solver iteration with the previous iterate and serves as an accelerated explicit scheme. The acceleration of the explicit scheme (28) is given as

$$\mathbf{u}^{m,k+1} = \alpha_k \cdot (\mathbf{I} - \tau \mathbf{P}(\mathbf{u}^{m,k})) \mathbf{u}^{m,k} + (1 - \alpha_k) \cdot \mathbf{u}^{m,k-1} , \tag{31}$$

where $\mathbf{u}^{m,-1} := \mathbf{u}^{m,0}$ and $\alpha_k = \frac{4k+2}{2k+3}$ for $k = 0, \ldots, n-1$. Here m stands for outer cycle, i.e. m-th cycle with inner cycle of length n. And for passing to the next outer cycle, we set $\mathbf{u}^{m+1,0} := \mathbf{u}^{m,n}$. The stability analysis requires the matrix $\mathbf{P}$ to be symmetric. This is satisfied since the diffusion tensor $\mathcal{D}$ is symmetric, and symmetric central discretizations are used. In our implementation, we used $n = 40$, and stopped iterating after the first outer cycle for which $\|\mathbf{u}^m - \mathbf{u}^{m-1}\|_2 < 10^{-4}$.

4 Experimental Results

To establish the usefulness of our proposed new model, we applied it to the reconstruction of images from a sparse subset of pixels (Sect. 4.1). Moreover, we evaluate performance for a more classic inpainting task, scratch removal (Sect. 4.2). We also demonstrate how results depend on the chosen diffusivity function and contrast parameter (Sect. 4.3).

4.1 Reconstruction From a Sparse Set of Pixels

Improving image reconstruction from a sparse set of known pixels was the main motivation behind our work. Therefore, we applied it to two well-known natural images, *toucan* and *peppers,* as well as to a medical image, a slice of a T_1 weighted brain MR scan *(t1slice).* For *toucan,* we kept a random subset of only 2% of the pixels. Due to the lower resolution of the *peppers* and *t1slice* images, we kept 5% and 20%, respectively.

In all three cases, results from our approach (FOEED) were compared to results from second-order EED, as well as from the two anisotropic fourth-order PDEs proposed by Li et al. [21]. In all experiments, we used the Charbonnier diffusivity function, we set the contrast parameter to $\lambda = 0.1$, and the pre-smoothing parameter to $\sigma = 1$.

Results for *toucan* are shown in Fig. 2, for *peppers* in Fig. 3, and for *t1slice* in Fig. 4. A quantitative evaluation in terms of MSE and AAE is presented in Table 2. In terms of the numerical results, our proposed method produced a more accurate reconstruction than any of the competing approaches. Visually, there is a clear difference between second-order (EED) and fourth-order approaches (Li1, Li2, FOEED). Especially, we found that the shapes of edges were reconstructed more accurately. For example, we noticed this around the body and face in the *toucan* image (Fig. 2). Similarly, the white and grey matter boundaries were better separated in the *t1slice* (Fig. 4).

As we expected based on the theoretical analysis in Sect. 3.2, visual differences between the fourth-order methods are more subtle. However, in the *peppers* image (Fig. 3), the tall and thin and the small and thick peppers in the foreground are much more clearly separated in the FOEED result than in any of the others.

In addition to experimenting with grayscale versions of the *toucan* and *peppers* images, we also applied EED and our FOEED filter channel-wise to the original RGB color versions. Results for *toucan* can be found in Fig. 5, for *peppers* in Fig. 6. Table 3 again provides a quantitative comparison. Similar observations can be made as in the grayscale images: Again, FOEED leads to lower reconstruction errors than EED, it visually reconstructs edges more accurately, and separates the peppers more clearly.

Finally, we reconstructed images from a larger number of pixels, to obtain visually cleaner results. Qualitative and numerical results are presented in Fig. 7 and Table 4, respectively. FOEED still yields lower numerical errors than EED. Unsurprisingly, the differences become smaller and less visually prominent as the mask density increases. The table also reveals that FOEED requires more CPU time compared to standard EED. However, due to the use of FSI in both cases, the difference in running times until convergence is much lower than the difference in time step sizes.
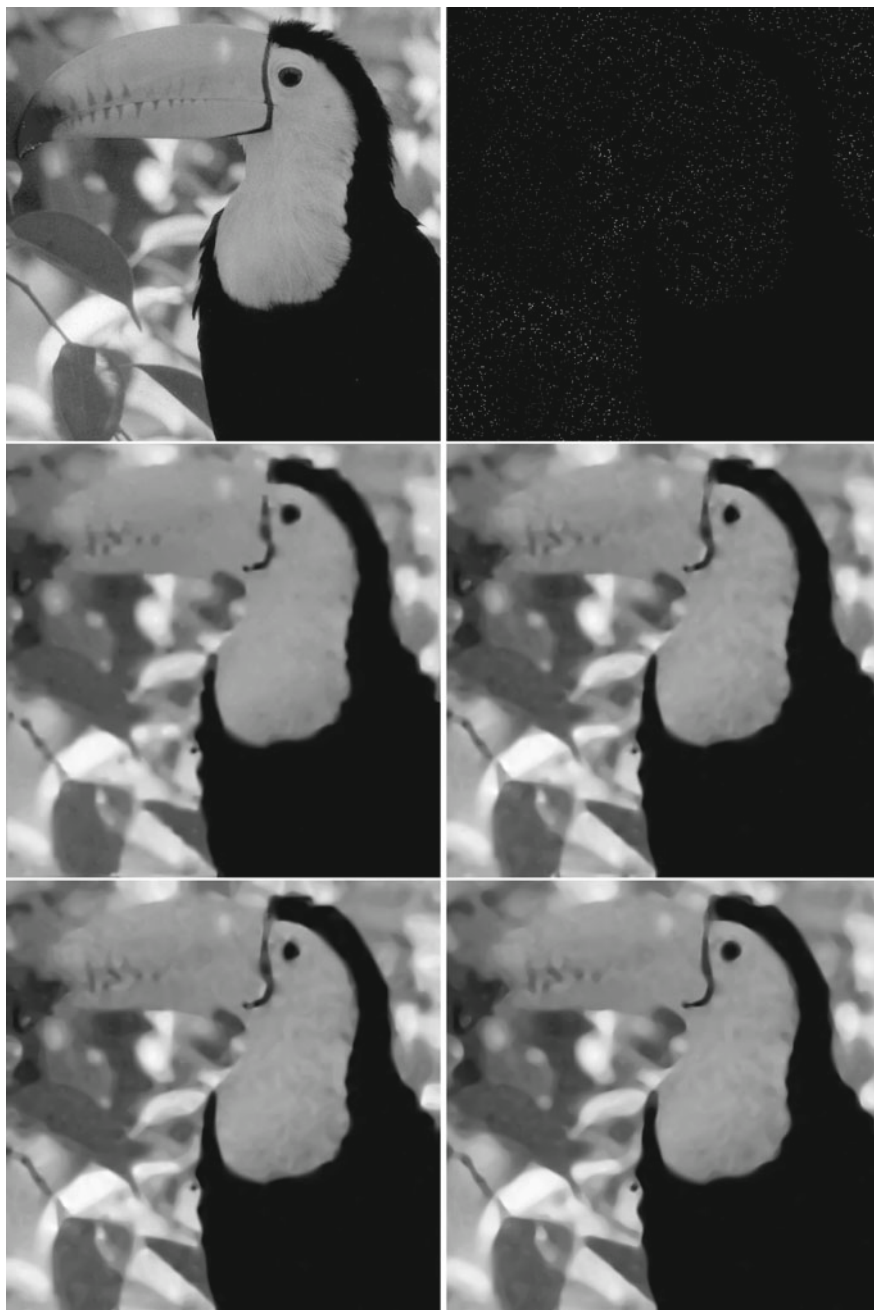
Fig. 2 1st row left: original *toucan* image of size 512 × 512; right: randomly chosen 2% of pixel values; 2nd row left: EED based inpainted image; right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; right: FOEED based inpainted image
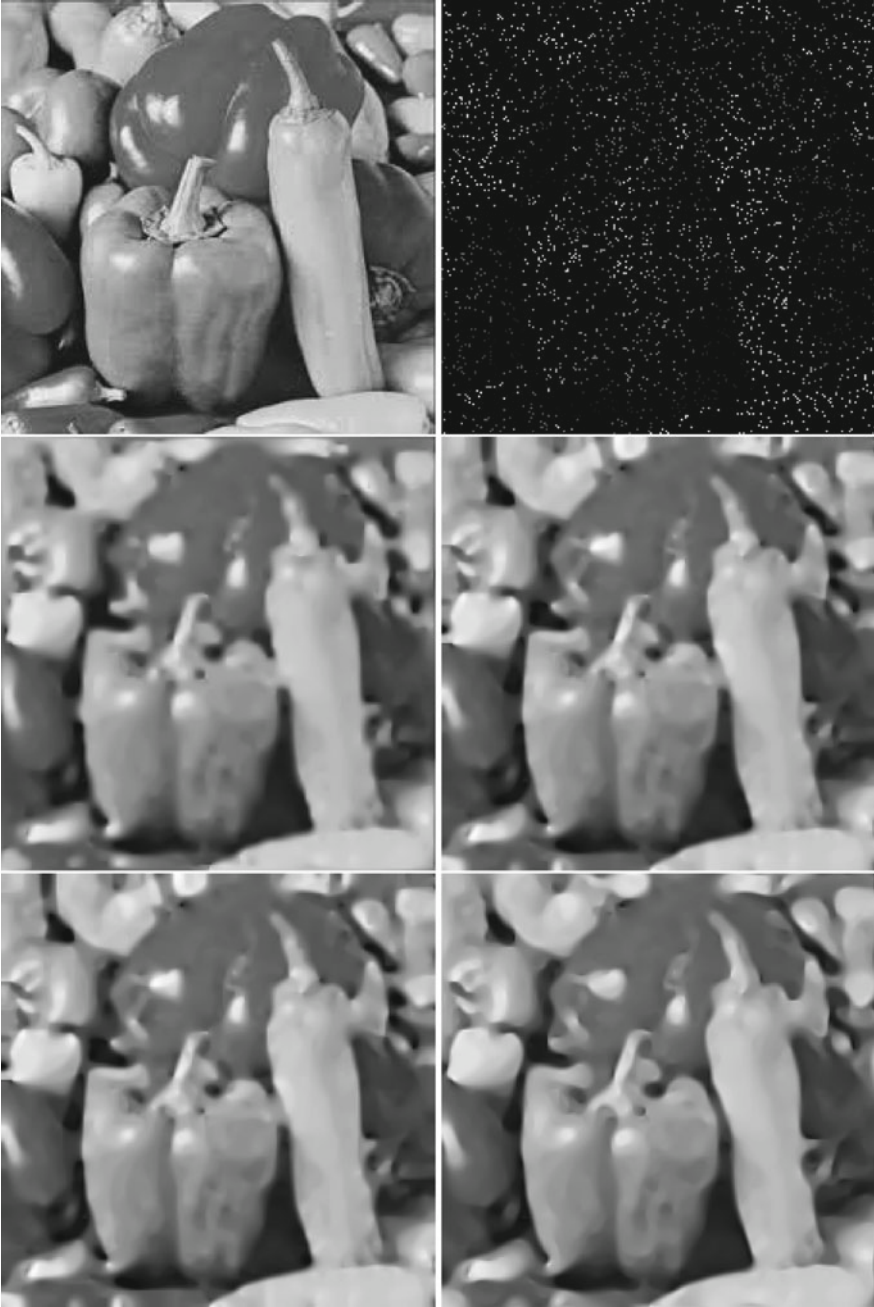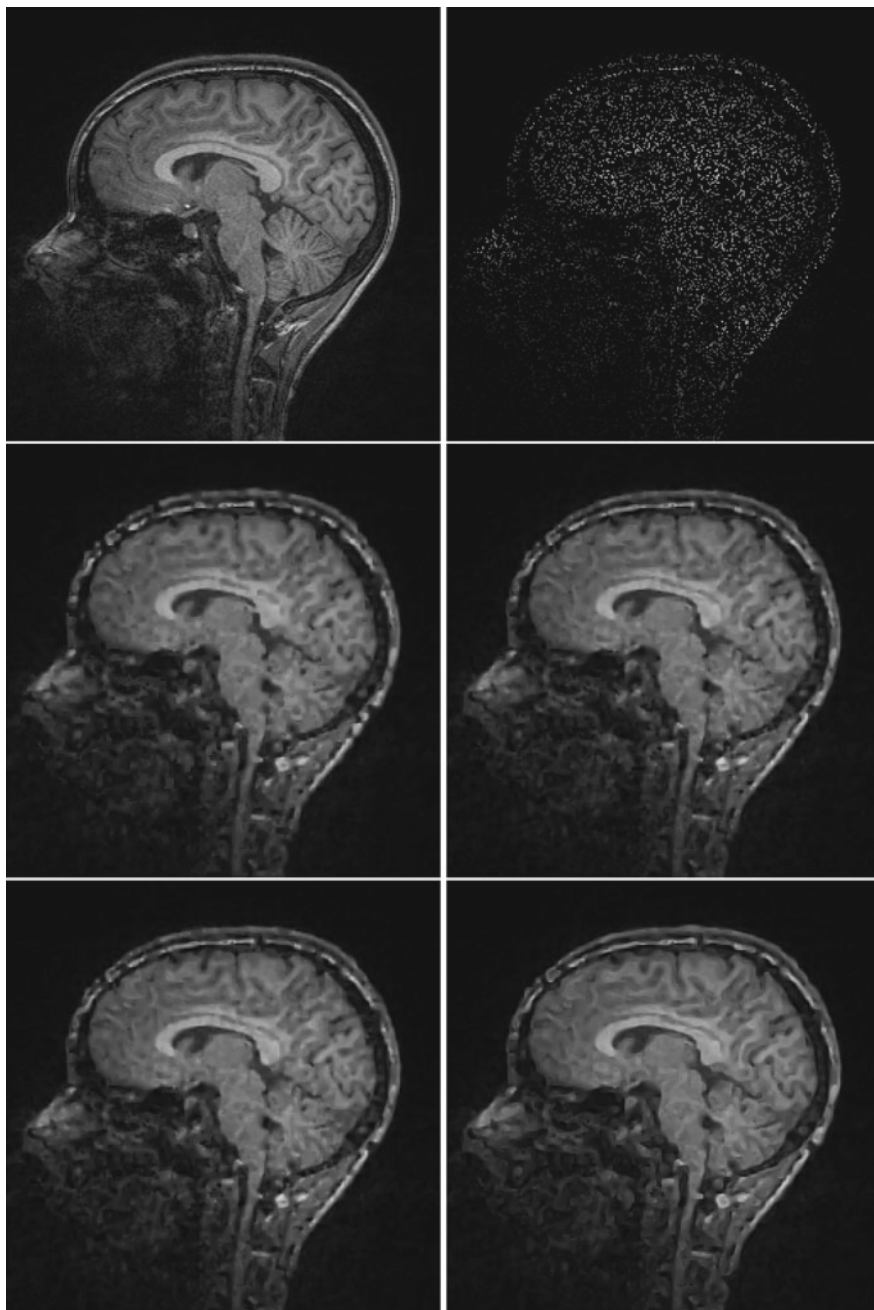
Fig. 3 1st row left: original *peppers* image of size 225 × 225; Right: randomly chosen 5% of pixel values; 2nd row left: EED based inpainted image; Right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; Right: FOEED based inpainted image

Fig. 4 1st row left: original *t1slice* image of size 256 × 256; Right: randomly chosen 20% of pixel values; 2nd row left: EED based inpainted image; Right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; Right: FOEED based inpainted image

Table 2 Numerical comparison of inpainting models for gray-valued images

Image	Errors	EED	FOEED	Li1	Li2
Toucan	MSE	105.37	**96.228**	100.994	102.665
	AAE	4.488	**4.164**	4.397	4.465
Peppers	MSE	467.261	**443.129**	455.633	459.606
	AAE	10.94	**10.523**	11.042	11.107
t1-slice	MSE	166.356	**150.002**	152.698	155.955
	AAE	5.895	**5.698**	5.789	5.853

Fig. 5 RGB *toucan* image, reconstructed from randomly chosen 2% of pixel values using EED (left) or FOEED (right)

Fig. 6 RGB *peppers* image, reconstructed from randomly chosen 5% of pixel values using EED (left) or FOEED (right)

Table 3 Numerical comparison of inpainting models for RGB images

Image	Errors	EED	FOEED
Toucan	MSE	119.062	**108.061**
	AAE	4.819	**4.594**
Peppers	MSE	478.799	**441.203**
	AAE	11.049	**10.543**

Table 4 Numerical comparison and computation times corresponding to Fig. 7

Image	Errors	EED	FOEED	CPU time
Toucan	MSE	18.029	**17.295**	53.060 (FOEED)
	AAE	1.696	**1.686**	21.259 (EED)
Peppers	MSE	113.5	**110.885**	20.999 (FOEED)
	AAE	4.565	**4.441**	19.79 (EED)
t1-slice	MSE	114.845	**107.323**	24.74 (FOEED)
	AAE	4.610	**4.553**	10.64 (EED)

4.2 Scratch Removal

Li et al. [21] proposed their anisotropic fourth-order PDE for more classical image inpainting tasks, such as scratch removal. We evaluated whether our more general filter can also provide a benefit in such a scenario by reconstructing a scratched version of the *peppers* image. Similar to Li et al., we first made the scratches rather thin, covering only 6% of all pixels. Results are shown in Fig. 8 and in Table 5. In this case, all methods work well: Numerical errors are small and similar between methods, and even though FOEED achieves the best numerical result, differences are difficult to discern visually.

Therefore, we created a more challenging version with thicker scratches, covering 18% of all pixels (Fig. 9). The corresponding numerical comparison is shown in Table 6. Here, FOEED achieves the most accurate reconstruction. Visually, we again observe that edges are reconstructed more accurately, and objects are more clearly separated, with fourth-order compared to second-order diffusion, and that steering it with a fourth-order diffusion tensor again provides small additional benefits over the previous methods.

Table 5 Numerical comparison for peppers with thinner scratches (Fig. 8)

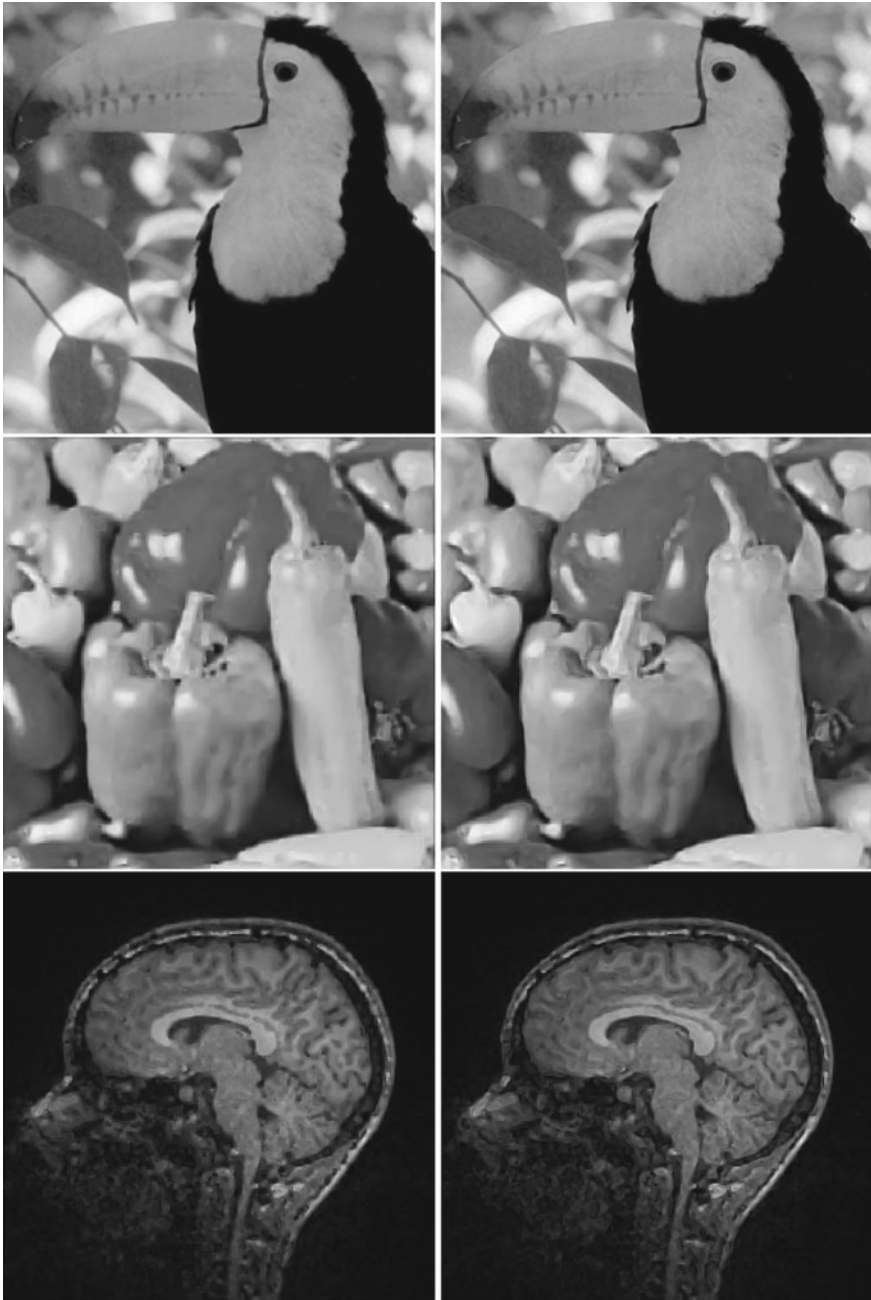| Image	Errors	EED	FOEED	Li1	Li2
Peppers	MSE	9.520	**7.813**	8.161	8.132
	AAE	0.363	**0.326**	0.346	0.346

Fig. 7 Higher quality reconstructions from a larger subset of pixels. 1st row: *toucan* image, reconstructed with EED (left) or FOEED (right) from randomly chosen 14% of pixels; 2nd row left: same for 20% of pixels from *peppers*; 3rd row left: same for 30% of pixels from *t1slice*. As expected, increasing the fraction of known pixels reduces the differences in the results of the two schemes
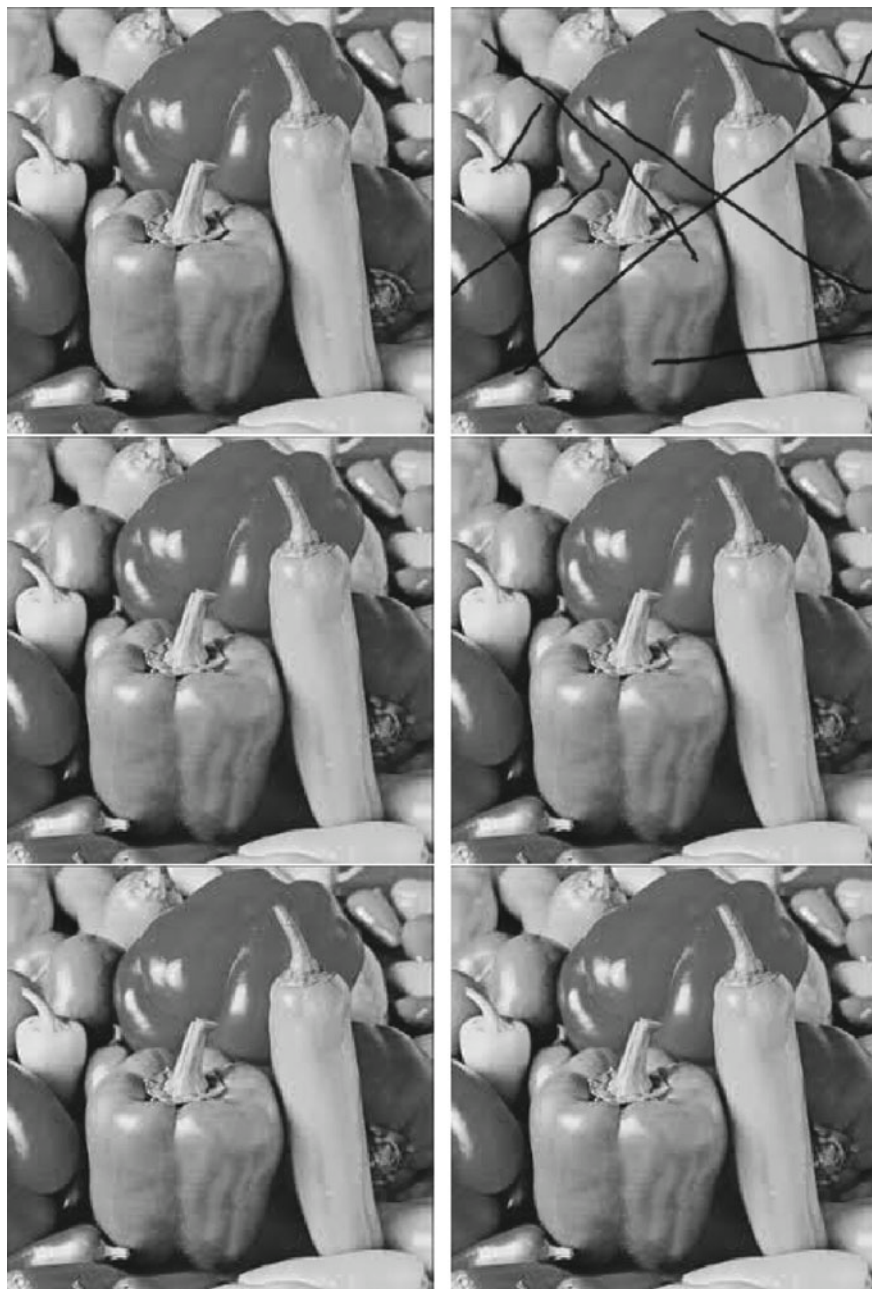
Fig. 8 1st row left: original *peppers* image of size 225×225; Right: corrupted image. 2nd row left: EED based inpainting; Right: Li1 based inpainting. 3rd row left: Li2 based inpainting; Right: FOEED based inpainting

Fig. 9 1st row left: original *peppers* image of size 225 × 225; Right: corrupted image. 2nd row left: EED based inpainting; Right: Li1 based inpainting. 3rd row left: Li2 based inpainting; Right: FOEED based inpainting

Table 6 Numerical comparison for peppers with thicker scratches (Fig. 9)

Image	Errors	EED	FOEED	Li1	Li2
Peppers	MSE	104.744	**78.761**	101.670	101.592
	AAE	2.455	**2.099**	2.465	2.450

4.3 Effect of Diffusivity Function and Contrast Parameter

For image inpainting with second-order PDEs, the Charbonnier diffusivity was previously found to work better than other established diffusivity functions. To assess whether this is still true in the fourth-order case, we repeated the reconstruction of the *peppers* image as shown in Fig. 3 with different diffusivities. Table 7 summarizes the results. We conclude that the Charbonnier diffusivity still appears to be optimal.

Finally, in Fig. 10, we illustrate how the reconstructed image depends on the contrast parameter λ. As expected, increasing λ leads to an increased blurring of edges. In the limit, the diffusivity function takes on values close to 1 over a substantial part of the image, and our model starts to approximate homogeneous fourth-order diffusion.

Table 7 Numerical comparison of FOEED with different diffusivity functions

Image	Errors	Charbonnier [8] $\frac{1}{\sqrt{1+(\frac{s}{\lambda})^2}}$	Aubert [7] $\frac{(\frac{s}{\lambda})^2}{(s^2+\lambda^2)^2}$	Perona-Malik [24] $\frac{1}{1+(\frac{s}{\lambda})^2}$	Perona-Malik2 [24] $e^{-(\frac{s}{\lambda})^2}$	Geman-Reynolds [14] $\frac{2\lambda^2}{(s^2+\lambda^2)^2}$
Peppers	MSE	**443.129**	458.961	478.411	491.153	491.186
	AAE	**10.523**	10.587	10.943	11.157	11.007

Fig. 10 From left to right: FOEED based inpainted image with $\lambda = 0.1$, $\lambda = 0.5$, $\lambda = 15.5$

5 Conclusions

We introduced a novel fourth-order PDE for edge enhancing diffusion (FOEED), steered by a fourth-order diffusion tensor. We implemented it using a fast semi-iterative scheme, and demonstrated that it achieved improved accuracy in several inpainting tasks, including reconstructing images from a small fraction of pixels, or removing scratches.

Our main motivation for using fourth-order diffusion in this context is the increased smoothness of results compared to second-order PDEs [35], which we expected to result in visually more pleasant reconstructions. The model in our current work is still based on a single edge direction at each pixel, extracted via a traditional second-order structure tensor. It is left as a separate research goal for future work to combine this with approaches for the estimation of complex structures such as crossings or bifurcations [1, 29], and with their improved reconstruction, e.g., by operating on the space of positions and orientations [4, 9, 11].

Finally, our current work only considered reconstructions from a random subset of pixels. A practical image compression codec that uses our novel PDE should investigate how it interacts with more sophisticated approaches for selecting and coding inpainting masks [27].

References

1. Aach, T., Mota, C., Stuke, I., Mühlich, M., Barth, E.: Analysis of superimposed oriented patterns. IEEE Trans. Image Process. **15**(12), 3690–3700 (2006)
2. Andris, S., Peter, P., Weickert, J.: A proof-of-concept framework for PDE-based video compression. In: Picture Coding Symposium (PCS), pp. 1–5. IEEE (2016)
3. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: International Conference on Learning Representations (ICLR) (2017)
4. Boscain, U., Chertovskih, R.A., Gauthier, J.P., Remizov, A.: Hypoelliptic diffusion and human vision: a semidiscrete new twist. SIAM J. Imaging Sci. **7**(2), 669–695 (2014)
5. Cagnazzo, M., Poggi, G., Verdoliva, L.: Region-based transform coding of multispectral images. IEEE Trans. Image Process. **16**(12), 2916–2926 (2007)
6. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**(1), 182–193 (1992)
7. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: International Conference on Image Processing, pp. 168–172 (1994)
8. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Trans. Image Process. **6**(2), 298–311 (1997)
9. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. J. Math. Imaging Vis. **24**(3), 307–326 (2006)
10. Didas, S., Weickert, J., Burgeth, B.: Properties of higher order nonlinear diffusion filtering. J. Math. Imaging Vis. **35**(3), 208–226 (2009)

11. Franken, E., Duits, R.: Crossing-preserving coherence-enhancing diffusion on invertible orientation scores. Int. J. Comput. Vis. **85**(3), 253 (2009)
12. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Towards PDE-based image compression. In: International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, pp. 37–48. Springer, Berlin (2005)
13. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. J. Math. Imaging Vis. **31**(2–3), 255–269 (2008)
14. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 367–383 (1992)
15. Gorgi Zadeh, S., Didas, S., Wintergerst, M.W.M., Schultz, T.: Multi-scale anisotropic fourth-order diffusion improves ridge and valley localization. J. Math. Imaging Vis. **59**(2), 257–269 (2017)
16. Greer, J.B., Bertozzi, A.L.: h^1 solutions of a class of fourth order nonlinear equations for image processing. Discret. Continuous Dyn. Syst. **10**, 349–366 (2003)
17. Hafner, D., Ochs, P., Weickert, J., Reißel, M., Grewenig, S.: FSI schemes: Fast semi-iterative solvers for PDEs and optimisation methods. In: German Conference on Pattern Recognition (GCPR), pp. 91–102 (2016)
18. Hajiaboli, M.R.: A self-governing hybrid model for noise removal. In: Pacific-Rim Symposium on Image and Video Technology, pp. 295–305. Springer, Berlin (2009)
19. Hajiaboli, M.R.: An anisotropic fourth-order diffusion filter for image noise removal. Int. J. Comput. Vis. **92**(2), 177–191 (2011)
20. Iijima, T.: Basic theory on the normalization of pattern (in case of typical one-dimensional pattern). Bull. Electro-Tech. Lab. **26**, 368–388 (1962)
21. Li, P., Li, S.J., Yao, Z.A., Zhang, Z.J.: Two anisotropic fourth-order partial differential equations for image inpainting. IET Image Process. **7**(3), 260–269 (2013)
22. Lysaker, M., Lundervold, A., Tai, X.C.: Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. IEEE Trans. Image Process. **12**(12), 1579–1590 (2003)
23. Mainberger, M., Weickert, J.: Edge-based image compression with homogeneous diffusion. In: International Conference on Computer Analysis of Images and Patterns, pp. 476–483 (2009)
24. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
25. Peter, P.: Three-dimensional data compression with anisotropic diffusion. In: German Conference on Pattern Recognition, pp. 231–236. Springer, Berlin (2013)
26. Scherzer, O.: Denoising with higher order derivatives of bounded variation and an application to parameter estimation. Computing **60**(1), 1–27 (1998)
27. Schmaltz, C., Peter, P., Mainberger, M., Ebel, F., Weickert, J., Bruhn, A.: Understanding, optimising, and extending data compression with anisotropic diffusion. Int. J. Comput. Vis. **108**(3), 222–240 (2014)
28. Schmaltz, C., Weickert, J., Bruhn, A.: Beating the quality of JPEG 2000 with anisotropic diffusion. In: Joint Pattern Recognition Symposium, pp. 452–461. Springer, Berlin (2009)
29. Schultz, T., Weickert, J., Seidel, H.P.: A higher-order structure tensor. In: Laidlaw, D.H., Weickert, J. (eds.) Visualization and Processing of Tensor Fields—Advances and Perspectives, pp. 263–280. Springer, Berlin (2009)
30. Thayammal, S., Selvathi, D.: A review on segmentation based image compression techniques. J. Eng. Sci. Technol. Rev. **6**(3) (2013)
31. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: International Conference on Learning Representations (ICLR) (2017)
32. Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. In: Bengio, Y., LeCun, Y. (eds.) International Conference on Learning Representations (ICLR) (2016)
33. Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5306–5314 (2017)

34. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner Stuttgart (1998)
35. You, Y.L., Kaveh, M.: Fourth-order partial differential equations for noise removal. IEEE Trans. Image Process. **9**(10), 1723–1730 (2000)
36. Zhang, F., Chen, Y., Xiao, Z., Geng, L., Wu, J., Feng, T., Liu, P., Tan, Y., Wang, J.: Partial differential equation inpainting method based on image characteristics. In: International Conference on Image and Graphics, pp. 11–19. Springer, Berlin (2015)