

# Mapping-Based Image Diffusion

Freddie Åström, Michael Felsberg and George Baravdish

## Journal Article



N.B.: When citing this work, cite the original article.

This is a copy of the original publication which is available at [www.springerlink.com](http://www.springerlink.com):

Freddie Åström, Michael Felsberg and George Baravdish, Mapping-Based Image Diffusion, Journal of Mathematical Imaging and Vision, 2017. 57(3), pp.293-323.

<http://dx.doi.org/10.1007/s10851-016-0672-6>

Copyright: Springer Verlag (Germany)

<http://www.springerlink.com/?MUD=MP>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-136620>

# Mapping-based Image Diffusion

Freddie Åström · Michael Felsberg · George Baravdish

The final publication is available at Springer via  
<http://dx.doi.org/10.1007/s10851-016-0672-6>

Received: date / Accepted: date

**Abstract** In this work, we introduce a novel tensor-based functional for targeted image enhancement and denoising. Via explicit regularization, our formulation incorporates application dependent and contextual information using first principles. Few works in literature treat variational models that describe both application dependent information and contextual knowledge of the denoising problem. We prove the existence of a minimizer and present results on tensor symmetry constraints, convexity, and geometric interpretation of the proposed functional. We show that our framework excels in applications where nonlinear functions are present such as in gamma correction and targeted value range filtering. We also study general denoising performance where we show comparable results to dedicated PDE-based state of the art methods.

**Keywords** image enhancement · denoising · PDE · diffusion · gradient energy tensor · structure tensor

---

F. Åström,  
Heidelberg University  
Heidelberg Collaboratory for Image Processing (HCI)  
Im Neuenheimer Feld 205  
69 120 Heidelberg, Germany  
Tel.: +49 (0) 662 1541 4841  
E-mail: [freddie.astroem@iwr.uni-heidelberg.de](mailto:freddie.astroem@iwr.uni-heidelberg.de)

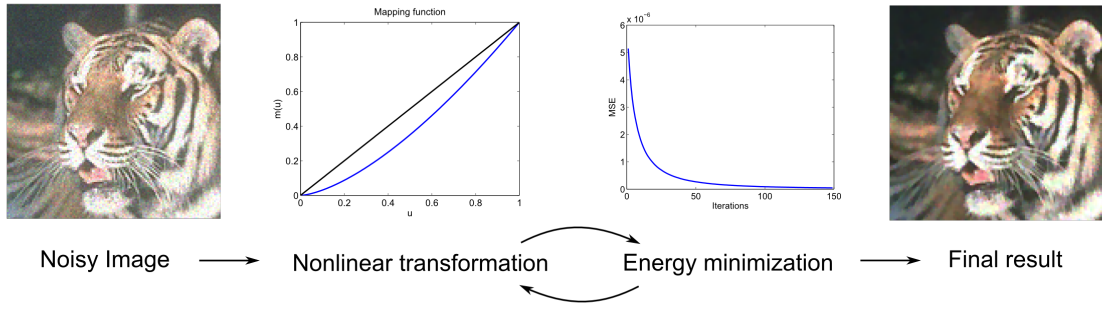
M. Felsberg  
Linköping University  
Department of Electrical Engineering  
581 83 Linköping, Sweden  
Tel.: +46 (0) 13 28 2460  
E-mail: [michael.felsberg@liu.se](mailto:michael.felsberg@liu.se)

G. Baravdish  
Linköping University  
Department of Science and Technology  
581 83 Linköping, Sweden  
Tel.: +46 (0) 11 36 3118  
E-mail: [george.baravdish@liu.se](mailto:george.baravdish@liu.se)

## 1 Introduction

A simple approach to noise reduction is to apply an isotropic convolution kernel to the image data, *e.g.*, normalized Gaussian filters or box filters [39]. Due to the simplicity of the approach, spatial and contextual information are neglected and therefore the output image is blurred. A common approach to adaptive filtering is to let the image gradient control the amount of filtering see, *e.g.*, the seminal work of Perona and Malik (P-M) [48]. They introduced an edge-stopping function to limit the filtering close to edges and corners. The P-M filter is often referred to as a nonlinear isotropic diffusion filter [56]. The filter's edge-stopping function acts to preserve image structure, thereby obtaining superior image quality compared to isotropic filtering. The successor to P-M filtering was introduced by Weickert [56] where additional adaptivity to image structure is achieved by using a second-order tensor. The main idea of the tensor-based formulation is to smooth parallel to image structures. The concept is based on the structure tensor [10,29]. The tensor is a windowed second moment matrix that describes the local orientation as a tensor field. Transforming the tensor field to align parallel to the image structure, one obtains a structure-preserving filtering scheme. The transformed tensor field is then used in the filtering scheme and is denoted as the diffusion tensor.

The linear diffusion scheme is the solution of the diffusion equation, a partial differential equation (PDE), and it is closely related to the notion of scale-space [45, 39]. Therefore it has been of interest to investigate also P-M formulation and its successors of adaptive image filtering in terms of a variational framework. The motivation is that the variational framework allows for a generic approach to model the denoising problem and



**Fig. 1** The presented framework studies image denoising from the viewpoint that the image data is not only corrupted by “noise” but also perturbed by some nonlinear function. In this figure, the nonlinear function corrects for the overexposed image. We incorporate this nonlinear function called a “mapping” as a feedback component in the iterative update scheme. For additional examples of the mapping function see Figure 2.

related image processing applications. By modifying the objective function this enables the inclusion of existing applications, *e.g.*, active contours [38], deblurring [22, 47], optical flow [34, 17, 36], and inpainting [7, 21]. Denote by  $u : \mathbb{R}^2 \rightarrow \mathbb{R}$  the image data. The variational approach to image diffusion is to model an energy functional,  $E$ , of the form

$$E(u) = F(u) + \lambda R(u), \quad (1)$$

where  $F$  is a fidelity term and the positive constant  $\lambda$  determines the influence of  $R$ , the regularization term. The aim is to find the minimizer of the energy functional  $E$ . Here we are interested in tensor-based formulations of the regularization term. In particular we study and introduce an application dependent, tensor-driven anisotropic diffusion scheme. Furthermore, we introduce the gradient energy tensor [28] as an alternative to the structure tensor [29].

### 1.1 Approach

Let  $\Omega$  be a two-dimensional domain, denoting the support of the image  $u$  in pixels. Assume  $u$  to be differentiable and denote the gradient operator as  $\nabla = (\partial_x, \partial_y)^\top$ , where  $^\top$  is the transpose. The gradient of  $u$  is defined as  $\nabla u = (\partial_x u, \partial_y u)^\top$ . Throughout this work we introduce a *mapping function*. We let this mapping be a differentiable nonlinear function  $m : \mathbb{R} \rightarrow \mathbb{R}$ , which describes a mapping of the *image value range*,  $u$ . For notational convenience we write  $\nabla m(u)$  rather than  $\nabla(m(u(x, y)))$  where  $(x, y) \in \Omega$  to express the differentiation of the mapping w.r.t. its argument. Given a noisy image, the mapping function can be estimated or defined from prior knowledge of the enhancement problem. Figure 1 illustrates the application of gamma correction. By letting the mapping function be an (estimated) prior of the nonlinear transform we obtain a

feedback component in the optimization scheme tailored for the specific application. This mapping is not only capable of preserving spatial features (*e.g.*, corners, lines) but also preserves (or smooth) image data in certain image intensity ranges, *i.e.*, the filter also depends on the image value range.

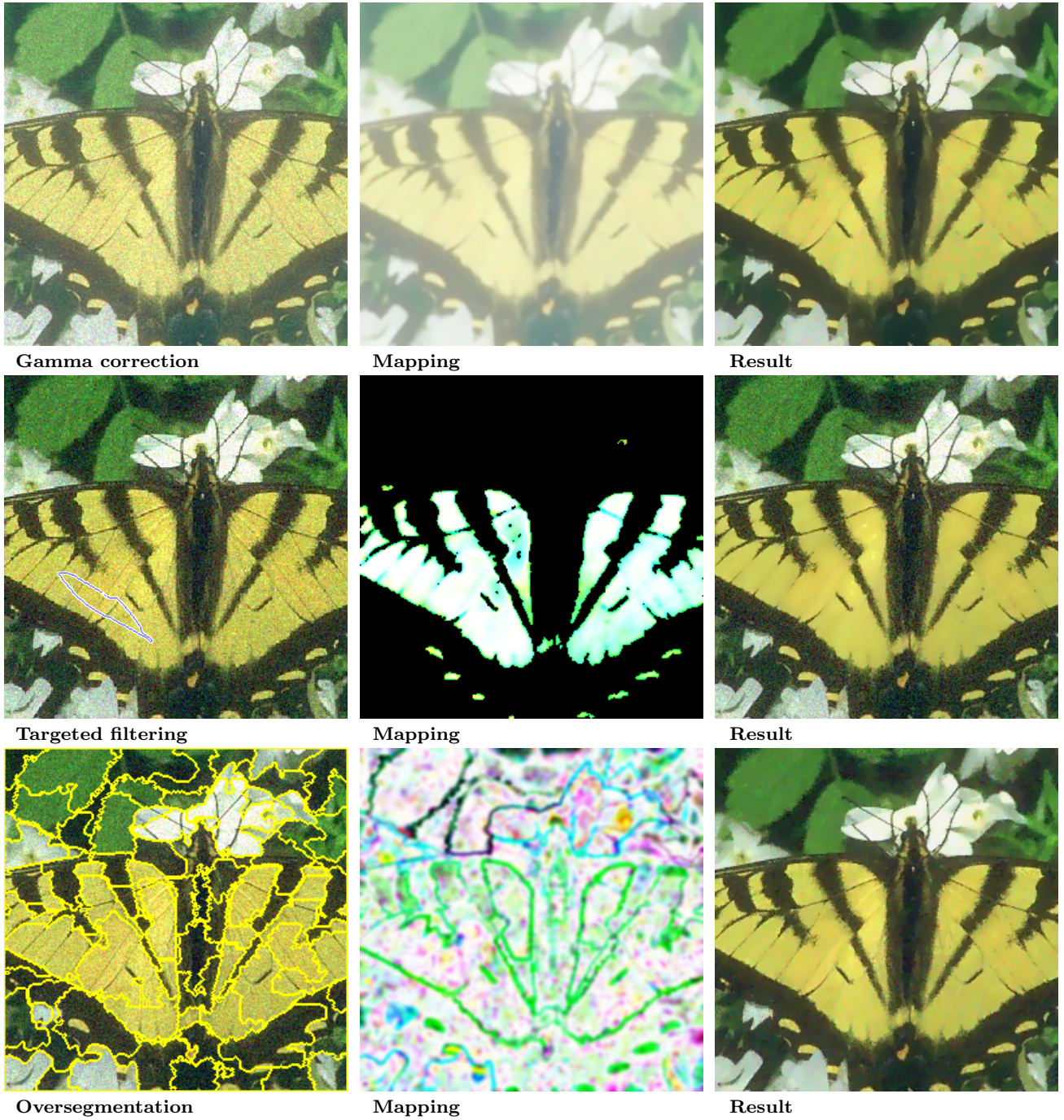
The mapping can be either *global* or *local* depending on the application. We consider a *global* gamma mapping (Section 7), *local* targeted region filtering (Section 8), and global region filtering via *local* oversegmentation maps (Section 9). A preview of these applications is given in Figure 2 and Table 1 shows the corresponding interpretations of the mapping function. We emphasize that the determination of a relevant mapping function is highly application dependent. For readers familiar with anisotropic diffusion [56, 48] we remark that the selection of mapping function can be thought as the accurate selection of a diffusivity function. However, while the diffusivity function is gradient and data dependent, the mapping function is intensity and application dependent. In connection to the variational model (1) we study regularization terms of the form

$$R(u) = \int_{\Omega} \nabla m(u)^\top W(\nabla m(u)) \nabla m(u) d\mathbf{x}, \quad (2)$$

where  $\mathbf{x} = (x, y)$  and  $W \in \mathbb{R}^{2 \times 2}$  is a positive semi-definite tensor that incorporates orientation dependent information.

### 1.2 Synthetic example

Figure 3 illustrates the advantage when performing noise reduction with targeted filtering. In this image, the noise component is only present in the background of the image. The foreground, or object, consisting of circles of different diameters and intensities are not corrupted by noise. Based on this prior knowledge we set,



**Fig. 2** In this work we introduce a mapping function in the energy (2). We consider three applications, each with different definitions of the mapping. First we perform gamma correction, illustrated on the first row. In the second application (second row) we perform targeted filtering. In the third application we perform general denoising by estimating the mapping from the image data via an oversegmentation map.

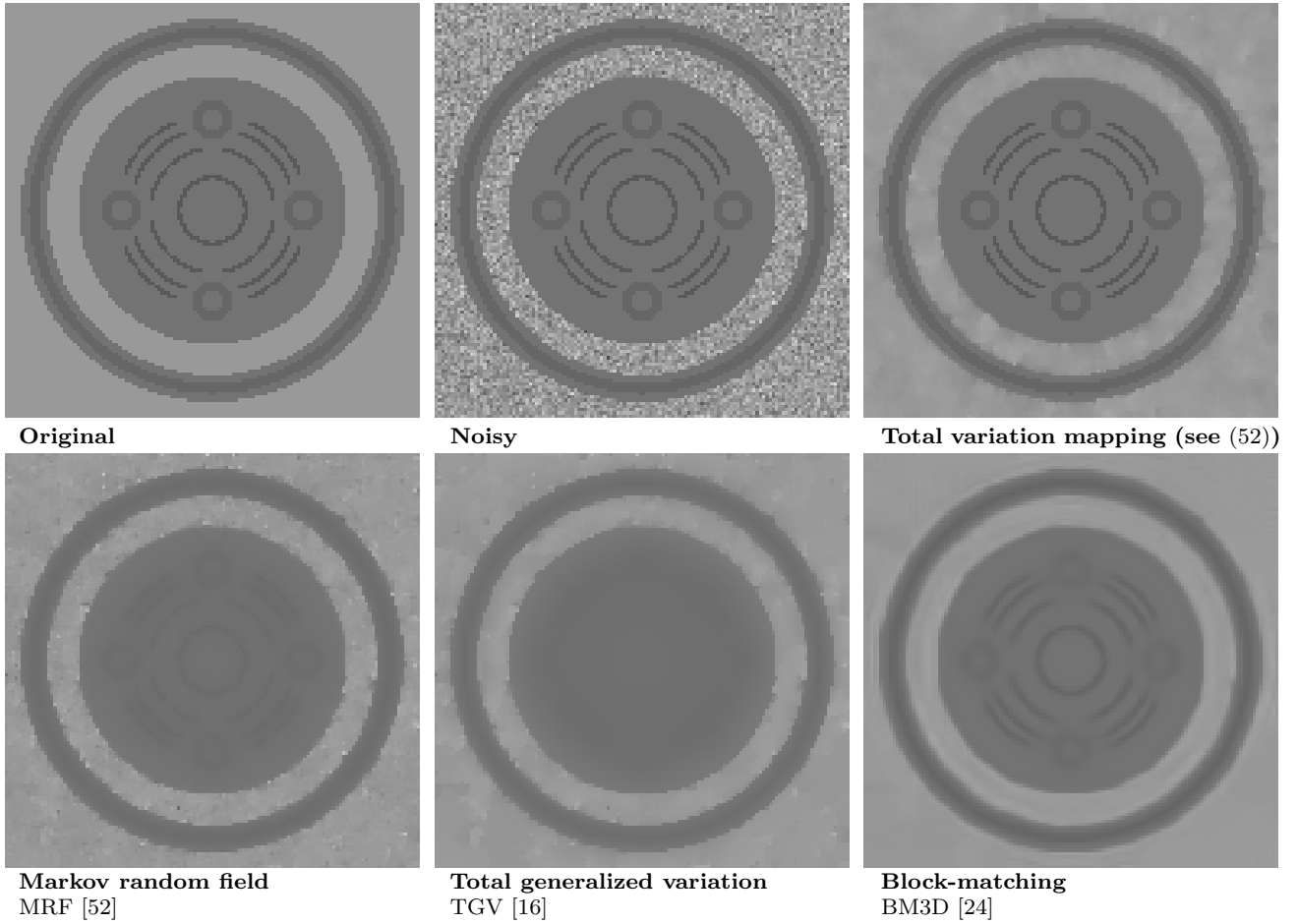
in this example, the mapping as to be sigmoidal, *i.e.*

$$m'(u; a, c) = (1 + \exp(-a(u - c)))^{-1}, \quad a, c > 0, \quad (3)$$

The parameters were set to  $a = 100$  and  $c = 1/2$  as  $u \in [0, 1]$ , these parameters correspond to a smooth approximation of a step-function translated to  $1/2$ . As an

instance of our framework, we use total variation with a mapping function (introduced in Section 4). Visually, comparing the result to Markov Random Field (MRF) [52], Total Generalized Variation (TGV) [16] and BM3D [24] our approach appears more similar to the noise free image. MRF, TGV and BM3D all oversmooth the four





**Fig. 3** Synthetic example where an image has been corrupted with additive Gaussian noise with standard deviation 20 in a specific value range. By appropriately defining the mapping in (2) we are able to preserve image details oversmoothed by MRF, TGV and BM3D. In this example the mapping function was of sigmoidal type, (3) with  $a = 100$  and  $c = 1/2$ .

small circles and details. We note however that in this simple example, assuming that the input image can be accurately partitioned in noise-free and noisy regions, one can naturally apply a restoration method on these partitions. However, in the case of more complex image regions, see Figure 2 second and third row, such heuristic partitioning of the image value range is in general a highly non-trivial task. Furthermore, by incorporating the mapping into the filtering process, one does not require an additional heuristic post-processing step to accurately fuse these independent regions.

### 1.3 Overview and summary of contributions

The main contribution of this work is to study the effects of the mapping function in the tensor-based regularization term (2). The corresponding minimizer is presented in Theorem 1. We show in the experimental part that the inclusion of the nonlinear mapping, as a component of the filtering scheme, improves the fil-

tering performance compared to applying the mapping as a separate pre- or post-processing step. The denoising applications we consider are gamma correction of the intensity range, targeted filtering, and adaptive denoising where the image value distributions are used to drive the filtering process. Furthermore, we introduce the gradient energy tensor (GET) [28] as an orientation dependent term into the regularization term. Our formulation allows us to utilize both the *eigenvalues* and *eigenvectors* of the tensor.

In Section 2 we give an overview of denoising approaches related to the current work. Section 3 further details the diffusion framework and introduces some additional notation. In Section 4 we present the main results of this paper, summarized in Theorem 1. To minimize the proposed functional we prove the existence of its Euler-Lagrange (E-L) equation. Furthermore, we derive properties in the subsequent analysis useful for image denoising applications. We show that the proposed functional describes several variational denoising

## Applications

### I. Gamma correction

*Global mapping applied pixel-wise*

### II. Targeted filtering

*Local mapping applied globally (on the value range)*

### III. Oversegmentation

*Local mappings applied locally (within one segment)*

**Table 1** Overview of the different applications considered in this work and their relation to local and global mappings. A discussion on the difference between these two classes of mappings is given in Section 10. Figure 2 shows examples of the different applications. For additional details, we refer to the respective section in the experimental evaluation.

methods as special cases and we extend these to include the mapping function. In particular we extend our gradient energy total variation formulation [2] to also include the mapping function. In the same section we also give some additional properties of the energy and its E-L equation. Section 5 briefly reviews the GET tensor and presents an empirical evaluation where the tensor is compared to the structure tensor for orientation estimation. The GET tensor is used to formulate the gradient energy total variation scheme with a mapping function in Section 6.

In the experimental evaluation, Section 7, we show empirically that providing the nonlinear mapping of the image data as a feedback component yields an improved result compared to (I) transforming the image value range and then denoise, or (II) denoise and then transform the image value range. In Section 8 we let the mapping define a user-set value range for targeted filtering. Section 9, we evaluate our framework on a third application where we let the filtering process be based on an oversegmentation map. Section 10 gives an extensive discussion on the difference between local and global mapping functions and their applicability in the context of image denoising. A preview of the above mentioned applications and their relation to the different classes of mappings are given in Table 1. Recall that the mappings are nonlinear. Section 11 concludes this work.

## 2 Related works

### 2.1 Variational methods

The connection between energy functionals and the corresponding E-L equation for image processing formulations have been investigated before, we refer to *e.g.*, [6] and references therein. The total variation (TV) formulation was introduced by Rudin et al. [50]. The

TV-model is well studied, both theoretically and practically, see [19] for an overview of the method and its application areas. One major draw-back with the TV-approach for image denoising applications is its tendency to show staircasing effects. The total generalized variation (TGV) introduced by Bredies et al. [16], which includes higher order differentials, does not suffer from these artifacts. The drawback of the approach is that the introduced smoothness of the higher-order derivatives may cause image features to be oversmoothed. Other relevant extensions of the TV model to include additional smoothness constraints are, *e.g.*, [12, 23, 13].

The extension of TV to variational tensor-based formulations was investigated by Roussos and Maragos [49], Lefkimmiatis et al. [43] and Grasmair and Lenzen [33]. These approaches consider the structure tensor [10, 29] and model the objective functions in terms of the tensor eigenvalues. Roussos and Maragos define their regularization term as  $R(u) = \int_{\Omega} \psi(\mu_1, \mu_2) d\mathbf{x}$ , where  $\psi$  is a strict convex function and  $\mu_1, \mu_2$  are the eigenvalues of the structure tensor, thus the regularization is only indirectly taking into account the image structure as it ignores the eigenvectors. Similar to Roussos and Maragos, Lefkimmiatis et al. consider the Schatten-norm of the structure tensor eigenvalues. In contrast, Grasmair and Lenzen define the regularization  $R(u) = \int_{\Omega} \sqrt{\nabla u^T A(u) \nabla u} d\mathbf{x}$ , where  $A$  is the positive semidefinite structure tensor with remapped eigenvalues. The objective function is then solved using a finite element method instead of deriving a variational solution. An alternative approach was proposed by Krajsek and Scharr [40] who formulated a linear anisotropic regularization term for tensor-valued image diffusion. The common factor by the aforementioned methods is that they use the structure tensor. Due to the fact that the structure tensor is defined as the integration of the outer product of the image gradient, the divergence theorem is not applicable when deriving the minimizer of the corresponding functional. We show this fact in Section 3.3, Proposition 1. To our knowledge, the only tensor-based total variation approach that also considered the eigenvectors, has been studied in [2], where the lesser known gradient energy tensor (GET) is used instead of the structure tensor.

In our previous work [4] we proposed to filter certain value ranges of the image data, *i.e.*, an image value range adaptive filter. In a related work, [5], we interpret the image data as observations from a stochastic process and use an oversegmentation of the input image to define the mapping function. The basic approach is to describe each locally homogeneous region using first and second statistical moments to drive a nonlinear filtering process. In this work we combine the mapping

function [4,5] and the tensor-based method [2], leading to novel results.

## 2.2 Block-matching methods

In addition to local, *e.g.*, tensor-based, methods, a different paradigm in the denoising literature has emerged called non-local methods. These methods rely on block-matching of neighbourhoods and have been shown to efficiently suppress image noise. The basic idea is to average similar patches in a local neighbourhood, where similarity is measured with respect to some criteria. The denoising methods block-matching by sparse 3-D transform (BM3D) [24] and non-local means (NLM) [18] are two such methods. NLM and BM3D are fundamentally similar since both use collaborative filtering. Rather than computing new pixel-centres from matched neighbourhoods as in NLM, BM3D considers patches and adaptively denoises a group of local neighbourhoods. In this context we mention Levin et al. [44] who studied image denoising quality w.r.t. patch-size and patch-complexity in view of natural image statistics, they conclude: while the restoration of homogeneous regions is straightforward by increasing the sample size, an increase of patch-size for textured regions has limited impact on the restoration quality as pixels in these regions are weakly correlated. Since, BM3D combines block-matching, linear transform thresholding and Wiener filtering, the noise can only be suppressed efficiently given that sufficiently many similar patches are found. Therefore, as a consequence of inaccurate template matching criteria, oversmoothing may occur in highly textured regions. We compare our framework with BM3D in the experimental evaluation.

In contrast to patch-matching methods, we present an altogether different view-point to image denoising by focusing on adaptive value range filtering and region specific noise removal. Application areas where a mapping function has shown beneficial are visualization of medical images [4] and colour-image denoising [5]. In the present work, we further study this approach by considering a wide range of applications: denoising overexposed images, targeted filtering and denoising of natural images. The potential improvement of image quality and interesting mathematical exposition motivates the subsequent presentation of a theoretical framework. However, before proceeding to the main results, we summarize the background on standard scalar- and tensor valued diffusion methods.

## 3 Review of image diffusion methods

To make this work self-contained we briefly review concepts from calculus of variations and its role in deriving standard diffusion methods. Moreover, we introduce some additional notation and terminology that is used in the remainder of the paper.

### 3.1 Concepts of image diffusion

The classical approach to image diffusion is to consider functionals, or energy models, consisting of a fidelity term,  $F$ , and a regularization term,  $R$ , see (1). Moreover, the energy often takes the form

$$E(u) = \frac{1}{2} \int_{\Omega} (u - u^0)^2 d\mathbf{x} + \lambda R(u), \quad (4)$$

where  $\mathbf{x} \in \Omega$  is the image domain,  $u^0$  is the observed (noisy) image. The constant  $\lambda$  is a positive scalar and determines the effect of the regularization. One well-studied choice of prior is the  $p$ -Dirichlet energy

$$R(u) = \int_{\Omega} \frac{1}{p} |\nabla u|^p d\mathbf{x}, \quad (5)$$

where  $p = 1$  corresponds to Total Variation [50] and  $p > 1$  has been studied in. *e.g.*, [41,8]. To minimize  $E$ , assuming it is convex, one finds the stationary point by solving the corresponding Euler-Lagrange (E-L) equation

$$\delta E(u) = 0, \quad (6)$$

and

$$\delta E(u)v = \lim_{\varepsilon \rightarrow 0} \frac{\partial}{\partial \varepsilon} E(u + \varepsilon v) \quad \text{in } \Omega. \quad (7)$$

With homogeneous Neumann boundary condition,  $\nabla u \cdot \mathbf{n} = 0$ , the E-L equation of (4), (5) is the following boundary value problem

$$\begin{cases} u - u^0 - \lambda \operatorname{div}(|\nabla u|^{p-2} \nabla u) = 0 & \text{in } \Omega \\ \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \quad (8)$$

where  $\mathbf{n}$  is the normal vector on the boundary  $\partial\Omega$ . If  $p = 2$ , the E-L equation describes linear (isotropic) diffusion [39] and if  $p = 1$  it describes the formal TV-formulation of Rudin et al. [50]. The  $u$  that minimizes (4), (5) is obtained by solving (8). The corresponding initial value problem is then

$$\begin{cases} \partial_t u - \operatorname{div}(|\nabla u|^{p-2} \nabla u) = 0 & \text{in } \Omega \\ \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \\ u(0) = u^0 & \text{on } \Omega, \end{cases} \quad (9)$$

and results in the diffusion equation [51,25,42]. In the case of isotropic filtering, the diffusion equation has a closed form solution.

### 3.2 Nonlinear diffusion

Isotropic filtering does not preserve characteristic image features such as edges and corners due to the rotational symmetry of the Laplacian operator. Therefore, when solving the E-L equation in (8), the image is blurred. Introducing a diffusivity function, also known as an “edge-stopping” function  $g : \mathbb{R} \rightarrow \mathbb{R}^+$  in (8), Perona and Malik [48] formulated the PDE

$$\begin{cases} u - u^0 - \lambda \operatorname{div}(g(|\nabla u|)\nabla u) = 0 & \text{in } \Omega \\ \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (10)$$

The function  $g(s)$ , with  $s = |\nabla u|$ , acts to reduce the filtering if the image gradient takes large values. The function’s principal behaviour is  $g(s) \rightarrow 1$  as  $s \rightarrow 0$  and  $g(s) \rightarrow 0$  as  $s \rightarrow \infty$  and a common choice is the negative exponential function

$$g(s) = \exp(-s^2/k^2), \quad (11)$$

where  $k > 0$  is an edge-stopping parameter. Although this formulation resolves many drawbacks of linear image diffusion, noise is preserved along edges since the filtering is simply “stopped” for large image gradients. An alternative to using the negative exponential function as the diffusivity function is to let the problem define which edge-stopping function is suitable. In the work on targeted iterative filtering [4] we let the mapping be defined by a user selected sigmoidal function, that compresses the value range of 12-bit image data for the purpose of medical visualization. The regularization term introduced in [4] is defined as

$$R(u) = \int_{\Omega} |\nabla m(u)|^2 d\mathbf{x}, \quad (12)$$

where the mapping  $m : \mathbb{R} \rightarrow \mathbb{R}$  and  $m \in \mathcal{C}^2(\Omega)$  is a sigmoidal function. Furthermore, note that  $\nabla m(u) = m'(u)\nabla u$ . In this case the mapping defines the visualization windows, thus replaces the ad-hoc selection of a potential function, *e.g.*, (11). The corresponding PDE equation is given by

$$u - u^0 - \lambda m'(u) \operatorname{div}(m'(u)\nabla u) = 0 \quad \text{in } \Omega. \quad (13)$$

Since  $u^0 \in L^2(\Omega)$  and  $m \in \mathcal{C}^2(\Omega)$ , it follows that  $u \in W^{1,2}(\Omega)$ , where  $W^{1,2}(\Omega)$  is the Sobolev space equipped with the norm  $\|u\|_{W^{1,2}(\Omega)}^2 = \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2$  (see, *e.g.*, [31]). Depending on the definition of  $m'$  the function can be understood as the diffusivity function  $g$  in (11). Note that the PDE (13) is equivalent to the formulation in [4] due to the novel Lemma 2 in Section 4.2.

### 3.3 Tensor-based diffusion

The main deficiency of the methods presented in Section 3.2 is that the filtering is simply stopped close to image features such as edges and corners. To filter parallel to line-structures Weickert [56] introduced a tensor-based anisotropic diffusion scheme. The filtering scheme is defined as the PDE

$$u - u^0 - \lambda \operatorname{div}(D(T)\nabla u) = 0 \quad \text{in } \Omega. \quad (14)$$

The adaptivity of the filter is determined by the diffusion tensor  $D : \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^{2 \times 2}$ , constructed from a nonlinear mapping of the structure tensor  $T : \mathbb{R}^2 \rightarrow \mathbb{R}^{2 \times 2}$  defined as

$$T(\nabla u) = w * (\nabla u \nabla u^\top), \quad (15)$$

where  $*$  is a convolution operator and  $w$  is a smooth kernel, *e.g.*, a Gaussian function [29, 10]. The tensor is a windowed second moment matrix, it is positive semidefinite, thus it estimates the local energy distribution and can be thought of being a covariance matrix [30]. The eigenvector of  $T$  corresponding to the largest eigenvalue is aligned orthogonal to the image structure. Therefore, to avoid blurring image structures, the diffusion tensor  $D$  is computed as  $D(T) = O^\top g(\Lambda) O$  where  $g$  is the diffusivity function (*e.g.*, (11)),  $O$  are the eigenvectors and  $\Lambda$  has the eigenvalues of  $T$  on its diagonal [26].

Before continuing we show why a straightforward variational formulation, which gives the PDE (14) without the convolution (15), cannot be extended into a variational formulation for (14) with the convolution (15) by inserting the convolution into the regularizer.

This problem has previously been studied by, *e.g.*, Scherzer and Weickert [51]. Here we tackle the energy functional directly and show that the failing component is the application of Green’s identity. Now, let  $F : \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^{2 \times 2}$  be a tensor-valued function and define the regularization term

$$R(u) = \int_{\Omega} \nabla u^\top F(\nabla u) \nabla u d\mathbf{x}. \quad (16)$$

To compute the variational derivative of  $R$  we need partial integration in higher dimensions defined by Green’s identity [31]: Let  $A$  be a vector field and  $v$  once continuously differentiable, then

$$\begin{aligned} \int_{\Omega} A(\nabla u)^\top \nabla v d\mathbf{x} &= \int_{\partial\Omega} v A(\nabla u) \cdot \mathbf{n} dS \\ &\quad - \int_{\Omega} v \operatorname{div}(A(\nabla u)) d\mathbf{x}. \end{aligned} \quad (17)$$

When  $F$  is defined by a nonlinear function and a convolution operator there is no operator  $A$  such that Green’s formula applies. This is shown in the following proposition.



**Proposition 1** *Let  $f : \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^{2 \times 2}$  be a differentiable tensor valued-function and  $w$  be a smooth (Gaussian) function. If*

$$F(\nabla u) = f(w * (\nabla u \nabla u^\top)), \quad (18)$$

*then the corresponding Euler-Lagrange equation of (16) does not exist.*

*Proof* Assume that the variational derivative of (16) exists. To show this assertion we set

$$s(\varepsilon) = w * (\nabla(u + \varepsilon v) \nabla(u + \varepsilon v)^\top), \quad (19)$$

and let

$$f(s(\varepsilon)) = F(\nabla(u + \varepsilon v)). \quad (20)$$

Then from (7) and (16) we have

$$\begin{aligned} \frac{\partial R(u + \varepsilon v)}{\partial \varepsilon} = \\ \frac{\partial}{\partial \varepsilon} \int_{\Omega} \nabla(u + \varepsilon v)^\top f(s(\varepsilon)) \nabla(u + \varepsilon v) \, d\mathbf{x}. \end{aligned} \quad (21)$$

From the product rule of differentiation we are interested in the following part from (21):

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} \int_{\Omega} \nabla u^\top f(s(\varepsilon)) \nabla u \, d\mathbf{x} \\ = \int_{\Omega} \nabla u^\top \left( \frac{\partial f(s(\varepsilon))}{\partial \varepsilon} \right) \nabla u \, d\mathbf{x}, \end{aligned} \quad (22)$$

from which we only need to consider the differentiation of  $f$  w.r.t.  $\varepsilon$ , i.e.,

$$\frac{\partial f(s(\varepsilon))}{\partial \varepsilon} = \frac{\partial f}{\partial s} \frac{\partial s}{\partial \varepsilon}. \quad (23)$$

With  $s$  as in (19) we have

$$\begin{aligned} \frac{\partial s}{\partial \varepsilon} &= w * \left( \frac{\partial}{\partial \varepsilon} \nabla(u + \varepsilon v) \nabla(u + \varepsilon v)^\top \right) \\ &= w * (2 \nabla u \nabla v^\top + 2 \varepsilon \nabla v \nabla v^\top). \end{aligned} \quad (24)$$

Now, we evaluate the limit  $\varepsilon \rightarrow 0$  and obtain from (22), (23), (24) the following result

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} \int_{\Omega} \nabla u^\top \left( \frac{\partial f(s(\varepsilon))}{\partial \varepsilon} \right) \nabla u \, d\mathbf{x} \\ = \int_{\Omega} \nabla u^\top \left( \frac{\partial f(s(0))}{\partial s} (w * (2 \nabla u \nabla v^\top)) \right) \nabla u \, d\mathbf{x} \\ \neq \int_{\Omega} A(\nabla u)^\top \nabla v \, d\mathbf{x}, \end{aligned} \quad (25)$$

for any operator  $A(\nabla u)$ . Since the element  $v$  cannot pass the convolution operator, Green's formula (17) can not be applied to give an E-L equation of (16).  $\square$

In the next section we present one of the main results of this work: the mapping function-based functional and its E-L equation.

## 4 Mapping-based Image Diffusion

In this section we integrate a nonlinear mapping into a framework of tensor-based image diffusion.

### 4.1 Variational formulation

In the following, let  $m \in \mathcal{C}^2(\Omega)$  be the mapping function which describes the nonlinear mapping of the image value range. Also let  $W : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  be a positive semidefinite tensor. Set  $\mathbf{s} = \nabla m(u)$ . Let the components of  $W(\mathbf{s})$  be differentiable, then  $W_{s_i}(\mathbf{s})$  is the component-wise differentiation of  $W(\mathbf{s})$  w.r.t.  $\mathbf{s}$ , such that  $W_{\mathbf{s}} \in \mathbb{R}^{n^2 \times n}$  is defined as

$$W_{\mathbf{s}}(\mathbf{s}) = \begin{pmatrix} W_{s_1}(\mathbf{s}) \\ \vdots \\ W_{s_n}(\mathbf{s}) \end{pmatrix}. \quad (26)$$

In order to simplify notation in the following calculations we introduce a prescript-notation for matrix contraction, we define

$$\nabla_u W_{\mathbf{s}}(\mathbf{s}) = \begin{pmatrix} \nabla u^\top W_{s_1}(\mathbf{s}) \\ \vdots \\ \nabla u^\top W_{s_n}(\mathbf{s}) \end{pmatrix}, \quad (27)$$

which contracts  $n^2 \times n$  matrices such as (26) to an  $n \times n$  matrix.

The next theorem generalizes our initial formulation [2] (cf. Theorem 1) which states the relation between a tensor-functional and its corresponding E-L equation. In this work we extend the theorem to dimensions  $n \geq 2$  and to include the mapping  $m$ . Compared with [2] we state the theorem in a simplified form, exploiting newly found identities which result in a generalized tensor-based application-driven image denoising framework.

**Theorem 1** *Let the regularization term  $R$ , be given by*

$$R(u) = \int_{\Omega} \nabla m(u)^\top W(\nabla m(u)) \nabla m(u) \, d\mathbf{x}, \quad (28)$$

*where  $u, m \in \mathcal{C}^2(\Omega)$  and  $W : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ , and  $\nabla m \mapsto W$  is a tensor-valued function. Set  $\mathbf{s} = \nabla m(u)$ , then the corresponding E-L equation is given by*

$$\begin{cases} u - u^0 - \lambda \left( (m')^2 \operatorname{div}(S_1 \nabla u) \right. \\ \quad \left. + \operatorname{div}((m')^2 S_2 \nabla u) \right) = 0 \text{ in } \Omega \\ \mathbf{n} \cdot (S_1 + S_2) \nabla u = 0 \text{ on } \partial \Omega \end{cases} \quad (29)$$

where

$$S_1 = 2W(\mathbf{s}) + m'_{\nabla u} W_{\mathbf{s}}(\mathbf{s}), \quad (30a)$$

$$S_2 = 2W(\mathbf{s})^\top + m'_{\nabla u} W_{\mathbf{s}}(\mathbf{s}). \quad (30b)$$

*Proof* The proof is a generalization of the results presented in [2] (cf. Theorem 1) and is given in Appendix A.

Depending on choice of mapping  $m$  and tensor  $W$  in (28), we show in Table 2 that the proposed scheme generalizes several diffusion methods from literature. The same table also shows the methods that are subject to further study in this paper. In the subsequent sections we investigate some additional selections of  $W$  and  $m$  that extend the gradient energy total variation scheme (GETV) presented in [2] to include the mapping. Also we extend the scalar TV formulation to include the mapping function. Before proceeding to specify  $W$  and  $m$ , consider the following properties of the proposed functional and corresponding E-L equation in Theorem 1.

#### 4.2 Properties

In the general case  $\nabla_u W_s$  is a non-symmetric tensor, however one may be interested in positive definiteness or other symmetry constraints. For example it is often desirable in orientation estimation that the magnitude response of the estimated orientation is equivalent in vertical and horizontal directions in the image plane. If  $S_1$  and  $S_2$  in (29) are symmetric, then it aids the subsequent analysis of otherwise complex expressions to determine positive-definiteness. Therefore, we derive necessary conditions for  $W$  to yield a symmetric  $\nabla_u W_s$  in this section. In the following, we show that a tensor  $S$  exists such that the tensors defined by  $S_1$  and  $S_2$  in the E-L equation are symmetric, *i.e.*,

$$S = S_1 = S_2. \quad (31)$$

It is easy to identify two additional constraints required for  $S$  to be of the form (31) from (30a),(30b), *i.e.*, it is necessary that  $W = W^\top$  and  $\nabla_u W_s = (\nabla_u W_s)^\top$ . Let

$$W = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \quad (32)$$

then the following corollary state the set of symmetric tensors  $W$  that yield a symmetric tensor  $S$  in (31).

**Corollary 1** *Let  $W \in \mathbb{R}^{2 \times 2}$  be the symmetric tensor (32) with differential components, then if*

- i)*  $b_{s_1} = a_{s_2}$
- ii)*  $c_{s_1} = b_{s_2}$

*are both satisfied,  $\nabla_u W_s$  is also a symmetric tensor.*

*Proof* Since  $W$  is symmetric we can write  $\nabla_u W_s$  as

$$\nabla_u W_s(s) = u_x \begin{pmatrix} a_{s_1} & b_{s_1} \\ a_{s_2} & b_{s_2} \end{pmatrix} + u_y \begin{pmatrix} b_{s_1} & c_{s_1} \\ b_{s_2} & c_{s_2} \end{pmatrix}. \quad (33)$$

Conditions *i)* and *ii)* follow directly from the constraint  $\nabla_u W_s = (\nabla_u W_s)^\top$ .  $\square$

The solution of *i)* and *ii)* is clearly not unique. For example, one solution is given by

$$b = \int a_{s_2} ds_1 = \int c_{s_1} ds_2. \quad (34)$$

If instead  $b$  is given, then  $a$  and  $c$  must satisfy

$$a = \int b_{s_1} ds_2, \quad (35)$$

$$c = \int b_{s_2} ds_1, \quad (36)$$

which results in

$$W = \begin{pmatrix} \int b_{s_1} ds_2 & b \\ b & \int b_{s_2} ds_1 \end{pmatrix}, \quad (37)$$

to which we can also add any constant symmetric matrix. We note that in the general case it may be difficult, or not interesting, to evaluate Corollary 1. However, as a theoretical tool it may serve the purpose to derive classes of tensors  $S$  that are symmetric.

The following lemmas are particularly useful and enable the reformulation of the E-L equation (29) in a single divergence form. In this case we set  $M = S_1 = S_2$  where  $M$  in contrast to  $S$  in (31) is not necessarily a symmetric tensor.

**Lemma 1** *Let  $M \in \mathbb{R}^{n \times n}$ , then*

$$\operatorname{div}(M \nabla u) = \operatorname{div}(M) \nabla u + \operatorname{tr}(M H u), \quad (38)$$

*where  $H$  is the Hessian matrix.*

*Proof* We first write the left hand-side of (38) as a sum, then by expanding its partial derivatives we get

$$\begin{aligned} \operatorname{div}(M \nabla u) &= \sum_{ij} \partial_{x_i} (M_{i,j} u_{x_j}) \\ &= \sum_{ij} (\partial_{x_i} M_{i,j}) u_{x_j} + M_{i,j} u_{x_i x_j} \\ &= \operatorname{div}(M) \nabla u + \operatorname{tr}(M H u), \end{aligned} \quad (39)$$

which shows the result.  $\square$

**Lemma 2** *Given a tensor  $M \in \mathbb{R}^{n \times n}$  with differentiable components, the following relation holds*

$$\begin{aligned} (m')^2 \operatorname{div}(M \nabla u) + \operatorname{div}((m')^2 M \nabla u) \\ = 2m' \operatorname{div}(m' M \nabla u). \end{aligned} \quad (40)$$

Methods <i>Without mapping</i>	Regularizer	$M$ in E-L equation $u - u_0 - \lambda \operatorname{div}(M \nabla u) = 0$	Comment
Isotropic	$\int_{\Omega}  \nabla u ^2 d\mathbf{x}$	$I$	
TV [50]	$\int_{\Omega}  \nabla u  d\mathbf{x}$	$I/ \nabla u $	
EAD [1]	$\int_{\Omega} \nabla u^\top W(\nabla u) \nabla u d\mathbf{x}$	$W + W^\top + W_s$	
GETV [2]	$\int_{\Omega}  \nabla u (\lambda_1(v \cdot \xi)^2 + \lambda_2(w \cdot \xi)^2) d\mathbf{x}$	$S$	see (63) with $m(u) = u$
<b>With mapping</b>		$u - u_0 - \lambda m'(u) \operatorname{div}(M \nabla u) = 0$	
TIF [4] D3 [5]	$\int_{\Omega}  \nabla m(u) ^2 d\mathbf{x}$	$m'(u)I$	
TVm ( <i>novel</i> )	$\int_{\Omega}  \nabla m(u)  d\mathbf{x}$	$I/ \nabla u $	See (49)-(52)
GETVm ( <i>novel</i> )	$\int_{\Omega}  \nabla m(u) (\lambda_1(v \cdot \xi)^2 + \lambda_2(w \cdot \xi)^2) d\mathbf{x}$	$m'(u)S$	see (63)

**Table 2** Examples of E-L equations for different choices of mapping functions and tensors. We have listed linear diffusion, total variation (TV), extended anisotropic diffusion (EAD), gradient energy total variation (GETV), targeted iterative filtering (TIF) and density driven diffusion (D3). The corresponding boundary conditions can be computed using (30a) and (30b). The methods TVm and GETVm are extensions, introduced in this work, of TV and GETV with a mapping function.

*Proof* From Lemma 1, the left hand-side of (40) can be written as

$$\begin{aligned}
& (m')^2 \left( \operatorname{div}(M) \nabla u + \operatorname{tr}(MHu) \right) \\
& \quad + \operatorname{div}((m')^2 M) \nabla u + \operatorname{tr}((m')^2 MHu) \\
& = \left( (m')^2 \operatorname{div}(M) + 2m'm'' \nabla u^\top M \right. \\
& \quad \left. + (m')^2 \operatorname{div}(M) \right) \nabla u + 2(m')^2 \operatorname{tr}(MHu) \\
& = 2 \left( (m')^2 \operatorname{div}(M) + m'm'' \nabla u^\top M \right) \nabla u \\
& \quad + 2(m')^2 \operatorname{tr}(MHu) \\
& = 2m' \left[ \operatorname{div}(m'M) \nabla u + \operatorname{tr}(m'MHu) \right],
\end{aligned}$$

which concludes the proof.  $\square$

Observe that depending on  $m$ , its derivative  $m'$  can in the general case obtain negative values. However, due to linearity of the divergence operator, any non-negative factors originating from  $m'$  cancel as indicated in the right-hand side of (40), *i.e.*,

$$(\pm m') \operatorname{div}((\pm m') M \nabla u) \iff +m' \operatorname{div}(m' M \nabla u).$$

Assuming that  $M$  is at least positive (semi)-definite, then the final discretized E-L equation will have a convergent behaviour.

In particular, with the sufficient condition for tensor symmetry of  $W$  and  $\nabla_u W_s$ , and the above simplification

of the divergence terms in Lemma 2, we obtain the E-L equation of the regularization term in Theorem 1 on the following (considerably) simplified form.

**Corollary 2** Let  $W \in \mathbb{R}^{n \times n}$  be a symmetric tensor, then the E-L equation (29) is given by

$$\begin{cases} u - u^0 - \lambda m' \operatorname{div}(m' M \nabla u) = 0 & \text{in } \Omega \\ \mathbf{n} \cdot M \nabla u = 0 & \text{on } \partial \Omega \end{cases} \quad (41)$$

where

$$M = 2W(\mathbf{s}) + m'_{\nabla u} W_s(\mathbf{s}). \quad (42)$$

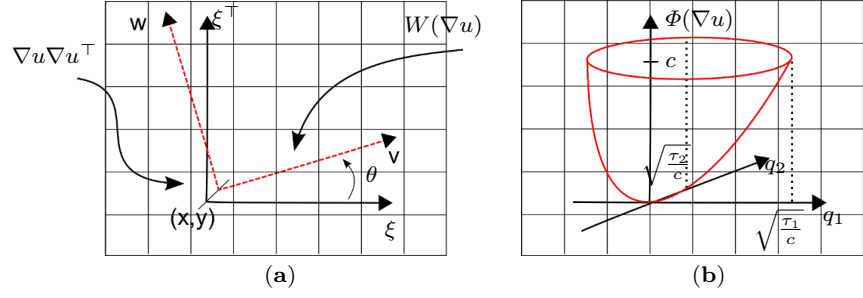
*Proof* Since  $W = W^\top$  it follows that  $S_1 = S_2$ , whereby Lemma 2 can be applied to (29).  $\square$

**Convexity** In this section we investigate and discuss convexity of the proposed functional in the case  $m(u) = u$  and  $W$  is a symmetric and positive semidefinite tensor. If  $m$  is a monotonically increasing function, the below results can be generalized using similar arguments, however, for a general function  $m$  the description of a convex function remains an open problem. Our approach is based on the eigenvalue decomposition of the tensor  $W$ .

Consider the quadratic form  $\nabla u^\top W(\nabla u) \nabla u$  rewritten in the eigendecomposition of  $W$  with eigenvalues  $\lambda_1, \lambda_2 \geq 0$  and eigenvectors  $v, w$ , then

$$\begin{aligned} \nabla u^\top W(\nabla u) \nabla u &= \nabla u^\top [\lambda_1 v v^\top + \lambda_2 w w^\top] \nabla u \\ &= \lambda_1 v^\top [\nabla u \nabla u^\top] v + \lambda_2 w^\top [\nabla u \nabla u^\top] w. \end{aligned} \quad (43)$$

**Fig. 4** (a) Illustration of eigenvector basis at coordinate  $(x, y)$ . Dashed (red) arrows indicate the eigenvectors of  $W$  and  $S$  and thick (black) arrows  $\nabla u \nabla u^\top$ . (b) Illustration of the paraboloid (47) where we set  $c = \tau_1 q_1^2 + \tau_2 q_2^2$  and  $\tau_1, \tau_2 \geq 0$ .



The product  $\nabla u \nabla u^\top$  has orthonormal eigenvectors  $\xi = (\xi_1, \xi_2)^\top$  and  $\xi^\perp = (\xi_2, -\xi_1)^\top$  such that  $P = (\xi, \xi^\perp)$  and  $\Lambda$  has the corresponding eigenvalues  $\kappa_1$  and  $\kappa_2$  on its diagonal. Note that by the spectral theorem [35], the eigendecomposition of  $\nabla u \nabla u^\top$  is *always* well-defined, *i.e.*,  $\xi$  is not singular. This is shown by the generalized definition of  $\xi$ , *i.e.*, in the case of  $|\nabla u| \neq 0$  then  $\xi = \nabla u / |\nabla u|$ , and in the case of  $|\nabla u| = 0$  we let  $P = I$  where  $I$  is the identity matrix.

In the following, we substitute the eigendecomposition of  $\nabla u \nabla u^\top = P^\top \Lambda P$  into (43):

$$\begin{aligned} \nabla u^\top W(\nabla u) \nabla u &= \lambda_1 v^\top P \Lambda P^\top v + \lambda_2 w^\top P \Lambda P^\top w \\ &= \lambda_1 v^\top P \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} P^\top v + \lambda_2 w^\top P \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} P^\top w, \end{aligned} \quad (44)$$

and insert  $\kappa_1 = |\nabla u|^2$  and  $\kappa_2 = 0$ . Then after rewriting (44) we obtain

$$\nabla u^\top W(\nabla u) \nabla u = \lambda_1 |\nabla u|^2 (v \cdot \xi)^2 + \lambda_2 |\nabla u|^2 (w \cdot \xi)^2. \quad (45)$$

In (45),  $v \cdot \xi = \cos(\theta)$  and  $w \cdot \xi = \sin(\theta)$ , where  $\theta$  is the rotation angle as shown in Figure 4 (a). This means that the scalar products define the rotation of  $W$  in relation to the image gradient direction. Note that as  $W$  describes the local directional information, its eigenvectors will be parallel to the orthonormal eigenvectors of  $\nabla u \nabla u^\top$ , *i.e.*,  $v \parallel \xi$  and  $w \parallel \xi^\perp$  if  $\theta = 0$ .

**Remark 1** We could have arrived at (45) directly by introducing  $|\nabla u|^2$  in the nominator and denominator of (43). However, this approach would not be valid since it introduces a singularity when  $|\nabla u| = 0$ . Instead we exploited properties from the spectral theorem and thus (45) is always well-defined even in the case  $|\nabla u| = 0$ .

The next result shows that the proposed functional is convex when  $W$  is symmetric and positive semidefinite with eigenvalues  $\lambda_1, \lambda_2 \geq 0$ . When the functional is strictly convex it is guaranteed from the theory of convex optimization that its stationary point gives the minimum energy, and thus results in the optimal solution [15]. Note that  $m(u) = u$ . Now consider the convexity of the proposed functional in the below Corollary.

**Corollary 3** *Let  $m(u) = u$ , then the functional,  $R$  in (28), is convex w.r.t. the element  $u$ .*

*Proof* To prove the convexity of  $R$  we write  $\Phi(u) = \nabla u^\top W(\nabla u) \nabla u$  in terms of the eigenvectors and eigenvalues of  $W$ . Then, from (45) it follows that

$$\begin{aligned} \Phi(\nabla u) &= |\nabla u|^2 (\lambda_1 \xi^\top (v v^\top) \xi + \lambda_2 \xi^\top (w w^\top) \xi) \\ &= |\nabla u|^2 \xi^\top (\lambda_1 v v^\top + \lambda_2 w w^\top) \xi \\ &= (V \xi)^\top \begin{pmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{pmatrix} V \xi, \end{aligned} \quad (46)$$

where  $V = (v, w)$  and  $\tau_i(\nabla u) = |\nabla u|^2 \lambda_i \geq 0$  for  $i = 1, 2$ . Let  $q = V \xi = (q_1, q_2)^\top$ , then

$$\Phi(\nabla u) = \tau_1 q_1^2 + \tau_2 q_2^2, \quad \tau_i \geq 0, \quad (47)$$

is a quadratic form in the basis of orthonormal eigenvectors  $V$  and  $\tau_i$ . The quadratic form is always well-defined due to the spectral theorem. The paraboloid (47) has positive curvature everywhere as illustrated in Figure 4 (b). Since  $u$  is mapped continuously to the paraboloid,  $R$  is convex in  $u$ .  $\square$

**Extending scalar-valued diffusion** It is straightforward to show that the quadratic form in (28) is related to the standard Euclidean p-norms:

**Proposition 2** *Let  $W = |\nabla m(u)|^q$  and  $q = p-2$  where  $1 \leq p < \infty$ , then*

$$\nabla m(u)^\top |\nabla m(u)|^q \nabla m(u) = |\nabla m(u)|^p. \quad (48)$$

*Proof* Substitute  $q = p-2$  in the left-hand side of (48). Since  $\nabla m(u)^\top \nabla m(u) = |\nabla m(u)|^2$ , the right-hand side is obtained after canceling common terms in the quotient.  $\square$

We obtain mapping-based isotropic diffusion by setting  $q = 0$  in (48), *i.e.*, it yields the regularization term (12) studied in [4]. In this case the E-L equation is given by (13). Additionally, from (48), if we select  $q = -1$  then

$$R(u) = \int_{\Omega} |\nabla m(u)| \, dx. \quad (49)$$

The regularization term (49) can also be interpreted in relation to the standard Rudin-Osher-Fatemi (ROF) TV-model [50], see definition in [6], pp. 39-43, but now with a mapping function.

The formulation (49) can also be obtained formally from the initial functional (28) in Theorem 1. Set

$$W(\nabla m(u)) = \frac{1}{|\nabla m(u)|} I, \quad (50)$$

then the differentiation with respect to each component of  $W$  yields

$$\nabla_u W_{\nabla m(u)}(\nabla m(u)) = -\frac{1}{(m')^2 |\nabla u|^3} \nabla u \nabla u^\top. \quad (51)$$

Since  $W$  in (50) is obviously symmetric, Corollary 2 is applicable and the resulting E-L equation is given by

$$\begin{cases} u - u^0 - \lambda m' \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) = 0 & \text{in } \Omega \\ \mathbf{n} \cdot \nabla u = 0 & \text{on } \partial\Omega \end{cases} \quad (52)$$

The standard isotropic diffusion and total variation formulations are obtained by setting  $m(u) = u$ , which yields  $m'(u) = 1$  in (48). Thus the standard methods are special cases of our presented framework. The introduction of the mapping function adds an additional nonlinear component that controls the amount of filtering based on some prior knowledge of the image value range.

Before continuing to the tensor-valued formulations of Theorem 1, we recall the gradient energy tensor [28] and present an empirical evaluation of the tensor compared to the structure tensor [10]. In [1] we proved that the connection between the functional and its E-L is not preserved when the structure tensor is subject to a nonlinear mapping of its eigenvalues. This has also been observed by others [40, 11], furthermore, an explicit derivation of this fact is given in Proposition 1, Section 3.3 of this work. Since the GET does not have a post-convolution of its tensor components we remove some obstacles to preserve the connection between the energy and the corresponding E-L equation.

In the next section we review the gradient energy tensor before introducing the gradient energy total variation scheme with a mapping function.

## 5 Gradient Energy Tensor

In this section we review the gradient energy tensor [28] (GET) and analyse how sensitive it is with regards to orientation estimation compared with the structure tensor. Our analysis show that the GET can approximate the structure tensor for image denoising.

### 5.1 Background and definition

The origin of the gradient energy tensor comes from the 1-D energy operator [14], and in particular, from the observation that the energy of a signal can be estimated from its squared magnitude [37]. The gradient energy tensor is a symmetric tensor, like the structure tensor, and it can be shown that GET is phase invariant, orientation equivariant second order tensor [27, 46]. The GET tensor is lesser known in the computer vision community and it was first introduced by Felsberg and Köthe [28]. The motivation for considering the GET in this work, as opposed to the structure tensor, is three-fold (I) the GET can potentially enable real-time image diffusion implementations which is not possible with the structure tensor (due to the post-convolution of its tensor-components, for details see [3]), (II) the GET preserves the relation between the energy and the E-L equation when the tensor is positive semi-definite and (III) to promote further studies on the tensor. Similarly to the structure tensor, GET is of rank 2 and determines the directional energy distribution of the signal gradient.

The classical GET is defined in terms of the image data in  $u$  [28]. Here we use an alternative, but equivalent, formulation of GET expressed in the image gradient. Let  $Hm(u) = \nabla(\nabla m(u)^\top)$  be the Hessian and  $\nabla(\Delta m(u)) = \nabla(\nabla^\top \nabla m(u))$ , where  $\Delta$  is the Laplace operator, then we define GET as

$$\begin{aligned} \text{GET}(\nabla m(u)) &= Hm(u)Hm(u) \\ &\quad - \frac{1}{2} \left( \nabla m(u) [\nabla \Delta m(u)]^\top + [\nabla \Delta m(u)] \nabla m(u)^\top \right). \end{aligned} \quad (53)$$

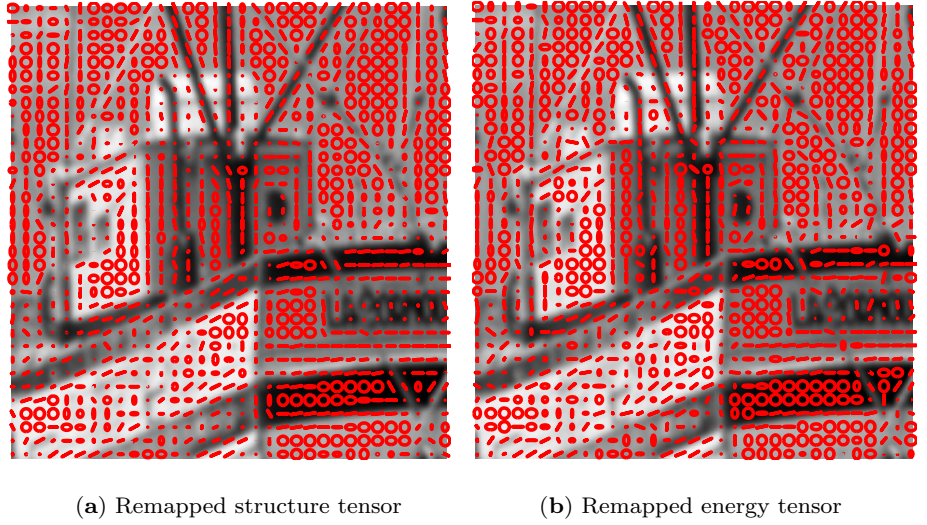
Since the GET components are expressed in terms of the image gradient, Theorem 1 is applicable. In contrast to the structure tensor (15), the gradient energy tensor (53) does *not* (necessarily) require a convolution operator to form a rank 2 tensor.

Example responses from the two tensors are illustrated in Figure 5. Note that due to the convolution operator, the structure tensor is not sensitive to structures smaller than the width of the averaging filter used to compute it (in this case the standard deviation of the Gaussian filter was set to 1). The presence of second and third-order derivatives in GET makes it slightly more sensitive to noise. However, it allows us to capture orientation of structures not possible to detect using the structure tensor.

### 5.2 Spectral properties of GET

An investigation of the positivity of the 1-D energy operator was presented in [14]. In the 2-D case, the pos-

**Fig. 5** Visualization of estimated local orientations for (a) the structure tensor and (b) the gradient energy tensor.



itivity of the operator is reflected in the sign of the eigenvalues. Let the components of the GET be  $a$ ,  $b$  and  $c$ , *i.e.*,

$$\text{GET} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \quad (54)$$

then the GET is positive semidefinite if the following lemma is satisfied.

**Lemma 3** *The GET is positive semidefinite if*

$$\text{tr}(Hm(u)Hm(u)) - \nabla m(u)^\top \nabla \Delta m(u) \geq \sqrt{l}, \quad (55)$$

where  $l = \text{tr}(\text{GET})^2 - 4\det(\text{GET}) \geq 0$ .

*Proof* Since GET is symmetric it has real eigenvalues. Thus by its eigenvalue decomposition it is sufficient to show that  $\text{tr}(\text{GET}) \geq \sqrt{l}$  (*i.e.*, (55)) in order for GET to be positive semidefinite.  $l$  is necessarily positive since  $l = (a - c)^2 + 4b^2 \geq 0$ .  $\square$

The GET-tensor can attain zero-values despite the gradient is not zero, and in some cases the eigenvalues may be negative. Here, we set the negative eigenvalues to positive in the  $\text{GET}^+$  tensor. We define

$$\text{GET}^+(\nabla m(u)) = vv^\top |\mu_1| + ww^\top |\mu_2|, \quad (56)$$

and  $v, w$  are the eigenvectors and  $\mu_1, \mu_2$  eigenvalues of the GET respectively. In the standard approach of anisotropic diffusion one considers the diffusion tensor, which is the structure tensor with transformed eigenvalues as described in Section 3.3. We do a similar mapping of (56) by using the negative exponential function, *i.e.*, we have

$$\begin{aligned} D(\nabla m(u)) &= \exp(-\text{GET}^+(\nabla m(u))/k^2) \\ &= \lambda_1 vv^\top + \lambda_2 ww^\top, \end{aligned} \quad (57)$$

and  $v, w$  are the same eigenvectors as in (56) and the eigenvalues  $\lambda_{1,2} = \exp(-|\mu_{1,2}|/k^2)$ . Now  $\text{GET}^+$ , is positive semidefinite as  $\exp$  is the matrix exponential function obtained by a Taylor series expansion. Figure 5 shows a visual comparison between the remapped energy tensor and the diffusion tensor. Visually, the local orientation estimation appears nearly identical between the two tensors. Next, we quantify this assertion.

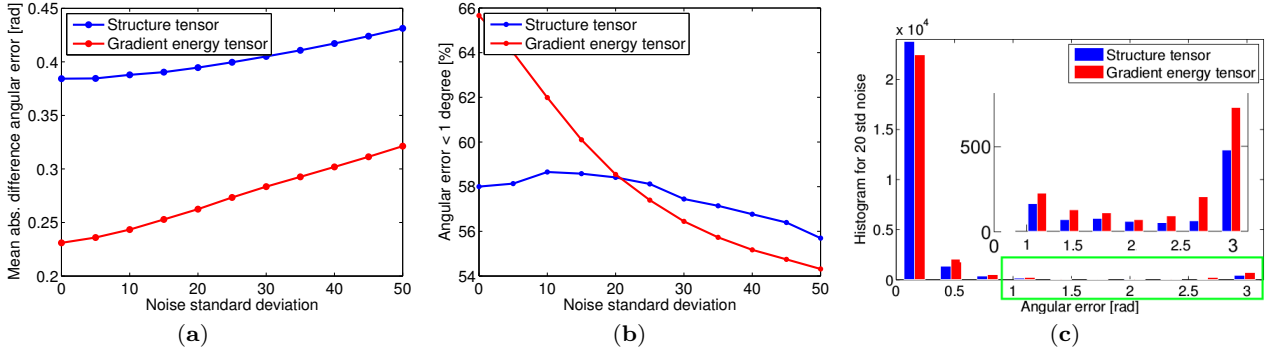
### 5.3 Analysis

In this part we discuss the effect of noise perturbation on the tensor's estimated orientations. We consider a radially symmetric test-pattern shown in Figure 7 (a). As a baseline for the evaluation we use the noise-free angles that have also been used to compute the test-pattern image. For each tensor and considered noise level, we compute the local neighbourhoods dominant distribution angle by projecting the tensor components on a vector  $z$  in the complex plane [32],

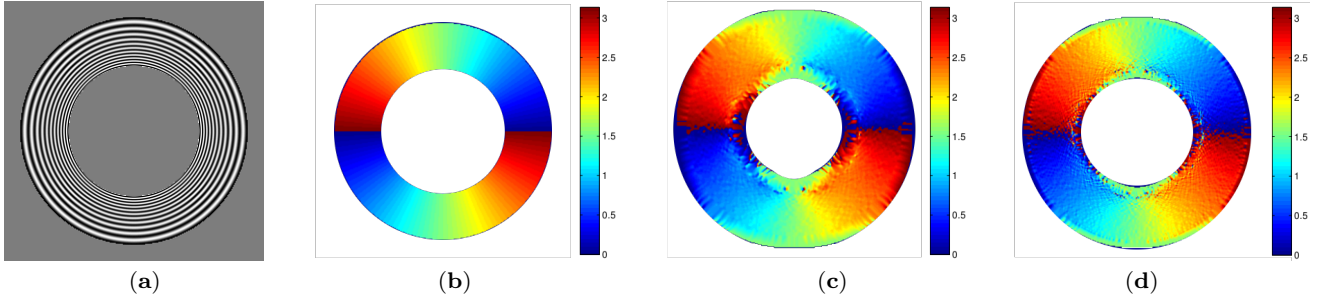
$$z = a - c + i2b, \quad (58)$$

where  $a, b$  and  $c$  are components of a symmetric  $2 \times 2$  matrix. We compare the mean absolute difference of the angular error in Figure 6 (a) for different noise levels. In (b) we show the percentage of pixels that have an angular error smaller than 1 degree. It is evident that the two tensors are marginally different. As the noise-level increases beyond standard deviation 20, the structure tensor performs better due to the post-convolution of the tensor components. At standard deviations 0 and 10 of noise the difference is approximately 8% and 3% respectively. Our third evaluation is a comparison of the histogram error quantized to 10 bins in the range





**Fig. 6** Results of sensitivity analysis of orientation estimation by using the structure tensor and the gradient energy tensor. We used the radially symmetric pattern (a) for different noise levels in Figure 7. (a) shows the mean absolute angular error. (b) illustrates the percentage of pixels which has an angular error smaller than 1 degree. (c) shows the histogram of angular errors divided into 10 bins from 0 to  $\pi$  radians. See text for details.



**Fig. 7** (a) Original, noise free circular pattern. (b) Ground truth orientation. Orientation in radians at standard deviation 20 of noise, (c) Structure tensor and (d) Gradient energy tensor.

0 –  $\pi$  at the intersection point seen in (b), *i.e.*, at the noise level of standard deviation 20. The histogram is shown in (c) and we see that the error distribution is indeed similar for the two tensors. We also see that for the angles less than 0.5 rad the structure tensor shows higher errors than the structure tensor due to the post-convolution of its components. However, considering the right most bins, the structure tensor has fewer pixels with large errors than the gradient energy tensor.

Objectively, the gradient energy tensor has more large magnitude errors than the structure tensor. However, this is expected since the energy tensor lacks a post-convolution of the tensor components to compensate for the noise. Nevertheless, considering that the difference between the two tensors mean angular error is small, as indicated in (a), it still suggests that the gradient energy tensor can be used in a diffusion framework, a claim that we further support in our numerical experiments, and was shown in [3]. In the following sections we use the gradient energy tensor to model a tensor-based total variation framework.

## 6 Mapping-based Gradient Energy Tensor Total Variation

In this section we extend our previous work [2, 1] on the gradient energy tensor total variation (GETV). In particular we introduce the mapping to GETV and denote the formulation as GETVm.

### 6.1 The objective function

The below corollary gives the definition of the GETVm scheme. The tensor that steer the filter is the transformed gradient energy tensor, *i.e.*, (57). In the following, let  $W$  be a tensor such that the following relation holds

$$D(\nabla m(u)) = W(\nabla m(u))|\nabla m(u)|^q, \quad (59)$$

where  $D$  is the tensor in (57) and has eigenvalues  $\lambda_1, \lambda_2$  and eigenvectors  $v, w$ . Then (59) can be reformulated into a modified expression of (45), *i.e.*,

$$\begin{aligned} \nabla m(u)^\top W(\nabla m(u)) \nabla m(u) \\ = \lambda_1 |\nabla m(u)|^{2-q} (v \cdot \xi)^2 + \lambda_2 |\nabla m(u)|^{2-q} (w \cdot \xi)^2. \end{aligned} \quad (60)$$

where the left hand side is the integrand of (28). Now we give the following statement

**Corollary 4** Set  $\mathbf{s} = \nabla m(u)$  and let  $q = 1$  in (60). Then the gradient energy total variation regularization term is given by

$$R(u) = \int_{\Omega} |\nabla m(u)| (\lambda_1 (v \cdot \xi)^2 + \lambda_2 (w \cdot \xi)^2) d\mathbf{x}, \quad (61)$$

and the tensor  $\nabla_u W_{\mathbf{s}}$  in (27) is

$$\begin{aligned} \nabla_u W_{\mathbf{s}}(\mathbf{s}) = & -\frac{1}{|\mathbf{s}|^3} \begin{pmatrix} s_1 \nabla u^\top D(\mathbf{s}) \\ s_2 \nabla u^\top D(\mathbf{s}) \end{pmatrix} \\ & + \frac{1}{|\mathbf{s}|} \begin{pmatrix} \nabla u^\top D_{s_1}(\mathbf{s}) \\ \nabla u^\top D_{s_2}(\mathbf{s}) \end{pmatrix}, \end{aligned} \quad (62)$$

where  $D$  is defined in (57).

*Proof* See Appendix B.  $\square$

Due to the definition of the tensor  $D$  in (59) the resulting functional describes an anisotropic total variation filtering method. The tensor acts as a weight limiting the TV filtering perpendicular to image structures. In a homogeneous region  $D(\nabla m(u)) \approx I$  the regularization term in the functional reduces to  $R(u) = \int_{\Omega} |\nabla m(u)| d\mathbf{x}$  previously described in (49). It is obvious that the GET-tensor will be isotropic in homogeneous regions.

In the section below we show how to derive the E-L equation corresponding to the selection of  $W$  in (59) that yields the diffusion scheme.

## 6.2 Diffusion formulation

To obtain the diffusion formulation of GETVm, note that  $W$  in (59) is symmetric. Then, according to Corollary 2, it is straightforward to show that the tensors in the E-L equation (29) are given by  $S = S_1 = S_2 = 2W + \nabla_u W_{\mathbf{s}} m'$ . With  $\nabla_u W_{\mathbf{s}}$  given in (62) we obtain

$$\begin{aligned} S = & \left\{ \frac{2D(\mathbf{s})}{|\mathbf{s}|} - \frac{1}{|\mathbf{s}|^3} \begin{pmatrix} s_1 \nabla u^\top D(\mathbf{s}) \\ s_2 \nabla u^\top D(\mathbf{s}) \end{pmatrix} \right. \\ & \left. + \frac{1}{|\mathbf{s}|} \begin{pmatrix} \nabla u^\top D_{s_1}(\mathbf{s}) \\ \nabla u^\top D_{s_2}(\mathbf{s}) \end{pmatrix} \right\} \Big|_{\mathbf{s}=\nabla m(u)} \\ = & \frac{M}{m'(u)|\nabla u|}, \end{aligned} \quad (63)$$

where

$$\begin{aligned} M = & 2D(\mathbf{s}) - \frac{1}{|\nabla u|^2} \begin{pmatrix} u_x \nabla u^\top D(\mathbf{s}) \\ u_y \nabla u^\top D(\mathbf{s}) \end{pmatrix} \\ & + \begin{pmatrix} \nabla u^\top D_{s_1}(\mathbf{s}) \\ \nabla u^\top D_{s_2}(\mathbf{s}) \end{pmatrix}. \end{aligned} \quad (64)$$

Now, from Corollary 2, (41) we get the E-L equation

$$\begin{cases} u - u^0 - \lambda m' \operatorname{div} \left( M \frac{\nabla u}{|\nabla u|} \right) = 0 & \text{in } \Omega \\ \mathbf{n} \cdot M \nabla u = 0 & \text{on } \partial\Omega \end{cases} \quad (65)$$

The difference to TV is the introduction of a mapping function and the tensor in front of  $\nabla u/|\nabla u|$  in the divergence. These weights control the anisotropy of the TV scheme: reducing the filtering orthogonal to image structures and accelerating the filtering parallel to image structures. The orientation estimation is achieved using the GET-tensor with positive eigenvalues, defined in (56). However, note that using GET is not unique, any tensor, also the diffusion tensor constructed from the structure tensor may be applicable in this framework.

The derivatives of  $D$  with respect to  $\partial_x m(u)$  and  $\partial_y m(u)$  in (64) are obtained by first computing the eigenvectors  $v$  and  $w$  and eigenvalues  $\lambda_1, \lambda_2$  of  $D$ . The differentiation of  $D$  in (59) w.r.t.  $s_1 = \partial_x m(u)$  reads

$$\begin{aligned} D_{s_1} = & (\partial_{s_1} v v^\top) \lambda_1 + v v^\top (\partial_{s_1} \lambda_1) \\ & + (\partial_{s_1} w w^\top) \lambda_2 + w w^\top (\partial_{s_1} \lambda_2), \end{aligned} \quad (66)$$

and  $\lambda_i = \exp(-|\mu_i|/k^2)$ ,  $i = 1, 2$  with the corresponding orthonormal eigenvectors ( $b \neq 0$ )

$$v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{|\tilde{v}|} \tilde{v}, \quad \text{where } \tilde{v} = \begin{pmatrix} 2b \\ c - a + \sqrt{l} \end{pmatrix}, \quad (67)$$

and

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \frac{1}{|\tilde{w}|} \tilde{w}, \quad \text{where } \tilde{w} = \begin{pmatrix} 2b \\ c - a - \sqrt{l} \end{pmatrix}, \quad (68)$$

and  $l$  is given in Lemma 3. Appendix C further details the components of (66). In the case  $b = 0$  the eigenvalues are given on the diagonal of GET and the orthonormal eigenvectors are given by the identity matrix. The implication of  $b = 0$ , is that derivatives with respect to  $\partial_s v v^\top$  and  $\partial_s w w^\top$  in (66) vanish.

## 6.3 Discretization

The PDEs we have introduced in this work can all be expressed on the form

$$u - u_0 - \lambda m'(u) \operatorname{div} \left( M \frac{\nabla u}{|\nabla u|} \right) = 0, \quad (69)$$

where  $\nabla u \mapsto M$  and  $M \in \mathbb{R}^{2 \times 2}$  is a tensor different for each application (selection) of  $W$  and  $m$ . Different selections of  $W$  are outlined in Table 2.

To approximate the PDEs we use a simple, but generic, PrimalDual approach of Chan, Golub and Mulet

	ISO/m, TV/m	GETV/m
$\lambda$	0.1	0.05
$\tau$	$\lambda/8$	
$\beta$	$10^{-12}$	
$\gamma$	1	
$k$	100	

**Table 3** Parameters of the numerical scheme were fixed for all applications. “/m” denotes with and without mapping. The abbreviations stand for isotropic diffusion (ISO), isotropic diffusion with a mapping function (ISOm) [5], total variation (TV) [50], total variation with a mapping function (TVm), see (52), gradient energy tensor total variation (GETV) [2] and GETV with a mapping function (GETVm), see (65).

[20]. The idea is straight-forward and the aim is to avoid explicit computation of the quotient in (69), therefore they introduce a dual variable,  $d$ . In the following, let  $\lambda, \beta, \gamma$  and  $\tau$  be small positive constants. In our case we define the dual variable as

$$d = \frac{M \nabla u}{\sqrt{|\nabla u|^2 + \beta}}, \quad (70)$$

where we introduce  $\beta$  in the denominator. This gives the nonlinear equation system

$$u - u_0 - \lambda m'(u) \operatorname{div}(d) = 0 \quad (71a)$$

$$d \sqrt{|\nabla u|^2 + \beta} - M \nabla u = 0. \quad (71b)$$

We update of the primal variable  $u$  as

$$u^{k+1} = u^k + \tau (u^k - u_0 - \lambda m'(u^k) \operatorname{div}(d^{k+1})), \quad (72)$$

where  $k$  is an iterator. To solve for the dual variable in (71b) we introduce the iterator  $i$ , and use the current  $u^k$  as initial value, *i.e.*, we have

$$\begin{cases} d_{i+1}^k &= d_i^k + \gamma (d_i^k \sqrt{|\nabla u^k|^2 + \beta} - M_k \nabla u^k) \\ d^{k+1} &\leftarrow d_{i+1}^k \end{cases} \quad (73)$$

The divergence term in (71a) is discretized using forward finite differences [57], *i.e.*,

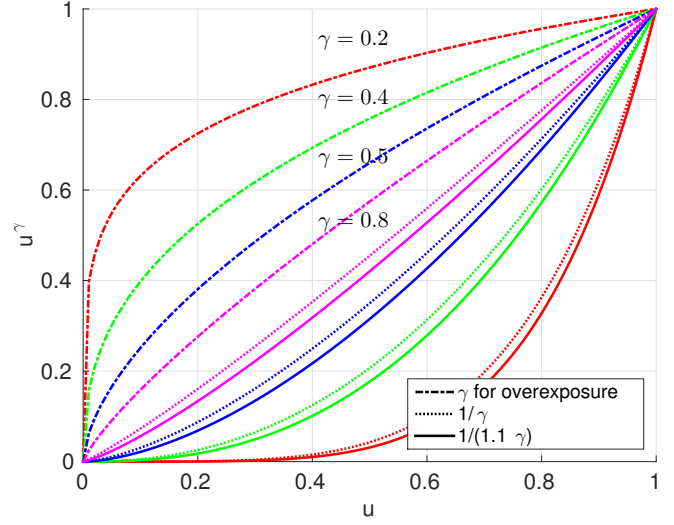
$$\operatorname{div}(d^{k+1}) = \partial_x^+ d^{k+1} + \partial_y^+ d^{k+1}, \quad (74)$$

where

$$\partial_x^+ u = u(x+1, y) - u(x, y) \quad (75a)$$

$$\partial_y^+ u = u(x, y+1) - u(x, y). \quad (75b)$$

In  $M$ , (69), we are required to compute terms such as  $1/|\nabla u|$ . Therefore, in practice we regularize the denominator term with a small positive constant, *i.e.*,  $|\nabla u|_\beta = \sqrt{u_x^2 + u_y^2 + \beta}$  as they appear in  $M$ . In the



**Fig. 8** Mapping function for gamma correction. Illustration of different gamma parameters (dash-dotted), corresponding compensation curves (dashed) and the compensation curve with 10% error (thick). The dash-dotted curve is used to obtain the noisy images and the thick curve is used to obtain the denoised results in Figure 9.

gradient energy tensor (53), we are required to compute third-order derivatives (see appendix (98a)-(98c)) for terms such as  $\partial_x \Delta u$ . However, rather than to expand these terms using derivative rules, we found that it is sufficient to approximate these terms using a central difference derivative. The finite differences are regularized with a smooth kernel. Here we use a Gaussian filter of standard deviation 1 before computing the finite differences. The filter size was kept constant for all images and all noise levels in the experimental evaluation. We also fixed the parameters  $\lambda, \beta, \gamma, \tau$  and the number of inner iterations  $k$  for all experiments. The parameter values are shown in Table 3.

## 7 Application I: Gamma correction

In this section we apply total variation with a mapping function (TVm), see (52), to the problem of denoising overexposed images corrupted with multiplicative noise. Multiplicative noise is the common noise model in, *e.g.*, ultrasound imaging, synthetic aperture radar (SAR) imaging, sonar and laser imaging, etc.

The intention with this application is not to show state of the art results denoising of multiplicative noise, rather we show empirically that (I) gamma correction (GC) and then denoise, or (II) denoising and then gamma correction is suboptimal to (III) using a combined filtering scheme compensating for the gamma correction in the evolution process of the minimiser. The parameters were set as shown in Table 3. We consider the following

Noisy $\gamma = 0.8$	Filter then GC	GC then filter	Mapping-based filter
			
PSNR: 8.6 SSIM: 0.50	24.5 0.74	24.4 0.70	26.2 0.79
$\gamma = 0.6$			
			
PSNR: 15.7 SSIM: 0.56	24.4 0.73	24.9 0.75	25.3 0.77
$\gamma = 0.4$			
			
PSNR: 11.9 SSIM: 0.56	24.4 0.74	25.1 0.75	25.8 0.79

**Fig. 9** Examples of combined gamma correction (GC) and denoising compared to “denoise and then gamma correct” and “gamma correct and then denoise”. If the degradation parameter is set to  $\gamma = 1$ , (77) is a one-to-one mapping and (78) is the identity. Instead of using the true gamma correction parameter  $1/\gamma$  we use  $1/(1.1\gamma)$  to represent the uncertainty in the estimate of an unknown inverse mapping. With a stronger nonlinearity, it can be seen that the mapping-based filter performs better than the other approaches, both visually and with respect to error measures.

noise model: let  $u^0$  be the degraded image, then

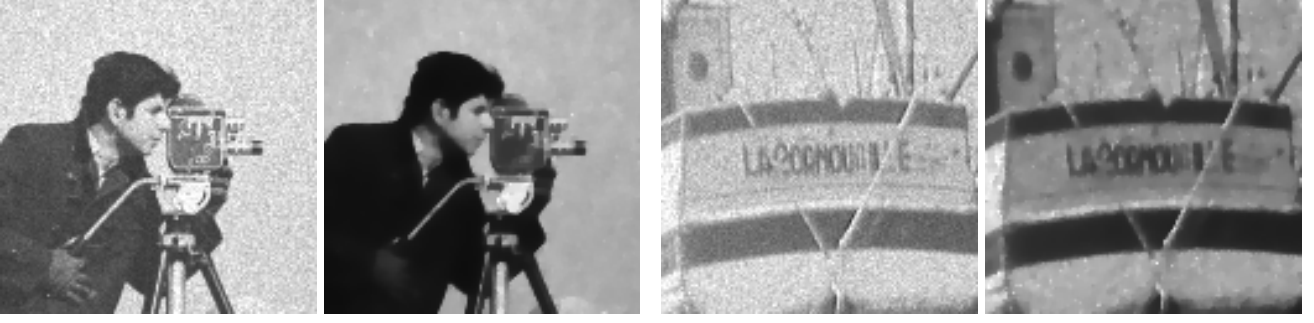
$$u^0 = \exp(\ln(u) + \eta(u)), \quad (76)$$

where  $\eta \sim \mathcal{N}(u|\mu, \sigma)$  and  $\mu$  is the mean-value and  $\sigma$  the standard deviation. The main difference to Gaussian noise is that multiplicative noise scales with the intensity; the lower the intensity, the less noise is present. Gaussian noise is uniform in the sense that the intensity

range is degraded by the same strength of the noise-components.

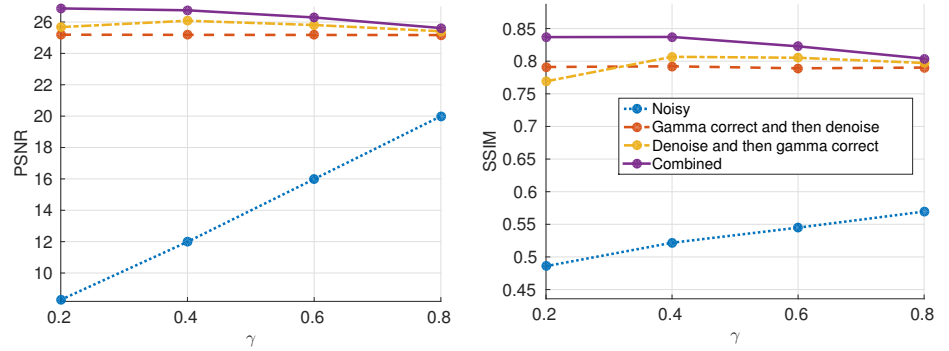
We use the “standard” gamma correction formula and define the mapping function to correct for an over-exposed image. In this sense, we define the mapping function that corrects for this degradation as

$$m(u) = u^{1/\gamma}, \quad (77)$$



**Fig. 10** Denoising results for the mapping-based filter when  $\gamma = 0.4$ . In each instance we accurately preserve the dominant features while suppressing the main parts of the noise. The noise is multiplicative and  $\sigma$  was set to 40 levels of the intensity range in the noise model (76).

**Fig. 11** Evaluation of gamma correction of overexposed images Cameraman, Barbara, Lena and Boat. As  $\gamma$  becomes smaller, *i.e.*, the nonlinearity is more pronounced, the mapping-based filter consistently performs better than the two alternatives “denoise and gamma correct” and “gamma correct and denoise”. We set  $\sigma = 40$  of the intensity range in (76).



where  $\gamma > 0$ . If  $\gamma < 1$  then (77) corrects for an overexposed image, and if  $\gamma > 1$  corrects for an underexposed image. Figure 8 shows the different values for  $\gamma$  and its inverse, we also show the difference to 10% offset of the inverse gamma value. The reason why we include this scale-factor is that in practical applications it is difficult, or impossible, to obtain an accurate estimate of the true inverse-mapping. In the E-L equation we are required to compute the derivative of the mapping, *i.e.*, we have

$$m'(u) = \frac{1}{\gamma} u^{1/\gamma-1}, \quad (78)$$

and normalize it by the maximum slope so that  $0 \leq m'(u) \leq 1$  to assure stability when solving the PDE (52).

## 7.1 Results

Figure 9 shows the result of denoising an overexposed image for the different cases of filtering order. Naturally, if  $\gamma = 1$ , then the case of “filter and then gamma correct” is equivalent to the mapping-based filter (as it should be). As  $\gamma$  decreases, *i.e.*, the nonlinearity become more severe, we observe that the mapping-based filter becomes superior to the two other approaches in all cases. The error measures in Figure 11 clearly illustrates this relation. In particular, Figure 11 shows the

error measures for varying  $\gamma$ -parameter and the different orders (I), (II), and (III) of filtering. The final error values were obtained at the peak structural similarity value (SSIM) [55]. The reason for the improved performance is easiest explained in terms of a dynamic system: by including the nonlinear mapping of the intensity range as a component in the filtering scheme, it acts as a feedback component adjusting the diffusivity to conform with the final visualized result. This behaviour is also illustrated in Figure 1, Section 1.

In addition to the error graph, we show the final result for the mapping-based filter of Cameraman and Boat as  $\gamma = 0.4$  in Figure 10. It is visible that most of the noise has been removed and that edges remain well-preserved.

## 8 Application II: Targeted value range filtering

In this section we present an application where the image value range is set from prior information based on user interaction.

Figure 3 in the introduction illustrates a fixed selection of the image value range where we know which value range is noisy prior to filtering. Thus, we can achieve high quality result where we preserve fine details in the image. These results are not possible to obtain with state of the art denoising methods such as





**Fig. 12** Two examples of targeted value range filtering from user selected regions. On the first row we focus on the text. As visible in the close-up in the Result-column the noise is removed in the white text whereas and preserved in the red background. On the second row we instead choose to filter the red color of the soda can, and as visible in the Result-column the red is denoised whereas the noise remains in the white text. The column labeled “Mapping” is computed from the fitted (normalized to 1) density functions in the right-most column. See text for details.

MRF [52], TGV [16] or BM3D [24] since they denoise the complete image value range.

### 8.1 User-based value-mapping

In the introduction we also gave an initial example of user-based region selected value range filtering, see the first row of Figure 2 which shows a result when filtering the butterfly’s wings. In the same figure, the corresponding mapping shows the region of interest that is processed by our algorithm. The denoised result is shown in the right column. It is visible that the noise in the desired region is reduced meanwhile noise in other value ranges are preserved.

We give an additional example in Figure 12. In the right most column we also show the histogram for the selected color region in the left column. The histogram shows the sample value-distribution of each color channel and the corresponding estimated Gaussian distribution. These distributions make up our channel specific mapping functions. In this targeted-value range application we first reduce the noise of the letters written on the soda can. This is achieved by selecting a sample region in the image outlined on the letter “C”, displayed on the first column in the first row. The resulting mapping shows that the algorithm is targeting the desired

region. A close-up in the result-column shows that the filtering is confined to the letters - leaving the noise in the remainder of the value range undisturbed.

Next, we mark a region with red color and filter the image. The corresponding close-up on the second row shows that the red color is free of noise and that the white color of the letters still contains noise, just as desired. This type of filtering process is beneficial when the application’s noise-component is known prior to filtering. Also, there may be cases when one can tolerate the preservation of noise meanwhile other regions are smoothed.

When performing the filtering process the fitted density functions are normalized in the same way as was done in the application of gamma correction in the previous section.

## 9 Application III: Denoising via an oversegmentation map

In the third application we treat the problem of global image denoising, *i.e.*, in contrast to applications I and II, we now consider the whole (spatial) image domain  $\Omega$ . Here, the mapping function will be estimated from locally homogeneous regions via an oversegmentation map.



### 9.1 Method definitions

We define the tensor  $W$  and the mapping in functional (28) to describe isotropic diffusion (ISO), isotropic diffusion with a mapping function (ISOm) [5], total variation (TV) [50], total variation with a mapping function (TVm), see (52), gradient energy tensor total variation (GETV) [2] and GETV with a mapping function (GETVm), see (65). The definitions of these methods are outlined in Table 2. The diffusivity parameter in (57) for GETV and GETVm is set according to

$$k^2 = \frac{\exp(1) - 1}{\exp(1) - 2} \sigma^2, \quad (79)$$

where  $\sigma^2$  is the image noise variance [26]. In the following evaluation the aim is not to show superiority over all existing denoising algorithms. We rather investigate how an oversegmentation map affects the final denoised result and we believe interesting results will follow from this approach. To get comparative performance result of the below outlined approach, we compare our formulation to original code of state of the art denoising methods BM3D [24], TGV [16] and MRF [52].

### 9.2 Approach and motivation

Objects in an image, or more specifically, their value distributions, can be described by a stochastic process. Such processes can be estimated from image segments, or homogeneous regions, via an oversegmentation map. Figure 13 shows one such partitioning of similar regions. In this case, each region (or object) is associated with its own set of features [54]. In image denoising the aim is to remove the noise while preserving image features. In other words we want to obtain a “clean” image without the noise component, *i.e.*, we want to infer the true underlying statistical process that describes the noise free image. Therefore, modelling the image filtering process based on estimated densities is a means to incorporate a prior into the filtering scheme. Here, the aim is to separate objects based on certainty estimates originating from the underlying distributions. In practice we rely on the principle of ergodicity to estimate first and second order moments of each segments sample distribution. Additionally, we assume uncorrelated pixels between segments to drive the diffusion process. The idea is to exploit properties of statistical sample distributions in a way such that we can reduce the uncertainty to which segment a pixel, located at the border of two segments, belongs to. This is achieved by incorporating the statistical properties of pixel density functions into the filtering scheme.

Based on this motivation we let the derivative of the mapping function, *i.e.*,  $m'$ , describe the sample distribution. Thus, the mapping function can be interpreted as the corresponding cumulative density function.

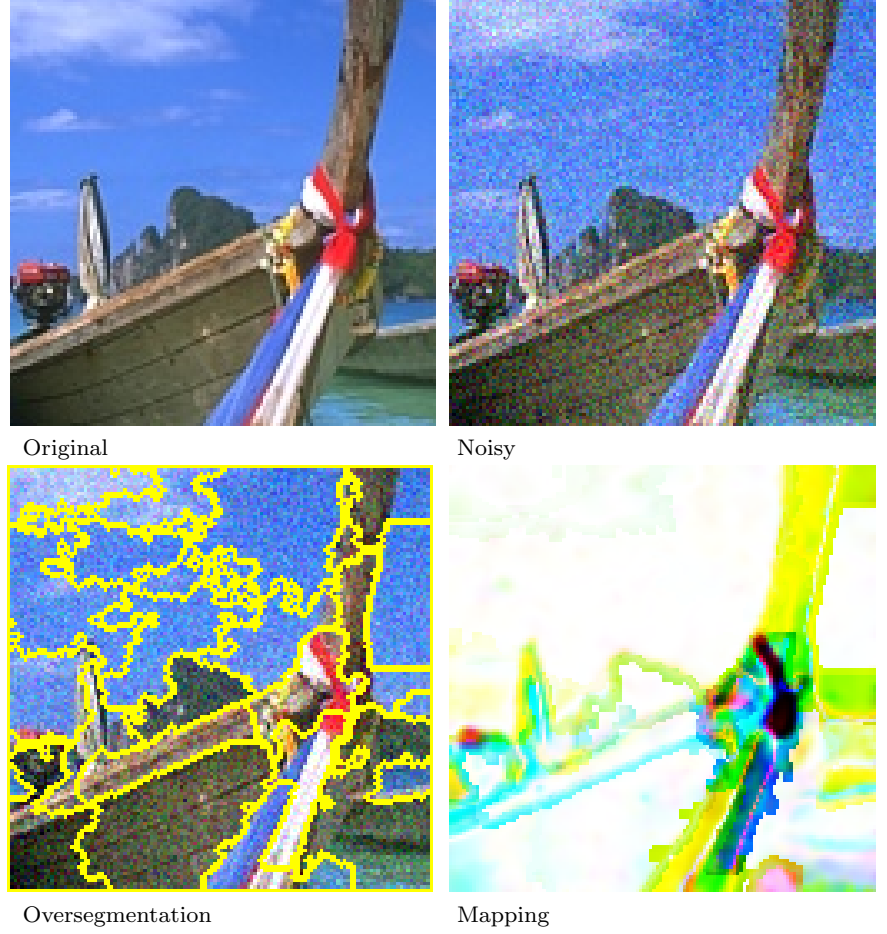
In this application, our approach to image denoising is a four step procedure: (I) Compute an oversegmentation that conforms to homogeneous regions. (II) In the case of natural colour images, transform the colour space into colour opponent components. (III) Construct the mapping derivative,  $m'$ , from estimated density functions. (IV) Solve the corresponding E-L equation for a given  $W$ . We use the CIELAB colour representation of the noisy images to decorrelate its RGB components [53]. In the following we introduce the concept of oversegmentation maps and show how they can be used to compute mapping functions to drive the diffusion process.

### 9.3 Oversegmentation

The introduction of a mapping into the diffusion framework was first presented in [5]. In this work we extend the initial formulation to include a tensor component, this is enabled by the functional in Theorem 1. However, before specifying the filtering method, we explain how to obtain an oversegmentation map to estimate the image statistics.

The selection of image regions suitable for image denoising is a non-trivial task. Much research has focused on devising efficient methods to partition images into homogeneous areas that adhere to image structure boundaries. It is not until recently these partitioning methods are efficient enough to produce sufficiently accurate regions for image denoising while achieving computation times suitable for practical applications. One oversegmentation method that satisfy these demands is SEEDS, superpixels extracted via energy-driven sampling [9]. With the assumption that each segment in the oversegmentation map is independent, we estimate local second order statistics for each segment. A sensitivity analysis on the selection of number of regions was done in [5], which showed that the filtering process is robust to the number of segments used in the oversegmentation map. Furthermore, note that the selection of oversegmentation method is not unique. We expect that as the oversegmentation methods continue to improve boundary localizations, they may further accelerate their applicability in denoising applications.

**Fig. 13** Image “Boat”. Top left original image, right is the degraded image corrupted with additive Gaussian noise with standard deviation 20. Down left, oversegmentation map and down right, is the obtained mapping function.



#### 9.4 Estimating a mapping

The spatial extent of a segment in the oversegmentation map should conform to approximately homogeneous regions. If the requirement is fulfilled, then the observed data (or pixels) within one region can be assumed to originate from the same stochastic process and modeled using the same density function. We use this idea to derive a confidence value that describes the certainty that a sample (pixel) belongs to a region or not, *e.g.*, a noisy observation would have a low confidence. We model the mapping  $m$  to reflect the confidence level that a sample belongs to a neighbourhood. To do so we define the mapping from a local neighbourhood  $\mathcal{L}$ . Assuming independence between each segment, we model the data sample in the neighbourhood using a Gaussian distribution with associated mean  $\mu$  and variance  $\sigma^2$ . In practice we estimate the statistical moments using the sample mean and variance of the segment. The mapping function is defined as

$$m'(u; \mu, \sigma) = \prod_{i \in \mathcal{L}} g(u; \mu_i, \sigma_i), \quad (80)$$

where

$$g(u; \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} \frac{(u - \mu_i)^2}{\sigma_i^2}\right). \quad (81)$$

Thus, instead of controlling the filtering based on edge-strength, *e.g.*, as in P-M (11), the filtering is now defined by  $m'$ , the certainty that an observed sample belong to the local neighbourhood. The selection of neighbourhood  $\mathcal{L}$  is arbitrary, but for simplicity we choose its size to be  $3 \times 3$ . In (80) we use the product of the probability density functions (PDF) rather than the sum which would yield a proper Gaussian mixture model (GMM). The motivation is that at edges, the GMM would not reduce the filtering as efficiently as in the case of a product mapping function. When filtering colour images we compute one oversegmentation of the noisy image. This map is then used to compute the density estimates in each channel of the colour image representation.



**Fig. 14** Comparison of visual quality. The Boat’s bow appears more clear where we have applied the mapping function, compared to ISO, TV and GETV. Visually, TGV appear similar to TVm and GETVm. While ISO and ISOm are oversmoothed, MRF and GETV exhibits color artifacts in the blue sky. Although, BM3D has good visual and good error values (Table 4), its approach to denoising is *fundamentally* different from the PDE-based method introduced in this work.

## 9.5 Results

We already mentioned in the introductory paragraph of this section that we do not aim to outperform state of the art denoising methods. Rather we study how the usage of an oversegmentation map influence the noise removal. Our PDE-framework defines a natural way of

reducing the filtering process at border regions rich with detail and texture, so as to not oversmooth corners and lines. We show that the mapping function is the key component for this approach.

Figure 14 illustrates a result for all considered methods, obtained at their maximum SSIM values. The right

	PSNR	SSIM
Noisy	22.10	0.47
ISO	25.09	0.75
ISOm	25.36	0.76
TV	26.71	0.80
TVm	26.66	0.80
GETV	27.60	0.77
GETVm	26.70	0.80
TGV	27.98	0.82
BM3D	31.36	0.90
MRF	28.26	0.81

**Table 4** Error values for the Boat image in Figure 13 and 14. We see that the mapping function in particular improves the SSIM values. Compared to TGV, BM3D and MRF our results are most similar to TGV and MRF.

illustration in Figure 13 shows the mapping for GETVm. We see that the red color-region on the bow of the Boat has high uncertainty, thus in this region, the mapping-based algorithms will reduce the filtering. Note that the color coding is in the RGB-space in the mapping-figure although the actual filtering is done in the CIELAB-space. This means that red color in the mapping corresponds to filtering in the CIELAB intensity channel whereas green and blue describe the opponent color channels. White color indicates that each CIELAB-value is well described by the estimated sample distributions and therefore we have high filtering in each CIELAB channel. Moreover, the color cyan indicates filtering of only the chroma information, not the structural information present in the intensity channel. Due to the mapping, the filtering is reduced in regions with high uncertainty. Example of such regions are the sail and the mountain in the background, see, *e.g.*, ISOm in Figure 14. Here, this gives an improved visual appearance without oversmoothing details in these regions while suppressing the noise in the background. With respect to the error measures for the Boat image, see Table 4, it is obvious that BM3D outperforms all compared methods. However, GETV is on-par with TGV and MRF. The mapping makes the most difference for GETV, showing more improvement than any of the other approaches. Although the PSNR value decreases marginally for TVm compared to TV, the denoised result in Figure 14 appears less oversmoothed for TVm.

Figure 15 shows the original and the noisy image of Lena. The same figure also depicts the mapping produced by the GETVm method. Figure 16 illustrates the result of filtering the Lena image for the respective method. In this case, image regions are well-preserved by the mapping, *e.g.*, the brim of the hat and the hair. Note that the mapping-based methods have a tendency

	PSNR	SSIM
Noisy	22.08	0.40
ISO	27.69	0.80
ISOm	28.09	0.81
TV	29.05	0.83
TVm	29.32	0.83
GETV	29.12	0.76
GETVm	29.42	0.84
TGV	30.03	0.85
BM3D	33.10	0.91
MRF	29.71	0.82

**Table 5** Error values for the Lena image in Figure 15 and 16. In this case the mapping improves all error values. Compared to TGV, BM3D and MRF our results are most similar to TGV and MRF.

to oversmooth features in highly textured regions, *e.g.*, the feathers in Lena’s hat. This is due to the violation of homogeneous partitioning of the oversegmentation map, clearly visible in Figures 13 and 15. This is a frequent problem for these types of image structure. With respect to the error measures, Table 5, we see that BM3D is again superior to all considered methods. We also note that the introduction of mapping function does improve the ISO, TV and GETV error values and visual appearance.

## 10 Discussion and summary of results

This work introduces an alternative viewpoint to image filtering, as such, we discuss the usability of the method and give some additional interpretations.

The novelty of our approach is the idea of including a mapping of the image value range in the variational framework. To this end, the resulting PDE does not only describe spatially-adaptive filtering but also depends on the image value range. Important to note is that the mapping  $m$  can be defined as either a *local* or a *global* function. By this we mean that the mapping can be defined locally, *i.e.*, in a region, or alternatively, globally as in the application of gamma correction where the mapping is applied pixel-wise. We shortly elaborate on the difference between these classes of mapping functions.

For interesting image processing applications, *e.g.*, the ones considered in this work, it is formally necessary to decouple the derived Euler-Lagrange equation from the initial variational formulation. The reason is that none of the complex definitions of the mapping and the nonlinear mappings of the tensor  $\text{GET}^+$  allows us obtain an energy functional on the form (17). This is due to the result in Proposition 1. However, note that

**Fig. 15** Image “Lena”. Top left original image, top right shows the degraded image corrupted with additive Gaussian noise of standard deviations 20. Down left shows the oversegmentation map of the noisy image and down right illustrates the obtained mapping function.



the decoupling is only in the case GET has negative eigenvalues. For an extended discussion of GET and its properties see [28]. If we instead consider the structure tensor: the energy and the E-L equation are *fully* decoupled due to the convolution operator, we again refer to Proposition 1.

The versatility of our framework allows a large number of image processing problems to be considered. In this work we illustrate applications of image denoising. We hypothesize that the mapping can be used in, *e.g.*, image value range compression, visualization, deblurring and inpainting, etc. We also see possible extensions to estimate (or learn) the mapping from datasets of images or self-similarity measures of patches in an image.

Naturally our approach is not the only way to achieve any of the considered applications. However, it is a reasonable assumption that coupling the filtering process with the nonlinear mapping is a preferred formulation. Our method is a framework that provides a natural and explicit coupling for image enhancement: it relates theoretical expositions with user-defined as well as data-specific priors. This versatility, comes at a cost of performance. In fact, one can tailor methods to solve each

application independently, this is what BM3D has done with great success for the application of image denoising. Although heuristic methods can give better performance w.r.t. error measures, their scalability to other applications remain limited.

To further motivate our approach and clarify its potential we wish to highlight specific differences between the mapping functions we have considered in the respective applications. In particular we discuss, what we call, *local* and *global* mapping functions and in what context the mapping is a suitable approach.

Table 1, Section 1, gives an overview of the different mapping functions considered in this work.

### 10.1 Global mappings

A global mapping is a mapping where each image sample (pixel) is mapped with the same definition of the mapping function, *i.e.*,  $u \mapsto m(u)$ .

This type of mapping is used in the first application on gamma correction. We emphasize that in this case the mapping does not provide any additional *explicit*





**Fig. 16** Comparison of visual quality. The brim of Lena’s hat appears more clear where we have applied the mapping function, compared to ISO, TV and GETV. As expected, ISO and ISOm are oversmoothed. TVm, GETVm and TGV all have similar visual appearance. In this example image, there are no apparent color artifacts. Visually, Lenas hair retains more details in BM3D, TGV and MRF compared with TVm and GETVm.

edge information of the image. However, the mapping does provide information how the image values *should* map, *e.g.*, to correct for the gamma distortion in the final result. Recall that the property of the mapping is that if a line/corner, or feature, is present in the image value range after the mapping, then the feature is

preserved. If the structure is not present in the value range after the mapping then the feature is not preserved. Again, although the purely global mapping formulation does not give an explicit edge-stopping effect, the *mapping will aid the discontinuity preservation* induced by the PDE given that the *feature information*



is present after the mapping. Moreover, since the mapping is nonlinear, a weak edge in the input data may be transformed to a strong edge as the mapping is applied. Thus, the mapping gives an *implicit* edge-preservation effect. Next we contrast the global mapping function with the local mapping.

## 10.2 Local mapping

We have considered two types of local mappings. First we defined a local mapping of the color image value range (Application II), where we target the filtering process to a certain color range. The selection of the color range is based on user-input but may also be generalized to prior value ranges based on application requirements. The advantage of having a coupled value and spatial-range filtering process is clear in this targeted application: one does not need to explicitly handle the transition (border) between filtered and unfiltered value range data. This is implicitly handled by the smoothness of the mapping function.

In the third application, we defined an oversegmentation map where each segment in the map describe a local homogeneous region. From these segments we estimate local density functions and via a mixture model we achieve an *edge stopping effect* between the segments. This edge-stopping effect can be seen in Figure 2 (second row, middle figure) and in the mappings in Figure 13 and 15, respectively. Individually, density functions do not provide any information about edge-localization, rather it is the *combination* of densities that prove edge-localization. This combination of mappings depends on a large number of pixels and their relation. This is comparable, but goes beyond the local estimation of the edge-stopping function in P-M diffusion etc. The method used to produce the oversegmentation map is not unique. Moreover, one can apply any method that produces an oversegmentation map. Naturally, the effect of the mapping in the filtering process will depend on the accuracy of detected segment boundaries.

## 11 Conclusions

This work introduced a mapping of the image value range into a variational framework. The key component is the mapping function that allows us to replace the ad-hoc edge-stopping function otherwise present in diffusion schemes. After adopting the mapping function to a set of image processing problems we conclude that the mapping contributes positively to the filtering process. Moreover, we have shown that the map-

ping can be estimated (or set) for each application in a non-parametric way. Our main finding is that context aware tensor-based image filtering can be derived from principles of energy minimization. However, with respect to general image denoising, state of the art denoising methods outperform our framework. If we consider targeted value range applications our methodology clearly demonstrates advantageous aspects by improving feature preservation while reducing the image noise. It is expected that, under appropriate mappings, related energy based problem will benefit from the presented work. Naturally, assuming a reasonable map can be formulated, the direct extension to these alternative applications is to generalize existing regularization terms.

**Acknowledgements** We thank the reviewers for their helpful comments and suggestions which have improved this work. This research has received funding from the Swedish Foundation for Strategic Research through the grant VPS and from Swedish Research Council through grants for the projects energy models for computational cameras (EMC<sup>2</sup>) and Visualization adaptive Iterative Denoising of Images (VIDI), all within the Linnaeus environment CADICS and the excellence network ELLIIT. Support by the German Science Foundation and the Research Training Group (GRK 1653) is gratefully acknowledged by the first author.

## A Proof of Theorem 1

To prove Theorem 1, compute the variational derivative of the functional (28). The first variation is given by

$$\delta R = \left. \frac{\partial}{\partial \varepsilon} R(u + \varepsilon v) \right|_{\varepsilon=0}. \quad (82)$$

Now, we let  $u \mapsto u + \varepsilon v$  in  $R(u)$  and we get

$$R(u + \varepsilon v) = \nabla m(u + \varepsilon v)^\top W(\nabla m(u + \varepsilon v)) \nabla m(u + \varepsilon v) \\ = \underbrace{m'(u + \varepsilon v)^2 \nabla(u + \varepsilon v)^\top}_{=A} \quad (83a)$$

$$\cdot \underbrace{W(\nabla m(u + \varepsilon v)) \nabla(u + \varepsilon v)}_{=B}, \quad (83b)$$

then, from the product rule of differentiation, we need to consider the following terms

$$\frac{\partial}{\partial \varepsilon} R(u + \varepsilon v) = \left( \frac{\partial}{\partial \varepsilon} A(u + \varepsilon v) \right) B \\ + A \left( \frac{\partial}{\partial \varepsilon} B(u + \varepsilon v) \right). \quad (84)$$

To simplify the notation define

$$z = \begin{pmatrix} z_1 = m'(u + \varepsilon v)(u_1 + \varepsilon v_1) \\ \vdots \\ z_d = m'(u + \varepsilon v)(u_d + \varepsilon v_d) \end{pmatrix}, \quad (85)$$

and note the relation

$$z|_{\varepsilon=0} = \nabla m(u) = s. \quad (86)$$

We first consider the differentiation of the  $B$ -component w.r.t  $\varepsilon$ . In order to differentiate the  $B$ -component use the chain rule of differentiation:

$$\frac{\partial W(z)}{\partial \varepsilon} = \frac{\partial W(z)}{\partial z_1} \frac{\partial z_1}{\partial \varepsilon} + \dots + \frac{\partial W(z)}{\partial z_d} \frac{\partial z_d}{\partial \varepsilon},$$

then

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} B(z) &= \frac{\partial}{\partial \varepsilon} (W(z) \nabla(u + \varepsilon v)) \\ &= \left( \frac{\partial}{\partial \varepsilon} W(z) \right) \nabla(u + \varepsilon v) + W(z) \nabla v, \end{aligned} \quad (87)$$

with

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} W(z) &= \left[ W_{z_1}(z) \left( m''(u + \varepsilon v) v (u_1 + \varepsilon v_1) \right. \right. \\ &\quad \left. \left. + m'(u + \varepsilon v) v_1 \right) \right. \\ &\quad \vdots \\ &\quad \left. + W_{z_d}(z) \left( m''(u + \varepsilon v) v (u_d + \varepsilon v_d) \right. \right. \\ &\quad \left. \left. + m'(u + \varepsilon v) v_d \right) \right], \end{aligned} \quad (88)$$

evaluating the limit  $\varepsilon \rightarrow 0$  in (87) with (88) we get

$$\begin{aligned} \left. \frac{\partial}{\partial \varepsilon} B(z) \right|_{\varepsilon=0} &= \left[ W_{s_1}(s) \left( m''(u) v u_1 + m'(u) v_1 \right) \right. \\ &\quad \vdots \\ &\quad \left. + W_{s_d}(s) \left( m''(u) v u_d + m'(u) v_d \right) \right] \nabla u \\ &\quad + W(s) \nabla v. \end{aligned} \quad (89)$$

Next, we focus on the second product in (84). Then, with (89) and after few rewrites, we obtain the following

$$\begin{aligned} A \left( \frac{\partial}{\partial \varepsilon} B(u + \varepsilon v) \right) \Big|_{\varepsilon=0} &= m'(u)^2 \nabla u^\top \left[ W_{s_1}(s) \left( m''(u) v u_1 + m'(u) v_1 \right) \right. \\ &\quad \vdots \\ &\quad \left. + W_{s_d}(s) \left( m''(u) v u_d + m'(u) v_d \right) \right] \nabla u \\ &\quad + m'(u)^2 \nabla u^\top W(s) \nabla v \end{aligned} \quad (90a)$$

$$\begin{aligned} &= m'(u)^2 \left[ \nabla u^\top W_{s_1}(s) \nabla u \left( m''(u) v u_1 + m'(u) v_1 \right) \right. \\ &\quad \vdots \\ &\quad \left. + \nabla u^\top W_{s_d}(s) \nabla u \left( m''(u) v u_d + m'(u) v_d \right) \right] \\ &\quad + m'(u)^2 \nabla u^\top W(s) \nabla v \end{aligned} \quad (90b)$$

$$\begin{aligned} &= m'(u)^2 \left\langle Q, \left( m''(u) v \nabla u + m'(u) \nabla v \right) \right\rangle \\ &\quad + m'(u)^2 \left\langle \nabla u, W(s) \nabla v \right\rangle \end{aligned} \quad (90c)$$

$$\begin{aligned} &= \underbrace{\left\langle m'(u)^2 m''(u) Q, \nabla u \right\rangle}_G v \\ &\quad + \left\langle \nabla v, \underbrace{\left( m'(u)^3 Q + m'(u)^2 W(s)^\top \right)}_{=H} \nabla u \right\rangle, \end{aligned} \quad (90d)$$

where we set

$$Q = \begin{pmatrix} \nabla u^\top W_{s_1}(s) \nabla u \\ \vdots \\ \nabla u^\top W_{s_d}(s) \nabla u \end{pmatrix} = \nabla_u W_s(s) \nabla u, \quad (91)$$

and  $\nabla_u W_s(s)$  is defined in (26). The definitions  $G$  and  $H$  will be used in the application of Green's formula (17). Now we focus on the differentiation of the  $A$ -component w.r.t.  $\varepsilon$  in (83a), i.e., we set  $u \mapsto u + \varepsilon v$  and get

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} A(u + \varepsilon v) &= 2m'(u + \varepsilon v) m''(u + \varepsilon v) v \nabla(u + \varepsilon v)^\top \\ &\quad + m'(u + \varepsilon v)^2 \nabla v^\top. \end{aligned} \quad (92)$$

With (92) we evaluate the limit  $\varepsilon \rightarrow 0$  of (83a) which results in

$$\begin{aligned} &\left( \frac{\partial}{\partial \varepsilon} A(u + \varepsilon v) \right) B \Big|_{\varepsilon=0} \\ &= \left( 2m'(u) m''(u) v \nabla u^\top + m'(u)^2 \nabla v^\top \right) W(\nabla m(u)) \nabla u \\ &= \underbrace{\left( 2m'(u) m''(u) \nabla u^\top W(\nabla m(u)) \nabla u \right)}_E v \\ &\quad + \nabla v^\top \underbrace{\left( m'(u)^2 W(\nabla m(u)) \right)}_F \nabla u, \end{aligned} \quad (93)$$

where the definitions  $E$  and  $F$  will be used in the application of Green's theorem. Summing (90d) and (93) and making use of  $G, H$  and  $E, F$  yields

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} R(u + \varepsilon v) \Big|_{\varepsilon=0} &= \int_{\Omega} (G + E) v \, dx \\ &\quad + \int_{\Omega} \nabla v^\top (H + F) \nabla u \, dx. \end{aligned} \quad (94)$$

The second integral of (94) is now on the form  $\nabla v^\top A(\nabla u)$ , i.e., we can integrate it w.r.t.  $\nabla v$  by using Green's formula (cmp. (17)) and get

$$\begin{aligned} &\int_{\Omega} \nabla v^\top (H + F) \nabla u \, dx \\ &= \int_{\partial \Omega} v (\mathbf{n} \cdot (H + F) \nabla u) \, dS - \int_{\Omega} v \operatorname{div}((H + F) \nabla u) \, dx, \end{aligned}$$

where the natural Nuemann boundary condition is given by  $\mathbf{n} \cdot (H + F) = 0$ . Substituting  $G, H, E, F$  defined in (90d) and (93) into the above expression, and after rearranging the terms and dropping the parentheses for improved clarity, we get

$$\begin{cases} m' m'' \nabla u^\top [2W + \nabla_u W_s(s) m'] \nabla u \\ - \operatorname{div}((m')^2 (W + W^\top + m' \nabla_u W_s(s)) \nabla u) = 0 \text{ in } \Omega \\ \mathbf{n} \cdot (F + H) \nabla u = 0 \text{ on } \partial \Omega \end{cases} \quad (95)$$

The above E-L equation can be simplified by considering the following expansion

$$\begin{aligned} &\operatorname{div}((m')^2 [2W + \nabla_u W_s(s) m'] \nabla u) \\ &= 2m' m'' \nabla u^\top [2W + \nabla_u W_s(s) m'] \nabla u \\ &\quad + (m')^2 \operatorname{div}((2W + \nabla_u W_s(s) m') \nabla u). \end{aligned} \quad (96)$$

Substituting the first term of the right hand side of (96) into the PDE of (95) we get:

$$\begin{aligned} & \operatorname{div}((m')^2[2W + \nabla_u W_s(s)m']\nabla u) \\ & - (m')^2 \operatorname{div}([2W + \nabla_u W_s(s)m']\nabla u) \\ & - 2 \operatorname{div}((m')^2[W + W^\top + m' \nabla_u W_s(s)]\nabla u) = 0. \end{aligned} \quad (97)$$

The final result is obtained after subtracting the last term from the first term in (97), which concludes the proof.  $\square$

## B Proof of Corollary 4

To derive the gradient energy tensor diffusion scheme we set  $W$  and  $D$  as specified in (59). With this relation between  $W$  and  $D$  we introduce  $E$  and set  $E = 1/|\nabla m(u)|$ , then  $W = DE$ . Before proceeding with the proof, we specify the components of GET. The components of GET (54) are

$$a = (\partial_x u_x)^2 + (\partial_x u_y)^2 - u_x(\partial_{xx} u_x + \partial_{xy} u_y) \quad (98a)$$

$$\begin{aligned} b &= (\partial_x u_x)(\partial_x u_y) + (\partial_y u_x)(\partial_y u_y) \\ & - \frac{1}{2}(u_x(\partial_{yx} u_x + \partial_{yy} u_y) + u_y(\partial_{xx} u_x + \partial_{xy} u_y)) \end{aligned} \quad (98b)$$

$$c = (\partial_y u_y)^2 + (\partial_y u_x)^2 - u_y(\partial_{yx} u_x + \partial_{yy} u_y). \quad (98c)$$

Now we need to compute  $\nabla_u W \nabla_u (\nabla u)$  in (27) with  $s = \nabla u$ , that is, we have

$$\nabla_u W \nabla_u (\nabla u) = \begin{pmatrix} \nabla u^\top E_{u_x} D \\ \nabla u^\top E_{u_y} D \end{pmatrix} + E \begin{pmatrix} \nabla u^\top D_{u_x} \\ \nabla u^\top D_{u_y} \end{pmatrix}. \quad (99)$$

The derivatives of  $E$  reads

$$E_{u_x} = \partial_{u_x} \frac{1}{|\nabla u|} = -\frac{1}{|\nabla u|^3} u_x, \quad (100a)$$

$$E_{u_y} = \partial_{u_y} \frac{1}{|\nabla u|} = -\frac{1}{|\nabla u|^3} u_y. \quad (100b)$$

The tensor  $\nabla_u W \nabla_u (\nabla u)$  in the E-L scheme, can now be expressed as (62).  $\square$

## C Eigendecomposition

This part decomposes  $D$  in its eigendecomposition and computes  $D_{u_x}$  and  $D_{u_y}$ . We have

$$D(\nabla u) = U \Lambda U^\top = \begin{pmatrix} v_1^2 & v_1 v_2 \\ v_1 v_2 & v_2^2 \end{pmatrix} \lambda_1 + \begin{pmatrix} w_1^2 & w_1 w_2 \\ w_1 w_2 & w_2^2 \end{pmatrix} \lambda_2,$$

where  $\lambda_{1,2} = \exp(-|\mu_{1,2}|/k^2)$  and  $\mu_1 = \frac{1}{2}(\operatorname{tr}(GET) + \alpha)$  and  $\mu_2 = \frac{1}{2}(\operatorname{tr}(GET) - \alpha)$  are the eigenvalues of GET with  $\alpha = \sqrt{(a-c)^2 + 4b^2}$ . We obtain the eigenvectors of GET by solving  $GET \tilde{v} = \mu_1 \tilde{v}$ , i.e., we have the following equation system

$$\begin{cases} (a - c - \alpha) \tilde{v}_1 + 2b \tilde{v}_2 = 0 \\ 2b \tilde{v}_1 + (c - a - \alpha) \tilde{v}_2 = 0 \end{cases}. \quad (101)$$

The orthonormal eigenvectors of GET are (67) and (68). Now, we focus on  $\partial_{u_x} D$  in (66). After expanding the derivatives of the eigenvectors we obtain

$$\partial_{u_x} \tilde{v}_1 = -(\partial_{yx} u_x + \partial_{yy} u_y), \quad (102a)$$

$$\partial_{u_x} \tilde{v}_2 = \partial_{xx} u_x + \partial_{xy} u_y + \partial_{u_x} \alpha, \quad (102b)$$

$$\partial_{u_x} \tilde{w}_1 = \partial_{u_x} \tilde{v}_1, \quad (102c)$$

$$\partial_{u_x} \tilde{w}_2 = \partial_{xx} u_x + \partial_{xy} u_y - \partial_{u_x} \alpha, \quad (102d)$$

and

$$\partial_{u_x} |\tilde{v}| = |\tilde{v}|^{-1}(\tilde{v}_1(\partial_{u_x} \tilde{v}_1) + \tilde{v}_2(\partial_{u_x} \tilde{v}_2)), \quad (103a)$$

$$\partial_{u_x} |\tilde{w}| = |\tilde{w}|^{-1}(\tilde{w}_1(\partial_{u_x} \tilde{w}_1) + \tilde{w}_2(\partial_{u_x} \tilde{w}_2)). \quad (103b)$$

The derivatives of the tensor's eigenvalues are:

$$\begin{aligned} \partial_{u_x} \lambda_{1,2} &= \partial_{u_x} \exp(-|\mu_{1,2}|/k^2) \\ &= -\frac{\operatorname{sgn}(\mu_{1,2})}{k^2} (\partial_{u_x} \operatorname{tr}(GET) \pm \partial_{u_x} \alpha) \lambda_{1,2}, \end{aligned} \quad (104)$$

where

$$\partial_{u_x} \alpha = \alpha^{-1} (\operatorname{tr}(GET) \partial_{u_x} \operatorname{tr}(GET) - 2 \partial_{u_x} \det(GET)) \quad (105)$$

and

$$\partial_{u_x} \operatorname{tr}(GET) = -(\partial_{xx} u_x + \partial_{xy} u_y), \quad (106a)$$

$$\begin{aligned} \partial_{u_x} \det(GET) &= \partial_{u_x} (ac - b^2) \\ &= -(\partial_{xx} u_x + \partial_{xy} u_y) c \\ &\quad + b(\partial_{yx} u_x + \partial_{yy} u_y). \end{aligned} \quad (106b)$$

In the case  $b = 0$  the difference to the above derivation is that  $\det(GET) = ac$ , thus  $\partial_{u_x} \alpha$  and  $\partial_{u_x} \det(GET)$  should be modified accordingly. The component  $\nabla u^\top D_{u_y}$  follows the same line of calculations and is therefore omitted here.

## References

- Åström, F., Baravdish, G., Felsberg, M.: On Tensor-Based PDEs and their Corresponding Variational Formulations with Application to Color Image Denoising. In: ECCV12, vol. 7574, pp. 215–228. LNCS, Springer Berlin/Heidelberg, Firenze (2012)
- Åström, F., Baravdish, G., Felsberg, M.: A tensor variational formulation of gradient energy total variation. In: X.C. Tai, E. Bae, T. Chan, M. Lysaker (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), LNCS, vol. 8932, pp. 307–320. Springer International Publishing (2015)
- Åström, F., Felsberg, M.: On the choice of tensor estimation for corner detection, optical flow and denoising. In: C. Jawahar, S. Shan (eds.) Computer Vision - ACCV 2014 Workshops, LNCS, vol. 9009, pp. 16–30. Springer International Publishing (2015)
- Åström, F., Felsberg, M., Baravdish, G., Lundström, C.: Targeted Iterative Filtering. In: SSVM2013, June, pp. 1–11. Springer Berlin/Heidelberg (2013)
- Åström, F., Zografos, V., Felsberg, M.: Density Driven Diffusion. In: SCIA, pp. 17–20 (2013)
- Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition), *Applied Mathematical Sciences*, vol. 147. Springer, Berlin (2006)
- Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-In by Joint Interpolation of Vector Fields and Gray Levels. *IEEE Transactions on Image Processing* **10**(8), 1200–1211 (2001)
- Baravdish, G., Svensson, O., Åström, F.: On Backward  $p(x)$ -Parabolic Equations for Image Enhancement. *Numerical Functional Analysis and Optimization* **36**(2), 147–168 (2015)

9. den Bergh, M.V., Boix, X., Roig, G., de Capitani, B., Gool, L.V.: SEEDS: Superpixels Extracted via Energy-Driven Sampling. In: A.W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) ECCV12, LNCS, vol. 7578, pp. 13–26. Springer (2012)
10. Bigün, J., Granlund, G.H.: Optimal Orientation Detection of Linear Symmetry. In: Proceedings of the IEEE First ICCV, pp. 433–438. London, Great Britain (1987)
11. Black, M.J., Rangarajan, A.: On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision. IJCV **19**(1), 57–91 (1996)
12. Blomgren, P., Chan, T.F., Mulet, P., Wong, C.: Total Variation Image Restoration: Numerical Methods and Extensions. In: in Proc. IEEE ICIP, pp. 384–387 (1997)
13. Bollt, E.M., Chartrand, R., Esedoğlu, S., Schultz, P., Vixie, K.R.: Graduated Adaptive Image Denoising: Local Compromise Between Total Variation and Isotropic Diffusion. Advances in Computational Mathematics **31**(1), 61–85 (2008)
14. Bovik, A.C., Maragos, P.: Conditions for positivity of an energy operator. Signal Processing, IEEE Transactions on **42**(2), 469–471 (1994)
15. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2004)
16. Bredies, K., Kunisch, K., Pock, T.: Total Generalized Variation. SIAM J. Img. Sci. **3**(3), 492–526 (2009)
17. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Variational optical flow computation in real time. Image Processing, IEEE Transactions on **14**(5), 608–615 (2005)
18. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: CVPR, vol. 2, pp. 60 – 65 vol. 2 (2005)
19. Chan, T., Esedoğlu, S., Park, F., Yip, A.: Total Variation Image Restoration: Overview and Recent Developments. In: N. Paragios, Y. Chen, O. Faugeras (eds.) Handbook of Mathematical Models in Computer Vision, chap. 2, pp. 17–31. Springer US (2006)
20. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. SIAM Journal on Scientific Computing **20**(6), 1964–1977 (1999)
21. Chan, T.F., Kang, S.H., Kang, Shen, J.: Euler’s elastica and curvature based inpaintings. SIAM J. Appl. Math **63**, 564–592 (2002)
22. Chan, T.F., Wong, C.K.: Total Variation Blind Deconvolution. IEEE Transactions on Image Processing **7**(3), 370–375 (1998)
23. Chen, Y., Levine, S., Rao, M.: Variable exponent, linear growth functionals in image processing. SIAM Journal on Applied Mathematics **66**, 1383–1406 (2004)
24. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
25. DiBenedetto, E.: Degenerate parabolic equations. Springer-Verlag, New York (1993)
26. Felsberg, M.: Autocorrelation-Driven Diffusion Filtering. Image Processing, IEEE Transactions on **20**(7), 1797–1806 (2011)
27. Felsberg, M., Granlund, G.: POI detection using channel clustering and the 2D energy tensor. In: Pattern Recognition: 26th DAGM Symposium. Volume 3175 of LNCS, pp. 103–110 (2004)
28. Felsberg, M., Köthe, U.: GET: The connection between monogenic scale-space and Gaussian derivatives. In: R. Kimmel, N. Sochen, J. Weickert (eds.) Scale Space and PDE Methods in Computer Vision, LNCS, vol. 3459, pp. 192–203. LNCS (2005)
29. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: ISPRS Intercommission, Workshop, Interlaken, pp. 149–155. (1987)
30. Gårding, J., Lindeberg, T.: Direct computation of shape cues using scale-adapted spatial derivative operators. IJCV **17**(2), 163–191 (1996)
31. Gilbarg, D., Trudinger, N.: Elliptic Partial Differential Equations of Second Order. Classics in Mathematics. Springer Berlin Heidelberg (2001)
32. Granlund, G.H., Knutsson, H.: Signal processing for computer vision. Kluwer (1995)
33. Grasmair, M., Lenzen, F.: Anisotropic Total Variation Filtering. Applied Mathematics & Optimization **62**(3), 323–339 (2010)
34. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Artificial Intelligence **17**, 185–203 (1981)
35. Horn, R.A., Johnson, C.R. (eds.): Matrix Analysis. Cambridge University Press, New York, NY, USA (1986)
36. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–7 (2007)
37. Kaiser, J.F.: On a simple algorithm to calculate the ‘energy’ of a signal. In: Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on, pp. 381–384 vol.1 (1990)
38. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**(4), 321–331 (1988)
39. Koenderink, J.J.: The Structure of Images. Biological Cybernetics **370**(50), 363–370 (1984)
40. Krajsek, K., Scharr, H.: Diffusion Filtering Without Parameter Tuning : Models and Inference Tools. In: CVPR2010, pp. 2536–2543. San Francisco (2010)
41. Kuijper, A.: Geometrical {PDEs} based on second-order derivatives of gauge coordinates in image processing. Image and Vision Computing **27**(8), 1023 – 1034 (2009)
42. Ladyženskaja, O.A., Solonnikov, V.A., Ural’ceva, N.N.: Linear and Quasilinear Equations of Parabolic Type. Translated from the Russian by S. Smith. Translations of Mathematical Monographs, Vol. 23. American Mathematical Society, Providence, R.I. (1968)
43. Lefkimmatis, S., Roussos, A., Unser, M., Maragos, P.: Convex Generalizations of Total Variation Based on the Structure Tensor with Applications to Inverse Problems. In: A. Kuijper, K. Bredies, T. Pock, H. Bischof (eds.) SSVM, pp. 48–60 (2013)
44. Levin, A., Nadler, B., Durand, F., Freeman, W.T.: ECCV 2012, chap. Patch Complexity, Finite Pixel Correlations and Optimal Denoising, pp. 73–86. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
45. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer international series in engineering and computer science: Robotics: Vision, manipulation and sensors. Springer (1993)
46. Nordberg, K.: Signal Representation and Processing using Operator Groups. Ph.D. thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (1995)
47. Oliveira, J.P., Bioucas-Dias, J.M., Figueiredo, M.A.: Adaptive total variation image deblurring: A majorization-minimization approach. Signal Processing **89**(9), 1683 – 1693 (2009)
48. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions, PAMI **12**(7), 629–639 (1990)

49. Roussos, A., Maragos, P.: Tensor-based image diffusions derived from generalizations of the total variation and beltrami functionals. In: ICIP, pp. 4141–4144 (2010)
50. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1-4), 259–268 (1992)
51. Scherzer, O., Weickert, J.: Relations Between Regularization and Diffusion Filtering. *Journal of Mathematical Imaging and Vision* **12**(1), 43–63 (2000)
52. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on mrfs in low-level vision. In: *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, pp. 1751–1758 (2010)
53. Sharma, G., Bala, R.: *Digital Color Imaging Handbook*. Electrical Engineering & Applied Signal Processing Series. Taylor & Francis (2014)
54. Torralba, A., Oliva, A.: Statistics of natural image categories. *Network: Computation in Neural Systems* pp. 391–412 (2003)
55. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing* **13**(4), 600–612 (2004)
56. Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, Germany (1998)
57. Weickert, J.: Nonlinear Diffusion Filtering. In: B. Jähne, H. Haussecker, P. Beissler (eds.) *Signal processing and pattern recognition, Handbook of Computer Vision and Applications*, chap. 15, pp. 423–451. Academic Press (1999)