



The Filter Echo: A General Tool for Filter Visualisation

Daniel Gaa¹ · Joachim Weickert¹ · Iva Farag¹ · Özgün Çiçek¹

Received: 15 September 2025 / Accepted: 10 April 2026 / Published online: 23 May 2026
© The Author(s) 2026

Abstract

To select suitable filters for a task or to improve existing filters, a deep understanding of their inner workings is vital. Diffusion echoes, which are space-adaptive impulse responses, are useful to visualise the effect of nonlinear diffusion filters. However, they have received little attention in the literature. There may be two reasons for this: firstly, the concept was introduced specifically for diffusion filters, which might appear too limited. Secondly, diffusion echoes have large storage requirements, which restricts their practicality. This work addresses both problems. We introduce the filter echo as a generalisation of the diffusion echo and use it for applications beyond adaptive smoothing, such as image inpainting, osmosis, and variational optic flow computation. We provide a framework to visualise and inspect echoes from various filters with different applications. Furthermore, we propose a compression approach for filter echoes, which reduces storage requirements by a factor of 20–100.

Keywords Diffusion echo · Impulse response · Filter kernel · Singular value decomposition

1 Introduction

Even in times where deep neural networks are highly popular, model-based approaches remain appealing due to their simplicity, transparency, and mathematical foundation. One of the simplest model-based approaches are linear shift-invariant (LSI) filters, for which it is well known that they can be represented by convolutions. Their filter action is not space variant and is determined by the so-called *impulse response*. It describes the filter output of an image with a unit impulse in the origin in the discrete case, or a delta peak in the continuous setting.

Since LSI filters may be too limited for various practical applications, nonlinear adaptive filters have been introduced. In the case of denoising applications, they allow e.g. to create structure-preserving results. Obviously, these highly adaptive filters cannot have a simple impulse response that characterises and visualises the filter behaviour at all locations. By design, their action is space variant and adapts itself to the original image. In some cases, this action is still transparent: for example, bilateral filtering [1–3] and nonlocal (NL) means [4] use space-variant adaptive filter kernels that are readily available and can thus be visualised in a straightforward way.

For filters based on nonlinear partial differential equations (PDEs) such as nonlinear diffusion filters [5–7], such an immediate intuition is not directly available. In 2001, Dam and Nielsen introduced the diffusion echo [8] as a means of intuitive understanding of diffusion filters. The (source) diffusion echo at a given location is obtained by evolving a unit impulse according to the given nonlinear diffusion evolution. Thus, diffusion echoes can be seen as the space-variant analogue of the impulse response [9] of an LSI filter. While the impulse response of an LSI filter is space invariant and does not depend on the input image, the diffusion echo differs between locations and changes with the input.

The concept is powerful: diffusion echoes carry the full information on the filtering process. If all echoes are known, they can be used to reconstruct the filtered image from the

Iva Farag and Özgün Çiçek were affiliated with the Mathematical Image Analysis Group during the time they worked on their student theses.

✉ Daniel Gaa
gaa@mia.uni-saarland.de
Joachim Weickert
weickert@mia.uni-saarland.de
Iva Farag
farag@mia.uni-saarland.de
Özgün Çiçek
cicek@mia.uni-saarland.de

¹ Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Saarland University, Campus E1.7, 66041 Saarbrücken, Germany

original. More importantly, the echoes are rich in information and offer a full understanding of the underlying diffusion process. Therefore, they can be used to analyse the filter behaviour in dependence on parameters or discretisations, or to investigate changes between different scales in the diffusion scale-space [10].

Unfortunately, in spite of its obvious merits, the diffusion echo has received little attention in the literature so far. This may have two reasons:

Firstly, the work of Dam and Nielsen does not go beyond diffusion filtering. However, other PDE-based filters might also benefit from an interpretation in terms of an echo. Furthermore, variational methods in image processing and computer vision are naturally connected to PDEs via their Euler–Lagrange equations or their gradient flow [11–15]. This suggests a generalisation of the diffusion echo to other filters.

Secondly, an apparent drawback of diffusion echoes are their high storage costs. Since the echo may differ from pixel to pixel, an image of size N exhibits N echoes of the same size, which means that storage requirements grow quadratically in the number of pixels. Thus, 4×10^9 floating point numbers, which equates to 16 gigabytes of data, must be stored for an image of size 256×256 . Therefore, a more compact representation of the diffusion echo is desirable.

1.1 Our Contribution

This paper highlights, generalises, and improves the practical applicability of the concept of the diffusion echo by addressing both problems. Firstly, we propose a general filter echo framework, which subsumes the diffusion echo formulation [8], and show how general smoothing filters and additional image processing and computer vision filters can be included in the framework. It comprises not only the previously mentioned filters, but also inpainting [12, 16, 17], osmosis [18], and optic flow [13] models. The basic matrix formulation in the framework is of general type and is commonly used [19]. Secondly, we propose a compression strategy for filter echoes, which mitigates their biggest drawback. Our approach is based on probabilistic algorithms [20–22] for computing a truncated singular value decomposition (SVD) [23]. We test this strategy on a number of filters and show that we can substantially reduce the storage requirements while allowing for a straightforward reconstruction in terms of a single matrix–vector multiplication.

The present manuscript builds upon our conference contribution [24], in which we review the concept of the diffusion echo [8], give an interpretation in terms of a typical numerical approach, and introduce our SVD-based compression approach for the efficient representation of isotropic nonlin-

ear diffusion echoes. We extend our conference paper [24] with the following additional contributions:

- We introduce a general framework for filter echoes. We show that it subsumes the diffusion echo formulation and cast a number of smoothing filters as well as general filters into the appropriate form.
- We display the generality and versatility of the filter echo as a tool for visualising filters. We show its usefulness for
 - highlighting differences between filters commonly used for the same task.
 - understanding and highlighting which components of filters make them specifically powerful for a given task.
 - interpreting complex filters in a more intuitive manner.
- We extend the echo compression experiments from [24] to other filters. We show that the case of isotropic nonlinear diffusion with Weickert diffusivity that we considered in [24] is of particular difficulty. Our results show that the compression potential of our approach is substantially higher for other diffusion filters.

1.2 Related Work

As already mentioned, the idea of the diffusion echo has been introduced by Dam and Nielsen [8]. In the context of inpainting-based image compression, the use of an *inpainting echo* has been proposed to optimise the tonal values of the stored data [17]. However, it has never been visualised, but has been used as a purely algorithmic tool.

The drain diffusion echo [8] can be interpreted as a space-variant local filter kernel. Such local kernels have been visualised, for example, for the bilateral filter [3], the nonlocal means filter [4], or the guided filter [25]. In contrast with diffusion filters, these methods explicitly design the shape of the kernels by assigning weights for the local averaging. Therefore, the local filter kernels can be retrieved directly.

The diffusion (and filter) echo gives an exact representation of the filtering process. However, this representation is costly. There exist approaches that aim to approximate nonlinear diffusion kernels, e.g. by using adaptively shaped Gaussians [26] or by learning space-variant integral kernels [27]. However, these approaches do not give quantifiable approximation results.

Other approaches can be linked to nonlinear diffusion using e.g. scaling limits or iterative applications, even though they were not initially designed to approximate diffusion processes. Examples are the Nitzberg–Shiota filter [28], the bilateral filter [1–3], or nonlocal linear diffusion scale spaces [29].

To compress filter echoes, we apply a truncated SVD [23] that we compute with a common method from probabilistic linear algebra, based on a randomised singular value decomposition (RSVD) [20–22]. By discarding components from the SVD, we can steer the reconstruction error in a consistent and quantifiable manner.

Loosely related to our work is the idea of Milanfar [30], who uses symmetric approximations of filters that he then decomposes with an eigendecomposition. The symmetrisation guarantees orthogonality of the eigenvectors, which enables the visualisation of the local effect of the filter on some exemplary shapes. Another mildly related work deals with the short time kernel for the Beltrami flow [31]. However, it uses a kernel that describes the increment in time, while our filter echo characterises the accumulated action over time.

Previous work by two of the coauthors was done in the form of two student theses that employ a principal component analysis (PCA) [32] to compress diffusion echoes [33, 34]. Although this approach is related, they work on a subset of the echo data and discard certain echoes, adding the additional task of selecting an appropriate subset. We work on the full matrix, which frees us from such considerations. This furthermore allows us to calculate the RSVD without explicitly computing and storing any echoes and without computing explicit matrix–vector multiplications.

1.3 Paper Organisation

In Sect. 2, we present the general filter echo framework, which allows us to transfer the concepts of the diffusion echo to a broader class of widely used image processing and even computer vision filters. In Sect. 3, we show how a large number of different filters can be formulated in such a way that they fit our framework. This enables us to consider meaningful echoes for these filters. In Sect. 4, we introduce our compression framework for the filter echo. Our experiments are presented in Sect. 5. We first display and discuss a number of different filter echoes. Then we evaluate the compression approach on different diffusion filters, showing that we can reduce the storage requirements by a considerable amount, making the entire concept more practical. Eventually, we conclude our work in Sect. 6.

2 The Filter Echo Framework

In the following, we introduce our general filter echo framework, which we define in the discrete setting. Although some of the filters that we present are derived in the continuous setting, they need to be discretised before being applied to digital images, which makes the discrete setting adequate for our considerations.

We define discrete images on a regular pixel grid of size $n_x \times n_y$ and stack them into vectors of size $N := n_x n_y$. In our framework, we consider filters that can be written as the result of a matrix–vector multiplication between a matrix $S \in \mathbb{R}^{N \times N}$ and the original image $f \in \mathbb{R}^N$. Then the resulting filtered image $u \in \mathbb{R}^N$ is given by

$$u = S f. \quad (1)$$

We call S the *state transition matrix* of the filter. It is important to note that S is not restricted to being a fixed shift-invariant linear filter, but might depend on the initial image f in the case of linear space-variant or nonlinear filters. S is quadratic in the number of pixels and is generally dense. It contains the complete information on the filtering process and maps the initial image f to the filtered solution u . In the following sections, we show that we can express a large number of different filters by means of such a matrix. However, storing it or sometimes even computing it can be very challenging in practice.

We now introduce the idea of a filter echo in terms of the discrete filter (1). The original idea goes back to Dam and Nielsen [8], who introduced the diffusion echo. However, they restrict themselves to the continuous setting and do not provide discrete theory or an embedding into a general discrete framework. Furthermore, they do not consider any filters beyond diffusion filters.

Adhering to their definitions [8], the *source echo* s_i at location i is the result of filtering a unit impulse e_i centred in i . Using our description of the filter in terms of the state transition matrix (1), we can express the source echo as

$$s_i = S e_i. \quad (2)$$

This shows that the *source echo* s_i of a filter is given by the i -th column of its state transition matrix S . The source echo describes how the grey value data from f_i is distributed by the filtering process. It is a space-variant generalisation of the impulse response of an LSI filter [9] and can be interpreted as the perspective of the “sender”.

Analogously, the *drain echo* [8] shows where the grey values u_j originated from. Using the formulation (1), the *drain echo* d_j is given by the j -th row of the state transition matrix S :

$$d_j = S^T e_j. \quad (3)$$

Therefore, the drain echo corresponds to the local, space-variant filter kernel. It can be interpreted as the “receiver” perspective.

By definition [8], the j -th component of the source echo in pixel i is the same as the i -th component of the drain echo in pixel j , that is, $(s_i)_j = (d_j)_i$. From the matrix-based

formulation, this is straightforward to see. Since the source and drain echoes correspond to the columns and rows of the state transition matrix S , equality between both echoes at a pixel position is only guaranteed for symmetric state transition matrices.

If all echoes are known, they can be used to reconstruct the filtered image u . From (1), it follows that

$$u = S f = \sum_{k=1}^N f_k s_k, \quad (4)$$

for source echoes, and that the component-wise reconstruction from drain echoes is given by

$$u_j = d_j^\top f, \quad (5)$$

for $j = 1, \dots, N$.

3 Analysing Filters within our Framework

In this section, we first show that our framework subsumes the diffusion echo formulation of Dam and Nielsen [8]. Next, we demonstrate that we can straightforwardly extend the approach to further smoothing filters. Afterwards, we show that other image processing and computer vision algorithms can also be reformulated such that they exhibit echoes.

3.1 Echoes for Smoothing Filters

We start by considering diffusion processes, which are given by a PDE. We show how an exemplary numerical solution strategy can be used to fit the continuous process into our discrete framework (1). This serves as a blueprint for later considerations.

Equation (1) is not new in terms of smoothing filters [19]. Well-established filters, such as the bilateral filter [1–3], the NL means filter [4], or the guided filter [25], are specifically designed as weighted averages. From a filter echo perspective, these are straightforward cases, as the echoes are directly obtained from the weights. However, note that this does not hold, for example, for the iterated bilateral filter [35], which can be expressed in terms of its echoes, but whose echoes cannot be retrieved directly. Nevertheless, we also include the bilateral filter and NL means into our considerations, since visualisations of the echoes are still interesting and instructive.

3.1.1 The Diffusion Echo

We now consider the original idea that is the basis of our work on a general filter echo: the *diffusion echo* [8]. Expressing

a diffusion process given by a PDE in terms of (1) is not straightforward, so we describe the necessary steps in the following.

Diffusion filters are typically derived in a continuous setting, on a two-dimensional, rectangular domain $\Omega \subset \mathbb{R}^2$, where grey value images are defined as mappings from Ω to $\mathbb{R}$. For the prototype of a diffusion evolution [7], we consider the parabolic initial boundary value problem

$$\partial_t u(x, t) = \operatorname{div}(D \nabla u(x, t)) \quad \text{for } x \in \Omega, \quad t \in (0, \infty), \quad (6)$$

$$u(x, 0) = f(x) \quad \text{for } x \in \Omega, \quad (7)$$

$$\partial_n u(x, t) = 0 \quad \text{for } x \in \partial\Omega, \quad t \in (0, \infty). \quad (8)$$

Here, $f: \Omega \rightarrow \mathbb{R}$ is the initial image at $t = 0$, and $u: \Omega \times [0, \infty) \rightarrow \mathbb{R}$ is the evolving image. Moreover, $\nabla = (\partial_x, \partial_y)^\top$ is the spatial gradient and $\operatorname{div}(v) = \partial_x v_1 + \partial_y v_2$ is the spatial divergence for $v = (v_1, v_2)^\top$. The matrix D is the *diffusion tensor*, a symmetric, positive definite matrix of size 2×2 , which steers the smoothing behaviour of the diffusion process. This diffusion tensor may depend on the evolving image u , in which case it renders the PDE nonlinear. Lastly, n is the outer normal at the boundary of the image domain $\partial\Omega$.

To solve (6) to (8), we discretise in space and time. Space discretisation is performed on a regular pixel grid as described in Sect. 2. This yields the semi-discrete (time-continuous and space-discrete) scheme

$$\frac{d\mathbf{u}(t)}{dt} = A(\mathbf{u}(t))\mathbf{u}(t), \quad (9)$$

$$\mathbf{u}(0) = f. \quad (10)$$

The matrix $A(\mathbf{u}(t))$ adequately discretises the spatial differential operators in (6) and includes the reflecting boundary conditions (8). There are five conditions (S1)–(S5) that were formulated for such matrices to guarantee that the process fulfils important theoretical properties, such as uniqueness of a solution, fulfilment of a maximum–minimum principle or preservation of the average grey value. These conditions on $A(\mathbf{u}(t))$ are Lipschitz-continuity in $\mathbf{u}$, symmetry, vanishing row sums, nonnegative off-diagonals, and irreducibility [7, Chapter 3].

Lastly, the time variable t is discretised with time step size τ . We use a semi-implicit scheme, where we fix the nonlinearity in each step:

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = A(\mathbf{u}^k)\mathbf{u}^{k+1}. \quad (11)$$

The upper index denotes the current time step k . To comply with the initial condition (10), we set $\mathbf{u}^0 = f$. In contrast

with an explicit scheme, the semi-implicit scheme is stable for arbitrary time step sizes τ , allowing us to use fewer time steps to compute the solution at a large time t . However, it requires us to solve a linear system of equations to compute the solution $\mathbf{u}^{k+1}$ from the current solution $\mathbf{u}^k$:

$$(\mathbf{I} - \tau \mathbf{A}(\mathbf{u}^k)) \mathbf{u}^{k+1} = \mathbf{u}^k. \quad (12)$$

Writing the solution explicitly yields

$$\mathbf{u}^{k+1} = \underbrace{(\mathbf{I} - \tau \mathbf{A}(\mathbf{u}^k))^{-1}}_{\mathbf{P}(\mathbf{u}^k, \tau)} \mathbf{u}^k. \quad (13)$$

Similar to the semi-discrete case, there are also conditions (D1)–(D6) formulated for the fully discrete scheme. They are continuity, symmetry, unit row sums, nonnegativity, irreducibility, and positive diagonal entries [7, Chapter 4]. One can show that $\mathbf{P}(\mathbf{u}^k, \tau)$ fulfils all of them and that the existence of the inverse in (13) is guaranteed for any $\tau > 0$, if $\mathbf{A}(\mathbf{u}^k)$ satisfies the five semi-discrete conditions (S1)–(S5) [7, Chapter 4]. Note that although $(\mathbf{I} - \tau \mathbf{A}(\mathbf{u}^k))$ is usually a sparse matrix, its inverse is generally not.

To make the diffusion models comply with our general filter echo framework (1), we write the solution after n steps in terms of the original image $\mathbf{f}$:

$$\mathbf{u}^n = \underbrace{\mathbf{P}(\mathbf{u}^{n-1}, \tau) \cdots \mathbf{P}(\mathbf{u}^1, \tau) \mathbf{P}(\mathbf{f}, \tau)}_{\mathbf{S}(\mathbf{f}, \tau, n)} \mathbf{f}. \quad (14)$$

The *state transition matrix* $\mathbf{S}(\mathbf{f}, \tau, n)$ for a given diffusion process then depends on the initial image $\mathbf{f}$, the time step size τ , and the number of steps n . These considerations show that the iterative solution of the discretised, nonlinear PDE can be expressed by a single matrix–vector multiplication with the large and dense state transition matrix.

Assuming that the nonnegativity, symmetry, and unit row sum conditions are satisfied, the matrices $\mathbf{P}(\mathbf{u}^k, \tau)$ are doubly stochastic (nonnegative, with unit row and column sums). It is easy to show that the product of doubly stochastic matrices and therefore the state transition matrix is doubly stochastic as well, which allows for an interpretation of drain and source echoes as probability distributions. This property of the state transition matrix has important consequences: Unit row sums and nonnegativity imply a discrete maximum–minimum principle, while unit column sums imply preservation of the average grey value [7, Chapter 4]. Note that symmetry of the state transition matrix is not guaranteed, even though the individual $\mathbf{P}(\mathbf{u}^k, \tau)$ are symmetric. *This means that source and drain echoes usually differ.*

For multiplication with the transposed state transition matrix, which is needed for drain echo calculation, the symmetric diffusion matrices $\mathbf{P}(\mathbf{u}^k, \tau)$ are simply applied in reverse order:

$$\mathbf{d}_j = \mathbf{S}(\mathbf{f}, \tau, n)^\top \mathbf{e}_j = \mathbf{P}(\mathbf{f}, \tau) \mathbf{P}(\mathbf{u}^1, \tau) \cdots \mathbf{P}(\mathbf{u}^{n-1}, \tau) \mathbf{e}_j. \quad (15)$$

In practice, this implies that the results from all steps need to be computed and stored, before the drain echoes can be calculated. The source echoes, on the other hand, can be computed along with the image evolution.

It is important to note that the considerations about the state transition matrix are mainly of theoretical nature. In practice, the fully discrete scheme is typically solved by solving the linear systems in each step. Although the system matrix $(\mathbf{I} - \tau \mathbf{A}(\mathbf{u}^k))$ is quadratic in the number of pixels, it is usually sparse. This can be exploited by using iterative solvers, such as the conjugate gradient (CG) method [36], which only rely on the evaluation of matrix–vector products and do not require the explicit formation of the system matrix. However, this means that it is neither desirable to explicitly compute $\mathbf{P}(\mathbf{u}^k, \tau)$, nor $\mathbf{S}(\mathbf{f}, \tau, n)$, and that multiplications with them should always be evaluated via the corresponding linear systems.

The preceding considerations entail a large number of different diffusion filters, some of which we will now examine in more detail.

Homogeneous Linear Diffusion

Let us first consider the simplest diffusion model, homogeneous linear diffusion [10]. It is obtained by setting the diffusion tensor to the identity, i.e. $\mathbf{D} = \mathbf{I}$, for which the diffusion PDE (6) simplifies to $\partial_t u = \Delta u = \partial_{xx} u + \partial_{yy} u$. It is well known that homogeneous diffusion in an infinite domain is equivalent to Gaussian convolution with a Gaussian kernel of standard deviation $\sigma = \sqrt{2t}$ [37]. Therefore, the diffusion echoes are Gaussian kernels that incorporate the reflecting boundary conditions.

Isotropic Nonlinear Diffusion

Space-adaptive, direction-independent (*isotropic*) smoothing behaviour can be achieved by setting the diffusion tensor to a multiple of the identity, with the magnitude changing depending on the location. A common choice is

$$\mathbf{D}(\nabla u) = g(|\nabla u|^2) \mathbf{I}, \quad (16)$$

where $|\cdot|$ denotes the Euclidean norm. This yields what we call isotropic nonlinear diffusion¹ [5]. The scalar function g is known as *diffusivity* [5]. It is a positive, decreasing function of the gradient magnitude and locally steers the strength of the diffusion activity using a nonlinear feedback mechanism depending on the evolving image u . It is common to use a Gaussian-smoothed version of the gradient in the diffusiv-

¹ Note that the terminology in the literature is not consistent. We differentiate between isotropic (direction-independent) and anisotropic (direction-dependent) diffusion processes.

ity [38], i.e. ∇u_σ , with σ being the standard deviation of the Gaussian.

Popular diffusivity functions include the Charbonnier diffusivity [39]:

$$g_{ch}(s^2) = \frac{1}{\sqrt{1 + s^2/\lambda^2}}, \quad (17)$$

the rational Perona–Malik diffusivity [5]:

$$g_{pm}(s^2) = \frac{1}{1 + s^2/\lambda^2}, \quad (18)$$

or the Weickert diffusivity [7]:

$$g_{we}(s^2) = \begin{cases} 1.0, & \text{if } s^2 = 0, \\ 1.0 - \exp\left(\frac{-3.3148}{s^8/\lambda^8}\right), & \text{else,} \end{cases} \quad (19)$$

with a positive contrast parameter λ .

Edge-Enhancing Anisotropic Nonlinear Diffusion

While the previous filter offers a space-variant smoothing behaviour, for certain applications, however, an *anisotropic* smoothing [40] may be preferable. A popular representative of anisotropic diffusion filters is the so-called edge-enhancing diffusion (EED) [40]. EED reduces the diffusion activity across image edges while still allowing full diffusive smoothing along them. To this end, one selects the diffusion tensor as

$$\begin{aligned} \mathbf{D}(\nabla u_\sigma) &= g(|\nabla u_\sigma|^2) \frac{\nabla u_\sigma}{|\nabla u_\sigma|} \left(\frac{\nabla u_\sigma}{|\nabla u_\sigma|} \right)^\top \\ &\quad + 1 \frac{\nabla u_\sigma^\perp}{|\nabla u_\sigma^\perp|} \left(\frac{\nabla u_\sigma^\perp}{|\nabla u_\sigma^\perp|} \right)^\top. \end{aligned} \quad (20)$$

This formulation implies that $\mathbf{D}$ has an eigenvector $\mathbf{v}_1 = \frac{\nabla u_\sigma}{|\nabla u_\sigma|}$ with corresponding eigenvalue $\lambda_1 = g(|\nabla u_\sigma|^2)$, which inhibits smoothing across strong image edges. The second eigenvector is orthogonal to the first with eigenvalue $\lambda_2 = 1$, which leads to full smoothing along the image edge. It should be noted that while discretising the continuous PDE works similarly to isotropic nonlinear diffusion, anisotropic PDEs require additional, careful considerations, since a standard discretisation in space may violate the nonnegativity condition [7]. As a remedy, L^2 -stable discretisations have been proposed [41], which bound the occurring over- and undershoots.

3.1.2 The Bilateral Filter Echo

Bilateral filtering averages pixels using weights that depend on *closeness* (distance in the domain) and *similarity* (distance in the co-domain) [1–3]. In the setting with a regular pixel grid and a grey value image, which we consider, the closeness

can be measured by the Euclidean distance and the similarity by the absolute difference of the grey values.

The discrete bilateral filter is then expressed for all $i = 1, \dots, N$ by the weighted average

$$u_i = \frac{\sum_{j=1}^N g(|f_i - f_j|) w(|\mathbf{x}_i - \mathbf{x}_j|) f_j}{\sum_{j=1}^N g(|f_i - f_j|) w(|\mathbf{x}_i - \mathbf{x}_j|)} = \sum_{j=1}^N p_{i,j} f_j, \quad (21)$$

with

$$p_{i,j} = \frac{g(|f_i - f_j|) w(|\mathbf{x}_i - \mathbf{x}_j|)}{\sum_{\ell=1}^N g(|f_i - f_\ell|) w(|\mathbf{x}_i - \mathbf{x}_\ell|)}. \quad (22)$$

For the tonal and spatial weighting functions g and w , we consider Gaussians with standard deviations σ_t and σ_s . The final weight $p_{i,j}$ describes the influence of pixel j of the input image $\mathbf{f}$ on pixel i of the filter output $\mathbf{u}$.

With this notation, we can rewrite (21) as

$$\mathbf{u} = \mathbf{P}(\mathbf{f})\mathbf{f}, \quad (23)$$

with a state transition matrix $\mathbf{P}(\mathbf{f}) = (p_{i,j})$, which for fixed weighting functions depends only on the initial image $\mathbf{f}$. We note that u_i is given as a convex combination of all f_j , with $\sum_j p_{i,j} = 1$ for all i and $p_{i,j} \geq 0$ for all i, j . Therefore, $\mathbf{P}$ has unit row sums, and all its entries are nonnegative, which implies that the bilateral filter fulfils a discrete maximum–minimum principle [7, Chapter 4]. Such matrices are also called *row stochastic*. However, due to the normalisation, $\mathbf{P}$ is nonsymmetric and may not have unit column sums, such that the average grey value may not be preserved [7, Chapter 4]. Since $\mathbf{P}$ is only row stochastic, we can interpret only its drain echoes as probability distributions.

3.1.3 The Nonlocal Means Echo

Nonlocal (NL) means [4] is related to bilateral filtering, in the sense that it is also given as a weighted average. However, its weights are based on the similarity between two patches $\mathcal{N}_i$ and $\mathcal{N}_j$, which can be defined as neighbourhoods (e.g. disc shaped) around the pixels i and j . The similarity between two patches with $|\mathcal{N}|$ pixels is given by the Euclidean distance between the corresponding grey value vectors $\mathbf{f}(\mathcal{N}_i) \in \mathbb{R}^{|\mathcal{N}|}$ and $\mathbf{f}(\mathcal{N}_j) \in \mathbb{R}^{|\mathcal{N}|}$. As for bilateral filtering, a weighting function g is applied. We consider a Gaussian with standard deviation σ . This yields the filtered result

$$u_i = \frac{\sum_{j=1}^N g(|\mathbf{f}(\mathcal{N}_i) - \mathbf{f}(\mathcal{N}_j)|) f_j}{\sum_{j=1}^N g(|\mathbf{f}(\mathcal{N}_i) - \mathbf{f}(\mathcal{N}_j)|)} = \sum_{j=1}^N p_{i,j} f_j. \quad (24)$$

with

$$p_{i,j} = \frac{g(|f(\mathcal{N}_i) - f(\mathcal{N}_j)|)}{\sum_{\ell=1}^N g(|f(\mathcal{N}_i) - f(\mathcal{N}_\ell)|)}. \quad (25)$$

The matrix $\mathbf{P}(f) = (p_{i,j})$ is the state transition matrix of the process. Its properties are the same as for bilateral filtering, i.e. it depends on f , is row stochastic and non-symmetric. Satisfaction of a maximum–minimum principle follows, while the preservation of the average grey value is not guaranteed.

3.2 The Inpainting Echo

We now diverge from smoothing filters and extend our filter echo framework to (sparse) PDE-based inpainting. The *inpainting echo* has originally been introduced as a tool to efficiently solve the tonal optimisation problem in image compression [17], and we will first review the basic theory behind it.

In the inpainting setting, we assume that image data are known only on a subset K of the image domain Ω . This subset is called *mask*. We aim at reconstructing the image in the unknown areas $\Omega \setminus K$ by solving a PDE using an inpainting operator L , which may or may not depend on the evolving image u .

We consider the elliptic inpainting formulation, which is instructive in terms of inpainting echoes, since some of their properties can be directly derived from the form of the state transition matrix. The full elliptic boundary value problem is given by

$$Lu(x) = 0 \quad \text{for } x \in \Omega \setminus K, \quad (26)$$

$$u(x) = f(x) \quad \text{for } x \in K, \quad (27)$$

$$\partial_n u(x) = 0 \quad \text{for } x \in \partial\Omega. \quad (28)$$

Examples of suitable inpainting operators include homogeneous diffusion inpainting ($Lu = \Delta u$) [16], nonlinear diffusion inpainting ($Lu = \operatorname{div}(g \nabla u)$) [42], or EED inpainting ($Lu = \operatorname{div}(\mathbf{D} \nabla u)$) [12, 43].

By introducing a mask function $c: \Omega \rightarrow \{0, 1\}$ that takes the value 1 at mask locations and 0 elsewhere, we can merge (26) and (27) into a single equation:

$$c(x)(u(x) - f(x)) - (1 - c(x))Lu(x) = 0. \quad (29)$$

Discretisation yields

$$\mathbf{C}(\mathbf{u} - \mathbf{f}) - (\mathbf{I} - \mathbf{C})\mathbf{L}(\mathbf{u})\mathbf{u} = \mathbf{0}, \quad (30)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{C} = \operatorname{diag}(c)$ is a diagonal matrix with c on the diagonal, and $\mathbf{L}$ is the discrete analogue of the differential operator with reflecting boundary

conditions. Reordering the terms yields a system of equations, which, depending on the differential operator, is linear or nonlinear:

$$(\mathbf{C} - (\mathbf{I} - \mathbf{C})\mathbf{L}(\mathbf{u})) \mathbf{u} = \mathbf{C} \mathbf{f}. \quad (31)$$

For a linear differential operator, we have $\mathbf{L}(\mathbf{u}) = \mathbf{L}$, and a linear system is solved to obtain the inpainted solution $\mathbf{u}$:

$$\mathbf{u} = \underbrace{(\mathbf{C} - (\mathbf{I} - \mathbf{C})\mathbf{L})^{-1} \mathbf{C}}_{\mathbf{S}(c)} \mathbf{f}. \quad (32)$$

For homogeneous diffusion inpainting, the existence of the inverse is guaranteed as long as we have at least one mask pixel [44]. For a fixed linear differential operator, the non-symmetric state transition matrix is uniquely determined by the mask configuration.

In the nonlinear case, we can solve (31) using the Kačanov method [45]. This means that we fix the nonlinearity in each step according to the current solution, leading to a number of linearised problems of similar form as (31):

$$(\mathbf{C} - (\mathbf{I} - \mathbf{C})\mathbf{L}(\mathbf{u}^k)) \mathbf{u}^{k+1} = \mathbf{C} \mathbf{f}, \quad (33)$$

where the upper index denotes the iteration step and where the initialisation $\mathbf{u}^0$ satisfies $\mathbf{C}\mathbf{u}^0 = \mathbf{C}\mathbf{f}$. Note that the final solution $\mathbf{u}^n$ after n steps depends only on $\mathbf{L}(\mathbf{u}^{n-1})$, which means that we do not have to store all intermediate solutions to compute the echoes. Assuming an invertible system matrix, we obtain

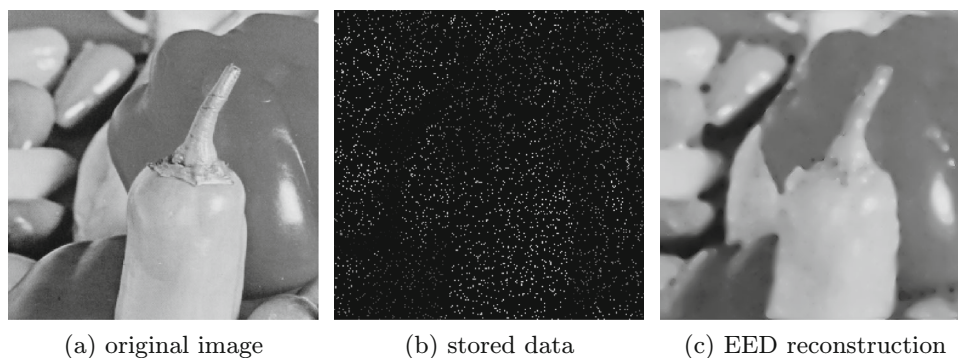
$$\mathbf{u}^n = \underbrace{(\mathbf{C} - (\mathbf{I} - \mathbf{C})\mathbf{L}(\mathbf{u}^{n-1}))^{-1} \mathbf{C}}_{\mathbf{S}(c, f|_c)} \mathbf{f}. \quad (34)$$

In addition to the mask, the state transition matrix $\mathbf{S}$ now also depends on the grey values at the mask locations $f|_c$.

Equation (34) is very insightful w.r.t. the echo configuration. Due to the right multiplication with $\mathbf{C}$ in the state transition matrix, the source echoes in nonmask pixels vanish entirely, which reflects that only the pixel values at mask locations have an influence on the final inpainting result. Accordingly, the drain echoes only take on nonzero values at mask locations. It was shown that for homogeneous diffusion inpainting, $\mathbf{S}$ has nonnegative entries and unit row sums, so the inpainting process satisfies a maximum–minimum principle [44]. Numerically, we compute the elliptic inpainting solution for nonlinear operators as the steady state of a parabolic evolution with automatic time step size adaptation.

Figure 1 shows an example of sparse inpainting. Only 5% of the pixels in *peppers* are stored and the rest is discarded. Then, EED inpainting with the parabolic formulation is used to restore the missing information. Even though the stored

Fig. 1 Inpainting example using the test image *peppers*. 5% randomly selected pixels are stored and the rest is discarded. Then the image is reconstructed using EED inpainting with the Charbonnier diffusivity ($\lambda = 0.8$, $\sigma = 1.0$)



pixels were chosen randomly, the inpainting reconstructs the image with an adequate quality and even restores the edges. Note that quality can be drastically improved, if the stored data are optimised [42].

3.3 The Osmosis Echo

In the next step, we consider linear osmosis modelled by a nonsymmetric drift diffusion process [18], which can create details and leads to nonconstant steady states. Applications range from shadow removal or image cloning [18] to image stitching [46].

Unlike in the diffusion case, we assume that we are given a *positive* initial image $f: \Omega \rightarrow \mathbb{R}_+^N$ and a *drift vector field* $\mathbf{d}: \Omega \rightarrow \mathbb{R}^2$ that steers the osmosis process. A family of filtered images $\{u(\mathbf{x}, t) \mid t \geq 0\}$ is obtained by solving

$$\partial_t u(\mathbf{x}, t) = \operatorname{div}(\nabla u(\mathbf{x}, t) - \mathbf{d}(\mathbf{x})u(\mathbf{x}, t)) \quad \text{for } \mathbf{x} \in \Omega, \quad t \in (0, \infty) \quad (35)$$

$$u(\mathbf{x}, 0) = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega, \quad (36)$$

$$0 = \mathbf{n}^\top (\nabla u(\mathbf{x}, t) - \mathbf{d}(\mathbf{x})u(\mathbf{x}, t)) \quad \text{for } \mathbf{x} \in \partial\Omega, \quad t \in (0, \infty). \quad (37)$$

Typically one is only interested in the steady state ($t \rightarrow \infty$) of this evolution.

Like in the diffusion case, we discretise our image on a regular pixel grid. Again, an adequate discretisation of the differential operators leads to a semi-discrete problem of a form similar to (9). However, the drift term renders the matrix $\mathbf{A}$ non-symmetric. The given process is linear, so $\mathbf{A}$ does not depend on $\mathbf{u}$. As in the diffusion case, there are conditions (SLO1)–(SLO3) on $\mathbf{A}$ in the semi-discrete case. These are zero column sums, nonnegativity of the off-diagonal entries, and irreducibility. More details can be found in [47].

For the time discretisation, we use an implicit scheme, which satisfies the relevant conditions (DLO1)–(DLO4) (unit column sums, nonnegativity, irreducibility, positive diagonal entries) [47] on the nonsymmetric matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ of the fully discrete scheme

$$\mathbf{u}^0 = \mathbf{f}, \quad (38)$$

$$\mathbf{u}^{k+1} = \underbrace{(\mathbf{I} - \tau \mathbf{A})^{-1}}_{\mathbf{P}(\tau)} \mathbf{u}^k, \quad (39)$$

for all $\tau > 0$. We solve the linear systems with BiCGSTAB [36]. We can write the result after n steps of size τ as

$$\mathbf{u}^n = \mathbf{P}^n(\tau) \mathbf{f} = \mathbf{S}(\tau, n) \mathbf{f}. \quad (40)$$

Since the iteration matrix $\mathbf{P}$ has unit column sums [47], the same holds for $\mathbf{S}$. Unit row sums are not guaranteed. Osmosis therefore preserves the average grey value, but does not fulfil a maximum–minimum principle.

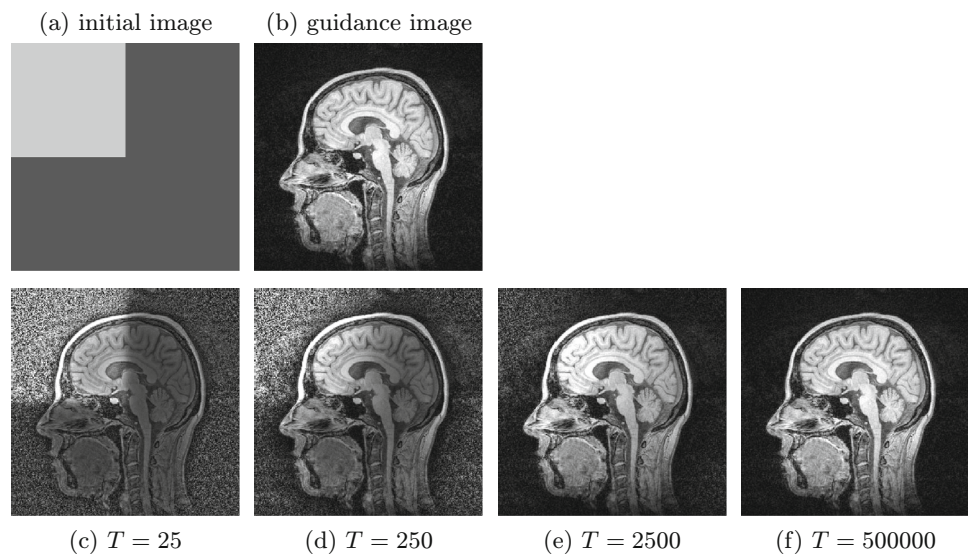
In this work, we only consider *the compatible case*: Setting the drift vector field to $\mathbf{d} = \nabla(\ln v)$, with some guidance image v , linear osmosis converges to $\mathbf{u} = \frac{\mu_v}{\mu_f} v$, with μ_v and μ_f being the average grey values of v and f , respectively [18]. However, note that the preceding analysis about the state transition matrix holds irrespective of the specific drift vector field.

In Fig. 2, we show the results of an osmosis process with the *head* test image as a guidance image, from which we derive the drift vector field, and a simple initial image. We rescale the initial image such that its average grey value matches that of the guidance image, so the evolving image converges exactly to the guidance image.

Although osmosis is a powerful process, its steady-state echoes are surprisingly simple. To show that, we make use of results of Proposition 1 in [47], which are based on the eigendecomposition $\mathbf{P} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^{-1}$, with $\mathbf{Q} = [\mathbf{q}_1 \cdots \mathbf{q}_N]$. The results, which mostly follow from the Perron–Frobenius theory for nonnegative matrices (see, e.g. [48]), state that $\mathbf{P}$ has a simple eigenvalue $\lambda_1 = 1$ and that all other eigenvalues are strictly smaller in absolute value. Furthermore, iterative application of $\mathbf{P}$ yields the steady state $\mathbf{v} \in \mathbb{R}_+^N$, which is the eigenvector $\mathbf{q}_1$ to the eigenvalue $\lambda_1 = 1$ [47]. Lastly, due to the column sum property, the eigenvector of $\mathbf{P}^\top$ that corresponds to $\lambda_1 = 1$ is a constant vector [47].

Repeated application of $\mathbf{P}$ yields the state transition matrix $\mathbf{S}$ of the steady state. Since its eigenvalues are given by powers of the eigenvalues of $\mathbf{P}$, we can conclude that it has a single nonzero eigenvalue $\lambda_1 = 1$. Thus, $\mathbf{S}$ is a rank-1 matrix,

Fig. 2 Osmosis evolution in the compatible case with semi-implicit scheme visualised at different stopping times T . Test image *square* is the initial image, and *head* the guidance image. The initial image is rescaled, such that its average grey value matches that of the guidance image. For $t \rightarrow \infty$, the guidance image is recovered



given by the outer product of $\mathbf{q}_1$ and the corresponding row of $\mathbf{Q}^{-1}$. Since $\mathbf{P}^\top = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1})^\top = (\mathbf{Q}^{-\top}\mathbf{\Lambda}\mathbf{Q}^\top)$, the column of $\mathbf{Q}^{-\top}$ corresponding to λ_1 (which is the corresponding eigenvector of $\mathbf{P}^\top$) is constant, so the corresponding row of $\mathbf{Q}^{-1}$ is also constant. It follows that all columns of the steady-state state transition matrix are the same. They are given by the eigenvector $\mathbf{q}_1$ of $\mathbf{P}$, which is the steady-state solution of the process.

Therefore, the steady-state drain echoes are constant, with intensity corresponding to the intensity of the respective pixel of the steady-state image divided by the sum of the entries of the initial image. Furthermore, all source echoes are given by the same version of the steady-state solution, i.e. the rescaled guidance image, again divided by the sum of the entries of the initial image to satisfy the unit column sum condition.

3.4 The Optic Flow Echo

Lastly, we take our ideas a step further by considering the echoes of a computer vision model: variational optic flow. It is widely acknowledged that variational optic flow models gain their power through the filling-in effect of the regulariser [49], which transports information to the areas where the flow is difficult to determine. In fact, Demetz et al. [50] show that certain linear variational optic flow models can be interpreted as Whittaker–Tikhonov regularisation of the normal flow, where the Euclidean norm in the regularisation term is replaced by some norm that is specific to the optic flow constraint. As we shall see, we can use a similar formulation to produce reasonable optic flow echoes with a slightly restructured formulation of (1).

Optic flow models [13] aim to estimate the motion between two subsequent frames of an image sequence at times t and $t + 1$. They produce a flow field $\mathbf{w} = (u, v)^\top$,

where u and v describe the motion along the x - and y -directions, respectively.

A basic assumption is that the grey values of a point in the first frame and of the corresponding point in the second frame are the same [13], which can be expressed as $f(x, y, t) = f(x + u, y + v, t + 1)$. Linearisation via a first-order Taylor expansion yields the constraint

$$f_x u + f_y v + f_t = \nabla f^\top \mathbf{w} + f_t = 0, \quad (41)$$

where $f_x := \partial_x f$ is the partial derivative of f w.r.t. x and $\nabla = (\partial_x, \partial_y)^\top$ is the spatial gradient.

We see that arbitrary components orthogonal to ∇f can be added to the flow field $\mathbf{w}$ without violating the optic flow constraint (41). This is called *aperture problem*. It follows that (41) can only determine the component parallel to ∇f , which is called the *normal flow*:

$$\begin{aligned} \begin{pmatrix} u_n \\ v_n \end{pmatrix} &= \mathbf{w}^\top \frac{\nabla f}{|\nabla f|} \frac{\nabla f}{|\nabla f|} \\ &= (f_x u + f_y v) \frac{\nabla f}{|\nabla f|^2}. \end{aligned} \quad (42)$$

From (41), we have $f_x u + f_y v = -f_t$, so the normal flow is given by

$$\mathbf{w}_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix} = -f_t \frac{\nabla f}{|\nabla f|^2} = -\frac{1}{|\nabla f|^2} \begin{pmatrix} f_x f_t \\ f_y f_t \end{pmatrix}, \quad (43)$$

and its regularised version [50], which avoids singularities for vanishing ∇f , by

$$\mathbf{w}_n = \frac{-f_t}{|\nabla f|^2 + \epsilon^2} \nabla f, \quad (44)$$

where $\epsilon > 0$ is a small constant. Since the optic flow constraint is insufficient to compute a dense flow field, variational optic flow models include additional smoothness constraints in terms of a regularisation term [51].

We consider two linear models as examples: The Horn–Schunck model [13] and the anisotropic, image-driven Nagel–Enkelmann model [52]. For both of them, the minimisation process leads to linear PDEs.

A suitable energy functional is given by

$$E(u, v) = \int_{\Omega} (f_x u + f_y v + f_t)^2 + \alpha V(\nabla f, \nabla u, \nabla v) dx dy, \quad (45)$$

with some regularisation term V and a regularisation parameter $\alpha > 0$, and the minimising flow field is found in terms of its the Euler–Lagrange equations, which are given by diffusion-like PDEs of the following form:

$$0 = f_x^2 u + f_x f_y v + f_x f_t - \alpha \operatorname{div}(\mathbf{D}(\nabla f) u), \quad (46)$$

$$0 = f_x f_y u + f_y^2 v + f_y f_t - \alpha \operatorname{div}(\mathbf{D}(\nabla f) v). \quad (47)$$

The Horn–Schunck model [13] uses $\mathbf{D}(\nabla f) = \mathbf{I}$, while the Nagel–Enkelmann model [52] employs $\mathbf{D}(\nabla f) = (\nabla f^\perp (\nabla f^\perp)^\top + \lambda^2 \mathbf{I}) / (f_x^2 + f_y^2 + 2\lambda^2)$. It only allows smoothing of the flow field in the direction orthogonal to the local image gradient and avoids smoothing across image discontinuities.

Rewriting (46) and (47) as a vector-valued equation yields

$$\begin{pmatrix} f_x^2 u + f_x f_y v - \alpha \operatorname{div}(\mathbf{D}(\nabla f) u) \\ f_x f_y u + f_y^2 v - \alpha \operatorname{div}(\mathbf{D}(\nabla f) v) \end{pmatrix} = \begin{pmatrix} f_x f_t \\ f_y f_t \end{pmatrix}. \quad (48)$$

For the discretisation, we assume that the partial derivatives of f are computed with suitable finite difference approximations, and the results are stacked into vectors $\mathbf{f}_x$, $\mathbf{f}_y$, and $\mathbf{f}_t$. Furthermore, multiplications between these vectors are to be understood in a component-wise manner. Assuming an appropriate discretisation of the differential operators that we denote by $\mathbf{L}(f)$, we get the following linear system of equations:

$$\underbrace{\begin{pmatrix} \operatorname{diag}(\mathbf{f}_x^2) - \alpha \mathbf{L}(f) & \operatorname{diag}(\mathbf{f}_x \mathbf{f}_y) \\ \operatorname{diag}(\mathbf{f}_x \mathbf{f}_y) & \operatorname{diag}(\mathbf{f}_y^2) - \alpha \mathbf{L}(f) \end{pmatrix}}_{\mathbf{B}} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\mathbf{f}_x \mathbf{f}_t \\ -\mathbf{f}_y \mathbf{f}_t \end{pmatrix}. \quad (49)$$

For an invertible matrix $\mathbf{B}$, which exists in all nontrivial scenarios, we can write the explicit solution formally as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{B}^{-1} \begin{pmatrix} -\mathbf{f}_x \mathbf{f}_t \\ -\mathbf{f}_y \mathbf{f}_t \end{pmatrix}. \quad (50)$$

Figure 3 shows an example of a flow field computed with the Horn–Schunck model [13]. The flow field displays the estimated motion between two frames of a test sequence from the Middlebury dataset [53].

Following the ideas of Demetz et al. [50], we write the optic flow computation in terms of the regularised normal flow, which is directly computable from the given frames:

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix} &= \mathbf{B}^{-1} \operatorname{diag}(\mathbf{f}_x^2 + \mathbf{f}_y^2 + \epsilon^2) \\ &\quad \operatorname{diag}(\mathbf{f}_x^2 + \mathbf{f}_y^2 + \epsilon^2)^{-1} \begin{pmatrix} -\mathbf{f}_x \mathbf{f}_t \\ -\mathbf{f}_y \mathbf{f}_t \end{pmatrix} \\ &= \underbrace{\mathbf{B}^{-1} \operatorname{diag}(\mathbf{f}_x^2 + \mathbf{f}_y^2 + \epsilon^2)}_{\mathbf{S}(\mathbf{f}_x, \mathbf{f}_y, \alpha, \epsilon)} \begin{pmatrix} u_n \\ v_n \end{pmatrix}. \end{aligned} \quad (51)$$

The state transition matrix $\mathbf{S}(\mathbf{f}_x, \mathbf{f}_y, \alpha, \epsilon)$ maps the normal flow to the resulting flow field and depends on the image derivatives, as well as on the regularisation parameters α and ϵ . While $\mathbf{B}^{-1}$ is symmetric, multiplication with the diagonal matrix renders the state transition matrix $\mathbf{S}$ nonsymmetric. Note that $\mathbf{S} \in \mathbb{R}^{2N \times 2N}$, so we get $2N$ echoes of size $2N$. In each position, there is an echo for the two components of the optic flow vector.

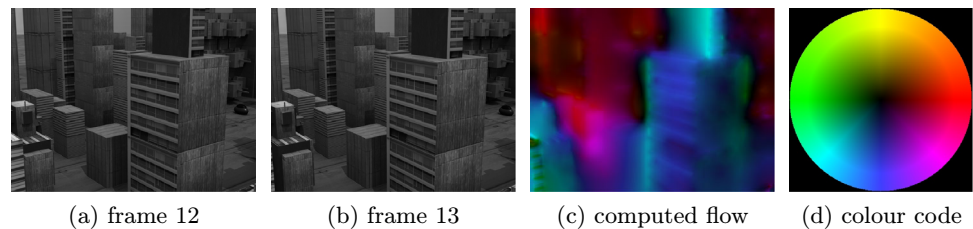
This formulation generalises our filter echo framework, since we are not filtering the original images, but rather the (regularised) normal flow. The matrix $\mathbf{B}$ contains terms that result from discretisations of diffusion operators, suggesting that the sparse normal flow experiences some sort of sophisticated smoothing. This highlights the relation of variational optic flow models to PDE-based smoothing and inpainting methods. In Sect. 5, we confirm that interpretation by visualising source echoes, which provide the information flow of the normal flow data.

4 Our Echo Compression Framework

We have seen that the filter echoes correspond to the columns and rows of the state transition matrix. An image with N pixels therefore has N source echoes and N drain echoes of size N each ($2N$ for optic flow), which constitutes a huge amount of data and makes storage costly. Furthermore, computing echoes at the time they are needed might not be feasible. Consequently, an effective compression approach for echoes is desirable. The goal is to come up with an alternative, more efficient representation, which further allows one to reconstruct the echoes in a short time.

As the source and drain echoes of a filter constitute its state transition matrix, it is natural to consider matrix approximation approaches for the echo compression task.

Fig. 3 Estimation of the optic flow field (colour-coded) using the Horn–Schunck method ($\alpha = 10000$) on frame 12 and frame 13 of the *Urban* test sequence from the Middlebury flow data set [53] and the used colour code



First attempts at such a compression have been made in some unpublished student theses by coauthors of this work [33, 34]. The basic idea of these approaches is to select a subset of echoes and perform a principal component analysis (PCA) [32].

Our goal is to work with the full, original state transition matrices, which frees us of the task to (empirically) decide on the most important echoes and has certain algorithmic advantages. To this end, we use the truncated singular value decomposition [23] in conjunction with a randomised linear algebra approach to compute it, which does not require the explicit formation of the state transition matrix [20–22]. We will provide details in the following subsections.

4.1 The Singular Value Decomposition

The singular value decomposition (SVD) [23] of a matrix $A \in \mathbb{R}^{m \times n}$ is given by

$$A = U \Sigma V^T. \quad (52)$$

The nonnegative singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$, with $p = \min(m, n)$, are the diagonal entries of the diagonal matrix $\Sigma \in \mathbb{R}^{m \times n}$. $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, containing the left and right singular vectors as columns.

If the rank r of the given matrix A is smaller than p , A has only r nonzero singular values, and we can write

$$A = U_r \Sigma_r V_r^T = \sum_{i=1}^r \sigma_i u_i v_i^T. \quad (53)$$

Here, the matrix $\Sigma_r \in \mathbb{R}^{r \times r}$ only contains the first r singular values, and the matrices $U_r \in \mathbb{R}^{m \times r}$ and $V_r \in \mathbb{R}^{n \times r}$ are obtained by discarding the last $m - r$ and $n - r$ columns of U and V . This representation is often called *compact SVD* in contrast to the *full SVD* introduced above.

4.1.1 Truncated Singular Value Decomposition

The fundamental Eckart–Young theorem [54] states that the matrix B_k^* of rank $k < r$, which minimises the approximation error to a given matrix A in the Frobenius norm, is given by the rank- k truncated SVD of A :

$$\begin{aligned} \operatorname{argmin}_{B \in \mathbb{R}^{m \times n}, \operatorname{rank}(B)=k} \|A - B\|_F \\ = \sum_{i=1}^k \sigma_i u_i v_i^T = U_k \Sigma_k V_k^T =: B_k^*, \end{aligned} \quad (54)$$

It furthermore quantifies the approximation error via the discarded singular values:

$$\|A - B_k^*\|_F = \left\| \sum_{i=k+1}^r \sigma_i u_i v_i^T \right\|_F = \sqrt{\sum_{i=k+1}^r \sigma_i^2}. \quad (55)$$

4.1.2 Randomised Singular Value Decomposition

To efficiently compute the truncated SVD, we use a probabilistic approximation, known as *randomised singular value decomposition (RSVD)* [20, 21] or *randomised subspace iteration (RSI)* [20, 22]. These methods compute the truncated SVD of $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ in two steps. First, they project A onto the column space of a matrix $Q \in \mathbb{R}^{m \times k}$, $k < r = \operatorname{rank}(A)$, with orthogonal columns:

$$\hat{A} = Q Q^T A. \quad (56)$$

Then they compute the compact SVD of $\hat{A}$, which only has k nonzero singular values. This is done efficiently by first computing the SVD $Q^T A = \bar{U}_k \Sigma_k V_k$. The desired SVD $\hat{A} = U_k \Sigma_k V_k$ with $U_k \in \mathbb{R}^{m \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$ and $V_k \in \mathbb{R}^{n \times k}$ can then be retrieved via $U_k = Q \bar{U}_k$.

The error of this approximation depends on the error $\|\hat{A} - A\|$, so the main task is to appropriately determine Q . This part of the algorithms is also known as the *randomised rangefinder* [20], as the span of the columns of Q should cover most of the range of A .

To compute Q , the given matrix is applied (once or multiple times) to a standard Gaussian random matrix $G \in \mathbb{R}^{n \times k}$, with columns $g_k \sim \mathcal{N}(0, I)$. The result is eventually orthogonalised via a QR decomposition [23]:

$$Q = \operatorname{orth} \left((A A^T)^{q-1} A G \right). \quad (57)$$

Choosing a larger parameter $q > 0$ for the exponent can improve the approximation quality for matrices with a slowly decaying singular value spectrum [20]. Furthermore,

to increase numerical stability, it is advisable to perform an orthogonalisation $\text{orth}(\cdot)$ after each application of the matrix. The standard *randomised singular value decomposition (RSVD)* [20, 21] applies the rangefinder with parameter $q = 1$, while for $q > 1$ the method is known as *randomised subspace iteration (RSI)* [20, 22].

Recent advances, known as *randomised block Krylov methods* [55], use the entire Krylov space generated through repeated matrix multiplications to construct $\mathbf{Q}$. While this enhances the quality of the approximation, it also increases the memory requirements.

Applying these methods as presented here requires us to specify the target rank k . In practice, one typically introduces an oversampling parameter ℓ , creates a matrix $\mathbf{Q} \in \mathbb{R}^{m \times (k+\ell)}$, and eventually discards the last ℓ singular values and vectors. This further increases the approximation quality, especially for slowly decaying spectra. In practice, it may suffice to choose the parameter as small as $\ell = 10$ [20]. The excellent work of Halko et al. [20] provides theoretical bounds on the approximation quality in dependence of the model parameters (i.e. k , q and ℓ). While we assume that the rank k is fixed, it is worth noting that one can also iteratively increase the size of the matrix $\mathbf{Q}$ and thus the rank of the final SVD, using a pre-selected target threshold for $\|\hat{\mathbf{A}} - \mathbf{A}\|$ [20].

4.2 The Echo Compression Approach

Our objective is to find an efficient representation $\hat{\mathbf{S}}$ of the state transition matrix $\mathbf{S}$ of the considered filtering process. This representation should minimise the Frobenius norm $\|\mathbf{S} - \hat{\mathbf{S}}\|_F$. This corresponds to a minimisation of the mean squared error (MSE) between the compressed and original versions of the source and drain echoes, which is a natural error measure if one considers each echo as an individual image.

To this end, we compute a truncated SVD of the state transition matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ by means of the randomised subspace iteration. Within the algorithm, matrix–matrix multiplications with the state transition matrix $\mathbf{S}$ need to be computed. Recall that $\mathbf{S}$ is dense and quadratic in the number of pixels. Fortunately, we can avoid constructing it explicitly by evaluating the matrix–matrix multiplications by applying the respective filter to each of the columns. For example, for diffusion filters this corresponds to computing a diffusion evolution. This approach allows us to compress filter echoes without ever needing to explicitly compute even a single echo, let alone the full state transition matrix.

Instead of N^2 floats for the full state transition matrix, the truncated SVD only requires $2Nk$ floats to store the matrices $\mathbf{U}_k \in \mathbb{R}^{N \times k}$ and $\mathbf{V}_k \mathbf{\Sigma}_k \in \mathbb{R}^{N \times k}$. This means that the representation is more efficient if $k < N/2$. We show in our evaluation in Sect. 5.2 that we can typically select a very

small k without deteriorating the quality of the echoes, so the compression gain can be substantial.

Let us briefly discuss the computational burden of our compression approach. The advantage of randomised methods for SVD computation is that the matrix decompositions are calculated on small matrices, and can be implemented using dedicated linear algebra packages. The bottleneck of the algorithm are the numerous matrix–vector multiplications with $\mathbf{S}$ and $\mathbf{S}^\top$ [56]. The algorithm presented in Sect. 4.1 requires $2q$ multiplications with $\mathbf{S}$ or $\mathbf{S}^\top$ (one for the projection and $2q - 1$ in the rangefinder). The matrices are of size $N \times (k + \ell)$. This results in $2q(k + \ell)$ diffusion evolutions. A naïve computation of all echoes would require N of those. Therefore, although not an initial objective of our work, the approach may even decrease the computational burden. In Sect. 5.2, we see that, generally, $2q(k + \ell) < N$ in our experiments.

Reconstruction of the echoes from the SVD representation is done by a single matrix–vector multiplication:

$$\mathbf{s}_i^n = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top \mathbf{e}_i = \mathbf{U}_k \left[(\mathbf{V}_k \mathbf{\Sigma}_k)^\top \right]_i, \quad (58)$$

and

$$\mathbf{d}_i^n = \left(\mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top \right)^\top \mathbf{e}_i = \mathbf{V}_k \mathbf{\Sigma}_k \mathbf{U}_k^\top \mathbf{e}_i = \mathbf{V}_k \mathbf{\Sigma}_k \left[\mathbf{U}_k^\top \right]_i, \quad (59)$$

where $[\mathbf{A}]_i$ denotes the i -th column of the matrix $\mathbf{A}$. This implies that the smaller the rank k (i.e. the more compressed the data), the faster the reconstruction.

We test our approach on a number of different diffusion filters and provide results and discussions in Sect. 5.2.

5 Experiments

5.1 Echo Visualisation

In the first part of the experiments, we demonstrate the visualisation qualities of the filter echo. We show that it can be beneficial for comparing similar filters and that it can be used to display the strengths or to understand the subtle details of complex filters. Furthermore, we show that if an adequate model is selected, it can also be used for tasks that may not directly come to mind.

5.1.1 Nonlinear Diffusion, Bilateral Filtering, and NL Means Echoes

Isotropic nonlinear diffusion [5], bilateral filtering [1–3], and nonlocal means [4] are prevalent classical smoothing filters

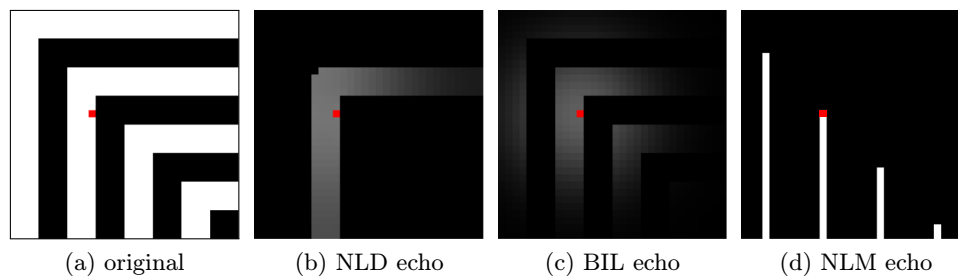


Fig. 4 Drain echo comparison for different smoothing filters. **a** Original image. **b** nonlinear diffusion with Weickert diffusivity (NLD, $t = 150$, $\lambda = 0.3$, $\sigma = 0.0$) echo. **c** bilateral filtering (BIL, $\sigma_t = 30$, $\sigma_s = 10$) echo. **d** NL means (NLM, patch radius 3, $\sigma = 10$) echo. The echo location is marked by the red dot. The three echoes are rescaled jointly,

i.e. the largest echo value among all three echoes is mapped to 255. The nonlinear diffusion echo uses only data from the same segment. Bilateral filtering also includes data from tonally similar, unconnected segments, but reduces weights for distant pixels. Nonlocal means uses information from the entire image, if the local neighbourhood is similar

that can be used for image denoising. Relations between nonlinear diffusion and bilateral filtering have been established in different works (see e.g. [57–59], or the survey paper [60] and the references therein) and bilateral filtering and NL means are similar in spirit, as they both explicitly model a weighted averaging of the image (see Sect. 3.1). Therefore, we use them as examples to show how the filter echo can be used to visualise and emphasise the differences between similar filters.

Tomasi and Manduchi [3] visualise the local kernel (drain echo) for bilateral filtering at a single exemplary artificial step edge to better understand the weight computation of the filter. Buades et al. [4] display NL means kernels for specific image features, and diffusion echoes have been displayed by Dam and Nielsen [8].

We use our filter echo framework to visualise and compare local filter kernels or drain echoes for all of them on an instructive test case in Fig. 4, which sheds light on the different philosophies behind the approaches. The parameters are selected such that all edges are preserved and the filtered images are identical to the original.

A discrete solution to nonlinear diffusion iteratively applies small, local filters, which are reduced at edges in order to avoid blurring. We see in the example that although the resulting kernel can become large in its extent, it stops at image edges, as long as their contrast is large enough. Therefore, the drain echo is given by a connected segment. Bilateral filters are capable of “jumping” across discontinuities. If a segment of similar brightness is spatially separated, it can still have nonzero weights. However, the spatial weighting function decreases for distant pixels. For NL means, such a spatial weight does not exist. If the search window is not restricted, it finds similar patches within the entire image, and therefore it acts truly global. The patch similarity, which is computed to determine the weights, ensures that only pixels with similar local structures are considered. For the drain echo in the test image *corners* in Fig. 4, this means that only

pixels at the right border of vertical white stripes are assigned a weight that is notably larger than zero.

5.1.2 Nonlinear Diffusion Echoes

It is known that for appropriate diffusivities and parameters, nonlinear diffusion can create segmentation-like results [7]. The Weickert diffusivity with a long stopping time and an appropriate contrast parameter is the correct choice for the task [7]. However, individual segments still have to be extracted from the filtered image. At this point, diffusion echoes are an option. They can be used to retrieve and identify individual segments [34, 61]. To this end, one simply has to extract a source echo, which is located within the segment of interest.

In Fig. 5, we show an example of such a segmentation. The *head* test image is smoothed with isotropic nonlinear diffusion with parameters selected such that only high-contrast edges are preserved. The segments of the cartoon-like result are then extracted using source echoes. We display two source echoes that correspond to distinct segments. Note how the segments match the structures in the filtered image and are adequate representations of semantically relevant structures in the original image.

5.1.3 Sparse Anisotropic Diffusion Inpainting Echoes

We now visualise some source echoes that help us understand why anisotropic diffusion performs so well for sparse inpainting [12, 42]. The source inpainting echoes describe how the known data from the mask pixels are distributed to the unknown areas to fill in the missing areas. They directly display the information flow and are the suitable echo choice for inpainting processes.

The goal of our experiments is to show the power of anisotropic EED inpainting given only sparse data points. We use the Charbonnier diffusivity [39], which is commonly

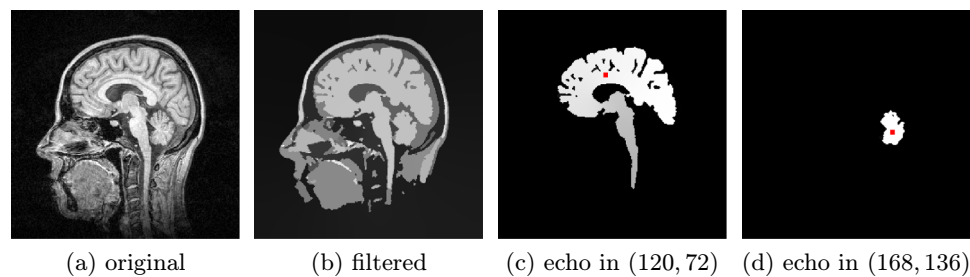


Fig. 5 Nonlinear diffusion echoes for image segmentation. **a** Original image. **b** Filtered by nonlinear diffusion with the Weickert diffusivity ($t = 15000$, $\lambda = 5.0$, $\sigma = 0.5$). **c** Source echo in (120, 72). **d** Source

echo in (168, 136). The echo locations are marked in red. The diffusivity creates a segmentation-like result. By computing the source echo of a pixel in a segment, we extract the segment from the filtered result

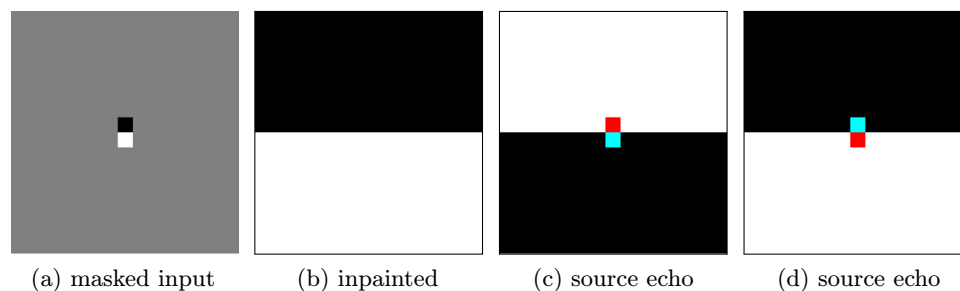


Fig. 6 Inpainting of the *dipole* test image. **a** Original image; the grey areas are unknown. **b** Inpainted result using EED inpainting ($\lambda = 0.01$, $\sigma = 0.1$). **c**, **d** Source echoes of the two mask pixels. The red dot marks

the echo location; the cyan dots mark the other mask pixels. EED spreads the information from the mask pixel to the entire half-plane, creating a sharp edge and filling in the image domain

used in diffusion-based inpainting [12, 42], and employ the parabolic inpainting scheme with a semi-implicit discretisation. The linear systems are solved with the conjugate gradient method. We perform shape completion experiments, inspired by [42], which show how EED inpainting is able to propagate edge information, thanks to its anisotropy and the Gaussian pre-smoothing in the structure tensor computation. By displaying the corresponding source echoes, we get a deeper understanding of this information propagation.

Firstly, we consider the *dipole* experiment [42] in Fig. 6. We see that EED is able to create a sharp edge, which separates the image into the two desired half-planes. The echoes of the two mask pixels show that the data are propagated only inside the corresponding half-plane and that there is no data flow across the boundary.

The *four dipoles* experiment [42] in Fig. 7 builds upon these results. EED is able to reconstruct a white disc with sharp boundaries from only eight mask points. The echoes show that each of the mask pixels has a truly global influence on the result. The discontinuity between the foreground and the background is perfectly respected by the information flow.

In Fig. 8, we compare EED to other diffusion-based inpainting strategies. The *rectangle* experiment shows how the strengths of EED allow it to accurately reconstruct edges

from limited data, which do not have to be as perfectly aligned as in the *dipole* and *four dipoles* experiments. We place mask pixels at staggered locations on both sides of the edges (see Fig. 8e). We then consider all echoes that are located left (i.e. in the dark segment) of the vertical edge and display them in a single image. We call this a *cumulative echo*. It shows how the filters behave along an edge of the image. We see that homogeneous diffusion inpainting spreads the data equally. However, in contrast with homogeneous diffusion smoothing (Sect. 3.1.1), its echo is space variant: it is affected by other mask pixels. This allows this simple inpainting method to perform well if the mask locations are optimised properly [17]. However, in this case, the imperfect mask placement and low density lead to poor edge reconstruction. The cumulative echo highlights the undesired data propagation across the edge. Isotropic nonlinear diffusion, on the other hand, reduces the smoothing near edges. Thus, the grey values do not bleed as much across the edge. However, it is still unable to properly connect the edge, as the smoothing is reduced in an isotropic way, meaning that also the smoothing along the edge is reduced. This is reflected by the cumulative echo. EED mitigates this. The cumulative echo shows how the data from the mask pixels is propagated along the edge, leading to a sharp reconstruction. Note that there is also some data propagation inside the rectangle.

Fig. 7 Inpainting of the *four dipoles* test image. **a** Original image; the grey areas are unknown. **b** Inpainted result using EED inpainting ($\lambda = 0.01$, $\sigma = 1.5$). **c**, **d** Source echoes of two mask pixels. The red dot marks the echo location; the cyan dots mark the other mask pixels. EED creates a sharp disc

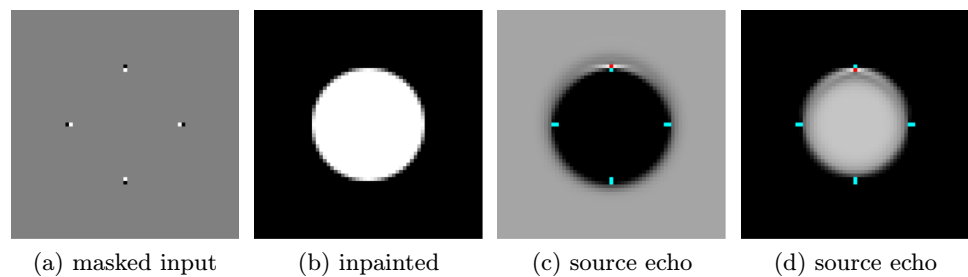
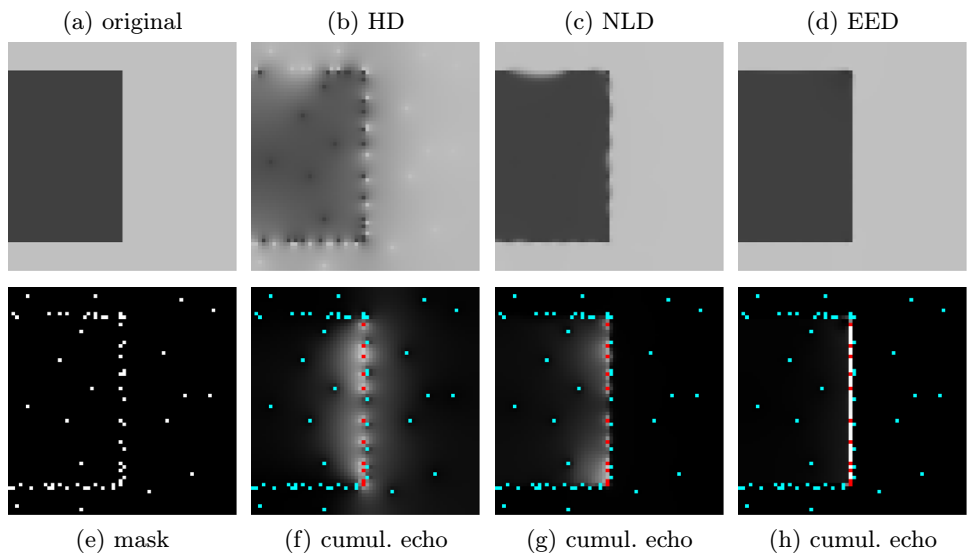


Fig. 8 Inpainting results of the test image *rectangle* shown in **a** using the mask in **e** with **b** homogeneous diffusion (HD) inpainting, **c** isotropic nonlinear diffusion (NLD) inpainting ($\lambda = 0.1$) with Charbonnier diffusivity, **d** EED inpainting ($\lambda = 0.1$, $\sigma = 0.5$), and corresponding cumulative echoes (**f–h**) along a prominent image edge. The mask pixels corresponding to echoes are marked in red, and the other mask pixels in cyan. The three cumulative echoes are rescaled jointly. The echoes show how EED is able to connect edges even if the mask density is low and the mask locations are suboptimal



Lastly, we consider a more realistic test example in Fig. 9. We use the test image *svalbard* and a mask with 1.5% density, which is created with a simple version of an optimisation strategy for homogeneous diffusion inpainting [62]. In this example, the mask pixels are not placed perfectly next to the considered edge but might sit a little further from it. We see that EED still reconstructs a sharp edge, although it is not perfectly straight at locations of low density. Nevertheless, the reconstruction is convincing.

5.1.4 Osmosis Echoes

We have seen in Sect. 3.3 that osmosis converges to a non-constant steady state, which is characterised by equal echoes at all locations. We illustrate this in Fig. 10 by considering the compatible case. We visualise the source and drain echoes at a given location at different times throughout the evolutions and show how the echoes converge towards the rescaled guidance image and a constant image, just as the theory prescribes.

5.1.5 Optic Flow Echoes

Optic flow models are complex, but understanding them as smoothed versions of the normal flow helps us to get a better

intuition. In the following, we display echoes that show how this regularisation acts. Similar to the inpainting case, since

$$w = \begin{pmatrix} u \\ v \end{pmatrix} = S \begin{pmatrix} u_n \\ v_n \end{pmatrix} = S w_n = \sum_{i=1}^{2N} w_{n,i} s_i, \quad (60)$$

only the source echoes s_i corresponding to locations where the normal flow is different from zero influence the result. However, in contrast with the inpainting case, there is no mechanism to avoid that the flow at these locations can be changed as well.

We visualise optic flow source echoes to see how the information from the normal flow is propagated throughout the whole image domain to create a dense flow field. To this end, we consider a simplistic test image of a moving rectangle.

In Fig. 11, we consider a simple shift by one pixel to the right and compute the flow field with the Horn–Schunck and Nagel–Enkelmann methods. The colour-coded ground truth solution is a red rectangle with sharp boundaries, which has the same size as the rectangle in the reference frame (first frame). We display the reference image and the sparse normal flow, which (aside from the corners) consists only of a horizontal component. Considering echoes from a pixel on the left edge, we see that a propagation and thus a smoothing of the normal flow takes place. For the Horn–Schunck method,

Fig. 9 Inpainting results of the test image *svalbard* shown in **a** using the mask in **e** with **b** homogeneous diffusion (HD) inpainting, **c** isotropic nonlinear diffusion (NLD) inpainting ($\lambda = 0.4$) with Charbonnier diffusivity, **d** EED inpainting ($\lambda = 0.3$, $\sigma = 1.0$), and zoom into corresponding cumulative echoes (**f–h**) along a prominent image edge. The mask pixels corresponding to echoes are marked in red, and the other mask pixels in cyan. The three cumulative echoes are rescaled jointly. We use a mask with 1.5% density that is optimised for homogeneous diffusion inpainting [62]

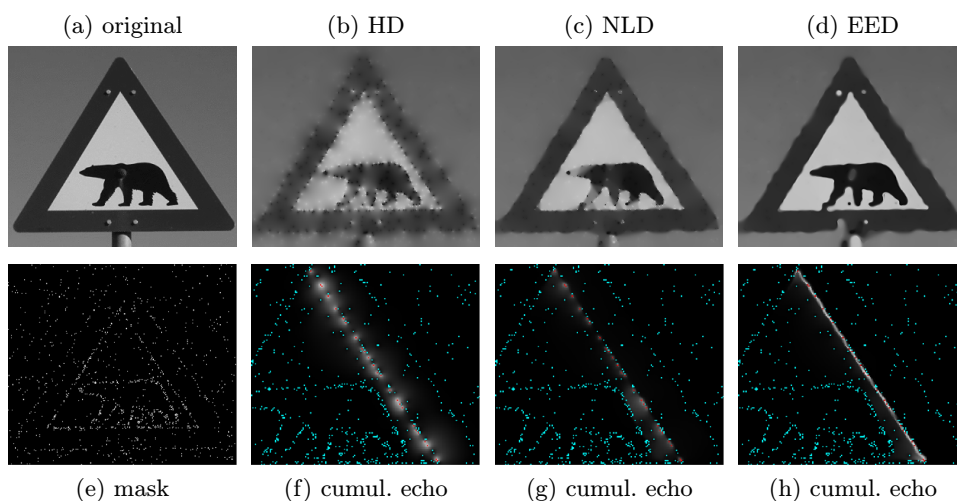
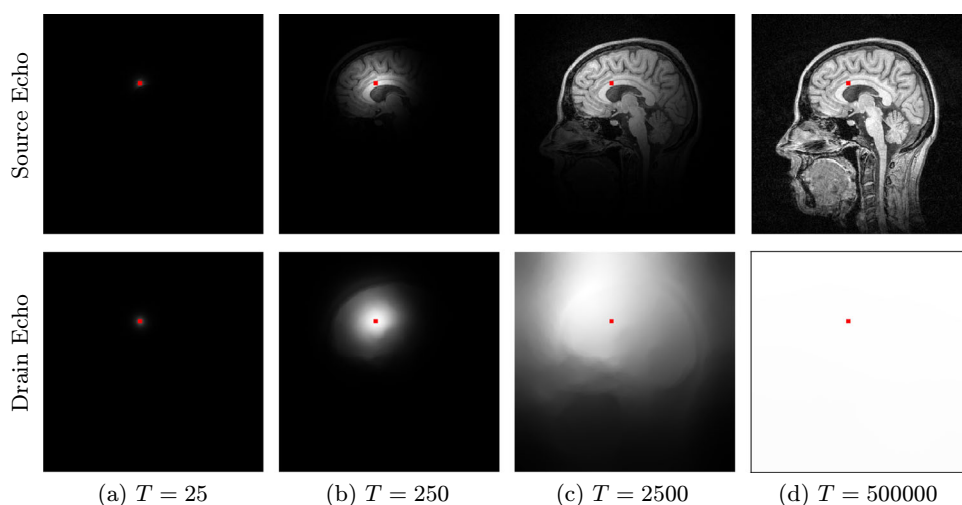


Fig. 10 Example echoes from osmosis evolution from Fig. 2. First row: source echo evolution in (112, 80). Second row: drain echo evolution in (112, 80). We see that the source echo converges to the (rescaled) guidance image, and the drain echo to a constant image



the regulariser propagates data to both sides of the edge of the rectangle, leading to a blurry result. The Nagel–Enkelmann method, steered by the discontinuities of the reference image, transports data only into the rectangle and thus provides a sharp flow field.

5.2 Echo Compression

We now evaluate the compression approach presented in Sect. 4.2.

For inpainting, the number of nonzero echoes is small, and for osmosis all echoes are redundant. Furthermore, we can quickly generate echoes for bilateral filtering and NL means from the explicit weights. However, for diffusion filters, echoes might vary from pixel to pixel, and a computation of an echo is costly, since it requires the application of the (transposed) state transition matrix, which equates to an entire diffusion process. Therefore, we test our approach with some of the diffusion filters presented in Sect. 3.1.1.

We use the 256×256 test image *head* with a grey value range of $[0, 255]$. The parameters of the methods are selected so that they perform a comparable amount of smoothing. We then compute the probabilistic truncated SVD for each of the filters, truncating at 0.5%, 1.25%, 2.5% and 5% of the singular values. We select $q = 3$ and $\ell = 10$. Lastly, we reconstruct the state transition matrix with the approximated singular values and vectors.

We use the conjugate gradient (CG) method to solve the linear systems in the semi-implicit schemes and make use of LAPACK [63] to compute the singular value and QR decompositions in the RSVD algorithm. The test image *head* and the filtered versions of three different diffusion filters, which we use as test cases, are shown in Fig. 12.

To evaluate the results, we proceed twofold. First, we compute the error $\|S - \hat{S}\|_F$ between the state transition matrix and the approximation. Since a calculation of this would require us to compute S explicitly, we instead apply Hutchinson's trace estimator [64] with Gaussian random vectors [65], using that $\|A\|_F^2 = \text{trace}(A^T A)$. We select the

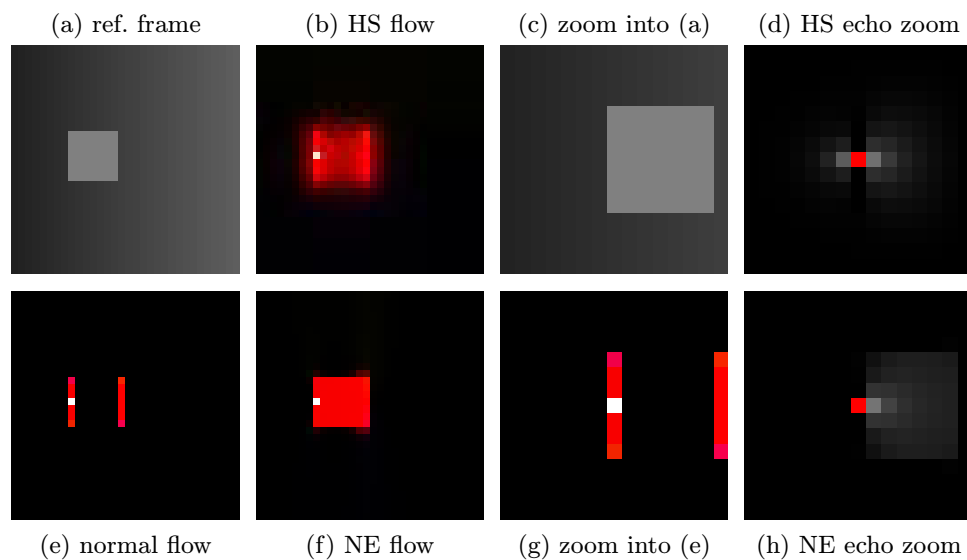


Fig. 11 Computed colour-coded optic flow of a square moving one pixel to the right and example echo. The flow is computed with **b** the Horn–Schunck (HS) model ($\alpha = 10$) and the **f** Nagel–Enkelmann (NE) method ($\alpha = 1$, $\lambda = 0.5$). We use the same colour code as in Fig. 3, so the ground truth motion would be a red square. We visualise echoes from the central pixel of the left vertical line of the normal flow. This location is marked in white in the normal flow and the computed flow fields and in red in the corresponding echoes. Since the normal flow in this pixel only has a horizontal direction (see (e)), we only consider the

echoes for the horizontal component. These echoes have a negligible vertical component, so instead of colour coding them, we logarithmically rescale the horizontal component and visualise them in greyscale. This enhances the visibility of the details. We display zooms of the echoes which correspond to the zooms in (c) and (g). The echoes (d) and (h) visualise how the normal flow information is locally transported to the inside of the square. However, (d) shows that the Horn–Schunck model also propagates flow information to the outside of the square, which leads to the blurry result from (b)

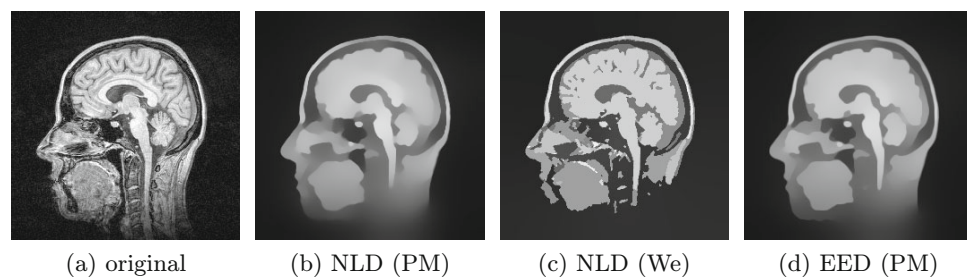


Fig. 12 Test cases for the echo compression method. **a** The original test image *head*, and filtered results with **b** isotropic nonlinear diffusion (NLD) with rational Perona–Malik diffusivity (PM, $t = 190$, $\lambda = 3$, $\sigma = 0.5$), **c** isotropic nonlinear diffusion (NLD) with Weickert diffusivity

(We, $t = 15000$, $\lambda = 5$, $\sigma = 0.5$), and **d** with edge-enhancing diffusion (EED) with rational Perona–Malik diffusivity ($t = 280$, $\lambda = 3$, $\sigma = 0.5$). The parameters are chosen such that a comparable smoothing effect is achieved

number of samples based on the results from Avron and Toledo [66]. Since we do not need the highest precision, we select the number so that the median relative approximation error is at most 10%. However, experiments in [66] suggest that we can expect substantially higher accuracy in practice. Nevertheless, we also complement this approximation-based quality criterion with a deterministic one: We compute the MSE between the filtered image $\mathbf{u}$ and the reconstruction $\hat{\mathbf{S}}\mathbf{f}$ based on the approximated state transition matrix. In addition to the quantitative evaluation, we visualise a few reconstructed source echoes to get a better intuition about the reconstruction quality.

Then we compare the singular value spectra of the methods. To this end, we plot the first 5% of the approximated singular values. Since the quality of the reconstruction is directly related to the spectrum (55), this gives us an idea of how the quality changes w.r.t. the compression ratio. Lastly, we visualise some of the singular vectors to get a better understanding of how the SVD-based representation works.

We see in Table 1 that the echoes corresponding to EED have the best compression potential, with isotropic nonlinear diffusion with the rational Perona–Malik diffusivity slightly behind. The results with nonlinear diffusion and the Weickert diffusivity stand in contrast to that. We have seen in the seg-

Table 1 Errors for RSVD-based compression

Percentage of singular values	Frobenius norm/MSE			
	0.5 %	1.25 %	2.5 %	5.0 %
NLD (PM)	2.198 / 0.84810	0.666 / 0.02864	0.074 / 0.00034	0.012 / 0.00010
NLD (We)	37.55 / 6451.53	30.95 / 3637.93	16.28 / 534.286	0.008 / 0.00291
EED (PM)	0.015 / 0.00012	0.012 / 0.00010	0.012 / 0.00010	0.012 / 0.00010

Estimated [64] Frobenius norm of the error and MSE of the reconstruction of the filtered image depending on the fraction of used singular values

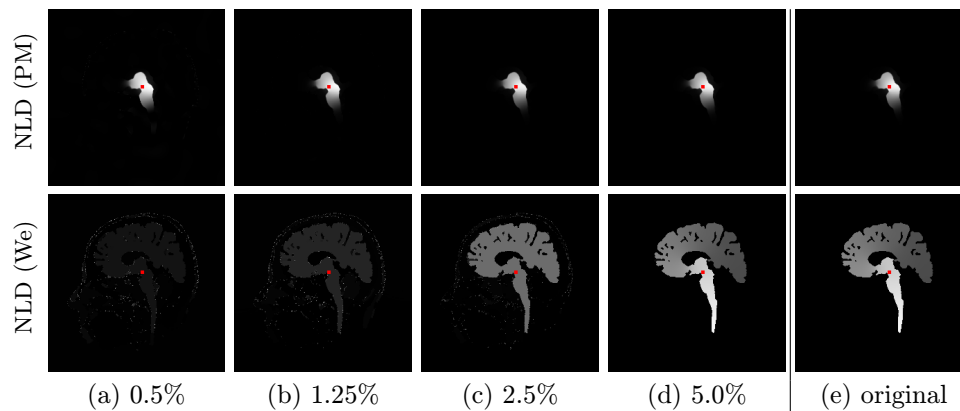


Fig. 13 Reconstructed source echoes for nonlinear diffusion at location (112, 136), using different percentages of the singular values and the original echo for comparison. The red dot marks the echo location. We see that 2.5% of singular values suffice for a visually error-free recon-

struction for the Perona–Malik diffusivity, while the reconstruction with 1.25% has almost negligible brightness differences. For the Weickert diffusivity, 5% of the singular values are required to adequately capture all details

mentation experiments in Sect. 5.1.2 that the echoes extend over full segments, which at first glance would imply a lot of redundancy and great compression potential. However, the results show that the echoes require the most data to be adequately reconstructed. The reason for this lies in a specific phenomenon: image edges are typically not perfectly sharp step edges, but may extend over a larger distance, affecting more pixels. If the edge is strong, the contrast between the individual pixels might still be high. This means that the inner edge pixels do not experience any smoothing at all, thus their corresponding echoes are localised almost entirely within the position. These impulse-like echoes are not easily encoded in the SVD basis and increase the number of large singular values, deteriorating the entire compression quality.

In Fig. 13, we visually compare reconstructions for the two isotropic nonlinear diffusion models. We see that the reconstruction for the rational Perona–Malik diffusivity which uses 2.5% of the singular values is visually indistinguishable from the original, while the reconstructions using 0.5% and 1.25% have minor brightness deviations. This gives us a feeling for the values of the error norm that we should target. For the Weickert diffusivity, only the reconstruction with 5% is adequate.

Let us now look at the spectra in Fig. 14. They not only confirm the results in Table 1, but also provide additional

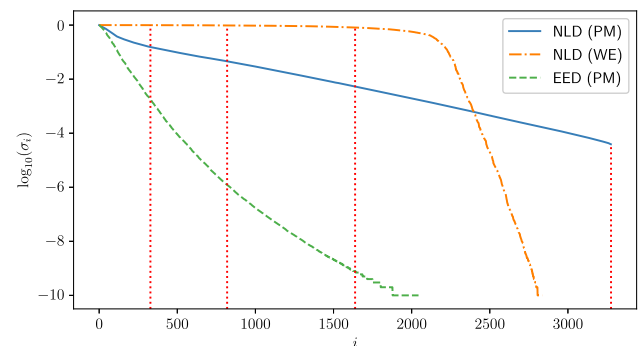
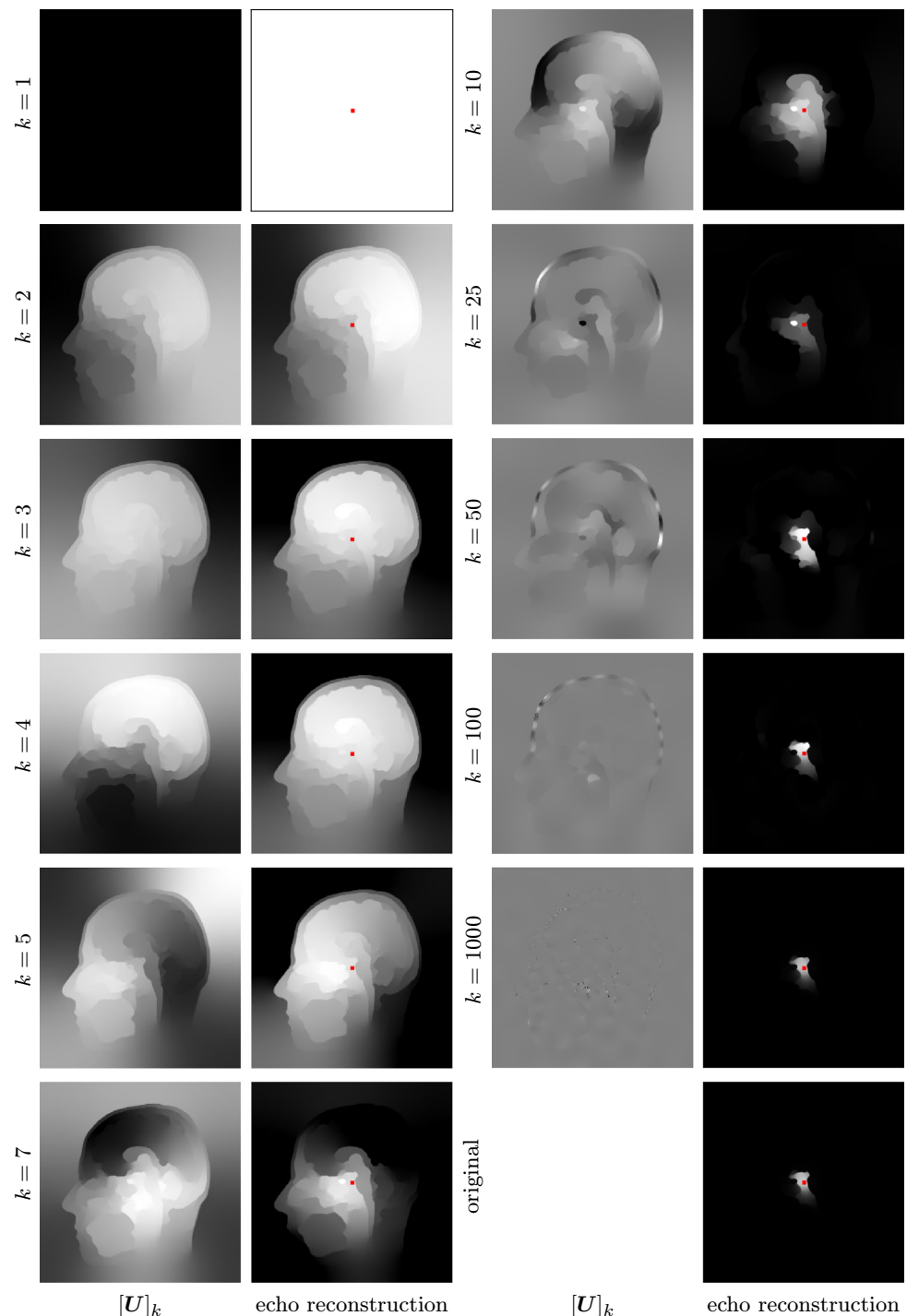


Fig. 14 Singular value spectra from the experiments. The vertical lines depict 0.5%, 1.25%, 2.5%, and 5.0% of the singular values. Note that the Weickert diffusivity leads to a lot of similarly large singular values and a fast drop, while the two methods with the rational Perona–Malik diffusivity have a more even decay

insights. It is clear to see that the Weickert diffusivity leads to a large number of singular values, before we get a sudden decay. The large singular values correspond to the individual segments and pointwise echoes that barely overlap, so they either can or cannot be represented adequately. Isotropic nonlinear diffusion with the rational Perona–Malik diffusivity leads to echoes with larger overlap, since the edge preservation mechanism is less prohibitive, and pointwise echoes

Fig. 15 Plot of some of the left singular vectors $[U]_k$ and the corresponding reconstructions of the source echo at (128, 128). We use the test case with nonlinear diffusion with the rational Perona–Malik diffusivity from the previous experiments. Note how the singular vectors become more and more detailed, allowing for an accurate reconstruction of small structures when using more data



are not an issue. This is reflected in the spectrum, which has a more even decay. For EED, we see that the singular values fall off even faster. Due to the additional smoothing along the edges, the echoes extend over a larger area, creating more redundancy. This enables a very efficient compression.

As a last step, plotting some of the singular vectors and the corresponding reconstructions using all the preceding SVD information shows how adding more SVD components

refines the echo reconstruction step by step. As an example, we select isotropic nonlinear diffusion with the rational Perona–Malik diffusivity. In Fig. 15, we plot some of the left singular vectors as well as the reconstructions of the source echo at a central pixel, using all the SVD data up to the respective index. Since singular vectors also have negative values, we shift the value of 0 to 127.5.

Table 2 Errors for isotropic nonlinear diffusion with the Weickert diffusivity using the proposed exclusion mechanism with $\epsilon \in \{0.0, 0.05, 0.1\}$. Estimated [64] Frobenius norm of the error and MSE of the reconstruction of the filtered image depending on the fraction of used singular values

percentage of singular values	Frobenius Norm / MSE			
	0.5%	1.25%	2.5%	5.0%
$\epsilon = 0.0$	37.55/6451.53	30.95/3637.93	16.28/534.286	0.008/0.00291
$\epsilon = 0.05$	22.38/2736.83	12.17/269.923	0.745/0.01346	0.745/0.01358
$\epsilon = 0.1$	16.62/1282.20	4.528/9.19979	1.596/0.05048	1.596/0.05048

The first singular vector corresponds to a flat image, since the process converges to the average grey value of the given image. The next singular vectors are an overlay of the most important structures of the image, where the structures become more and more detailed and smaller in their spatial extent. The reconstruction of the echo converges to the original by discarding more and more irrelevant information, which is possible due to the fine details contained in later singular vectors. We see that the 100-th singular vector contains small-scale details, which is why the reconstruction is already fairly accurate. The reconstruction using 1000 singular vectors is almost indistinguishable from the original, which is in line with the results from Table 1 and Fig. 14.

5.2.1 An Extension for Rapidly Decaying Diffusivities

We have seen in the results in Table 1 and the spectrum in Fig. 14 that the Weickert diffusivity suffers from the point-wise echoes. To mitigate this, we propose an exclusion mechanism.

To this end, we detect echoes for which the central pixel (i.e. the corresponding diagonal element of the state transition matrix S) is larger than some threshold $1 - \epsilon$. We simply store the corresponding echo location (i.e. the two pixel coordinates) and exclude the corresponding row and column from the state transition matrix, leading to a smaller matrix, which we compress with the method described in Sect. 4.2. At decompression, we simply add the discarded echoes as unit impulses. The parameter ϵ leads to a trade-off between the number of echoes that we exclude (*efficiency*) and the error that we make by describing the corresponding echo as a unit impulse (*accuracy*).

To find the pixels that should be excluded, we first need to compute the corresponding echoes, increasing the cost of the encoding phase. However, we can reduce this additional cost by decreasing the number of potential exclusion candidates for which the echoes must be computed. To this end, recall that the state transition matrix S for diffusion processes has nonnegative entries and unit row sums (see Sect. 3.1.1), and that the initial image f is also nonnegative. Then it follows from (5) that if the diagonal element $s_{ii} > 1 - \epsilon$, we must have $(1 - \epsilon) f_i + \epsilon f_{\min} < u_i < (1 - \epsilon) f_i + \epsilon f_{\max}$ for the corresponding pixel u_i of the filtered image. If any of these

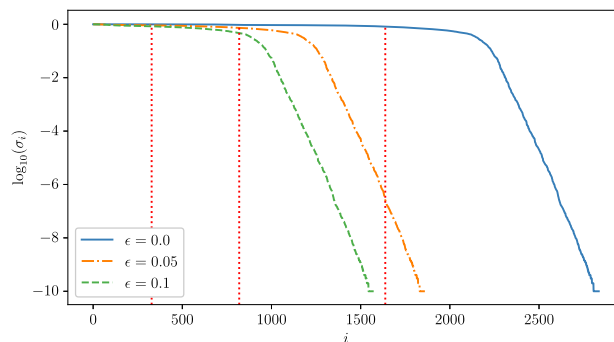


Fig. 16 Singular value spectra using our exclusion approach with different values for ϵ . The vertical lines depict 0.5%, 1.25% and 2.5% of the singular values. We see that the proposed exclusion mechanism shifts the spectrum and decreases the number of large singular values. Note that the spectrum does not reflect the approximation error that is made by describing an echo as a unit impulse

inequalities is violated, we can conclude that $s_{ii} \leq 1 - \epsilon$ and the corresponding echo will not be excluded. However, fulfilling the inequalities does not guarantee $s_{ii} > 1 - \epsilon$, so we need to manually check by explicitly computing the echo.

We test our proposed approach by choosing $\epsilon \in \{0.0, 0.05, 0.1\}$ and compressing the echoes from isotropic nonlinear diffusion with the Weickert diffusivity and the parameters from before.

The results in Table 2 and Fig. 16 confirm the effectiveness of our proposed exclusion mechanism. Table 2 shows that we can get good results at high compression ratios, since few singular vectors now suffice to capture the most important structures of the echoes. This is confirmed by the singular value spectra in Fig. 16, which show a shift in the curve and a decay that occurs much earlier. However, one must keep in mind that this approach introduces a lower bound on the reconstruction error, which becomes apparent for the results at 5% of the singular values. Therefore, the parameter ϵ should be adjusted to the requirements.

Although the required precomputation of the echoes for exclusion increases the cost of the encoding phase, it should be further noted that the exclusion approach even reduces the storage cost slightly, since each excluded point reduces the size of the singular vectors while only requiring the storage of two integers.

6 Conclusions

We have introduced the filter echo as a general visualisation framework that can be applied to a large number of filters from image processing and even computer vision. While it includes filters for which a visualisation of the kernel has been done previously, we have shown that it can be readily extended to other filters, such as osmosis or inpainting, and even to optic flow models.

We have presented its capabilities of visualising the inner workings of complex filters, which enables a better understanding, comparability between filters, and can even be helpful for segmentation.

In addition, in the present paper and its conference predecessor [24], we have proposed the first compression approach specifically tailored towards filter echoes, which counteracts the extensive storage cost of the filter echo. We have shown on a number of test cases with diffusion models that an SVD-based approach can drastically reduce the storage requirements while preserving visual quality. Furthermore, we have seen that reconstruction is simple and can be achieved by a single matrix–vector multiplication. This compression approach completes the filter echo framework and makes it relevant for practical applications where a large number of echoes must be readily available.

In our future work, our aim is to compute filter echoes for further filters for which the interpretation of the echo is not straightforward.

Author Contributions DG and JW both developed concepts and provided theory. DG took care of all implementations, partially based on code by JW, and performed all experiments. DG wrote the paper with feedback and modifications from JW. IF and ÖÇ performed earlier research in their student theses, which gave the inspiration for some parts of the conducted work. All authors have read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the

permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aurich, V., Weule, J.: Non-linear Gaussian filters performing edge preserving diffusion. In: Sagerer, G., Posch, S., Kummert, F. (eds.) *Musterverkennung 1995*, pp. 538–545. Springer, Berlin (1995)
2. Smith, S.M., Brady, J.M.: SUSAN: a new approach to low-level image processing. *Int. J. Comput. Vis.* **23**(1), 45–78 (1997)
3. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proceedings of Sixth International Conference on Computer Vision*, pp. 839–846. Narosa Publishing House, Bombay (1998)
4. Buades, A., Coll, B., Morel, J.-M.: A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **4**(2), 490–530 (2005)
5. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 629–639 (1990)
6. ter Haar Romeny, B.M. (ed.): *Geometry-Driven Diffusion in Computer Vision. Computational Imaging and Vision*, vol. 1. Kluwer, Dordrecht (1994)
7. Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart (1998)
8. Dam, E., Nielsen, M.: Exploring non-linear diffusion: the diffusion echo. In: Kerckhove, M. (ed.) *Scale-Space and Morphology in Computer Vision. Lecture Notes in Computer Science*, vol. 2106, pp. 264–272. Springer, Berlin (2001)
9. Proakis, J., Manolakis, D.: *Digital Signal Processing*, 4th edn. Pearson International, London (2013)
10. Iijima, T.: Basic theory on normalization of pattern (in case of typical one-dimensional pattern). *Bull. Electrotech. Lab.* **26**, 368–388 (1962). (In Japanese)
11. Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, 2nd edn. Applied Mathematical Sciences, vol. 147. Springer, New York (2006)
12. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.-P.: Image compression with anisotropic diffusion. *J. Math. Imaging Vis.* **31**(2–3), 255–269 (2008)
13. Horn, B., Schunck, B.: Determining optical flow. *Artif. Intell.* **17**, 185–203 (1981)
14. Ikeuchi, K., Horn, B.K.P.: Numerical shape from shading and occluding boundaries. *Artif. Intell.* **17**(1), 141–184 (1981)
15. Welk, M., Theis, D., Brox, T., Weickert, J.: PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. In: Kimmel, R., Sochen, N., Weickert, J. (eds.) *Scale Space and PDE Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 3459, pp. 585–597. Springer, Berlin (2005)
16. Carlsson, S.: Sketch based coding of grey level images. *Signal Process.* **15**(1), 57–83 (1988)
17. Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for homogeneous diffusion inpainting. In: Bruckstein, A., Haar Romeny, B., Bronstein, A., Bronstein, M. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 6667, pp. 26–37. Springer, Berlin (2011)
18. Weickert, J., Hagenburg, K., Breuß, M., Vogel, O.: Linear osmosis models for visual computing. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *Energy Minimisation Methods in Computer Vision and Pattern Recognition. Lecture Notes in Computer Science*, vol. 8081, pp. 26–39. Springer, Berlin (2013)

19. Milanfar, P.: A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Process. Mag.* **30**(1), 106–128 (2013)
20. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**(2), 217–288 (2011)
21. Papadimitriou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: a probabilistic analysis. *J. Comput. Syst. Sci.* **61**(2), 217–235 (2000)
22. Rokhlin, V., Szlam, A., Tygert, M.: A randomized algorithm for principal component analysis. *SIAM J. Matrix Anal. Appl.* **31**(3), 1100–1124 (2010)
23. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
24. Gaa, D., Weickert, J., Farag, I., Çiçek, Ö.: Efficient representations of the diffusion echo. In: Bubba, T.A., Gaburro, R., Gazzola, S., Papafitsoros, K., Pereyra, M., Schönlieb, C.-B. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 15668, pp. 324–336. Springer, Cham (2025)
25. He, K., Sun, J., Tang, X.: Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1397–1409 (2013)
26. Dam, E., Olsen, O.F., Nielsen, M.: Approximating non-linear diffusion. In: Griffin, L.D., Lillholm, M. (eds.) *Scale Space Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 2695, pp. 117–131. Springer, Berlin (2003)
27. Fischl, B., Schwartz, E.: Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(4), 342–352 (1997)
28. Nitzberg, M., Shiota, T.: Nonlinear image filtering with edge and corner enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 826–833 (1992)
29. Cárdenas, G.M., Weickert, J., Schäffer, S.: A linear scale-space theory for continuous nonlocal evolutions. In: Aujol, J.-F., Nikolova, M., Papadakis, N. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 9087, pp. 103–114. Springer, Berlin (2015)
30. Milanfar, P.: Symmetrizing smoothing filters. *SIAM J. Imag. Sci.* **6**(1), 263–284 (2013)
31. Spira, A., Kimmel, R., Sochen, N.: Efficient Beltrami flow using a short time kernel. In: Griffin, L.D., Lillholm, M. (eds.) *Scale Space Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 2695, pp. 511–522. Springer, Berlin (2003)
32. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**(11), 559–572 (1901)
33. Baykova, I.: PCA-based representation of diffusion echoes. Bachelor thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany (2016)
34. Çiçek, Ö.: Efficient computation and representation of the diffusion echo. Master thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany (2014)
35. Elad, M.: On the bilateral filter and ways to improve it. *IEEE Trans. Image Process.* **11**(10), 1141–1151 (2002)
36. van der Vorst, H.A.: *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge (2003)
37. Colton, D.: *Partial Differential Equations*. Random House, New York (1998)
38. Catté, F., Lions, P.-L., Morel, J.-M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.* **29**(1), 182–193 (1992)
39. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.* **6**(2), 298–311 (1997)
40. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. In: Kropatsch, W., Klette, R., Solina, F., Albrecht, R. (eds.) *Theoretical Foundations of Computer Vision. Computing Supplement*, vol. 11, pp. 221–236. Springer, Vienna, Austria (1996)
41. Weickert, J., Welk, M., Wickert, M.: L2-stable nonstandard finite differences for anisotropic diffusion. In: Kuijper, A., Bredies, K., Pock, T., Bischof, H. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 7893, pp. 380–391. Springer, Berlin (2013)
42. Schmaltz, C., Peter, P., Mainberger, M., Ebel, F., Weickert, J., Bruhn, A.: Understanding, optimising, and extending data compression with anisotropic diffusion. *Int. J. Comput. Vis.* **108**(3), 222–240 (2014)
43. Weickert, J., Welk, M.: Tensor field interpolation with PDEs. In: Weickert, J., Hagen, H. (eds.) *Visualization and Processing of Tensor Fields*, pp. 315–325. Springer, Berlin (2006)
44. Mainberger, M., Bruhn, A., Weickert, J., Forchhammer, S.: Edge-based compression of cartoon-like images with homogeneous diffusion. *Pattern Recogn.* **44**(9), 1859–1873 (2011)
45. Kachanov, L.M.: Variational methods of solution of plasticity problems. *J. Appl. Math. Mech.* **23**(3), 880–883 (1959)
46. Bungert, P., Peter, P., Weickert, J.: Image blending with osmosis. In: Calatroni, L., Donatelli, M., Morigi, S., Prato, M., Santacesaria, M. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 14009, pp. 652–664. Springer, Cham (2023)
47. Vogel, O., Hagenburg, K., Weickert, J., Setzer, S.: A fully discrete theory for linear osmosis filtering. In: Kuijper, A., Bredies, K., Pock, T., Bischof, H. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 7893, pp. 368–379. Springer, Berlin (2013)
48. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (1990)
49. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vis.* **61**(3), 211–231 (2005)
50. Demetz, O., Weickert, J., Bruhn, A., Zimmer, H.: Optic flow scale space. In: Bruckstein, A., Haar Romeny, B., Bronstein, A., Bronstein, M. (eds.) *Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 6667, pp. 713–724. Springer, Berlin (2012)
51. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. *Int. J. Comput. Vis.* **45**(3), 245–264 (2001)
52. Nagel, H.-H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 565–593 (1986)
53. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **92**(1), 1–31 (2011)
54. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. *Psychometrika* **1**(3), 211–218 (1936)
55. Tropp, J.A., Webber, R.J.: Randomized algorithms for low-rank matrix approximation: design, analysis, and applications. *arXiv preprint arXiv:2306.12418* (2023)
56. Martinsson, P.-G., Tropp, J.A.: Randomized numerical linear algebra: foundations and algorithms. *Acta Numer* **29**, 403–572 (2020)
57. Buades, A., Coll, B., Morel, J.-M.: Neighborhood filters and PDE's. *Numer. Math.* **105**(1), 1–34 (2006)
58. Didas, S., Weickert, J.: From adaptive averaging to accelerated nonlinear diffusion filtering. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) *Pattern Recognition. Lecture Notes in Computer Science*, vol. 4174, pp. 101–110. Springer, Berlin (2006)
59. Sochen, N., Kimmel, R., Bruckstein, F.: Diffusions and confusions in signal and image processing. *J. Math. Imaging Vis.* **14**(3), 195–210 (2001)

60. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: theory and applications. *Found. Trends Comput. Graph. Vis.* **4**(1), 1–73 (2009)
61. Jennewein, S.: Interpretation nichtlinearer bildverarbeitungsmethoden anhand ihres filterechos. High school teacher thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany (2014) **(in German)**
62. Belhachmi, Z., Bucur, D., Burgeth, B., Weickert, J.: How to choose interpolation data in images. *SIAM J. Appl. Math.* **70**(1), 333–352 (2009)
63. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: *LAPACK Users' Guide*, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999)
64. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat. Simul. Comput.* **19**(2), 433–450 (1990)
65. Silver, R.N., Röder, H.: Calculation of densities of states and spectral functions by chebyshev recursion and maximum entropy. *Phys. Rev. E* **56**(4), 4822–4829 (1997)
66. Avron, H., Toledo, S.: Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM* **58**(2), 1–34 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Daniel Gaa is a Ph.D. student with the Mathematical Image Analysis Group at Saarland University (Saarbrücken, Germany). He obtained a B.Sc. degree in Biomedical Engineering from Mannheim University of Applied Sciences (Germany) in 2018, and a M.Sc. degree in Visual Computing from Saarland University in 2021. His research focuses on PDE- and learning-based image processing and computer vision methods.



Joachim Weickert is professor of Mathematics and Computer Science at Saarland University (Saarbrücken, Germany), where he heads the Mathematical Image Analysis Group. He graduated and obtained his Ph.D. from the University of Kaiserslautern (Germany) in 1991 and 1996. He worked as post-doctoral researcher at the University Hospital of Utrecht (The Netherlands) and the University of Copenhagen (Denmark), and as assistant professor at the University of Mannheim

(Germany). Joachim Weickert has developed many models and efficient algorithms for image processing and computer vision using partial differential equations and variational methods. His scientific work covers more than 350 refereed publications. He has been editor-in-chief of the *Journal of Mathematical Imaging and Vision* and has

served in the editorial boards of ten international journals and book series. He has received a Gottfried Wilhelm Leibniz Prize and an ERC Advanced Grant.



Iva Farag earned her M.Sc. degree from Saarland University (Saarbrücken, Germany), specializing in Image Processing and Data Mining. Following her graduate studies, she completed a research program at Honda Research Institute in Japan, where her work focused on video processing and segmentation. She currently works as a Software Engineer at Google, where she develops cutting-edge Google Cloud services.



Özgün Çiçek is a Research Scientist at Robert Bosch GmbH Corporate Research. She received her doctoral degree from the University of Freiburg (Germany), where she was supervised by Prof. Thomas Brox and co-supervised by Dr. Olaf Ronneberger. Her research spans three broad areas: volumetric segmentation and medical image analysis, including sparse annotation learning and cell segmentation; autonomous driving perception, covering motion estimation, uncertainty estimation, and future prediction; and open-vocabulary scene understanding, focusing on vision-language models and 3D occupancy estimation. She is a co-author of 3D UNet: Learning Dense Volumetric Segmentation from Sparse Annotation (MICCAI 2016), a widely cited contribution to volumetric deep learning.