

Universität des Saarlandes



Fachrichtung 6.1 – Mathematik

Preprint Nr. 370

**Turning Diffusion-based Image Colorization
into Efficient Color Compression**

Pascal Peter, Lilli Kaufhold and Joachim Weickert

Saarbrücken 2015

Turning Diffusion-based Image Colorization into Efficient Color Compression

Pascal Peter

Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
`peter@mia.uni-saarland.de`

Lilli Kaufhold

Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
`kaufhold@mia.uni-saarland.de`

Joachim Weickert

Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
`weickert@mia.uni-saarland.de`

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-Mail: preprint@math.uni-sb.de
WWW: <http://www.math.uni-sb.de/>

Abstract

The work of Levin et al. (2004) popularised stroke-based methods that add color to gray value images according to a small amount of user-specified color samples. Even though such reconstructions from sparse data suggest a possible use in compression, only few attempts were made so far in this direction. Diffusion-based compression methods pursue a similar idea: They store only few image pixels and inpaint the missing regions. Despite this close relation and a lack of diffusion-based color codecs, colorization ideas were so far only integrated into transform-based approaches such as JPEG.

We address this missing link with two contributions. First, we show the relation between the discrete colorization of Levin et al. and continuous diffusion-based inpainting in the YCbCr color space. It decomposes the image into a luma (brightness) channel and two chroma (color) channels. Our luma-guided diffusion framework steers the diffusion inpainting in the chroma channels according to the structure in the luma channel. We show that making the luma-guided colorization anisotropic outperforms the method of Levin et al. significantly. Secondly, we propose a new luma preference codec that invests a large fraction of the bit budget into an accurate representation of the luma channel. This allows a high-quality reconstruction of color data with our colorization technique. Simultaneously we exploit the fact that the human visual system is more sensitive to structural than to color information. Our experiments demonstrate that our new codec outperforms the state-of-the-art in diffusion-based image compression and is competitive to transform-based codecs.

1 Introduction

Colorization is a practically relevant image processing task that has its origins in the movie industry. In the 1970s, many monochromatic films were remastered manually to include color. Šykora et al. [1] provide a good overview over the history of early colorization methods. Since this task is time-consuming and tedious, researchers aim at minimizing the amount of user-interaction that is required while still allowing artists to influence the result. The method of Levin et al. [2] is a classical discrete model for *stroke-based* colorization where the user prescribes a few color scribbles and the algorithm fills in the missing colors.

In image compression, at first glance an unrelated field, diffusion-based codecs pursue similar ideas. They create a sparse representation of the image by carefully selecting and storing only a small fraction of pixels. A physically

inspired diffusion process propagates this known information into the missing areas to inpaint (reconstruct) the full image in decompression. The R-EED codec of Schmaltz et al. [3] combines the reconstruction capabilities of edge-enhancing anisotropic diffusion (EED) [4] with efficient entropy coding to outperform the established transform-based coders JPEG [5] and JPEG2000 [6] on grayscale images. A naïve extension to color images in an RGB color space also exists [3]. However, it is not competitive to the JPEG family, since it does not provide the option to benefit from the properties of the human visual system. Both JPEG and JPEG2000 exploit the fact that the human perception values structure higher than color. In particular, they use color spaces that decompose the image into brightness and color information to compress color channels in a coarser manner.

Although diffusion-based codecs appear to be close in spirit to the concept of Levin et al. w.r.t. filling in missing information, colorization has so far only been integrated into JPEG and JPEG2000. In our paper, we close this gap with a new continuous diffusion-framework for colorization and a dedicated color codec based on R-EED.

1.1 Our Contribution.

While the colorization method of Levin et al. is formulated purely in a discrete setting, we present a new continuous framework for colorization. We consider the problem in YCbCr space where the luma (brightness) channel Y is given by the grayscale input image. Our *luma-guided* diffusion adapts to the structure of the brightness channel and propagates the user-specified color scribbles in the chroma (color) channels to the missing areas. We evaluate four different diffusion models from our general framework, namely space-variant isotropic and space-variant anisotropic diffusion, as well as two newly proposed higher-order counterparts to the aforementioned models. We show that the discrete method of Levin et al. is closely related to our continuous higher-order isotropic diffusion. Our experiments on well-known test images demonstrate that our anisotropic models outperform isotropic methods significantly.

For image compression, we combine our luma-guided diffusion with the R-EED codec to a *luma preference (LP) mode*. It relies on the core idea that the structural information in the luma channel is more important for the human visual system than the color components. Therefore, this channel should be stored with higher accuracy than the chroma channels Cb and Cr. To this end, we dedicate a large part of our file size budget to the luma channel. The luma-guided diffusion benefits from the accurate structure in the brightness channel and allows to reconstruct the chroma channels from

a smaller amount of known data.

Two of the authors have presented preliminary color compression results with luma-guided EED in a conference paper [7]. Here we go beyond pure compression: Our continuous colorization framework offers a new area of application, three additional luma-guided diffusion models, and we establish connections to prior discrete methods. Moreover, we conduct experiments with diffusion-based compression on a real-world image database [8].

1.2 Related Work.

Colorization. Luan et al. [9], classify colorization methods as *stroke-* or *example-*based. In stroke-based methods [2, 10, 9, 11, 12, 13], the user manually specifies a small amount of color scribbles on a single image or a sequence of images. In contrast, *example-*based algorithms [1, 14, 15, 16] require one or multiple fully colored images that are similar to the grayscale image to be colorized. This approach is useful for video sequences and can be combined with stroke-based approaches: A single frame can be colored with scribbles and serve as an example for the remaining frames. Due to our focus on single images, we only consider the stroke-based approach. For a review for colorization by example, we refer to Luan et al. [9].

In our paper, we establish a close relationship of our diffusion framework to the work of Levin et al. [2]. They minimize a discrete energy that relies on a simple assumption: Neighboring pixels with similar grayvalues also have similar colors. First they transform the image to YUV space to obtain a structure-color-decomposition. Then they minimize the difference of each unknown color pixel in YUV space to a weighted average of its 3×3 neighborhood.

Several contributions try to stop color propagation across brightness edges that is prominent in the model of Levin et al. To this end, Huang et al. [10] compute thin, closed edges with a Sobel operator and apply multiple post-processing steps. Hua et al. [17] interpret colorization from a gradient domain perspective and propose a framework with edge-preserving constraints that includes colorization among other tasks. Casaca et al. [13] achieve sharp edges with a completely different approach: They first use the color scribbles to partition the image into segments and then propagate color according to segment labels. This approach also allows easy corrections by iterated user-interaction. In our work, we aim to address edge preservation directly in a continuous energy formulation.

There are also various *non-local* approaches for colorization: They have the advantage that they can propagate color globally to all image areas that have similar structure. For instance, Yatziv and Sapiro [18] average known

colors globally according to intensity, while Luan et al. [9] group pixels only locally by intensity and globally by texture similarity. A mapping between brightness and color values is obtained by Quang et al. [11] by means of a machine learning approach, and Yang [12] propagates color information with a recursive bilateral filter. For the specialized class of monochromatic manga comics, Qu et al. [19] use clustering according to texture similarity. While non-local approaches can yield better results, they also tend to achieve this at the cost of computational complexity which can be a limiting factor in time-critical applications like video decompression. In our compression codec we rely on local, diffusion-based colorization.

The higher-order approaches that we consider in Section 2 are very rare in the literature. To our best knowledge, only Greer et al. [20] have proposed a nonlinear higher-order model for heat flow on surfaces. It is unrelated to colorization.

Compression. In the second part of our paper, we propose a new diffusion-based compression approach. This class of codecs started out with a proof-of-concept codec by Galić et al. [21]. It combined edge-enhancing anisotropic diffusion (EED) [4] with a triangular subdivision of the image in order to find and store known data efficiently. There are many different diffusion models. For instance, Chan and Shen [22] proposed to use the total variation approaches of Rudin et al. [23] without conducting actual experiments. When Schmalz et al. [3] introduced their PDE-based compression codec, they conducted an evaluation of a wide range of inpainting models and concluded EED to be the best choice. In particular, they found that good inpainting methods rely on nonlocality and anisotropy. Therefore, widely used local and isotropic operators such as total variation [22, 23] do not perform well. We discuss their codec in more details in Section 2.1. So far, there are no dedicated color codecs, but there is a variety of other specialized compression methods, for example for cartoons [24], depth maps [25, 26, 27], 3-D data [3], or texture [28]. For a full review of diffusion-based compression, we refer to the in-depth survey of Hoeltgen et al. [29].

The transform-based codecs JPEG [5] and JPEG2000 [6] rely on a completely different concept, since they apply a discrete cosine or wavelet transform to create a sparse image representation, while diffusion-based codecs operate in the spatial domain. The JPEG family exploits the properties of the human visual system in order to improve the perceived fidelity of compressed color images: JPEG performs a subsampling of the chroma channels in YCbCr space, while JPEG2000 omits fine-scale wavelet coefficients for the color components of YUV space. With our method, we also want to exploit perception. However, we go one step further than the JPEG family, since we compensate the reduced amount of known data for the color

channels by reusing structural information from the brightness component. Similarly, some transformation-based codecs exploit colorization ideas for compression. Most of them rely on the method of Levin et al. [2]. In [30], Horiuchi and Tominagaan use an extended version of the method of Levin et al. in combination with JPEG to restore color information from samples in CIELab space. Cheng and Vishwanathan [31] investigate a machine learning approach to the problem. They interpret the method of Levin et al. as a learning algorithm and incorporate a modified version into JPEG. Ono et al. [32] explore finding optimal representative pixels for colorization with the approach of Levin et al. in compression. Furthermore, variations of a colorization method based on Markov random fields have been applied in a post-processing step to JPEG [33] and JPEG2000 [34]. Lee et al. [35] also compress the luma channel with conventional transform-based coding. They formulate the colorization part as a L_1 minimization problem for choosing a small set of representative color values.

1.3 Organization of the Paper

We first review the core concepts of diffusion-based inpainting and compression in Section 2.1 since we need them for our new contributions. In Section 2, we propose our new luma-guided colorization framework, establish a connection to the method of Levin et al., and evaluate it experimentally. Section 3 is dedicated to our luma-preference mode for diffusion-based compression. We describe our new codec and demonstrate its potential with experiments on a database of well-known test images. Finally, we conclude our paper in Section 4 with a summary and an outlook on future work.

2 From Inpainting to Colorization

2.1 Review: Diffusion-based Inpainting

Let us consider a rectangular image domain $\Omega \subset \mathbb{R}^2$, and an image $\mathbf{f} = (f_R, f_G, f_B)^\top$ with the RGB channels $f_c : \Omega \rightarrow \mathbb{R}$, $c \in \{R, G, B\}$. In inpainting, we know the image content only on a subset $K \subset \Omega$ of the image domain, the so-called *inpainting mask*. Our task is now to reconstruct the missing areas in the *inpainting domain* $\Omega \setminus K$. In order to propagate the known data to the inpainting domain, we consider the image evolution which is described

by the general diffusion equation

$$\partial_t u_c = \operatorname{div}(\mathbf{D}\nabla u_c) \text{ on } \Omega \setminus K, \quad (1)$$

$$u_c = f_c \text{ on } K, \quad (2)$$

with reflecting boundary conditions on the outer image boundaries $\partial\Omega$. According to Eq. (1), the diffusion process equilibrates contrast differences between the pixels over time t . In denoising applications, $t \rightarrow \infty$ leads to a flat steady state containing the average pixel value of the channel. However, the Dirichlet boundary conditions of Eq. (2) fix the known data on K which allows the propagation of information from points in K to points in $\Omega \setminus K$, but not the other way around. This leads to a non-trivial steady state: the reconstructed image (see e.g. [21]). Experiments for the models that we consider show that the initialization of u on $\Omega \setminus K$ at time $t = 0$ has no influence on this reconstruction.

Specific diffusion models described by Eq. (1) differ in their choice of the diffusion tensor \mathbf{D} , a positive definite 2×2 matrix that steers the propagation. Following the taxonomy of Weickert [36], we distinguish diffusion models by two core properties: If the tensor \mathbf{D} depends on the evolving image \mathbf{u} , the method is referred to as *nonlinear*, and *linear* otherwise. If the eigenvalues λ_1 and λ_2 of \mathbf{D} are equal, the propagation behavior is the same for all directions and the method is called *isotropic*. For instance, Gerig et al. [37] proposed the following nonlinear isotropic model for color images:

$$\partial_t u_c = \operatorname{div} \left(g \left(\sum_{\ell \in \{R,G,B\}} |\nabla u_\ell|^2 \right) \nabla u_c \right), \quad c \in \{R, G, B\}. \quad (3)$$

Here the diffusion tensor degenerates to a scalar, monotonously decreasing *diffusivity* $g(\sum_{\ell \in \{R,G,B\}} |\nabla u_\ell|^2)$. The sum of the gradient magnitude over all channels acts as an edge detector: At locations of high magnitude, the diffusivity reduces the amount of diffusion compared to flat regions. We use the Charbonnier diffusivity [38]

$$g(s) := \frac{1}{\sqrt{1 + s/\lambda^2}} \quad (4)$$

with some contrast parameter $\lambda > 0$. Note that the scalar diffusivity stops diffusion in all directions. However, as the evaluation of Schmalz et al. [3] has shown, *anisotropic* models are more powerful in compression. To this end, we also consider edge-enhancing anisotropic diffusion (EED) [4]. It adapts the diffusion tensor \mathbf{D} to the local image structure according to Di

Zenno’s structure tensor for color images [39]:

$$\mathbf{J}_{RGB} := \sum_{c \in \{R, G, B\}} \nabla u_{c, \sigma} \nabla u_{c, \sigma}^\top, \quad (5)$$

where $u_{k, \sigma}$ denotes a convolution of the channel u_k with a Gaussian of standard deviation σ . It propagates the structural information to its neighborhood. The nonnegative eigenvalues $\mu_1 \geq \mu_2$ of \mathbf{J}_{RGB} measure the contrast in the directions of the eigenvectors \mathbf{v}_1 and \mathbf{v}_2 . To reflect the image structure, the diffusion tensor \mathbf{D} uses the same eigenvectors. In order to reduce diffusion *across* edges, we apply the diffusivity from Eq. (4) to μ_1 . Thus, the first eigenvalue of the diffusion tensor is $\lambda_1 := g(\mu_1)$. For full diffusion *along* edges we use a constant second eigenvalue $\lambda_2 := 1$.

2.2 Relation to the Method of Levin et al.

In the following, we show that the *discrete* colorization approach of Levin et al. [2] is closely related to the *continuous* diffusion-based inpainting methods of Section 2.1. This connection motivates the introduction of a new class of diffusion-based colorization methods in Section 2.3.

In contrast to inpainting, a full original gray value image is known for colorization. In addition, a user can specify some color information at arbitrary positions, for instance some manually drawn color “scribbles” (see Fig. 1 (b)). The RGB space from the previous section is not adequate to describe this situation, since each channel carries color information. Instead, color spaces like YCbCr provide a separation of the image data into intensity information in the *luma channel* Y and color information in the *chroma channels* Cb and Cr; see e.g. [40]. We use the following transform of an RGB image $\mathbf{f}_{RGB} = (f_R, f_G, f_B)^\top$ to YCbCr space:

$$(f_Y, f_{Cb}, f_{Cr})^\top = (0, 127.5, 127.5)^\top + \mathbf{T} \mathbf{f}_{RGB} \quad (6)$$

$$\mathbf{T} := \begin{pmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{pmatrix}. \quad (7)$$

With this transformation, we can describe the colorization task adequately: We want to obtain a colored version $\mathbf{u}: \Omega \rightarrow \mathbb{R}^3$ of a grayscale image $f: \Omega \rightarrow \mathbb{R}$. In YCbCr space, f corresponds to the Y channel of \mathbf{u} , which yields $u_Y = f$ on the entire domain Ω . On a subset $K \subset \Omega$, color scribbles $u_{Cb} = f_{Cb}$ and $u_{Cr} = f_{Cr}$ are given in addition. Our task is now to reconstruct the missing colors in $\Omega \setminus K$.

Levin et al. [2] base their colorization algorithm on the assumption that for any pixel \mathbf{r} from the discrete image domain $\tilde{\Omega} = [1, \dots, m] \times [1, \dots, n]$, the points in its 3×3 neighborhood $\mathcal{N}(\mathbf{r})$ should have similar colors unless there is a huge discrepancy in the corresponding luma values. Imposing this constraint onto the reconstructed colors for $c \in \{Cb, Cr\}$ comes down to minimising the discrete energy

$$E(\mathbf{u}_c) = \sum_{\mathbf{r} \in \tilde{\Omega} \setminus \tilde{K}} \left(\left(\sum_{\mathbf{s} \in \mathcal{N}(\mathbf{r})} w_{\mathbf{r},\mathbf{s}} \right) u_{c,\mathbf{r}} - \sum_{\mathbf{s} \in \mathcal{N}(\mathbf{r})} \left(w_{\mathbf{r},\mathbf{s}} u_{c,\mathbf{s}} \right) \right)^2, \quad (8)$$

$$w_{\mathbf{r},\mathbf{s}} \sim 1 + \frac{1}{\sigma_{\mathbf{r}}^2} (u_Y(\mathbf{r}) - \mu_{\mathbf{r}})(u_Y(\mathbf{s}) - \mu_{\mathbf{r}}). \quad (9)$$

Levin et al. define confidence weights $w_{\mathbf{r},\mathbf{s}}$ in such a way that they sum up to 1. They are based on the luma channel u_Y in the considered pixels as well as the mean $\mu_{\mathbf{r}}$ and variance $\sigma_{\mathbf{r}}$ in a window around the center pixel \mathbf{r} . Interestingly, this model resembles the standard finite-difference approximation [41] of a divergence term as it appears in the diffusion equation (3). On a discrete grid with grid size h , we approximate $\text{div}(g \nabla u_c)$ in each pixel $\mathbf{r} := (i, j)$ of the channel c . According to [41], we use finite differences of the diffusivity g and the evolving image u_c . If $u_{c,i,j}$ and $g_{i,j}$ denote approximations to u_c and g in pixel (i, j) , we obtain

$$\begin{aligned} \text{div}(g \nabla u_c)_{i,j} \approx & \frac{1}{h} \left(\frac{g_{i+1,j} + g_{i,j}}{2} \frac{u_{c,i+1,j} - u_{c,i,j}}{h} \right. \\ & - \frac{g_{i,j} + g_{i-1,j}}{2} \frac{u_{c,i,j} - u_{c,i-1,j}}{h} \\ & + \frac{g_{i,j+1} + g_{i,j}}{2} \frac{u_{c,i,j+1} - u_{c,i,j}}{h} \\ & \left. - \frac{g_{i,j} + g_{i,j-1}}{2} \frac{u_{c,i,j} - u_{c,i,j-1}}{h} \right). \end{aligned} \quad (10)$$

Since all diffusivities $g_{\mathbf{s}}$ with $\mathbf{s} \in \mathcal{N}(\mathbf{r})$, the 3×3 neighborhood of \mathbf{r} , appear in Eq. (10), we can change notation to

$$w_{\mathbf{r},\mathbf{s}} = \begin{cases} \frac{1}{2} \frac{g_{\mathbf{s}}}{h^2} & (\mathbf{r} \neq \mathbf{s}), \\ 2 \frac{g_{\mathbf{s}}}{h^2} & (\mathbf{r} = \mathbf{s}), \end{cases} \quad (11)$$

and obtain the same structure as Eq. (8) for a fixed $\mathbf{r} \in \tilde{\Omega} \setminus \tilde{K}$:

$$\text{div}(g \nabla u_c)_{\mathbf{r}} \approx \left(\sum_{\mathbf{s} \in \mathcal{N}(\mathbf{r})} w_{\mathbf{r},\mathbf{s}} \right) u_{c,\mathbf{r}} - \sum_{\mathbf{s} \in \mathcal{N}(\mathbf{r})} \left(w_{\mathbf{r},\mathbf{s}} u_{c,\mathbf{s}} \right) \quad (12)$$

This implies that the model of Levin et al. can be interpreted as a discretization of the continuous energy

$$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} (\operatorname{div}(g(|\nabla u_Y|^2) \nabla u_c))^2 d\mathbf{x} \quad (13)$$

with natural boundary conditions

$$\mathbf{n}^\top \nabla \operatorname{div}(g \nabla u_c) = 0, \quad \operatorname{div}(g \nabla u_c) = 0, \quad (14)$$

where \mathbf{n}^\top denotes the outer normal for a boundary point in $\partial\Omega$. Finding a minimizer u_c with gradient descent leads to the following partial differential equation on $\Omega \setminus K$:

$$\partial_t u_c = \operatorname{div}(g(|\nabla u_Y|^2) \nabla (\operatorname{div}(g(|\nabla u_Y|^2) \nabla u_c))). \quad (15)$$

On the inpainting mask K , we have Dirichlet boundary conditions $u_c = f_c$. According to Eq. (15), the model of Levin et al. can be understood as a higher-order variant of the isotropic diffusion model in Eq. (3). This *bi-isotropic* inpainting also bases its diffusivities purely on the luma channel u_Y . In the following we use this observation to derive new diffusion-based colorization algorithms.

2.3 Generalized Diffusion-based Colorization Models

In Section 2.1, we have learned that anisotropic inpainting models are able to reconstruct edges more accurately than their isotropic counterparts. Therefore, we can hope to improve the method of Levin et al. by replacing the scalar-valued diffusivity in Eq. (13) by a diffusion tensor. We adapt our colorization locally to the image structure with the *luma tensor*

$$\mathbf{J}_Y := \nabla u_{Y,\sigma} \nabla u_{Y,\sigma}^\top, \quad (16)$$

which describes only the known image structure in the brightness channel. Now we can formulate the colorization in YCbCr space as a variational problem. We impose a smoothness constraint on the missing parts of the chroma channels. For a given channel $c \in \{Cb, Cr\}$, we minimize the energy

$$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} \left(\operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla u_c) \right)^2 d\mathbf{x}. \quad (17)$$

Note that the first eigenvector \mathbf{v}_1 of the luma tensor \mathbf{J}_Y is parallel to the luma gradient $\nabla u_{Y,\sigma}$ and its second eigenvector \mathbf{v}_2 is perpendicular to it.

Table 1: Overview of our Diffusion-based Colorization Methods.

model	energy	diffusion equation
isotropic	$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} g(\nabla u_Y ^2) \nabla u_c ^2 d\mathbf{x}$	$\partial_t u_c = \operatorname{div}(g(\nabla u_Y ^2) \nabla u_c)$
bi-isotropic	$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} (\operatorname{div}(g(\nabla u_Y ^2) \nabla u_c))^2 d\mathbf{x}$	$\partial_t u_c = \operatorname{div}(g(\nabla u_Y ^2) \nabla (\operatorname{div}(g(\nabla u_Y ^2) \nabla u_c)))$
EED	$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} \nabla^\top u_c \mathbf{D}(\mathbf{J}_Y) \nabla u_c d\mathbf{x}$	$\partial_t u_c = \operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla u_c)$
Bi-EED	$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} (\operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla u_c))^2 d\mathbf{x}$	$\partial_t u_c = \operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla (\operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla u_c)))$

The corresponding eigenvalues are $\mu_1 = |\nabla u_{Y,\sigma}|^2$ and $\mu_2 = 0$. As in the previous sections, we choose the eigenvectors of $\mathbf{D}(\mathbf{J}_Y)$ to be \mathbf{v}_1 and \mathbf{v}_2 . Using once more variational calculus as in Section 2.2, we obtain the corresponding second-order diffusion process

$$\partial_t u = \operatorname{div}\left(\mathbf{D}(\mathbf{J}_Y) \nabla (\operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla c))\right). \quad (18)$$

Note that in contrast to the inpainting models in Section 2.1, the diffusion tensor relies only on information from the luma channel and thus the process is *linear*. Its ability to preserve edges in the chroma channels relies on two factors: The structural information of the luma tensor \mathbf{J}_Y and the location and value of the specified chroma information. Thereby, brightness information can be reused, but the selection of known data still allows the user to influence the formation of edges in the Cb and Cr channels. We call this new model *luma-guided Bi-EED*.

Since first-order models are already successful in inpainting [21, 3], we also want to investigate if the higher-order approach of Levin et al. can be simplified. To this end, we propose to minimise the energy

$$E(u_c) = \frac{1}{2} \int_{\Omega \setminus K} \nabla^\top u_c \mathbf{D}(\mathbf{J}_Y) \nabla u_c d\mathbf{x}. \quad (19)$$

and obtain the corresponding diffusion process

$$\partial_t u_c = \operatorname{div}(\mathbf{D}(\mathbf{J}_Y) \nabla u_c), \quad c \in \{Cb, Cr\}. \quad (20)$$

As in Section 2.1, we can obtain both isotropic and anisotropic processes from this model by choosing appropriate eigenvalues for $\mathbf{D}(\mathbf{J}_Y)$. We denote these two colorization models as *isotropic* and *EED*. Table 1 provides an overview of the four energies and their corresponding diffusion PDEs.

2.4 Colorization Experiments

In the following, we perform two sets of experiments: a quantitative comparison and a purely visual one. For our implementation, we use the finite

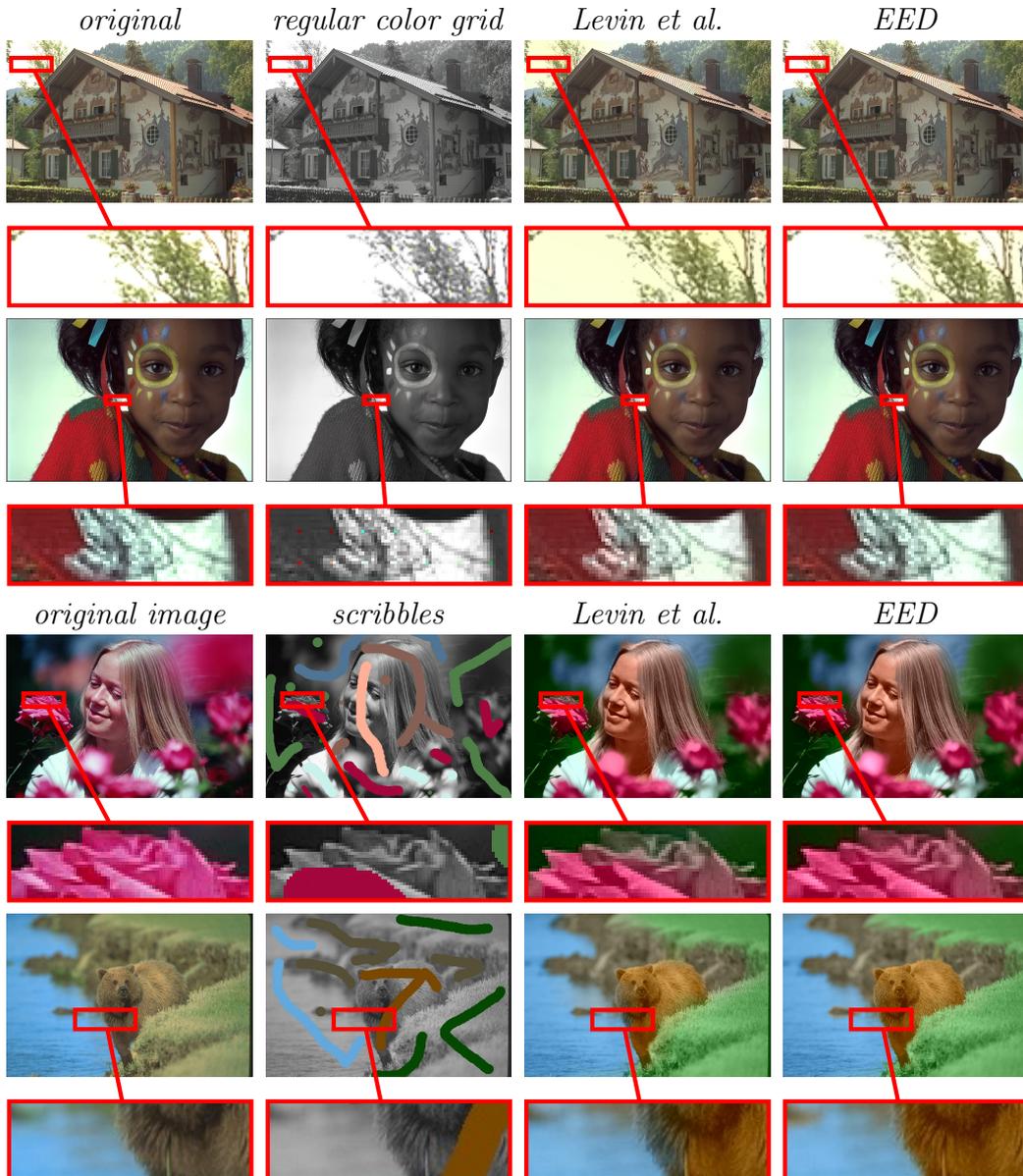


Figure 1: **Colorization** (a) **Rows 1 and 2:** Quantitative experiments on the Kodak images 24 and 15. Only 1% of the original color data is given on a regular grid and error values are computed w.r.t. the original image. (b) **Rows 3 and 4:** Colorization with manual scribbles on images 100099 and 206062 of the Berkeley image database. These two images should be considered as recoloring since we do not use the original colors for the scribbles.

difference framework of Weickert et al. [42]. We consider the evolution of the parabolic PDEs from the previous section and solve the inpainting problem iteratively with fast explicit diffusion (FED) [43] combined with a coarse-to-fine initialization. We stop the iterative scheme as soon as the norm of the

Table 2: Average MSE for colorization on the Kodak database.

	isotropic	bi-isotropic	Levin et al.	EED	Bi-EED
RGB-MSE	20.338	20.328	22.503	13.836	13.831
CIELab error	2.7894	2.7907	2.8155	2.2976	2.2974

residual has decreased by a factor 10^{-5} . Experimentally, we have determined that a small contrast parameter seems to consistently yield the best results for all four of our diffusion models. Thereby, the contrast parameter is fixed to $\lambda = 0.01$ for all results.

First, we compare the quality of our four methods luma-guided isotropic and bi-isotropic diffusion, EED, and Bi-EED against the reference implementation provided by Levin et al. [2]. For a quantitative evaluation, on the Kodak database [8] we keep only 1% of the color data, colorize the images, and compute the RGB-MSE w.r.t. the originals. In order to avoid bias towards one of the algorithms, we keep the known data on a regular grid. As an additional color distortion measure, we compute the average Euclidean distance in the CIELab color space [44] (observation angle 2° , illumination D65) that has been designed to reflect perceived color differences.

Table 2 demonstrates that higher-order models do not yield a significant advantage over their simpler counterparts. Empirically we found that their respective evolutions differ, but the steady state is almost identical. Therefore, we prefer isotropic diffusion and EED for colorization since they converge significantly faster. Higher-order models might still be viable for applications that rely on the evolution, for instance denoising. Compared to Levin et al., the isotropic PDEs perform better on average, but the anisotropic models yield by far the best quantitative results. Fig. 1(a) shows that the edge-enhancing properties of anisotropic diffusion avoid the substantial colour-bleeding of the approach of Levin et al. in areas with complex, high-contrast structures (e.g. tree-branches, hair).

In a second evaluation, we compare EED and Levin et al. on images from the Berkeley database [45] in a practical colorization scenario: In Fig. 1(b) we manually specify some rough color scribbles without considering any ground truth images. These experiments confirm that EED avoids color bleeding also with rough scribbles: For instance, the bear’s fur in *image 100099* is consistently brown for EED, while blue color from the water bleeds in for Levin et al. Similar phenomena occur at the flower-petals in *image 206062*. Due to its consistent performance we use EED colorization for our compression applications in Section 3.

3 From Colorization to Compression

3.1 Review: Compression with R-EED

The practical application of our PDE-based colorization in compression builds on successful concepts from the R-EED codec of Schmaltz et al. [3]. In contrast to the inpainting case described in Section 2.1, the whole original image is known and the inpainting mask K can be chosen freely. The selection of known data affects the reconstruction quality significantly (see [46, 47]) and influences the coding cost.

R-EED achieves a good balance between coding efficiency and inpainting accuracy by pursuing a subdivision strategy. It defines a fixed point pattern which includes the four corner points and the midpoint of a given rectangular image. With this simple inpainting mask, it reconstructs the image. If the resulting mean squared error (MSE) exceeds a given threshold a , the algorithm splits the image in its largest dimension. For each rectangular half, R-EED adds the corresponding point pattern to the mask, inpaints this subimage locally, and compares the result again with the threshold. We repeat this successively until the error threshold is respected by all subimages or none of them can be split any further. This approach restricts the inpainting mask to a regular adaptive grid (see Fig. 3(a)), which achieves two goals: It limits the search space, thus improving runtime, and allows to store the positions of known data efficiently. Each subdivision corresponds to a node in a binary tree, and the leaves of the whole splitting tree encode the relative positions for point patterns.

Now that the inpainting mask has been selected, the codec performs a coarse quantization of each known pixel value to q different, equally spaced intensity values from the interval $[0, 255]$. These quantized values can finally be stored efficiently with a suitable entropy coder such as arithmetic coding [48] or PAQ [49]. Note that R-EED needs to optimize the number of quantized values q , the subdivision threshold, and the contrast parameter λ of EED. In addition, an important post-processing step in R-EED is the so-called *brightness optimization*. Instead of quantizing the original pixel values, R-EED chooses the optimal value at each mask position from the quantized co-domain. While this introduces an error to the sparse set K of known data, it has been shown that such brightness optimization can greatly improve the inpainting quality on the inpainting domain $\Omega \setminus K$ [29]. For further technical details of the original R-EED codec, we refer to [3]. In the following, we describe how to use our colorization methods for an efficient color extension of this codec.

3.2 Color Compression with Luma Preference Mode

Currently, R-EED by Schmaltz et al. [3] only supports compression of color images in RGB space. It treats all channels equally by inpainting with the coupled multi-channel EED from Section 2.1. The locations of known data are shared between these channels. This reduces the overhead since only one inpainting mask needs to be stored, but it also implicitly encodes the image structure (such as edges) in all three channels. In the following, we apply our luma-guided colorization in YCbCr space to eliminate this redundancy. Since we use YCbCr space, we can also exploit the fact that the human visual system is much more sensitive to errors in the luma channel than to deviations in the chroma components (see e.g. [50]). Therefore, we can increase the perceived quality by storing the luma channel with higher accuracy compared to the chroma information.

Motivated by the two observations above, we propose a *luma preference (LP) mode* for color compression with R-EED. In order to reach our goal of accurate luma compression, we prescribe a target compression ratio $R : 1$ and dedicate a larger amount of the resulting bit budget to the brightness channel. We describe this weighting of the luma and chroma components by the *LP ratio* $r \geq 1$. This free parameter expresses the file sizes s_{Cb} and s_{Cr} for color data as multiples of the luma size s_Y :

$$s_Y = r \cdot s_{Cb} = r \cdot s_{Cr}. \quad (21)$$

Therefore, the desired compression ratio $R : 1$ also determines s_Y . Due to Eq. (21), the total size of the compressed file is $(1 + 2r)s_Y$. Consequently, for an original file size s_O , LP mode reaches the target ratio if s_Y satisfies

$$s_Y := \frac{s_O}{(1 + 2r)R}. \quad (22)$$

Thus, our *channel weighting* with the LP ratio introduces perceptive coding to R-EED in the same sense as chroma subsampling in JPEG. However, this only addresses one of our two goals. Instead of just improving the quality of structural information at the cost of color accuracy, we want to exploit correlations between the channels. Even though the edges are not identical in the Y, Cb, and Cr components, a luma edge is still a good indicator for a chroma edge, as the high-fidelity colorizations in Section 2.4 demonstrate. Therefore, we use the compressed luma channel as the guidance image for our colorizations method from Section 2. Moreover, we can still influence the formation of chroma edges by choosing the colorization mask K_c and optimising the chroma contrast parameter λ_{CbCr} for the diffusivity from Eq.(4).

In contrast to Eq. (20), we restore all channels with the luma diffusion tensor and allow individual known data K_c for every channel. On $\Omega \setminus K_c$, we compute the steady state of the evolution

$$\partial_t u_c = \operatorname{div}(\mathbf{D}(\mathbf{J}_Y)\nabla u_c), \quad c \in \{Y, Cb, Cr\}. \quad (23)$$

Thus, we perform non-linear EED in the luma channel and linear luma-guided EED in the chroma channels. Finally, we combine the channel weighting and luma-guided diffusion to our luma preference codec. In the following, we describe its compression and decompression pipelines.

Compression in LP mode consists of two sequential steps.

Step 1: Luma Compression. For the luma channel, we obtain the mask with the R-EED subdivision scheme from Section 2.1. We optimize the contrast parameter λ_Y , and the number of quantized gray values q_Y in the luma channel for the best brightness MSE at the target file size s_Y . We approximate s_Y by compressing the positions of known data represented by a binary *luma tree* and the quantized gray values with PAQ [49]. In addition, we optimize the intensity values at the locations of the final luma inpainting mask.

Step 2: Chroma Compression. In this compression step, we also perform subdivision and optimize a contrast parameter λ_{CbCr} and a quantization parameter q_{CbCr} . However, there are some important differences. To find the colorization mask, we employ the same subdivision scheme as in R-EED. However, we use the accurate structural information of the already encoded luma tensor for luma-guided EED colorization. Since this reduces the importance of exact mask positions, we use a joint *chroma tree* for the Cb and Cr channels. The reduced cost of the shared tree frees up bits for additional data points and allows finer quantization. We also optimize our parameters, known data, and chroma values w.r.t. the final MSE after transformation of the reconstruction back to RGB space.

Decompression is very straightforward and strictly sequential. A PAQ decoding yields access to the parameters and known data. The trees enable us to reconstruct the masks for all channels. After inpainting the luma channel, we have the structural information of the reconstructed brightness component and can colorize the image. A transformation from YCbCr to RGB space concludes the reconstruction.

File Format. We first write all data sequentially in a binary file. The header contains the image dimensions m, n , the contrast parameters λ_L and λ_{CbCr} , and the LP factor r , each as a binary number of adequate size (8-16 bit). As in the original R-EED, we store the subdivision trees in terms of their minimal and maximal depth and a binary sequence that indicates the

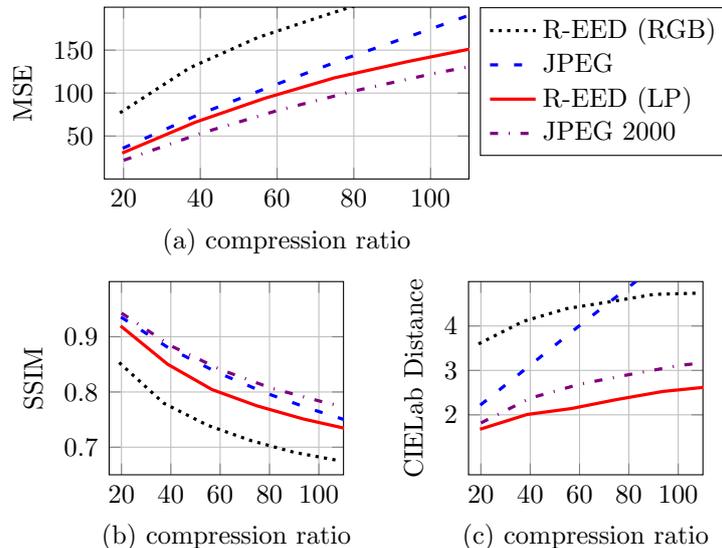


Figure 2: Error distribution on Kodak image database: **(a) Top: RGB-MSE** over all RGB channels (lower is better), **(b) Bottom Left: SSIM** in the luma channel (higher is better), **(c) Bottom Right: Euclidean CIE Lab Distance** in the a and b channels (lower is better).

tree structure in-between. Then we append the quantized luma or chroma values row-by-row as a sequence of raw binary numbers. Finally, we apply the entropy coder PAQ [49] to the whole file. Note that in comparison to our conference publication [7], we have simplified the file format. Originally, we used entropy encoding for luma and chroma values separately. The joint PAQ container allows us to write all data sequentially without the need for jump addresses and can reduce the file size slightly for high compression ratios.

3.3 Compression Experiments

We compare our luma preference codec to the state-of-the-art in PDE-based compression, namely R-EED in RGB space, and the established transform coders JPEG and JPEG2000. For the implementation of our codec we use the FED-approach with coarse-to-fine initialization from Section 2 and parallelize it on the GPU. Compared to the original R-EED implementation, which took days to fully optimise a single 256×256 image, R-EED-RGB compression now takes ≈ 2 min to find the mask and ≈ 1 h for tonal optimisation. In LP mode, this time is further reduced to ≈ 1 min for tree-building and ≈ 30 min for tonal optimisation (on *Intel Xeon CPU W3565@3.20GHz* with *Nvidia Geforce GTX 460*). In both cases, a typical decoding takes ≈ 0.8 seconds

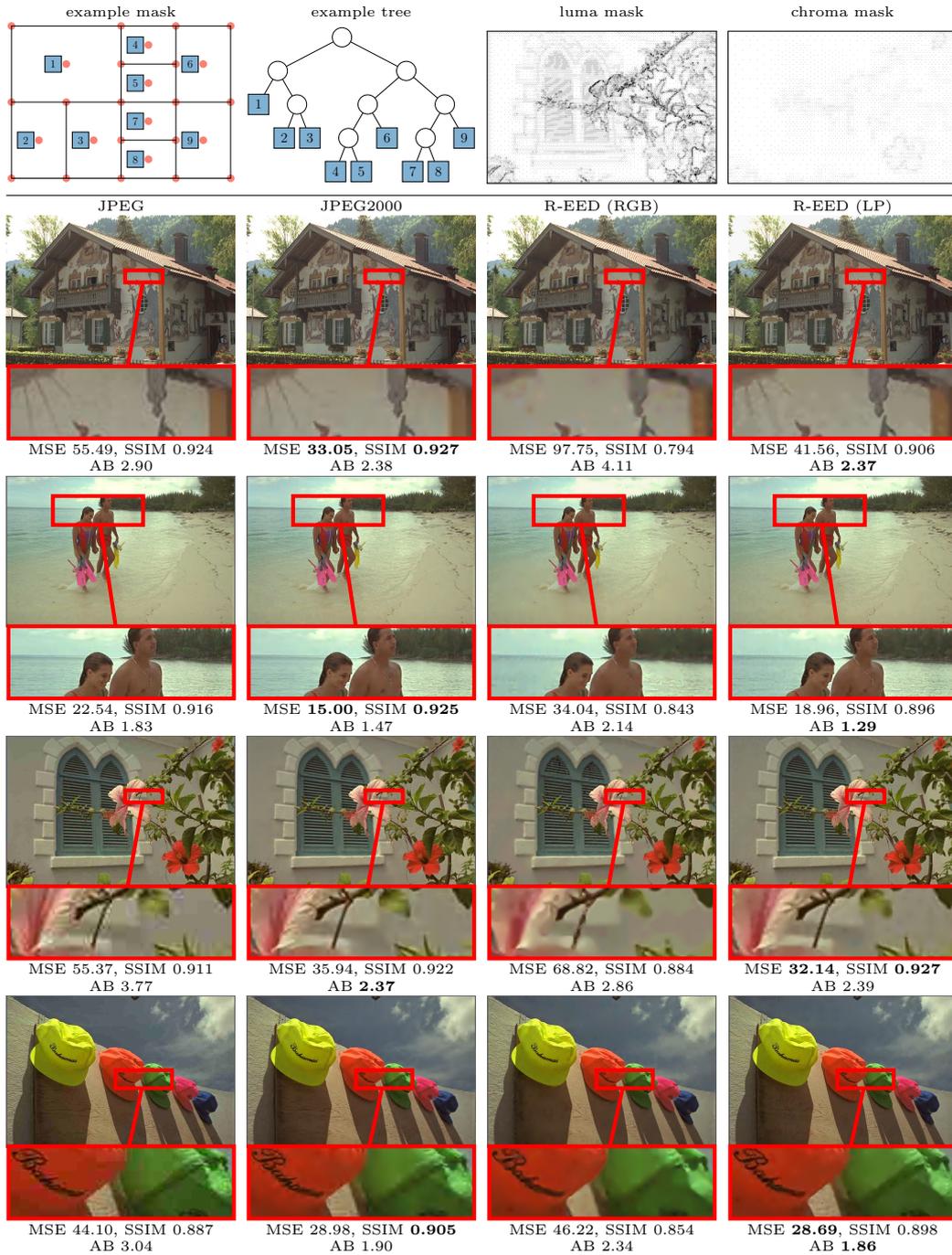


Figure 3: (a) **Top row:** Illustration of mask generation. From left to right: example subdivision with corresponding mask points in red; subdivision tree with numbered leaf nodes that correspond to subimages; luma and chroma masks for Kodak image 7 at 60:1. (b) **Remaining Rows:** Results for the Kodak images 24 at 20:1, 12 at 40:1, 7 at 60:1, and 3 at 80:1 (768 × 512 pixels). Three error measures are used: For MSE in RGB space and Euclidean distance AB in the a and b channels of CIELab lower is better, for SSIM in the luma channel higher is better.

which amounts primarily to PAQ decompression.

In our conference paper [7] we have focused on classic test images with low amounts of texture. Here we consider the Kodak database [8]: It is frequently used for the evaluation of compression methods and contains also highly textured images. For a quantitative evaluation, we consider mainly the mean squared error (MSE) over all three channels of RGB space. Since structural information is particularly important for perception, we also use the structural similarity index (SSIM), a perceptually motivated measure for grayscale-converted color images [51]. Moreover, we measure color distortion in the a and b channels of CIELab space.

The quantitative evaluation in Fig. 2 demonstrates that our LP-mode can indeed outperform standard R-EED by a large margin in all three error measures. This shows that LP-mode is not merely a perceptual improvement, but also makes better use of the given bit budget. In particular, the additional known data in the luma channel enables it to reconstruct more small-scale details in textured images than R-EED. This helps to close the gap towards the transform-based coders which have the advantage that they can deal better with textured images.

Compared to JPEG, R-EED (LP) is consistently better w.r.t the overall RGB-MSE. JPEG achieves a slight advantage in the luma channel by reducing the color accuracy significantly. On average, R-EED (LP) does not quite reach the quality of JPEG2000 on the full database. However, it is able to surpass it for those images of the Kodak database that have only moderate amounts of texture. Fig. 3 illustrates the advantages and limitations of R-EED (LP): On textured images with low compression rates, JPEG2000 still yields better results. However, for images with moderate amounts of texture, it preserves more detail and accurate colors without introducing blocking artifacts like JPEG and JPEG2000. In particular, R-EED (LP) offers the best CIELab distance consistently over all compression ratios.

4 Conclusion and Outlook

We have presented a new diffusion-based colorizations framework in YCbCr space and have shown that the method of Levin et al. is related to isotropic diffusion inpainting. Our luma-guided anisotropic diffusion outperforms the approach of Levin et al. and can reconstruct color from very sparse known data. This makes it a valuable tool for compression: Our luma preference mode introduces perceptive coding to PDE-based compression. This provides another important step towards reaching the same sophisticated level of engineering as transform-based codecs. An evaluation on the Kodak database

demonstrates that on images with medium amounts of texture, LP mode can beat both PDE- and transform-based competitors. Even for highly textured images, our method outperforms JPEG and comes close to the quality of JPEG2000.

Our contributions open up interesting perspectives for future work: In combination with optic flow or 3-D diffusion, luma-guided diffusion is promising for video colorization. Moreover, the new higher-order models could be evaluated in different applications. On the compression side, the performance on textured images can be addressed by applying texture synthesis in the luma channel. Our general concept of LP mode can be extended with any new reconstruction approaches such as promising hybrid approaches with non-local and PDE-based inpainting [28]. Overall, our contributions demonstrate that there is still a lot of unexplored potential for new applications in diffusion-based image processing.

References

- [1] D. Šykora, J. Buriánek, and J. Žára, “Unsupervised colorization of black-and-white cartoons,” in *Proc. 3rd International Symposium on Non-Photorealistic Animation and Rendering*, Annecy, France, Jun. 2004, pp. 121–127.
- [2] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” in *Proc. SIGGRAPH 2004*, New York, NY, Aug. 2004, pp. 689–694.
- [3] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, and A. Bruhn, “Understanding, optimising, and extending data compression with anisotropic diffusion,” *International Journal of Computer Vision*, vol. 108, no. 3, pp. 222–240, Jul. 2014.
- [4] J. Weickert, “Theoretical foundations of anisotropic diffusion in image processing,” *Computing Supplement*, vol. 11, pp. 221–236, 1996.
- [5] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York: Springer, 1992.
- [6] D. S. Taubman and M. W. Marcellin, Eds., *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Boston: Kluwer, 2002.
- [7] P. Peter and J. Weickert, “Colour image compression with anisotropic diffusion,” in *Proc. 21st IEEE International Conference on Image Processing*, Paris, France, Oct. 2014, pp. 4822–4862.

- [8] Eastman Kodak Company. (1999) Kodak true color image suite. [Online]. Available: <http://r0k.us/graphics/kodak/>
- [9] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, “Natural image colorization,” in *Proc. 18th Eurographics Conference on Rendering Techniques*, Grenoble, France, 2007, pp. 309–320.
- [10] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, “An adaptive edge detection based colorization algorithm and its applications,” in *Proc. 13th Annual ACM International Conference on Multimedia*, Singapore, Nov. 2005, pp. 351–354.
- [11] M. H. Quang, S. H. Kang, and T. M. Le, “Image and video colorization using vector-valued reproducing kernel Hilbert spaces,” *Journal of Mathematical Imaging and Vision*, vol. 37, no. 1, pp. 49–65, 2010.
- [12] Q. Yang, “Recursive bilateral filtering,” in *Computer Vision – ECCV 2012, Part I*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin: Springer, 2012, vol. 7572, pp. 399–413.
- [13] W. Casaca, M. Colnago, and L. G. Nonato, “Interactive image colorization using Laplacian coordinates,” in *Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science, vol. 9257. Berlin: Springer, 2015, pp. 675–686.
- [14] R. Irony, D. Cohen-Or, and D. Lischinski, “Colorization by example,” in *Proc. 16th Eurographics Symposium on Rendering*, Konstanz, Germany, Jun. 2005, pp. 201–210.
- [15] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentation by expectation-maximization,” in *Proc. 2005 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, San Diego, CA, Jun. 2005, pp. 747–754.
- [16] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 277–280, 2002.
- [17] M. Hua, X. Bie, M. Zhang, and W. Wang, “Edge-aware gradient domain optimization framework for image filtering by local propagation,” in *Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, Jun. 2014, pp. 2838–2845.

- [18] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.
- [19] Y. Qu, T.-T. Wong, and P.-A. Heng, “Manga colorization,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1214–1220, 2006.
- [20] J. B. Greer, A. L. Bertozzi, and G. Sapiro, “Fourth order partial differential equations on general geometries,” *Journal of Computational Physics*, vol. 216, no. 1, pp. 216–246, 2006.
- [21] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel, “Image compression with anisotropic diffusion,” *Journal of Mathematical Imaging and Vision*, vol. 31, no. 2–3, pp. 255–269, 2008.
- [22] T. Chan and J. Shen, “Mathematical models for local nontexture inpaintings,” *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, Jul. 2002.
- [23] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D*, vol. 60, pp. 259–268, Nov. 1992.
- [24] M. Mainberger, A. Bruhn, J. Weickert, and S. Forchhammer, “Edge-based compression of cartoon-like images with homogeneous diffusion,” *Pattern Recognition*, vol. 44, no. 9, pp. 1859–1873, 2011.
- [25] J. Gautier, O. L. Meur, and C. Guillemot, “Efficient depth map compression based on lossless edge coding and diffusion,” in *Proc. 29th Picture Coding Symposium*, Krakow, Poland, May 2012, pp. 81–84.
- [26] Y. Li, M. Sjöström, U. Jennehag, and R. Olsson, “A scalable coding approach for high quality depth image compression.” in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Zurich, Switzerland, Oct. 2012, pp. 1–4.
- [27] S. Hoffmann, M. Mainberger, J. Weickert, and M. Puhl, “Compression of depth maps with segment-based homogeneous diffusion,” in *Scale-Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds. Berlin: Springer, 2013, vol. 7893, pp. 319–330.
- [28] P. Peter and J. Weickert, “Compressing images with diffusion- and exemplar-based inpainting,” in *Scale-Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, J.-F. Aujol,

- M. Nikolova, and N. Papadakis, Eds. Berlin: Springer, 2015, vol. 9087, pp. 154–165.
- [29] L. Hoeltgen, M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, S. Setzer, D. Johannsen, F. Neumann, and B. Doerr, “Optimising spatial and tonal data for PDE-based inpainting,” in *Variational Methods in Image Analysis*. Berlin: De Gruyter, 2016, to appear.
- [30] T. Horiuchi and S. Tominaga, “Color image coding by colorization approach,” *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 158273, 2008.
- [31] L. Cheng and S. Vishwanathan, “Learning to compress images and videos,” in *Proc. 24th International Conference on Machine Learning*, Corvallis, OR, Jun. 2007, pp. 161–168.
- [32] S. Ono, T. Miyata, and Y. Sakai, “Colorization-based coding by focusing on characteristics of colorization bases,” in *Proc. Picture Coding Symposium (PCS) 2010*, Dec. 2010, pp. 230–233.
- [33] H. Noda, N. Takao, and M. Niimi, “Colorization in YCbCr space and its application to improve quality of JPEG color images,” in *Proc. 14th IEEE International Conference on Image Processing*, vol. 4, San Antonio, TX, Sep. 2007, pp. 385–388.
- [34] H. Noda and M. Niimi, “Local MAP estimation for quality improvement of compressed color images,” *Pattern Recognition*, vol. 44, no. 4, pp. 788–793, 2011.
- [35] S. Lee, S.-W. Park, P. Oh, and M. G. Kang, “Colorization-based compression using optimization,” *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2627–2636, 2013.
- [36] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart: Teubner, 1998.
- [37] G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz, “Nonlinear anisotropic filtering of MRI data,” *IEEE Transactions on Medical Imaging*, vol. 11, pp. 221–232, 1992.
- [38] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, “Deterministic edge-preserving regularization in computed imaging,” *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.

- [39] S. Di Zenzo, “A note on the gradient of a multi-image,” *Computer Vision, Graphics and Image Processing*, vol. 33, pp. 116–125, 1986.
- [40] W. K. Pratt, *Digital Image Processing: PIKS Inside*, 3rd ed. New York: Wiley, 2001.
- [41] J. Weickert, “Nonlinear diffusion filtering,” in *Handbook on Computer Vision and Applications, Vol. 2: Signal Processing and Pattern Recognition*, B. Jähne, H. Haußecker, and P. Geißler, Eds. San Diego: Academic Press, 1999, pp. 423–450.
- [42] J. Weickert, M. Welk, and M. Wickert, “L2-stable nonstandard finite differences for anisotropic diffusion,” in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds. Berlin: Springer, 2013, vol. 7893, pp. 380–391.
- [43] J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn, “Cyclic schemes for PDE-based image analysis,” *International Journal of Computer Vision*, vol. 118, no. 3, pp. 275–299, Jul. 2016.
- [44] International Organization for Standardization, “ISO 11664-4:2008 (CIE S 014-4/E:2007): Colorimetry – part 4: CIE 1976 L*a*b* colour space,” Standard, 2008.
- [45] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. Eighth International Conference on Computer Vision*, Vancouver, Canada, Jul. 2001, pp. 416–423.
- [46] Z. Belhachmi, D. Bucur, B. Burgeth, and J. Weickert, “How to choose interpolation data in images,” *SIAM Journal on Applied Mathematics*, vol. 70, no. 1, pp. 333–352, 2009.
- [47] Y. Chen, R. Ranftl, and T. Pock, “A bi-level view of inpainting-based image compression,” in *Proc. 19th Computer Vision Winter Workshop*, Křtiny, Czech Republic, Feb. 2014, pp. 19–26.
- [48] J. J. Rissanen, “Generalized Kraft inequality and arithmetic coding,” *IBM Journal of Research and Development*, vol. 20, no. 3, pp. 198–203, 1976.

- [49] M. Mahoney, “Adaptive weighing of context models for lossless data compression,” Florida Institute of Technology, Melbourne, Florida, Tech. Rep. CS-2005-16, Dec. 2005.
- [50] C. J. van den Branden Lambrecht, Ed., *Vision Models and Applications to Image and Video Processing*. Boston: Kluwer, 2001.
- [51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.