



Published in Image Processing On Line on 2016-04-20.  
 Submitted on 2015-07-02, accepted on 2016-02-25.  
 ISSN 2105-1232 © 2016 IPOL & the authors CC-BY-NC-SA  
 This article is available online with supplementary materials,  
 software, datasets and online demo at  
<https://doi.org/10.5201/ipol.2016.141>

# On the Implementation of Collaborative TV Regularization: Application to Cartoon+Texture Decomposition

Joan Duran<sup>1</sup>, Michael Moeller<sup>2</sup>, Catalina Sbert<sup>1</sup>, Daniel Cremers<sup>2</sup>

<sup>1</sup> Department of Mathematics and Computer Science, University of Balearic Islands, Spain  
 (joan.duran@uib.es, catalina.sbert@uib.es)

<sup>2</sup> Department of Mathematics and Computer Science, Technical University of Munich, Germany  
 (michael.moeller@in.tum.de, cremers@tum.de)

*Communicated by* Antonin Chambolle and Sandra Doucet      *Demo edited by* Jose-Luis Lisani



This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:  
 J. Duran, M. Moeller, C. Sbert, D. Cremers. "Collaborative Total Variation: A General Framework for Vectorial TV Models" *SIAM Journal on Imaging Sciences*, 9(1), pp. 116–151, 2016.

## Abstract

This paper deals with the analysis, implementation, and comparison of several vector-valued total variation (TV) methods that extend the Rudin-Osher-Fatemi variational model to color images. By considering the discrete gradient of a multichannel image as a 3D structure with dimensions corresponding to the spatial extent, the differences to other pixels, and the color channels, we introduce in [J. Duran, M. Moeller, C. Sbert, and D. Cremers, "Collaborative Total Variation: A General Framework for Vectorial TV Models", *SIAM Journal on Imaging Sciences*, 9(1), pp. 116–151, 2016] collaborative sparsity enforcing norms for penalizing the resulting tensor. We call this class of regularizations collaborative total variation (CTV). We first analyze the denoising properties of each collaborative norm for suppressing color artifacts while preserving image features and aligning edges. We then describe the primal-dual hybrid gradient method for solving the minimization problem in detail. The resulting CTV-L2 variational model can successfully be applied to many image processing tasks. On the one hand, an extensive performance comparison of several collaborative norms for color image denoising is provided. On the other hand, we analyze the ability of different CTV methods for decomposing a multichannel image into a cartoon and a textural part. Finally, we also include a short discussion on alternative minimization methods and compare their computational efficiency.

## Source Code

ANSI C source code to produce the same results as the demo is accessible at the IPOL web part of this article<sup>1</sup>.

**Keywords:** denoising; cartoon + texture decomposition; color images; vectorial total variation; collaborative norm; proximity operator; primal-dual hybrid gradient algorithm

<sup>1</sup><https://doi.org/10.5201/ipol.2016.141>

# 1 Introduction

The most common degradation model in image restoration relates the observed data  $f$  to the underlying true image  $u$  by means of

$$f = Au + \eta, \quad (1)$$

where  $\eta$  is an i.i.d. zero-mean Gaussian variable representing the noise and  $A$  is a linear operator modelling the degradation of  $u$  caused, for instance, by optical blurring. Image denoising models may be seen as a special case by taking  $A$  to be the identity. Unfortunately, the problem of recovering  $u$  from (1) is ill-posed in the sense of Hadamard [22] in many practical applications. It is therefore necessary to introduce some *a priori* knowledge on the solution. In several mathematical frameworks, this prior often takes the form of a regularization term for an energy functional which is to be minimized. Observing that quadratic regularization did not allow recovering sharp discontinuities, Rudin, Osher and Fatemi proposed [33] the total variation (TV) regularization.

Let  $\Omega \subset \mathbb{R}^2$  be a bounded domain, then the total variation of a function  $u \in L^1(\Omega, \mathbb{R})$  is given by

$$\text{TV}(u) := \sup_{\xi \in \Xi} \left\{ \int_{\Omega} u(x, y) \operatorname{div}(\xi(x, y)) \, dx \, dy \right\},$$

where  $(x, y) \in \Omega$  and  $\Xi$  is a set of continuously differentiable and bounded functions with compact support in  $\Omega$ . The space of functions of bounded variation is then defined as

$$\text{BV}(\Omega) := \{u \in L^1(\Omega, \mathbb{R}) : \text{TV}(u) < \infty\}.$$

It is well known [1] that  $\text{TV}(u)$  is finite if and only if its distributional derivative  $Du$  is a finite Radon measure, in which case  $\text{TV}(u) = |Du|(\Omega)$ . If  $\nabla u \in L^1(\Omega, \mathbb{R}^2)$ , then  $\text{TV}(u) = \int_{\Omega} |\nabla u|$ . Without risk of ambiguity,  $\text{TV}(u)$  is usually denoted by  $\int_{\Omega} |\nabla u|$  even when  $u$  is not smooth enough to justify this integral notation. In this setting, TV denoising consists of solving the minimization problem

$$\min_{u \in \text{BV}(\Omega)} \int_{\Omega} |\nabla u| + \frac{\lambda}{2} \|u - f\|_2^2, \quad (2)$$

where  $\lambda$  is a positive weighting constant and  $f \in L^2(\Omega, \mathbb{R})$ .

In the grayscale setting, the total variation can be defined differently depending on the norm in which the functions in  $\Xi$  are bounded. Indeed,  $\Xi = \{\xi \in C_c^1(\Omega, \mathbb{R}^2) : \|\xi(x, y)\|_2 \leq 1, \forall (x, y) \in \Omega\}$  leads to the isotropic TV

$$\int_{\Omega} \sqrt{(\partial_x u(x, y))^2 + (\partial_y u(x, y))^2} \, dx \, dy,$$

where  $\nabla u(x, y) = (\partial_x u(x, y), \partial_y u(x, y))$  denotes the gradient in  $\mathbb{R}^2$ . On the other hand, the anisotropic TV follows from the choice  $\Xi = \{\xi \in C_c^1(\Omega, \mathbb{R}^2) : \|\xi(x, y)\|_{\infty} \leq 1, \forall (x, y) \in \Omega\}$

$$\int_{\Omega} (|\partial_x u(x, y)| + |\partial_y u(x, y)|) \, dx \, dy.$$

For color images  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^C$ , where  $C$  denotes the number of channels, several extensions of TV have been proposed depending not only on the coupling of the spatial derivatives but also on the coupling of color channels. The first known vectorial TV is due to Blomgren and Chan [6], who coupled the global weight of the TV contributions across channels. On the other hand, Attouch et al. [4] suggested the channel-independent variant

$$\sum_{k=1}^C \int_{\Omega} \sqrt{(\partial_x u_k(x, y))^2 + (\partial_y u_k(x, y))^2} \, dx \, dy,$$

which uses an  $\ell^1$  channel coupling. A special case of TV proposed in [35] and further studied in [8] follows from coupling both the spatial derivatives and the color channels with the  $\ell^2$  norm

$$\int_{\Omega} \sqrt{\sum_{k=1}^C \left( (\partial_x u_k(x, y))^2 + (\partial_y u_k(x, y))^2 \right)} dx dy.$$

Other versions of vectorial TV are based on Schatten  $p$ -norms, which penalize the  $\ell^p$  norm of the singular values of the Jacobian at each point [23, 35]. For typical color images, the singular values of matrices of the form

$$\begin{pmatrix} \partial_x u_1(x, y) & \partial_x u_2(x, y) & \partial_x u_3(x, y) \\ \partial_y u_1(x, y) & \partial_y u_2(x, y) & \partial_y u_3(x, y) \end{pmatrix} \quad (3)$$

are penalized. It was shown in [19] that the natural choice for vectorial TV from geometric measure theory is the Schatten  $\infty$ -norm, that is, the penalization of the largest singular value. Another important case is the nuclear norm that arises when one penalizes the singular values with the help of the  $\ell^1$  norm. It is a convex relaxation of minimizing the rank of the matrix in (3), so that the gradients (jumps) of different color channels are encouraged to point into the same direction.

Although the TV regularization was originally designed for image denoising, it has become one of the most popular regularizations for several image processing problems and has sparked a tremendous amount of research. In particular, the problem of decomposing an image into a cartoon part, which consists of the geometric and smoothly-varying component of the original image, and a textural part, which captures essentially the oscillating patterns (texture, details, and noise), can be convincingly solved by means of (2). Indeed, Meyer stated in [26] that the right decomposition within a variational framework is the one where the cartoon part has minimal TV while the oscillatory component has a minimal norm in a dual space of BV. In the end, the oscillatory component is simply computed as the difference between the original image and its cartoon part. We refer the reader to [9, 36] and references therein for more details.

We proposed in [16, 17] to take a generalized viewpoint that unifies most vectorial TV based models proposed in the literature. By considering the derivatives of a color image as a linear operator, one obtains a 3D data structure of the gradient: one dimension corresponding to the pixels, one dimension corresponding to the derivatives, and one dimension corresponding to the color channels. We penalize this structure using a different norm along each dimension.

In this paper, we provide an extensive performance comparison of different vectorial TV regularizations based on collaborative norms for both denoising and cartoon + texture decomposition of color images. We use the primal-dual hybrid gradient (PDHG) method [12, 18, 37] for solving the resulting minimization problems. Since the key to obtaining a fast PDHG algorithm is an efficient evaluation of the proximity operators arising from the collaborative norms, they will be considered and provided in detail. We also include a short discussion on alternative minimization methods and compare their computational efficiency.

## 2 Mathematical Background on Convex Analysis

In a general framework, assume that  $X$  is a finite dimensional Hilbert space endowed with an inner product  $\langle \cdot, \cdot \rangle$  and induced norm  $\|\cdot\|$ . By the Riesz representation theorem, we identify the dual space of  $X$  with  $X$  itself. Furthermore, assume throughout this section that  $f : X \rightarrow \mathbb{R} \cup \{\infty\}$  is a proper convex function. We refer to [21, 32] and references therein for further details.

The *convex conjugate* of  $f$ , also known as the *Legendre-Fenchel transform* of  $f$ , is the function  $f^* : X \rightarrow \mathbb{R} \cup \{\infty\}$  defined as

$$f^*(x^*) = \sup_{x \in X} \{ \langle x, x^* \rangle - f(x) \}, \quad x^* \in X.$$

For example, the Legendre-Fenchel transform of  $f(x) = \|x\|$  is

$$f^*(y) = \begin{cases} 0 & \text{if } \|y\|_* \leq 1, \\ +\infty & \text{otherwise,} \end{cases}$$

that is, the indicator function of the dual norm ball, with the dual norm given by

$$\|y\|_* = \sup_{x \in X} \{\langle x, y \rangle : \|x\| \leq 1\}.$$

Suppose now that  $f$  is further lower-semicontinuous. The *proximity operator* of  $f$  with parameter  $\tau > 0$ , denoted by  $\text{prox}_{\tau f} : X \rightarrow X$ , is defined as

$$\text{prox}_{\tau f}(x) = \arg \min_{y \in X} \left\{ \frac{1}{2} \|y - x\|_2^2 + \tau f(y) \right\}, \quad x \in X.$$

Since the function minimized on the righthand side is strongly convex, there exists a unique minimizer. The above definition indicates that  $\text{prox}_{\tau f}(x)$  is a point that compromises between minimizing  $f$  and being close to  $x$ . For differentiable functions and small values of  $\tau$ , the proximal mapping behaves as a kind of gradient step,  $\text{prox}_{\tau f}(x) \simeq x - \tau \nabla f(x)$ . This suggests that proximal operators may be useful in optimization. Finally, we recall the celebrated *Moureau's identity* that connects the proximity operator of a function and of its Legendre-Fenchel transform in the following way

$$x = \text{prox}_{\tau f}(x) + \tau \text{prox}_{\frac{1}{\tau} f^*} \left( \frac{x}{\tau} \right). \quad (4)$$

### 3 Collaborative Total Variation

For the rest of the paper, we will focus on the discrete setting. We consider a color image as a 2D matrix of size  $N \times C$  denoted by  $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_C) \in \mathbb{R}^{N \times C}$ , where  $N$  is the number of pixels,  $C$  is the number of channels, and  $\mathbf{u}_k = (u_{1,k}, \dots, u_{N,k})^\top$ . Up to a permutation of the dimensions, let us define the discrete gradient of  $\mathbf{u}$  to be a 3D matrix or tensor of the form  $D\mathbf{u} \in \mathbb{R}^{N \times M \times C}$  with  $M$  denoting the number of spatial derivatives (typically,  $M = 2$  and the differences in  $x$ - and  $y$ -direction are considered).

It makes sense to look at vectorial TV as applying *collaborative norms* to the discrete gradient of the image as we define in [17] and reproduce below for the sake of completeness.

**Definition 1.** Let  $\|\cdot\|_a : \mathbb{R}^N \rightarrow \mathbb{R}$  be any vector norm and  $\|\cdot\|_{\vec{b}} : \mathbb{R}^{M \times C} \rightarrow \mathbb{R}$  any matrix norm. Then, the collaborative norm of  $A \in \mathbb{R}^{N \times M \times C}$ , which will be denoted by  $\|\cdot\|_{\vec{b},a} : \mathbb{R}^{N \times M \times C} \rightarrow \mathbb{R}$ , is defined as

$$\|A\|_{\vec{b},a} := \|v\|_a, \quad \text{with } v_i = \|A_{i,\cdot,\cdot}\|_{\vec{b}}, \quad \forall i \in \{1, \dots, N\}, \quad (5)$$

where  $A_{i,\cdot,\cdot}$  is the (two-dimensional) submatrix obtained by stacking the second and third dimensions of  $A$  at the  $i$ -th position in the first dimension.

By setting  $\|\cdot\|_a := \ell^r$  and  $\|\cdot\|_{\vec{b}} := \ell^{p,q}$  in (5), we obtain the  $\ell^{p,q,r}$  norm defined as

$$\|A\|_{p,q,r} = \left( \sum_{i=1}^N \left( \sum_{j=1}^M \left( \sum_{k=1}^C |A_{i,j,k}|^p \right)^{q/p} \right)^{r/q} \right)^{1/r},$$

where any of the indices  $p$ ,  $q$  or  $r$  being equal to infinity means taking the maximum of the absolute values along the corresponding dimension. Since the previous norm is in general non invariant to

permutations, it is important to clearly state the order of the dimensions of the 3D structure. Let  $col$ ,  $der$ , and  $pix$  be abbreviations for *color*, *derivative*, and *pixel*, respectively. We propose to denote by  $\ell^{p,q,r}(dim_1, dim_2, dim_3)$ , where  $dim_1, dim_2, dim_3 \in \{col, der, pix\}$ , the collaborative norm resulting from first applying the  $\ell^p$  norm along the  $dim_1$  dimension, then the  $\ell^q$  norm along the  $dim_2$  dimension of the remaining 2D matrix, and, finally, the  $\ell^r$  norm to the remaining vector.

We also propose to generalize the framework introduced in [35] by setting  $\|\cdot\|_a := \ell^q$  and  $\|\cdot\|_b := \text{Schatten-}p$  in (5), which leads to the collaborative  $(S^p, \ell^q)$  norm defined as

$$(S^p, \ell^q)(A) = \left( \sum_{i=1}^N \left\| \begin{pmatrix} A_{i,1,1} & \cdots & A_{i,1,C} \\ \vdots & \ddots & \vdots \\ A_{i,M,1} & \cdots & A_{i,M,C} \end{pmatrix} \right\|_{S^p}^q \right)^{1/q}.$$

We will write  $(S^p(dim_1, dim_2), \ell^q(dim_3))$  to make clear that we first penalize the singular values of each 2D matrix arising from each position along the  $dim_1$  dimension, and then the  $\ell^q$  norm is applied to the remaining vector.

The proposed framework unifies several pre-existing definitions for vectorial TV regularization. Table 1 gives an overview of all local vectorial TV methods proposed in the literature the authors are aware of. Since many of these methods only involve either the isotropic or the anisotropic variant, we complete the table by including the missing parts. Interestingly, there are cases where one can imagine different versions due to an ambiguity of the permutation of the dimensions. We refer the reader to [17] for a more detailed discussion on the literature of vectorial total variation methods.

### 3.1 Denoising Behaviour of Collaborative TV Regularization

The question about which TV methods work well depends on the characteristics of the data, so we cannot expect one regularization to be the best choice for all types of images. This makes the understanding of what prior hypothesis different collaborative norms are based on even more important. To simplify, we will focus on what kind of color coupling is most suitable for certain types of data.

Figure 1 shows the results for an image with independent color channels. In this case, the  $\ell^{1,1,1}$  regularization is superior to the others because of penalizing each channel independently, although the restored image provided by the  $\ell^{2,1,1}$  norm is on the same footing (which makes sense since the difference between both is weak). On the contrary, Figure 2 displays the performance of the same collaborative norms on an almost black-and-white image. Since there is a strong inter-channel correlation, the  $\ell^{\infty,1,1}$  regularization provides the best denoised image whereas the  $\ell^{1,1,1}$  one exhibits the strongest color artifacts. We can see that, although it performs not as well as  $\ell^{\infty,1,1}$ , the  $(S^1, \ell^1)$  approach – which forces the gradient vectors of all channels to be parallel – almost suppresses all color artifacts. Finally, Figure 3 shows the results on an image where we leave open if the colored wave pattern is signal content or noise. We see that the  $\ell^{1,1,1}$  regularization eliminates all noise from constant regions but keeps the wave pattern. On the contrary, the  $\ell^{\infty,1,1}$  norm eliminates the colored waves entirely. The regularization based on an  $\ell^2$  channel coupling suppresses but does not eliminate them. Therefore, we expect a color coupling with an  $\ell^p$  norm to be stronger the larger  $p$  is.

## 4 Numerical Minimization

In this section, we present the primal-dual hybrid gradient (PDHG) method [12, 18, 37], a powerful optimization algorithm that breaks complex problems into simple sub-steps. For a better readability, we use the notations  $X = \mathbb{R}^{N \times C}$  and  $Y = \mathbb{R}^{N \times M \times C}$ .

Literature	Continuous Formulation	Our Framework
[4]	$\sum_{k=1}^C \int_{\Omega} \sqrt{(\partial_x u_k(x, y))^2 + (\partial_y u_k(x, y))^2} dx dy$	$\ell^{2,1,1}(der, pix, col)$
Anisotropic variant	$\sum_{k=1}^C \int_{\Omega} ( \partial_x u_k(x, y)  +  \partial_y u_k(x, y) ) dx dy$	$\ell^{1,1,1}(der, pix, col)$
[8, 34]	$\int_{\Omega} \sqrt{\sum_{k=1}^C ((\partial_x u_k(x, y))^2 + (\partial_y u_k(x, y))^2)} dx dy$	$\ell^{2,2,1}(der, col, pix)$
Anisotropic variants	$\int_{\Omega} \left( \sqrt{\sum_{k=1}^C (\partial_x u_k(x, y))^2} + \sqrt{\sum_{k=1}^C (\partial_y u_k(x, y))^2} \right) dx dy$	$\ell^{2,1,1}(col, der, pix)$
	$\int_{\Omega} \sqrt{\sum_{k=1}^C ( \partial_x u_k(x, y)  +  \partial_y u_k(x, y) )^2} dx dy$	$\ell^{1,2,1}(der, col, pix)$
Strong coupling	$\int_{\Omega} \left( \max_{1 \leq k \leq C}  \partial_x u_k(x, y)  + \max_{1 \leq k \leq C}  \partial_y u_k(x, y)  \right) dx dy$	$\ell^{\infty,1,1}(col, der, pix)$
	$\int_{\Omega} \max_{1 \leq k \leq C} ( \partial_x u_k(x, y)  +  \partial_y u_k(x, y) ) dx dy$	$\ell^{1,\infty,1}(der, col, pix)$
Isotropic variants	$\int_{\Omega} \sqrt{\left( \max_{1 \leq k \leq C}  \partial_x u_k(x, y)  \right)^2 + \left( \max_{1 \leq k \leq C}  \partial_y u_k(x, y)  \right)^2} dx dy$	$\ell^{\infty,2,1}(col, der, pix)$
	$\int_{\Omega} \max_{1 \leq k \leq C} \sqrt{(\partial_x u(x, y))^2 + (\partial_y u(x, y))^2} dx dy$	$\ell^{2,\infty,1}(der, col, pix)$
Supremum variant	$\int_{\Omega} \left( \max_{1 \leq k \leq C} \left( \max \{  \partial_x u_k(x, y) ,  \partial_y u_k(x, y)  \} \right) \right) dx dy$	$\ell^{\infty,\infty,1}(col, der, pix)$
[34, 23]	$\int_{\Omega} \sum_{i=1}^r  \sigma_i(\nabla u(x, y))  dx dy$	$(S^1(col, der), \ell^1(pix))$
Frobenius norm	$\int_{\Omega} \sqrt{\sum_{i=1}^r (\sigma_i(\nabla u(x, y)))^2} dx dy$	$(S^2(col, der), \ell^1(pix))$
[34, 19]	$\int_{\Omega} \max_{1 \leq i \leq r} \sigma_i(\nabla u(x, y)) dx dy$	$(S^{\infty}(col, der), \ell^1(pix))$

Table 1: Overview of the different vectorial TV approaches proposed in the literature as well as a completion regarding their isotropic and anisotropic relatives.

## 4.1 The Primal-Dual Hybrid Gradient Method

The general convex minimization problem we are concerned with is

$$\min_{\mathbf{u} \in X} G(\mathbf{u}) + F(D\mathbf{u}), \quad (6)$$

where  $G : X \rightarrow \mathbb{R}$  and  $F : Y \rightarrow \mathbb{R}$  are proper convex lower-semicontinuous functionals, and  $D : X \rightarrow Y$  is the linear operator associated to the 3D structure of the gradient.

Given  $(\mathbf{u}^0, \mathbf{g}^0, \mathbf{q}^0) \in X \times Y \times Y$  and  $\bar{\mathbf{u}}^0 = \mathbf{u}^0$  as initializations, the PDHG algorithm iteratively

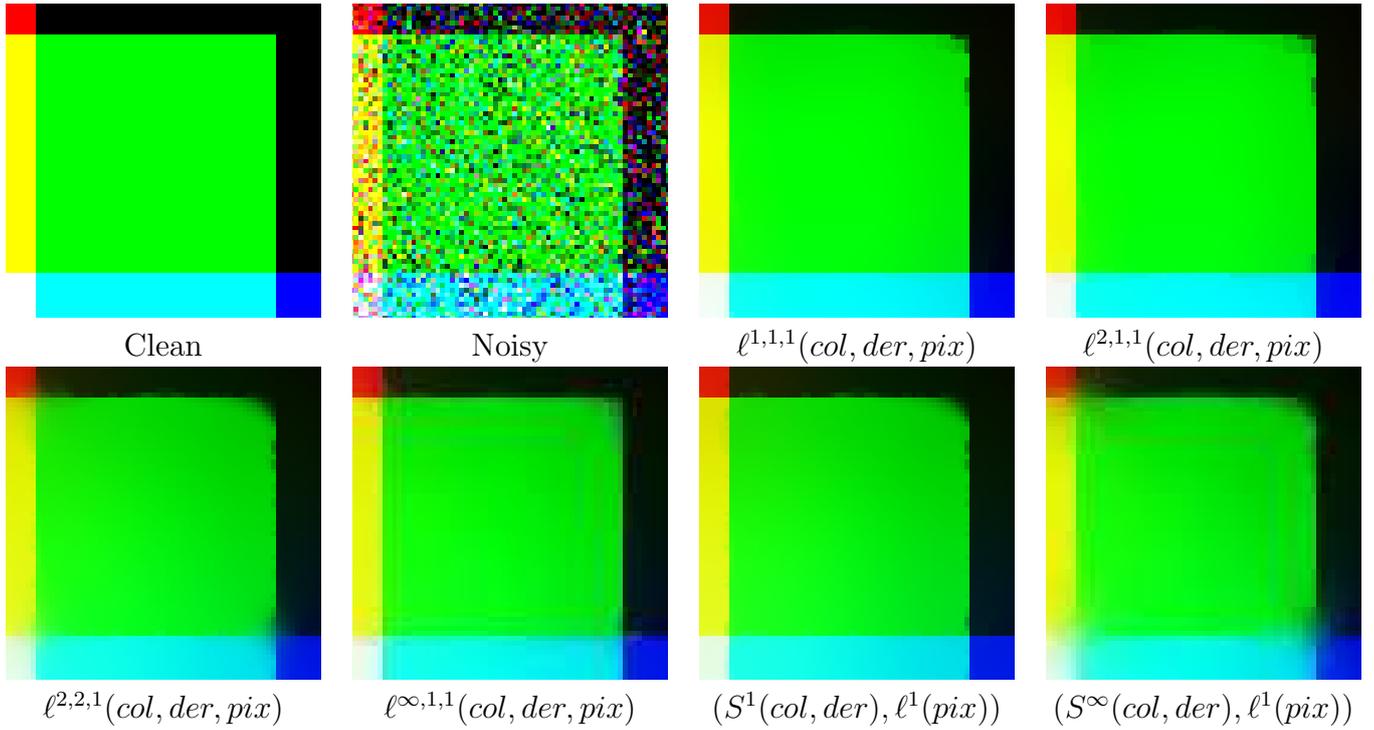


Figure 1: Denoising a synthetic image with totally uncorrelated channels. Observe that the TV regularization that makes use of the  $\ell^{1,1,1}$  norm is superior to the others because it penalizes each color channel independently.

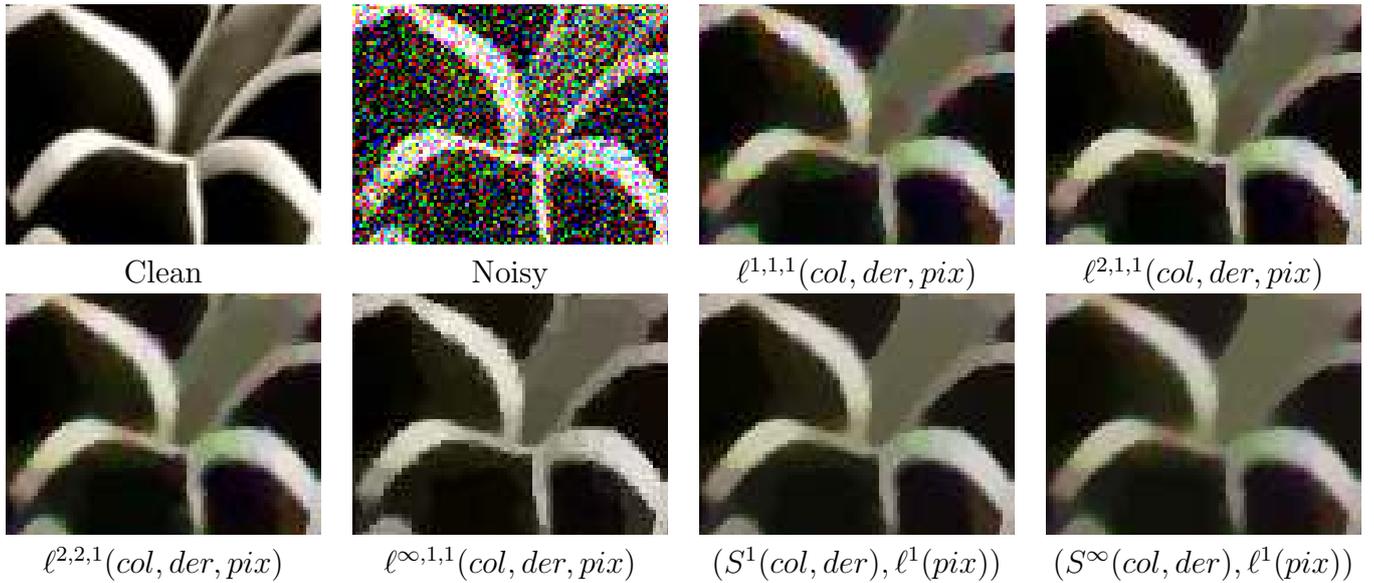


Figure 2: Denoising a synthetic image with high inter-channel correlation. Since the color channels are highly correlated, the  $\ell^{\infty,1,1}$  norm provides the best denoised image contrary to the  $\ell^{1,1,1}$  norm, which exhibits the strongest color artifacts.

computes the solution of (6) by means of

$$\begin{aligned}
 \mathbf{u}^{n+1} &= \text{prox}_{\tau_n G}(\mathbf{u}^n - \tau_n D^\top \mathbf{q}^n), \\
 \bar{\mathbf{u}}^{n+1} &= \mathbf{u}^{n+1} + (\mathbf{u}^{n+1} - \mathbf{u}^n), \\
 \mathbf{g}^{n+1} &= \text{prox}_{\frac{1}{\sigma_n} F}\left(D\bar{\mathbf{u}}^{n+1} + \frac{1}{\sigma_n} \mathbf{q}^n\right), \\
 \mathbf{q}^{n+1} &= \mathbf{q}^n + \sigma_n (D\bar{\mathbf{u}}^{n+1} - \mathbf{g}^{n+1}),
 \end{aligned} \tag{7}$$

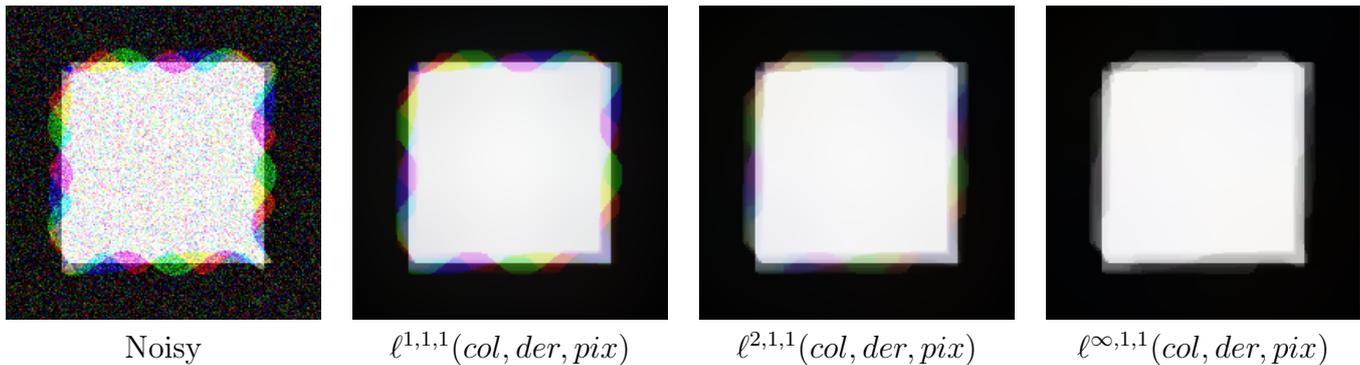


Figure 3: Denoising a synthetic image where we leave open if the colored wave pattern is signal content or noise. One observes that the  $\ell^{1,1,1}$  norm keeps the colored waves but the  $\ell^{\infty,1,1}$  norm eliminates them totally. In between both approaches, the  $\ell^2$  norm suppresses but does not eliminate the wave pattern.

where  $\tau_n$  and  $\sigma_n$  are step-size parameters that can, for instance, be chosen fixed with  $\tau\sigma\|D\|^2 \leq 1$  to guarantee convergence. We refer the reader to [18, 10, 31] for more detailed information about the algorithm as well as its connection to other first order methods. In subsection 4.4 we will discuss a strategy for accelerating the above algorithm by choosing  $\tau$  and  $\sigma$  adaptively.

## 4.2 Proximal Operator of the Data Term

By introducing collaborative total variation regularizations, the extension of the ROF model to color images reads as follows

$$\min_{\mathbf{u} \in X} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{f}\|_2^2 + \|D\mathbf{u}\|_{\vec{b},a}, \quad (8)$$

where  $\mathbf{f} \in X$  is the noisy image and  $\lambda > 0$  controls the trade-off between a good fit to  $\mathbf{f}$  and a (piecewise) smooth solution. Casting (8) in the form of (6), one obtains that  $G(u) = \frac{\lambda}{2} \|\mathbf{u} - \mathbf{f}\|_2^2$  is the data-fidelity term and  $F(D\mathbf{u}) = \|D\mathbf{u}\|_{\vec{b},a}$  is the regularization term.

The proximity operator of  $G$  has a closed-form solution

$$\hat{\mathbf{u}} = \text{prox}_{\tau G}(\mathbf{u}) \Leftrightarrow \hat{\mathbf{u}} = \arg \min_{\mathbf{v} \in X} \left\{ \frac{1}{2} \|\mathbf{v} - \mathbf{u}\|_2^2 + \tau \frac{\lambda}{2} \|\mathbf{v} - \mathbf{f}\|_2^2 \right\} \Leftrightarrow \hat{\mathbf{u}} = \frac{\mathbf{u} + \tau \lambda \mathbf{f}}{1 + \tau \lambda}.$$

Therefore, the solution of  $\mathbf{u}^{n+1} = \text{prox}_{\tau_n G}(\mathbf{u}^n - \tau_n D^\top \mathbf{q}^n)$  is given by

$$\mathbf{u}^{n+1} = \frac{\mathbf{u}^n + \tau_n (\text{div } \mathbf{q}^n + \lambda \mathbf{f})}{1 + \tau_n \lambda}, \quad (9)$$

where the discrete divergence operator,  $\text{div} : Y \rightarrow X$ , is defined in analogy with the continuous setting as  $\langle -\text{div } \mathbf{q}, \mathbf{u} \rangle_X = \langle \mathbf{q}, D\mathbf{u} \rangle_Y$ .

## 4.3 Proximal Operators of Collaborative TV Regularizations

It is remarkable that all variants of different collaborative norms imposed on the gradient of the color channels in (8) can be solved efficiently by the PDHG method (7). The only thing that changes when changing the regularization term is the proximity operator of  $F$ .

### 4.3.1 Proximity Operators of $\ell^{p,q,r}$ Norms

In the following, we use the colon to denote all elements along one dimension. For example, the  $\ell^p$  norm of  $A \in \mathbb{R}^{M \times N \times C}$  with respect to its third dimension reads  $\|A_{i,j,:}\|_p^p = \sum_{k=1}^C |A_{i,j,k}|^p$ .

**The  $\ell^{1,1,1}$  norm.** The proximity operator decouples in all variables and each problem just contains an absolute value penalty. It is well known that the resulting proximity operator becomes

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{1,1,1}}(A)\right)_{i,j,k} = \max\left(|A_{i,j,k}| - \frac{1}{\sigma}, 0\right) \text{sign}(A_{i,j,k}).$$

**The  $\ell^{2,1,1}$  norm.** The case of an inner  $\ell^2$  coupling is also well known in the image processing literature and its proximity operator is often called (generalized) shrinkage. By a short computation or via the application of [17, Theorem 7] we find

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{2,1,1}}(A)\right)_{i,j,k} = \max\left(\|A_{i,j,:}\|_2 - \frac{1}{\sigma}, 0\right) \frac{A_{i,j,k}}{\|A_{i,j,:}\|_2}.$$

**The  $\ell^{2,2,1}$  norm.** Similarly to the previous case, the proximity operator of the  $\ell^{2,2,1}$  norm is

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{2,2,1}}(A)\right)_{i,j,k} = \max\left(\|A_{i,:}\|_{2,2} - \frac{1}{\sigma}, 0\right) \frac{A_{i,j,k}}{\|A_{i,:}\|_{2,2}}.$$

**The  $\ell^{\infty,1,1}$  norm.** Whenever the supremum norm is involved it is more convenient to use Moreau's identity (4) to express the proximity operator by the proximity operator of its dual, which results in a projection onto the  $\ell^1$  ball. We obtain

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{\infty,1,1}}(A)\right)_{i,j,k} = A_{i,j,k} - \frac{1}{\sigma} \text{sign}(A_{i,j,k}) \left(\text{proj}_{\|\cdot\|_1 \leq 1}(\sigma|A_{i,j,:}|)\right)_{i,j,k}.$$

where  $|A_{i,j,:}|$  denotes the component-wise absolute value of vector  $A_{i,j,:}$ . Although the projection onto the  $\ell^1$  ball does not admit a closed form solution, several efficient algorithms have been proposed in the literature and we refer the reader to [15] for an example.

**The  $\ell^{\infty,\infty,1}$  norm.** Similarly to the previous norm, it follows that

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{\infty,\infty,1}}(A)\right)_{i,j,k} = A_{i,j,k} - \frac{1}{\sigma} \text{sign}(A_{i,j,k}) \left(\text{proj}_{\|\cdot\|_1 \leq 1}(\sigma|A_{i,:}|)\right)_{i,j,k}.$$

with  $\text{proj}_{\|\cdot\|_1 \leq 1}$  denoting the projection operator onto the unit  $\ell^{1,1}$  ball.

**The  $\ell^{\infty,2,1}$  norm.** Analogously, the proximity operator of the  $\ell^{\infty,2,1}$  norm is

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{\infty,2,1}}(A)\right)_{i,j,k} = A_{i,j,k} - \frac{1}{\sigma} \text{sign}(A_{i,j,k}) \left(\text{proj}_{\|\cdot\|_{1,2} \leq 1}(\sigma|A_{i,:}|)\right)_{i,j,k}.$$

where  $\text{proj}_{\|\cdot\|_{1,2} \leq 1}$  denotes the projection operator onto the unit  $\ell^{1,2}$  ball. Opposed to the previous case, a projection onto the unit  $\ell^{1,2}$  ball is much more expensive to compute than an  $\ell^{1,1}$  ball projection and requires the solution of a convex optimization problem.

**The  $\ell^{2,\infty,1}$  norm.** If the order of norms of the previous case is switched such that the inner norm is  $\ell^2$ , we can use [17, Theorem 7] to obtain the proximity operator as follows

$$\left(\text{prox}_{\frac{1}{\sigma}\|\cdot\|_{2,\infty,1}}(A)\right)_{i,j,k} = \frac{A_{i,j,k}}{\|A_{i,j,:}\|_2} \max\left(\|A_{i,j,:}\|_2 - \frac{1}{\sigma}v_{i,j}, 0\right),$$

where

$$v_{i,j} = \left(\text{proj}_{\|\cdot\|_1 \leq 1}\left(\sigma(\|A_{i,j,:}\|_2)_j\right)\right)_{i,j}.$$

In the above formula,  $(\|A_{i,j,:}\|_2)_j$  stands for the vector obtained by stacking  $\|A_{i,j,:}\|_2$  for all  $j \in \{1, \dots, M\}$ . Once more,  $\text{proj}_{\|\cdot\|_1 \leq 1}$  is merely the projection onto the  $\ell^1$  ball.

### 4.3.2 Proximity Operators of $(S^p, \ell^q)$ Norms

Let us finally discuss the proximity operators of the  $(S^1(\text{col}, \text{der}), \ell^1(\text{pix}))$  and  $(S^\infty(\text{col}, \text{der}), \ell^1(\text{pix}))$  norms. For this purpose, define the submatrix

$$B_i := \begin{pmatrix} A_{i,1,1} & \cdots & A_{i,1,C} \\ \vdots & \ddots & \vdots \\ A_{i,M,1} & \cdots & A_{i,M,C} \end{pmatrix},$$

for each  $i \in \{1, \dots, N\}$ . Since the considered regularizations have an outer  $\ell^1$  norm, the computation of their proximity operators decouples at each pixel. We are thus left with a problem of the form

$$\min_{Z \in \mathbb{R}^{C \times M}} \frac{1}{2} \|Z - B\|_2^2 + \alpha \|Z\|_{S^p}, \quad (10)$$

where  $B := B_i^T \in \mathbb{R}^{C \times M}$ . Let  $U \Sigma_0 V^T$  be the singular value decomposition of  $B$ , with  $U$  and  $V$  being orthonormal and  $\Sigma_0$  being diagonal. Since  $S^p$  norms are invariant with respect to multiplication with orthonormal matrices, our optimization problem is equivalent to

$$\min_{Z \in \mathbb{R}^{C \times M}} \frac{1}{2} \|U^T Z V - \Sigma_0\|_2^2 + \alpha \|U^T Z V\|_{S^p}.$$

Instead of minimizing over all  $Z$ , we can introduce a new variable  $\Sigma := U^T Z V$ , minimize over all  $\Sigma$ , and set  $\widehat{Z} = U \widehat{\Sigma} V^T$  after finding the minimizing argument  $\widehat{\Sigma}$ . Note that  $U$  and  $V$  are invertible, such that we are still minimizing over the full matrix space. The resulting optimization problem is

$$\min_{\Sigma \in \mathbb{R}^{r \times r}} \frac{1}{2} \|\Sigma - \Sigma_0\|_2^2 + \alpha \|\Sigma\|_{S^p},$$

where  $r$  is the number of nonzero singular values of  $B$ . The argument that minimizes the above expression has to be a diagonal matrix due to Mirsky's inequality [27]. In this case, the  $S^p$  norm coincides with the  $\ell^p$  norm of the diagonal, which is the reason why we can finally rewrite (10) as

$$\min_{s \in \mathbb{R}^r} \frac{1}{2} \|s - s_0\|_2^2 + \alpha \|s\|_p, \quad (11)$$

where  $s_0$  and  $s$  denote the diagonals of  $\Sigma_0$  and  $\Sigma$  respectively.

We would like to emphasize that it is not necessary to compute  $U$  in the singular value decomposition of  $B$ . Indeed, let  $\widehat{\Sigma}$  be the diagonal matrix which has the argument  $\widehat{s}$  that solves (11) as its

diagonal. From the singular value decomposition one has that  $BV = U\Sigma_0$ , and the commutation of diagonal matrices leads to the equalities

$$BV\widehat{\Sigma} = U\Sigma_0\widehat{\Sigma} = U\widehat{\Sigma}\Sigma_0.$$

Let  $\Sigma_0^\dagger$  be the pseudo-inverse matrix of  $\Sigma_0$ , which is given by

$$(\Sigma_0^\dagger)_{i,j} = \begin{cases} \frac{1}{(\Sigma_0)_{i,i}} & \text{if } i = j \text{ and } (s_0)_{i,i} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Multiplying each righthand side of the relation  $BV\widehat{\Sigma} = U\widehat{\Sigma}\Sigma_0$  by  $\Sigma_0^\dagger$ , one obtains

$$BV\widehat{\Sigma}\Sigma_0^\dagger = U\widehat{\Sigma}.$$

Finally, since  $\widehat{\Sigma}$  has at most as many nonzero diagonal entries as  $\Sigma_0$ , we conclude that the solution to (10) is

$$BV\widehat{\Sigma}\Sigma_0^\dagger V^T = U\widehat{\Sigma}V^T = \widehat{Z}. \quad (12)$$

Considering the left hand side of (12), we can see that only the matrices  $V$  and  $\Sigma_0$  are needed for solving the original optimization problem. Let us explain in detail how to compute them for usual  $3 \times 2$  matrices as arising from applying  $(S^p, \ell^q)$  norms to RGB color images. It is well known that the characteristic polynomial of the square symmetric matrix  $B^T B \in \mathbb{R}^{2 \times 2}$  can be written as

$$\sigma^2 - \text{tr}(B^T B)\sigma + \det(B^T B) = 0,$$

where  $\text{tr}(B^T B)$  and  $\det(B^T B)$  denote the trace and determinant of  $B^T B$ , respectively. Therefore, the singular values of  $B$  are explicitly given by<sup>2</sup>

$$\sigma_1 = \sqrt{\frac{\text{tr}(B^T B) + \Delta}{2}} \quad \text{and} \quad \sigma_2 = \sqrt{\frac{\text{tr}(B^T B) - \Delta}{2}},$$

with  $\Delta = \sqrt{\text{tr}^2(B^T B) - 4\det(B^T B)}$ . On the other hand, from the relation  $(B^T B)v_i = \sigma_i v_i$  one easily deduces that the eigenvectors of  $B^T B$ , displayed as columns of matrix  $V$ , are

$$V = \begin{cases} \begin{pmatrix} \sigma_1 - (B^T B)_{22} & \sigma_2 - (B^T B)_{22} \\ (B^T B)_{21} & (B^T B)_{21} \end{pmatrix}, & \text{if } (B^T B)_{21} \neq 0, \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{if } (B^T B)_{21} = 0 \text{ and } (B^T B)_{11} \geq (B^T B)_{22}, \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, & \text{if } (B^T B)_{21} = 0 \text{ and } (B^T B)_{11} < (B^T B)_{22}. \end{cases}$$

#### 4.4 Adaptive Step-Size Parameters

A drawback of the PHDG scheme is that the condition  $\tau\sigma\|D\|^2 \leq 1$  has to be met to ensure convergence. For complicated linear operators  $D$  the quantity  $\|D\|$  can be difficult to estimate tightly. Additionally, the convergence speed heavily depends on the particular choices of  $\tau$  and  $\sigma$ .

<sup>2</sup>Recall that the singular values of a real matrix  $B \in \mathbb{R}^{m \times n}$  are the square roots of the eigenvalues of the square symmetric matrix  $B^T B \in \mathbb{R}^{n \times n}$ .

Therefore, we incorporate the idea of Goldstein et al. in [20], in which the step-sizes are chosen adaptively leading to a faster convergence of the algorithm and making the computation of  $\|D\|$  unnecessary. For the sake of completeness, let us summarize the adaptive choice proposed in [20].

Using Moreau's identity (4), the optimality conditions of the algorithm given in (7) are

$$\begin{aligned} \frac{1}{\tau_n} (\mathbf{u}^n - \mathbf{u}^{n+1}) - D^\top \mathbf{q}^n &\in \partial G(\mathbf{u}^{n+1}), \\ \frac{1}{\sigma_n} (\mathbf{q}^n - \mathbf{q}^{n+1}) + D\bar{\mathbf{u}}^{n+1} &\in \partial F^*(\mathbf{q}^{n+1}). \end{aligned}$$

After some straightforward computations, one has

$$\begin{aligned} \frac{1}{\tau_n} (\mathbf{u}^n - \mathbf{u}^{n+1}) - D^\top (\mathbf{q}^n - \mathbf{q}^{n+1}) &\in \partial G(\mathbf{u}^{n+1}) + D^\top \mathbf{q}^{n+1}, \\ \frac{1}{\sigma_n} (\mathbf{q}^n - \mathbf{q}^{n+1}) - D (\mathbf{u}^n - \mathbf{u}^{n+1}) &\in \partial F^*(\mathbf{q}^{n+1}) - D\mathbf{u}^{n+1}. \end{aligned} \tag{13}$$

Considering that the two expressions on the lefthand side being equal to zero yields the optimality condition to the original problem (6), we define the following sequence of primal and dual residuals

$$\begin{aligned} \mathcal{R}_{n+1}^p &:= \frac{1}{\tau_n} (\mathbf{u}^n - \mathbf{u}^{n+1}) - D^\top (\mathbf{q}^n - \mathbf{q}^{n+1}), \\ \mathcal{R}_{n+1}^d &:= \frac{1}{\sigma_n} (\mathbf{q}^n - \mathbf{q}^{n+1}) - D (\mathbf{u}^n - \mathbf{u}^{n+1}). \end{aligned}$$

Accordingly, the PDHG scheme (7) converges if

$$\lim_{n \rightarrow \infty} (\|\mathcal{R}_{n+1}^p\|_1^2 + \|\mathcal{R}_{n+1}^d\|_1^2) = 0$$

and, thus, one would like to choose  $\tau_n$  and  $\sigma_n$  so that the residuals are as small as possible. Note that there is a trade-off between small primal residuals  $\mathcal{R}_{n+1}^p$  and small dual residuals  $\mathcal{R}_{n+1}^d$ . Indeed, a large value of  $\tau_n$  leads to a strong minimization in  $u$  but a slow maximization in  $w$ , resulting in a small primal residual at the cost of a large dual residual. On the contrary, a small value of  $\tau_n$  leads to a small dual residual at the cost of a large primal residual. By assuming that  $\mathcal{R}_{n+1}^p$  and  $\mathcal{R}_{n+1}^d$  increase and decrease monotonically with  $\tau_n$  and  $\sigma_n$ , Goldstein et al. [20] suggested to enforce the residual balancing principle, that is,

$$\|\mathcal{R}_{n+1}^p\|_1 \simeq \|\mathcal{R}_{n+1}^d\|_1.$$

Residual balancing methods work by adapting parameters after each iteration to approximately maintain this equality. In order to control the adaptivity level of the algorithm, a sequence of parameters  $\{\alpha_n\}$  is introduced, with  $\alpha_n \in (0, 1)$  for all  $n \geq 0$ . If the primal and dual residuals differ by a factor greater than  $\Delta$ , with  $\Delta > 1$ , then the step-size related to the larger residual is increased by a factor of  $(1 - \alpha_n)^{-1}$  and the step-size related to the smaller residual is decreased by a factor of  $(1 - \alpha_n)$ , (for image intensity values in  $[0, 255]$ ). If both residuals are comparable, then the step-sizes remain the same. Every time the step-size parameters are updated to fit the balancing principle, we define  $\alpha_{n+1} = \eta\alpha_n$ , for some  $\eta < 1$ , so that the adaptivity decreases during the iterations.

Finally, the convergence of the algorithm is guaranteed if the following ‘‘backtracking’’ condition is met at each iteration

$$b_{n+1} = \frac{2\tau_n\sigma_n \langle D(\mathbf{u}^{n+1} - \mathbf{u}^n), \mathbf{q}^{n+1} - \mathbf{q}^n \rangle}{\gamma\sigma_n\|\mathbf{u}^{n+1} - \mathbf{u}^n\|^2 + \gamma\tau_n\|\mathbf{q}^{n+1} - \mathbf{q}^n\|^2} \leq 1, \tag{14}$$

where  $\gamma \in (0, 1)$  is a constant. If the step-sizes are too large and, consequently, the above condition does not hold,  $\tau_n$  and  $\sigma_n$  are decreased by a factor  $\beta b_{n+1}$ , with  $\beta \in (0, 1)$  being a constant. It is also shown in [20] that condition (14) is only activated a finite number of times.

## 5 Implementation Details

For practical purposes, we assume that an image channel is given by a matrix of size  $N_w \times N_h$  rearranged from left to right and from top to bottom into a vector of size  $N$ . In this setting,  $u(x, y)$  shall denote the value  $u_{yN_w+x}$  in the image vector. Furthermore, we have that  $M = 2$ , since the horizontal and vertical spatial derivatives are used, and  $C = 3$ , since the red, green and blue channels are considered.

### 5.1 Discrete Gradient and Divergence Operators

Consider an RGB image as  $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \in \mathbb{R}^{N \times 3}$ , with  $\mathbf{u}_k = (u_{1,k}, \dots, u_{N,k})^\top \in \mathbb{R}^N$ . We define the discrete gradient operator  $D\mathbf{u} \in \mathbb{R}^{N \times 2 \times 3}$  via forward differences. Therefore, the horizontal derivatives are computed as

$$(D\mathbf{u})_{i,1,k} = \begin{cases} u_{i+1,k} - u_{i,k} & \text{if } i \not\equiv 0 \pmod{N_w}, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

and the vertical derivatives are given by

$$(D\mathbf{u})_{i,2,k} = \begin{cases} u_{i+N_w,k} - u_{i,k} & \text{if } i \not\equiv 0 \pmod{N_h}, \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

for each channel  $k \in \{1, 2, 3\}$  and at each pixel  $i \in \{1, \dots, N\}$ . According to the previous definitions, the discrete divergence  $\text{div } \mathbf{q} \in \mathbb{R}^{N \times 3}$  is explicitly given by

$$(\text{div } \mathbf{q})_{i,k} = \begin{cases} q_{i,1,k} & \text{if } i \equiv 1 \pmod{N_w}, \\ -q_{i-1,1,k} & \text{if } i \equiv 0 \pmod{N_w}, \\ q_{i,1,k} - q_{i-1,1,k} & \text{otherwise,} \end{cases} + \begin{cases} q_{i,2,k} & \text{if } i \equiv 1 \pmod{N_h}, \\ -q_{i-N_w,2,k} & \text{if } i \equiv 0 \pmod{N_h}, \\ q_{i,2,k} - q_{i-N_w,2,k} & \text{otherwise,} \end{cases}$$

where  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$  is the dual variable.

### 5.2 PDHG Algorithm

Algorithm 1 displays the pseudo-code of the proposed backtracking PDHG method. The adaptive procedure described in Section 4.4 requires several parameters, which are fixed to  $\Delta = 1.5$ ,  $\eta = 0.95$ ,  $\alpha_0 = 0.2$ ,  $\beta = 0.95$ , and  $\gamma = 0.75$ . The primal and dual step-sizes are initialized with  $\tau_0 = \sigma_0 = 0.5$ . As a stopping criterion we used a tolerance value  $\varepsilon$  for the average primal plus dual residuals per pixel (which are computed for the step-size adaptation anyways). In any case, we stop the algorithm after 500 iterations even if the tolerance was not reached. The pseudo-code for the step-size adaptation function `adaptSteps` is given as Algorithm 2. Note that its use is optional and Algorithm 1 works with fixed step-sizes as well. The implementation of the proximity operators `proxG` and `proxF` is detailed in the next subsection.

### 5.3 Implementation of Proximal Operators

The proximity operator of the data fidelity-term is given by (9) or Algorithm 3 respectively. On the other hand, the implementation of the proximal mappings related to the norms described in Section 4.3 are detailed in Algorithms 4–11. One simply replaces `proxF` in Algorithm 1 by the desired proximity operator. In Algorithms 12–13 we additionally show the numerical schemes we propose for computing the projection of a non-negative vector onto the unit ball of the  $\ell^1$  and  $\ell^{1,2}$  norms, respectively. We used an alternating directions method of multipliers to solve the  $\ell^{1,2}$  projection subproblem. For the sake of readability, we have moved all algorithmic descriptions from this subsection to Appendix A.

---

**Algorithm 1:** PDHG Scheme for Color TV Denoising Using Collaborative Norms

---

**input** : Noisy image  $\mathbf{f} \in \mathbb{R}^{N \times 3}$ , trade-off parameter  $\lambda > 0$ , tolerance  $\varepsilon > 0$ , proximal operator function  $\text{proxF}$ , step-size adaptation function  $\text{adaptSteps}$ , and parameters  $(\tau_0, \sigma_0, \alpha_0)$ .  
**output**: Denoised image  $\mathbf{u} \in \mathbb{R}^{N \times 3}$ .

$(\mathbf{u}^0, \bar{\mathbf{u}}^0, \mathbf{g}^0, \mathbf{q}^0, n) \leftarrow (\mathbf{0}, \mathbf{u}^0, \mathbf{0}, \mathbf{0}, 0)$  *Initializations*

**while** *not stopping criterion* **do**

$\mathbf{u}^{n+1} \leftarrow \text{proxG}(\mathbf{u}^n + \tau_n \text{div } \mathbf{q}^n, \mathbf{f}, \lambda, \tau_n)$  *Optimize for u*

$\bar{\mathbf{u}}^{n+1} \leftarrow \mathbf{u}^{n+1} + (\mathbf{u}^{n+1} - \mathbf{u}^n)$  *Over-relaxation on u*

$\mathbf{g}^{n+1} \leftarrow \text{proxF}\left(D\bar{\mathbf{u}}^{n+1} + \frac{1}{\sigma_n} \mathbf{q}^n, \sigma_n\right)$  *Optimize for g*

$\mathbf{q}^{n+1} \leftarrow \mathbf{q}^n + \sigma_n (D\bar{\mathbf{u}}^{n+1} - \mathbf{g}^{n+1})$  *Update dual variable q*

*Adapt step-sizes*

$(\mathbf{u}^{n+1}, \mathbf{g}^{n+1}, \mathbf{q}^{n+1}, \tau_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \leftarrow$   
 $\text{adaptSteps}(\mathbf{u}^n, \mathbf{u}^{n+1}, \mathbf{g}^n, \mathbf{g}^{n+1}, \mathbf{q}^n, \mathbf{q}^{n+1}, \tau_n, \sigma_n, \alpha_n) n \leftarrow n + 1$

---



---

**Algorithm 2:**  $\text{adaptSteps}(\mathbf{u}^n, \mathbf{u}^{n+1}, \mathbf{g}^n, \mathbf{g}^{n+1}, \mathbf{q}^n, \mathbf{q}^{n+1}, \tau_n, \sigma_n, \alpha_n)$  – Step-sizes Adaptation

---

**input** : Current iterates  $(\mathbf{u}^{n+1}, \mathbf{g}^{n+1}, \mathbf{q}^{n+1})$ , previous iterates  $(\mathbf{u}^n, \mathbf{g}^n, \mathbf{q}^n)$ , current step-sizes  $(\tau_n, \sigma_n)$ , and adaptation parameter  $\alpha_n$ . The parameters  $(\nu, \Delta, \alpha_0, \beta)$  are prefixed.  
**output**: Adapted iterates  $(\mathbf{u}^{n+1}, \mathbf{g}^{n+1}, \mathbf{q}^{n+1})$  and adapted parameters  $(\tau_{n+1}, \sigma_{n+1}, \alpha_{n+1})$ .

$\mathcal{R}_{n+1}^p \leftarrow \left\| \frac{1}{\tau_n} (\mathbf{u}^n - \mathbf{u}^{n+1}) + \text{div} (\mathbf{q}^n - \mathbf{q}^{n+1}) \right\|_1$  *Primal residual*

$\mathcal{R}_{n+1}^d \leftarrow \left\| \frac{1}{\sigma_n} (\mathbf{q}^n - \mathbf{q}^{n+1}) - D(\mathbf{u}^n - \mathbf{u}^{n+1}) \right\|_1$  *Dual residual*

$b_{n+1} \leftarrow \frac{2\tau_n \sigma_n \langle D(\mathbf{u}^{n+1} - \mathbf{u}^n), \mathbf{q}^{n+1} - \mathbf{q}^n \rangle}{\gamma \sigma_n \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_2^2 + \gamma \tau_n \|\mathbf{q}^{n+1} - \mathbf{q}^n\|_2^2}$  *Backtracking condition*

**if**  $b_{n+1} > 1$  **then** *Decrease step-sizes to fit balancing principle*

$(\tau_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \leftarrow \left( \beta \tau_n b_{n+1}, \frac{\beta \sigma_n}{b_{n+1}}, \alpha_0 \right)$

$(\mathbf{u}^{n+1}, \mathbf{g}^{n+1}, \mathbf{q}^{n+1}) \leftarrow (\mathbf{u}^n, \mathbf{g}^n, \mathbf{q}^n)$

**else if**  $\mathcal{R}_{n+1}^p > \Delta \mathcal{R}_{n+1}^d$  **then** *Increase primal and decrease dual step-sizes*

$(\tau_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \leftarrow \left( \frac{\tau_n}{1 - \alpha_n}, \sigma_n (1 - \alpha_n), \eta \alpha_n \right)$

**else if**  $\mathcal{R}_{n+1}^p < \frac{1}{\Delta} \mathcal{R}_{n+1}^d$  **then** *Decrease primal and increase dual step-sizes*

$(\tau_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \leftarrow \left( \tau_n (1 - \alpha_n), \frac{\sigma_n}{1 - \alpha_n}, \eta \alpha_n \right)$

**else** *Keep primal and dual step-sizes*

$(\tau_{n+1}, \sigma_{n+1}, \alpha_{n+1}) \leftarrow (\tau_n, \sigma_n, \alpha_n)$

---

## 6 Discussion on Alternative Minimization Methods

Although our numerical experiments, the online code, as well as the experiments in the companion SIAM paper [17] were done with the PDHG backtracking algorithm described in the previous subsection, there are several other minimization techniques which could be more suitable to a particular type of TV regularized problem. While the PDHG backtracking approach is extremely versatile as it obtains good runtimes on a wide variety of problems without changing anything but the proximity operators, a particular structure of the optimization problem such as the strongly convex collaborative TV denoising problem make accelerated schemes attractive.

To provide the reader with some intuition of possible gains of using minimization methods that exploit the strong convexity of the problem, we conduct a small comparison in MATLAB. Besides the PDHG scheme with fixed stepsizes and the PDHG backtracking scheme (PDHG-BT), we additionally compare to the following methods:

1. The fast iterative shrinkage thresholding algorithm FISTA [5], which is based on ideas of Nesterov in [28], on the dual TV denoising problem. Instead of the original stepsize rule proposed in [5], we used a variant which was for instance considered in [11] and led to faster convergence in our numerical experiments. The updates are conducted as follows

$$\begin{aligned}\mathbf{q}^{n+1} &= \arg \min_{\mathbf{q}} \frac{1}{2} \left\| \mathbf{q} - \left( \bar{\mathbf{q}}^n - \tau D \left( D^\top \bar{\mathbf{q}}^n - \frac{1}{\lambda} \mathbf{f} \right) \right) \right\|_2^2 + F^*(\mathbf{q}), \\ \bar{\mathbf{q}}^{n+1} &= \mathbf{q}^{n+1} + \frac{n-1}{n+2} (\mathbf{q}^{n+1} - \mathbf{q}^n).\end{aligned}$$

The projections corresponding to the proximity operators of  $F^*$  can easily be obtained from our description of the proximity operators of  $F$  using Moreau's identity (4).

2. An acceleration of the PDHG algorithm for strongly convex problems given in [12], which iterative adapts the step sizes and the amount of extrapolation. Since pseudocode for this variant is provided in [12, Algorithm 2], we do not restate the algorithm here.
3. The alternating directions method of multipliers (ADMM), see for instance [7] for an overview. Similar to the way (7) states the PDHG algorithm in primal form, one can derive the ADMM algorithm in primal-dual form as

$$\begin{aligned}\mathbf{u}^n &= \arg \min_{\mathbf{u}} G(\mathbf{u}) + \frac{\sigma}{2} \left\| D(\mathbf{u} - \mathbf{u}^{n-1}) + \frac{1}{\sigma} \bar{\mathbf{q}}^n \right\|_2^2, \\ \mathbf{q}^n &= \arg \min_{\mathbf{q}} \frac{1}{2} \left\| \mathbf{q} - \mathbf{q}^{n-1} + \sigma D \mathbf{u}^n \right\|_2^2 + \sigma F^*(\mathbf{q}), \\ \bar{\mathbf{q}}^{n+1} &= -2\mathbf{q}^n + \mathbf{q}^{n-1},\end{aligned}\tag{17}$$

where, in our case,  $G(\mathbf{u}) = \frac{\lambda}{2} \|\mathbf{u} - \mathbf{f}\|_2^2$  is the data fidelity term.

4. Motivated by the primal-dual form of the ADMM algorithm in (17), it seems natural to use exactly the same strongly convex acceleration scheme as in [12, Algorithm 2] not only for the

PDHG, but also for the ADMM method. The latter yields the algorithm

$$\begin{aligned} \mathbf{u}^n &= \arg \min_{\mathbf{u}} \left\{ G(\mathbf{u}) + \frac{\sigma_n}{2} \left\| D\mathbf{u} - D\mathbf{u}^{n-1} + \frac{1}{\sigma_n} \bar{\mathbf{q}}^n \right\|_2^2 \right\}, \\ \mathbf{q}^n &= \arg \min_{\mathbf{q}} \frac{1}{2} \left\| \mathbf{q} - \mathbf{q}^{n-1} + \sigma_n D\mathbf{u}^n \right\|_2^2 + \sigma_n F^*(\mathbf{q}), \\ \theta_n &= \frac{1}{\sqrt{1 + 2\gamma \frac{1}{\sigma_n}}}, \quad \sigma_{n+1} = \frac{\sigma_n}{\theta_n}, \\ \bar{\mathbf{q}}^{n+1} &= -\mathbf{q}^n - \theta_n (\mathbf{q}^n - \mathbf{q}^{n-1}). \end{aligned} \tag{18}$$

Although the idea for the acceleration scheme (18) is straight-forward for the ADMM algorithm written in its primal-dual form (17), we are not aware of a publication investigating algorithm (18) above. Upon submission of our manuscript, we were informed through private communication that the same algorithm will also appear in the overview paper by Chambolle and Pock [13], which is in preparation and will contain a more detailed analysis of the convergence properties. Acceleration techniques for linearized ADMM algorithms have recently been investigated in [30]. We will refer to algorithm (18) as ADMM-ACC.

Table 2 shows the results of our small runtime comparison. As we can see, the accelerated ADMM scheme requires the lowest number of iterations in all test scenarios. Depending on the difference in the number of iterations and the complexity of the proximity operator either the accelerated ADMM or the accelerated primal-dual scheme led to the fastest runtime. Additionally, we can see that the acceleration schemes clearly outperform the two primal-dual algorithms that do not exploit the strong convexity. Nevertheless, the backtracking approach leads to significantly fewer iterations than the standard PDHG scheme while being equally versatile. In summary, we used and described the PDHG backtracking approach detailed in this paper because a great variety of different imaging problems (for instance the ones we considered in [17]) can be solved without changing the algorithm or adapting the step sizes. Considering Table 2 the price we pay for this flexibility are suboptimal convergence speeds on problems such as TV denoising.

## 7 Image Denoising Results

In this section, we present a detailed performance comparison of several collaborative norms for color image denoising. For our experiments, we used 8-bit reference images and all results were saved in integer values relatives to the intensity range  $[0, 255]$ . Furthermore, we ran the algorithm until the error was below  $\varepsilon = 10^{-5}$  or the maximum number of 500 iterations was reached. In what follows, all norms are given by  $\ell^{p,q,r}(col, der, pix)$  or  $(S^p(col, der), \ell^q(pix))$ , except  $\ell^{2,\infty,1}(der, col, pix)$ .

### 7.1 Influence of the Balancing Parameter

As a first step, we want to determine the general behaviour of each collaborative TV regularization with respect to tuning the balancing parameter  $\lambda$ . For these tests, we artificially added zero-mean Gaussian noise of standard deviation 25 to the first noise-free color image from Figure 8.

Figure 4 shows the plots of the peak signal-to-noise ratio (PSNR)<sup>3</sup> each method achieves for certain values of  $\lambda$ . One observes that the peaks of the PSNR curves of the regularizations using the

<sup>3</sup>The *peak signal-to-noise ratio* (PSNR) is the ratio between the reference signal and the distortion signal in an image, given in decibels. The higher the PSNR, the closer the distorted image is to the original one. Its value is defined as  $\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right)$ , where MSE is the mean-squared error between the original and the distorted images.

CTV	$\lambda$		PDHG	PDHG-BT	PDHG-ACC	ADMM	FISTA	ADMM-ACC
$\ell^{1,1,1}$	3	iter.	3211	907	1080	209	1299	<b>183</b>
		time	17.37	13.21	5.84	4.19	7.86	<b>3.61</b>
	6	iter.	3215	764	559	196	685	<b>124</b>
		time	18.07	11.09	3.11	4.02	4.11	<b>2.45</b>
	12	iter.	2620	529	299	159	369	<b>85</b>
		time	14.23	7.65	<b>1.62</b>	3.20	2.25	1.67
$\ell^{2,2,1}$	3	iter.	1133	404	236	88	508	<b>75</b>
		time	7.26	6.09	<b>1.51</b>	1.89	3.65	1.62
	6	iter.	876	253	135	62	271	<b>49</b>
		time	5.63	3.86	<b>0.87</b>	1.30	2.00	1.03
	12	iter.	430	122	63	35	116	<b>28</b>
		time	2.76	1.86	<b>0.40</b>	0.74	0.84	0.59
$(S^1, \ell^1)$	3	iter.	3601	867	436	221	1094	<b>132</b>
		time	132.37	39.80	16.04	11.24	41.20	<b>6.73</b>
	6	iter.	2932	557	227	179	600	<b>89</b>
		time	107.82	25.47	8.36	9.11	22.55	<b>4.50</b>
	12	iter.	1695	307	109	104	289	<b>53</b>
		time	62.28	14.02	3.99	5.26	10.84	<b>2.66</b>

Table 2: Runtime comparison on the minimization of (8) using MATLAB. We ran the alternating directions method of multipliers (ADMM) algorithm for 5000 iterations and computed the energy of the final result. We used the latter as a baseline and determined after how many iterations (*iter*) and after what runtime in seconds (*time*) each of the algorithms got within 0.01% of the baseline energy. The comparison refers to solving the TV denoising problem with varying data fidelity parameters  $\lambda$  and varying types of CTV regularizations. The algorithms involved in the comparison are the primal-dual (PDHG) method from [12] with  $\sigma = 2$  and  $\tau = 1/18$ , the accelerated PDHG method from [12] for strongly convex problems (PDHG-ACC), the PDHG method with backtracking from [20] (PDHG-BT), the ADMM with a Lagrange parameter of 0.03, the fast gradient projection algorithm with heavy-ball scheme (FISTA), and the proposed accelerated ADMM scheme (ADMM-ACC).

$\ell^{\infty,1,1}$ ,  $(S^1, \ell^1)$ ,  $\ell^{2,1,1}$ , and  $\ell^{\infty,2,1}$  norms achieve the highest values. Interestingly, although the  $\ell^{1,1,1}$  norm shows one of the lowest performances in terms of the maximal PSNR, its corresponding curve seems to drop slower as one overestimates  $\lambda$ .

To illustrate how the choice of the balancing parameter influences visually the final result, Figure 5 exhibits the performance of the  $\ell^{\infty,1,1}$  norm for several values of  $\lambda$ . When  $\lambda$  is too small, noise is completely removed but the image becomes over-smoothed and, thus, significant information is lost – see the result with  $\lambda = 0.01$ . On the contrary, if  $\lambda$  is chosen too large it makes the influence of the regularization term insignificant and the method is not able to hide color artifacts efficiently – see the result with  $\lambda = 0.1$ . The optimal value of the balancing parameter in terms of PSNR, which in this case is  $\lambda = 0.04$ , compromises between removing noise and preserving signal content.

## 7.2 Runtime Comparison among CTV Methods

In this subsection, we compare the computational cost of all CTV methods described in the implementation details. For these tests, we added zero-mean Gaussian noise of standard deviation 30 to the twelfth image from Figure 6. The tolerance was fixed to  $5 \cdot 10^{-3}$  since we experimentally checked

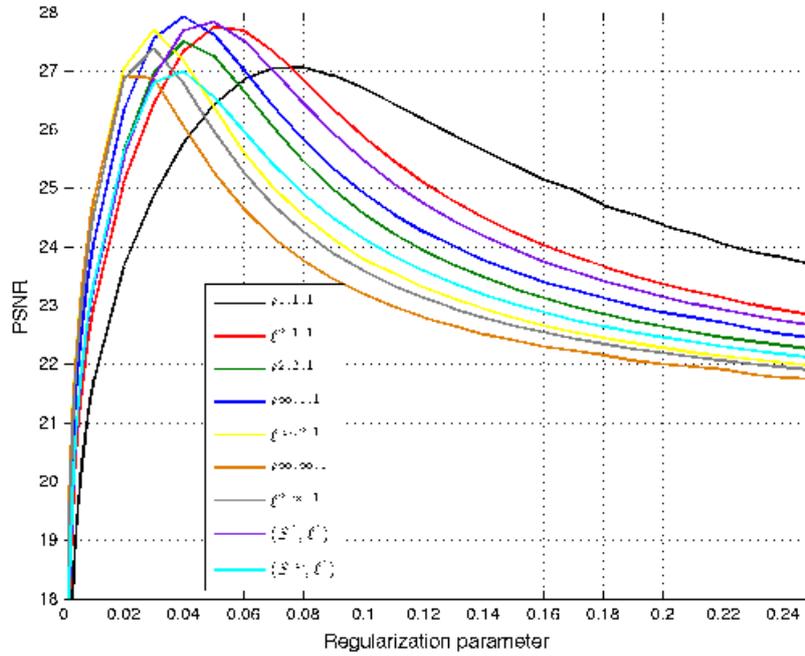


Figure 4: Comparison of CTV methods using different values of the trade-off parameter  $\lambda$ . The peaks of the curves of  $\ell^{\infty,1,1}$ ,  $(S^1, \ell^1)$ ,  $\ell^{2,1,1}$  and  $\ell^{\infty,2,1}$  norms achieve the highest values. Although  $\ell^{1,1,1}$  norm shows one of the lowest performances in terms of the maximal PSNR, its curve drop is slower as one overestimates  $\lambda$ .

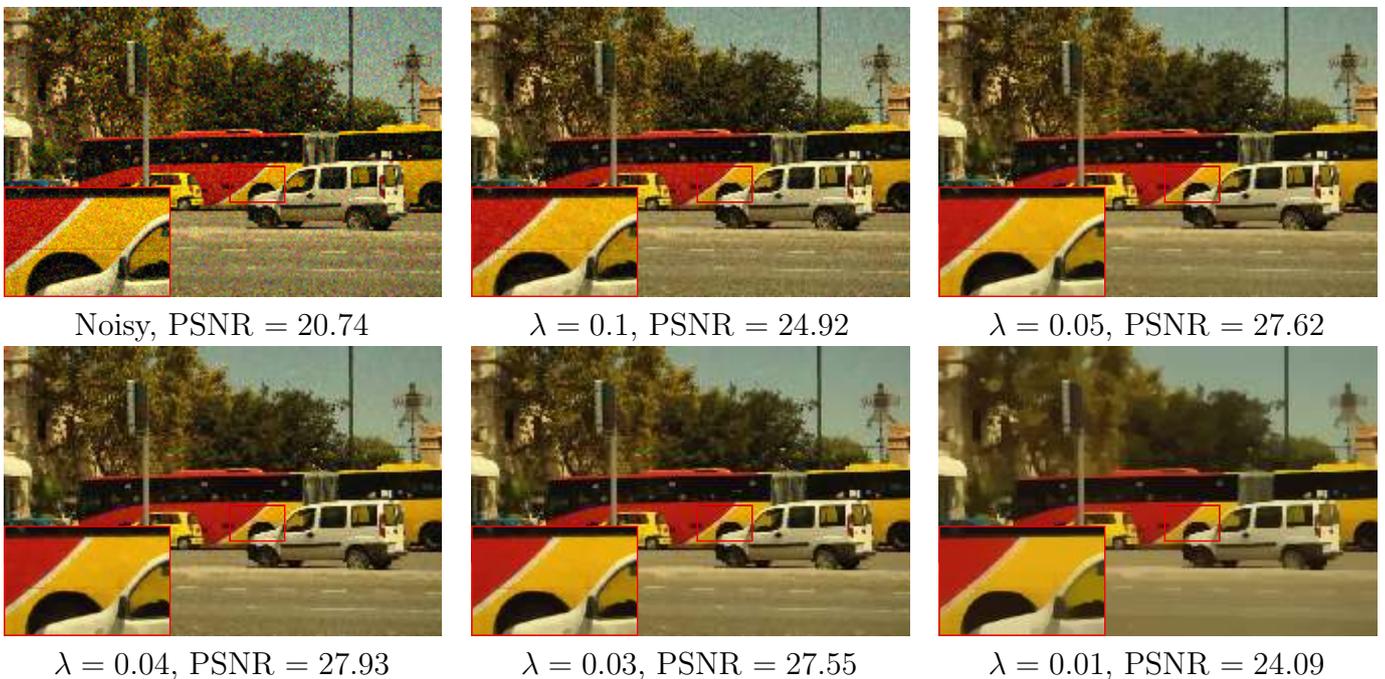


Figure 5: Results from  $\ell^{\infty,1,1}$  norm. For  $\lambda = 0.01$ , the noise disappears but at the cost of over-smoothing. If  $\lambda = 0.1$ , the method is not able to hide color artifacts. The optimal value of  $\lambda$  in terms of the PSNR ( $\lambda = 0.04$ ) gets a compromise between removing noise and preserving signal content.

CTV	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
CPU	8.939	8.221	7.685	15.931	244.017	19.785	13.749	11.344	10.947
PSNR	30.14	31.00	30.92	31.13	30.91	30.71	30.97	31.05	30.46
$\lambda$	0.048	0.034	0.026	0.025	0.019	0.015	0.018	0.031	0.024

Table 3: For a  $768 \times 512$  Kodak image (the twelfth from Figure 6), the running times in seconds of the CTV regularizations are displayed. All methods were run on a laptop with one core Processor 2.5 GHz Intel Core i5 with 4 GB 1600 MHz DDR3 RAM. Note that  $\ell^{2,2,1}$  is the most efficient regularization, whereas  $\ell^{\infty,2,1}$  involves high computational costs because of the projection onto the unit ball of the  $\ell^{1,2}$  norm. Although the  $\ell^1$  projection also requires an iterative process, the computational cost of the  $\ell^{\infty,1,1}$  regularization is not far from that of  $(S^1, \ell^1)$  but still providing better denoising results in terms of the PSNR.

that the best PSNR values were reached at this point in all cases. All methods were run with the optimal trade-off parameters on a laptop with one core Processor 2.5 GHz Intel Core i5 with 4 GB 1600 MHz DDR3 RAM.

The running times as well as the obtained PSNR values are displayed in Table 3. It is important to make clear that, for each method, the computational cost decreases as  $\lambda$  increases because the solution is required to be closer and closer to the initial data. We observe that  $\ell^{2,2,1}$ ,  $\ell^{2,1,1}$ , and  $\ell^{1,1,1}$  are the most efficient methods, although they do not provide the best PSNR values. On the other hand, we realize that the projection onto the unit ball of the  $\ell^{1,2}$  norm needed by the  $\ell^{\infty,2,1}$  regularization is computationally expensive, which does not happen in such considerable way with the projections onto the unit balls of the  $\ell^1$  and  $\ell^{1,1}$  norms involved in the methods using the  $\ell^{\infty,1,1}$  and  $\ell^{\infty,\infty,1}$  collaborative norms, respectively. The computational costs of the Schatten- $p$  norms are midway between the most efficient methods and those coupling the color channels with the help of the  $\ell^\infty$  norm.

### 7.3 Performance Comparison Using Several Image Databases

We now provide an extensive performance comparison of all CTV methods on different image collections with different inherent properties. In each table of this subsection we determined the value of the parameter  $\lambda$  at which the highest PSNR was reached on the first image and then used the same parameter on the others. To create the noisy images, we artificially added zero-mean Gaussian noise of standard deviation 15 to the reference noise-free images. On the other hand, we used noise of standard deviation 25 in all figures to make the differences among regularizations more apparent. For a better readability, all figures have been moved to Appendix B.

#### 7.3.1 Kodak Database

Images in the Kodak collection<sup>4</sup> have few color saturated regions, but are challenging due to fine scale details close to the Nyquist frequency. The mean saturation of these images is 15 and the mean value of the gradient of the chromatic components is 1.75, computed in the YUV color space. Figure 6 displays all Kodak images used in our tests.

Table 4 displays the PSNR values for each CTV regularization obtained on the Kodak image data set. As we can see, the  $\ell^{\infty,1,1}$  norm is superior for 10 out of the 12 examples. The large inter-channel correlation of Kodak images explains why a strong channel coupling, which encourages jumps that occur in all color coordinates, performs best. Along the same lines, the  $\ell^{\infty,2,1}$  norm behaves well, in fact, it gives the best numerical result on the fifth and twelfth images. Not only

<sup>4</sup><http://r0k.us/graphics/kodak>

Kodak	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
1	24.78	28.14	29.07	28.51	<b>29.90</b>	29.19	28.60	29.07	29.20	27.96
2	24.76	28.54	29.48	29.22	<b>30.18</b>	29.87	29.36	29.66	29.83	28.62
3	24.80	29.20	30.15	29.81	<b>30.85</b>	30.51	29.84	30.25	30.33	29.24
4	24.68	30.92	32.22	31.80	<b>32.73</b>	32.71	31.54	32.13	32.32	31.01
5	24.71	31.50	32.75	32.41	33.13	<b>33.30</b>	32.10	32.64	32.81	31.65
6	24.72	27.36	28.19	27.98	<b>29.01</b>	28.64	28.29	28.52	28.59	27.47
7	24.71	29.46	30.39	30.12	<b>30.86</b>	30.71	29.99	30.35	30.57	29.53
8	24.96	31.08	32.10	31.84	<b>32.41</b>	32.40	31.62	32.02	32.20	31.22
9	25.68	30.92	31.74	31.54	<b>32.10</b>	32.00	31.49	31.78	31.85	31.11
10	24.66	29.75	30.81	30.49	<b>31.48</b>	31.29	30.52	30.94	31.05	29.84
11	24.66	30.14	31.10	30.84	<b>31.49</b>	31.46	30.68	31.07	31.22	30.25
12	24.71	31.85	33.15	32.84	33.45	<b>33.69</b>	32.47	33.03	33.25	32.05
Avg.	24.82	29.91	30.93	30.62	<b>31.47</b>	31.31	30.54	30.96	31.10	30.00

Table 4: PSNR denoising results on Kodak dataset (Figure 6).

the  $\ell^{\infty,1,1}$  outperforming in general  $\ell^{\infty,2,1}$ , but also  $\ell^{2,1,1}$  beating  $\ell^{2,2,1}$  leads to the conclusion that anisotropy is better suited than isotropy for this dataset. Table 4 also reveals that  $\ell^{\infty,2,1}(col, der, pix)$  provides better results than  $\ell^{2,\infty,1}(der, col, pix)$ , which reinforces the importance of the order of the dimensions when penalizing the gradient of the image. Furthermore, the  $\ell^{1,1,1}$  norm shows one of the worst performances since it neither couples the colors nor the derivatives. In our experiments, the Schatten  $(S^\infty, \ell^1)$  norm does not work very well either. It seems that imposing jumps of different color channels to point into the same direction can more effectively be enforced by the convex relaxation  $(S^1, \ell^1)$  for minimizing the rank of the derivative matrix, than having a single direction in the dual variable as in the  $(S^\infty, \ell^1)$  approach.

In general, a low performance in the PSNR also entails a rejection by a human visual inspection. In spite of this, any numerical criterion cannot fully replace human evaluation, which still is an important criterion for judging the performance of image denoising algorithms. Figure 11 illustrates a comparative visual quality assessment of the optimal results obtained on parts of the first Kodak image. We see that most of tested TV regularizations lead to significant color artifacts. For instance, note that many spots remain around the graffiti or on the window frames. Only the method based on the  $\ell^{\infty,1,1}$  norm succeeds in hiding most of the remaining noise from the human observer. Although  $(S^1, \ell^1)$  shows nice denoising properties, a derivative matrix which has two vectors being equal to zero also has rank one and, as a consequence, colored edges are not actively suppressed.

### 7.3.2 IMAX Database

The IMAX database [24] has many more saturated colors and edges separating colored regions than Kodak. Contrary to the previous case, the mean saturation is 30 while the mean value of the gradient of the chromatic components is 6.21. Figure 7 shows all images from IMAX collection used here.

Table 5 shows that the order of the regularizations from best to worst performances in terms of the PSNR changes considerably with respect to the Kodak collection. Images in the IMAX dataset have lower channel-correlation and, thus, methods based on a strong channel coupling are not expected to be the best choice in this case. Because of the Euclidean norm being clearly weaker than the supremum norm,  $\ell^{2,1,1}$  and  $\ell^{2,2,1}$  outperform  $\ell^{\infty,1,1}$  and  $\ell^{\infty,2,1}$ , which gave the best results on Kodak images instead. One also realizes that the method based on the  $(S^1, \ell^1)$  norm is superior for all tested images. Therefore, imposing the gradients of different channels to point in the same direction benefits

IMAX	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
1	25.32	29.29	29.83	29.64	29.74	29.52	28.97	29.25	<b>29.98</b>	29.16
2	24.90	27.80	28.41	28.26	28.43	28.32	27.80	28.02	<b>28.60</b>	27.75
3	25.46	30.44	30.96	30.84	30.78	30.66	30.16	30.39	<b>31.17</b>	30.33
4	25.14	29.26	29.91	29.75	29.95	29.82	29.30	29.54	<b>30.13</b>	29.22
5	25.62	31.11	31.46	31.40	30.97	30.84	30.33	30.55	<b>31.64</b>	30.89
6	25.01	29.83	30.49	30.32	30.34	30.13	29.55	29.84	<b>30.74</b>	29.68
7	25.21	30.96	31.63	31.48	31.41	31.21	30.66	30.98	<b>31.80</b>	30.87
8	25.34	31.98	32.72	32.60	32.50	32.30	31.78	32.15	<b>32.88</b>	31.99
9	25.21	32.54	33.36	33.32	33.08	32.93	32.50	32.85	<b>33.53</b>	32.70
10	24.69	32.26	33.06	33.02	32.70	32.54	32.10	32.49	<b>33.20</b>	32.37
11	25.55	30.21	30.85	30.75	30.87	30.73	30.35	30.59	<b>30.98</b>	30.29
12	25.21	30.58	31.18	30.99	31.11	30.87	30.36	30.69	<b>31.30</b>	30.50
Avg.	25.22	30.52	31.16	31.03	30.99	30.82	30.32	30.61	<b>31.33</b>	30.48

Table 5: PSNR denoising results on IMAX dataset (Figure 7).

from images having many edges separating saturated colored regions. Although the performance of the  $\ell^{1,1,1}$  norm increases with respect to the Kodak set, the improvement is not enough to become competitive. Finally, the regularizations that make use of the  $\ell^{\infty,\infty,1}$  and  $(S^\infty, \ell^1)$  norms are not efficient on this collection either.

The optimal result each method gave on parts of the first IMAX image is shown in Figure 12. Note that all regularizations with an  $\ell^\infty$  channel coupling lead to color artifacts in smooth (but saturated) areas, such as the green flowers, and around image features. The same conclusions can be drawn from the result given by the  $(S^\infty, \ell^1)$  norm. In particular, the above mentioned methods fail to recover the underlying true image at edges separating saturated colored regions. On the other hand, methods based on the  $(S^1, \ell^1)$  and  $\ell^{2,1,1}$  norms provide the best visual performances since the contours are almost noise-free and look clear. Interestingly, the result of the  $\ell^{1,1,1}$  norm is relatively close to the two previous ones although the denoised image suffers from over-smoothing in some parts.

### 7.3.3 IPOL Database

In this subsection we compare the performance of CTV methods on a set of noise-free images (the real standard deviation of the inherent noise is less than one) introduced by Colom and Buades [14] for IPOL. All RGB pictures were taken with a Canon EOS 30D reflex camera of scenes under good lighting conditions and with a low ISO level. Figure 8 shows the 12 images used in our tests. Unlike the Kodak and IMAX collections, there are no common properties among these images that allow us to preset the requirements a regularization method should satisfy for a good performance on it.

The PSNR values of the denoised images are given in Table 6. As expected, there is no regularization being the best choice for all images, although  $(S^1, \ell^1)$  provides the best average followed closely by  $\ell^{\infty,1,1}$  and  $\ell^{\infty,2,1}$ . We observe that  $\ell^{\infty,2,1}$  is superior for images with main smooth regions, since a strong channel coupling with isotropic diffusion is well suited. The performance of all methods on images with outstanding vegetation is not satisfactory due to the general inability of distinguishing texture and noise. Finally,  $\ell^{1,1,1}$ ,  $(S^\infty, \ell^1)$  and  $\ell^{\infty,\infty,1}$  norms give the worst numerical results.

Figure 13 shows the optimal results for the fourth IPOL image. For a better comparative visual quality assessment, we have zoomed in on the lower left corner of the dice. First, we observe that the  $\ell^{\infty,\infty,1}$  and  $(S^\infty, \ell^1)$  norms do not perform well because, not only strong color artifacts remain

IPOLE	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
1	25.01	29.86	30.61	30.27	<b>30.81</b>	30.53	29.84	30.18	30.77	29.71
2	25.04	30.51	31.22	30.90	31.26	31.07	30.32	30.68	<b>31.34</b>	30.33
3	25.11	30.43	31.32	31.20	31.62	31.56	30.99	31.25	<b>31.68</b>	30.62
4	24.99	34.91	36.06	36.01	35.96	<b>36.61</b>	35.30	35.77	36.22	35.36
5	25.63	29.68	30.18	30.02	30.20	29.97	29.58	29.76	<b>30.42</b>	29.55
6	25.16	27.12	28.02	27.85	<b>28.54</b>	28.35	27.95	28.17	28.34	27.29
7	25.06	33.69	34.71	34.45	34.76	<b>34.90</b>	33.88	34.35	34.80	33.85
8	25.78	33.15	33.76	33.73	33.57	33.67	33.10	33.33	<b>33.90</b>	33.33
9	24.62	33.65	34.71	34.42	34.87	<b>35.07</b>	33.96	34.47	34.80	33.80
10	24.83	26.45	27.19	27.09	27.32	27.29	26.79	26.99	<b>27.37</b>	26.57
11	24.90	29.52	30.32	30.07	<b>30.72</b>	30.51	29.94	30.24	30.50	29.56
12	25.26	31.49	32.21	31.92	32.25	32.08	31.40	31.75	<b>32.31</b>	31.40
Avg.	25.12	30.87	31.69	31.49	31.82	31.80	31.09	31.41	<b>31.87</b>	30.95

Table 6: PSNR denoising results on IPOLE dataset (Figure 8).

inside the white dot, but also the edges have been clearly damaged during the restoration process. Furthermore,  $\ell^{1,1,1}$ ,  $\ell^{2,1,1}$  and  $\ell^{2,2,1}$  are not able to hide noise effectively. Although the remaining methods perform similarly, the underlying image being piecewise smooth favours the regularizations based on a  $\ell^2$  derivative coupling, such as  $\ell^{\infty,2,1}$  and  $\ell^{2,\infty,1}$ . Finally, the  $(S^1, \ell^1)$  norm provides one of the best visual results, particularly at image edges.

### 7.3.4 ARRI Database

The ARRI collection<sup>5</sup> [2] contains high-resolution color image sequences captured with a professional digital cinema camera in raw format. For our experiments, we extracted clipped pictures of  $750 \times 750$  pixels and transformed them into 8-bit images. Figure 9 displays the set of images we used.

The results displayed in Table 7 support the claim that the question which regularization is better for suppressing noise depends on the properties of the image being considered. In general, the performance of all methods on this collection is numerically superior to the three previous databases. Furthermore, the  $\ell^{1,1,1}$  norm providing the worst PSNR values indicates that images in Figure 9 are inter-channel correlated to varying degrees. In general, we see that  $(S^1, \ell^1)$ ,  $\ell^{\infty,1,1}$  and  $\ell^{\infty,2,1}$  lead to the best numerical results. It is interesting to remark that  $\ell^{\infty,1,1}$  and  $\ell^{\infty,2,1}$  are beaten by  $\ell^{2,2,1}$  on images taken under difficult lighting conditions like the number seven.

It is well known that a drawback of the total variation is the *staircase effect* [29], a tendency to produce flat regions separated by artificial edges. To a greater or lesser extent, this phenomenon is observed for all methods at the contour of the dice in Figure 13, but can also be seen in the warning signs in Figure 14, which shows the optimal denoising results for the first ARRI image. Note that while the staircase effect is least visible for  $(S^1, \ell^1)$  regularization, the best noise suppression results are obtained by the  $\ell^{\infty,1,1}$  norm. In particular, even the border of the exclamation mark shows little color artifacts.

### 7.3.5 BSDS Database

We finally present experiments on the BSDS database, a superset of  $481 \times 321$  RGB pictures of a wide variety of natural scenes introduced in [25] and further extended in [3] for image segmentation.

<sup>5</sup>[ftp://ftp.arri.de/](http://ftp.arri.de/)

ARRI	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
1	24.85	30.93	31.79	31.61	<b>31.94</b>	31.79	31.29	31.63	31.90	31.03
2	24.79	33.26	34.42	34.06	<b>34.55</b>	34.40	33.54	34.08	34.43	33.33
3	24.84	33.89	34.81	34.83	34.94	<b>35.23</b>	34.69	34.89	34.99	34.33
4	25.23	34.61	35.40	35.40	35.43	<b>35.59</b>	35.16	35.37	35.49	35.01
5	24.66	33.43	34.18	34.12	34.13	<b>34.24</b>	33.76	34.04	34.20	33.70
6	24.74	29.33	30.23	30.11	30.35	30.27	29.80	30.10	<b>30.46</b>	29.53
7	25.21	33.17	<b>33.81</b>	33.68	33.56	33.50	32.95	33.28	<b>33.81</b>	33.25
8	24.65	31.30	32.19	31.82	<b>32.36</b>	31.95	31.25	31.75	32.18	31.18
Avg.	24.87	32.49	33.35	33.20	33.41	33.37	32.81	33.14	<b>33.43</b>	32.67

Table 7: PSNR denoising results on ARRI dataset (Figure 9).

BSDS	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
1	24.88	29.70	30.56	30.41	<b>30.82</b>	30.72	30.17	30.46	30.80	29.85
2	25.02	30.01	30.98	30.52	<b>31.54</b>	31.03	30.44	30.87	31.12	29.95
3	25.04	30.26	31.03	30.86	<b>31.43</b>	31.26	30.78	31.04	31.24	30.41
4	24.96	32.59	33.73	33.66	33.99	<b>34.01</b>	33.36	33.72	34.00	32.93
5	24.72	30.16	30.88	30.75	<b>31.18</b>	31.07	30.62	30.87	31.01	30.32
6	25.03	29.24	30.19	29.77	<b>30.89</b>	30.36	29.84	30.22	30.37	29.27
7	24.65	29.12	30.11	29.74	<b>30.88</b>	30.44	29.81	30.22	30.32	29.15
8	24.71	30.57	31.62	31.51	<b>32.11</b>	32.09	31.44	31.75	31.92	30.82
9	24.70	31.05	31.94	31.75	<b>32.11</b>	32.01	31.32	31.69	32.04	31.20
10	25.42	31.19	31.93	31.87	31.90	31.86	31.34	31.57	<b>32.10</b>	31.37
11	24.72	28.06	29.06	28.92	<b>30.02</b>	29.69	29.48	29.60	29.64	28.36
12	24.64	30.82	31.86	31.58	<b>32.19</b>	31.97	31.20	31.67	32.03	30.89
Avg.	24.87	30.23	31.16	30.95	<b>31.59</b>	31.38	30.82	31.14	31.38	30.38

Table 8: PSNR denoising results on BSDS dataset (Figure 10).

	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Avg.	24.98	30.80	31.66	31.46	<b>31.86</b>	31.74	31.12	31.45	31.82	30.90

Table 9: Average of PSNR values over all databases.

Figure 10 shows the twelve BSDS images we tested.

The numerical results displayed in Table 8 show that all CTV regularizations perform similar to the Kodak database. In this regard, the  $\ell^{\infty,1,1}$  norm is significantly superior for 10 out of the 12 images, which indicates that the color channels of pictures in Figure 10 are highly correlated. On images 4 and 10, the  $\ell^{\infty,2,1}$  and  $(S^1, \ell^1)$  regularizations obtain the highest PSNR values, respectively. In the first case, the isotropic diffusion is favoured by prevailing smooth areas, whereas the nuclear norm has a better performance on images having color saturated regions.

In Figure 15, one can assess the visual quality of the optimal results obtained on parts of the third BSDS image. Note that the regularization making use of the  $\ell^{\infty,1,1}$  norm is the most successful in suppressing noise. In particular, it removes most of the colored edges between the tire and the wheel rim, which clearly remain in all others results to a greater or lesser extent.

### 7.3.6 Average over Databases

We finally display the PSNR average over all databases in Table 9. As expected, the regularizations using either the  $\ell^{\infty,1,1}$  norm or the  $(S^1, \ell^1)$  norm yield the best numerical results. Accordingly, we conclude that strong inter-channel coupling and the modeling assumption of the jumps in all channels being parallel give the best chances for denoising color image. If the type of images to be reconstructed is known, one can choose a regularization that is tailored to the particular type of color and derivative correlation.

## 7.4 Performance Comparison Using Several Noise Levels

The goal of this subsection is to evaluate the ability of each vectorial TV model for denoising color images corrupted with several noise levels. In order not to give advantage to a particular regularization due to the inherent properties of the data, the study was led on five natural images from different collections. More concretely, we used the last picture from each of the previous databases. We optimized the value of the balancing parameter for each test, so that one expects smaller values of  $\lambda$  as the noise level increases.

Table 10 shows the PSNR values of the denoised images provided by each regularization for several input noise standard deviations. By and large, we realize that the methods based on  $\ell^{\infty,1,1}$ ,  $(S^1, \ell^1)$ ,  $\ell^{2,1,1}$  and  $\ell^{\infty,2,1}$  norms outperform all other approaches. Interestingly, one observes that the superiority of a strong channel coupling becomes even more evident as the noise level increases. In this regard, note that the  $\ell^{\infty,1,1}$  norm gives the best numerical results for the IPOL, ARRI and BSDS collections, whereas the  $\ell^{\infty,2,1}$  norm does so for the Kodak database. However, the less inter-channel correlation and the more saturated colored regions of images in the IMAX collection make the  $\ell^\infty$  channel coupling unsuitable for denoising this type of data. In this case, the  $(S^1, \ell^1)$  norm or even the  $\ell^{2,1,1}$  norm provide better results. Furthermore, the nuclear norm performs best for almost all images in case the noise level is small.

Finally, Figure 16 in Appendix B shows the same results but considering noise with standard deviation 50. We observe that strong color artifacts appear on the parrot cheek in all results except for the  $\ell^{\infty,1,1}$  norm. In fact, this method is visually superior to the regularization based on the  $\ell^{\infty,2,1}$  norm which, interestingly, gave the best PSNR value.

## 8 Cartoon + Texture Decomposition Results

We present now examples of decomposing a color image  $\mathbf{f}$  into a cartoon  $\mathbf{u}$  part and a textural  $\mathbf{v}$  part,  $\mathbf{f} = \mathbf{u} + \mathbf{v}$ , by using different collaborative sparsity enforcing norms. For that purpose, the cartoon part is defined as the solution to the minimization problem (8). Then, the textural part is simply computed as the difference between the original data and its cartoon part. For visualization purposes,  $\mathbf{v}$  is linearly rescaled from  $[-20, 20]$  to  $[0, 255]$ . Differences outside this range are saturated to 0 and 255 respectively. The algorithm parameters and the stop criterion are the same as in the image denoising case. All figures from this section have been moved to Appendix B.

### 8.1 Influence of the Balancing Parameter

As described previously, the balancing parameter  $\lambda$  defines the contributions of the regularization and fidelity terms to the total energy given in (8). Since the cartoon part is computed as the minimizer of (8), the value of  $\lambda$  determines the minimal collaborative total variation which  $\mathbf{u}$  is allowed to have. In the variational model we are dealing with, the CTV of  $\mathbf{u}$  is an increasing function of  $\lambda$ . Consequently, one expects that less and less amount of textural details remain in the cartoon

Noise standard deviation 5										
Image	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Kodak	34.19	39.02	39.83	39.70	40.02	39.94	39.39	39.68	<b>40.09</b>	39.17
IMAX	34.38	36.92	37.33	37.11	37.22	37.03	36.49	36.74	<b>37.50</b>	36.68
IPOL	34.34	37.90	38.53	38.16	38.41	38.15	37.43	37.76	<b>38.69</b>	37.64
ARRI	34.14	37.41	38.14	37.71	38.01	37.66	36.94	37.34	<b>38.17</b>	37.13
BSDS	34.16	37.41	38.23	37.90	<b>38.60</b>	38.30	37.63	38.01	38.45	37.39
Avg.	34.24	37.73	38.41	38.12	38.45	38.22	37.58	37.91	<b>38.58</b>	37.60
Noise standard deviation 10										
Image	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Kodak	28.21	35.52	36.43	36.32	36.60	36.55	35.99	36.31	<b>36.68</b>	35.76
IMAX	28.58	32.87	33.42	33.20	33.39	33.19	32.60	32.91	<b>33.56</b>	32.71
IPOL	28.58	33.83	34.54	34.21	34.53	34.30	33.56	33.94	<b>34.67</b>	33.66
ARRI	28.15	33.42	<b>34.28</b>	33.86	34.27	33.93	33.16	33.66	<b>34.28</b>	33.21
BSDS	28.14	33.23	34.21	33.87	<b>34.56</b>	34.27	33.53	33.99	34.39	33.26
Avg.	28.33	33.77	34.58	34.29	34.67	34.45	33.77	34.16	<b>34.72</b>	33.72
Noise standard deviation 20										
Image	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Kodak	22.25	32.12	33.04	32.96	33.19	33.19	32.68	32.97	<b>33.26</b>	32.43
IMAX	22.82	29.23	29.80	29.62	29.78	29.62	29.11	29.41	<b>29.89</b>	29.20
IPOL	22.93	29.90	30.57	30.32	30.63	30.45	29.90	30.22	<b>30.65</b>	29.86
ARRI	22.18	29.87	30.77	30.44	<b>30.89</b>	30.57	29.96	30.44	30.75	29.83
BSDS	22.16	29.37	30.43	30.12	<b>30.77</b>	30.51	29.81	30.28	30.57	29.46
Avg.	22.47	30.10	30.92	30.69	<b>31.05</b>	30.87	30.29	30.66	31.02	30.16
Noise standard deviation 30										
Image	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Kodak	18.88	30.13	30.99	30.91	31.12	<b>31.16</b>	30.69	30.95	<b>31.16</b>	30.44
IMAX	19.49	27.09	27.60	27.46	27.58	27.45	27.03	27.30	<b>27.66</b>	27.11
IPOL	19.72	27.49	28.08	27.90	<b>28.16</b>	28.02	27.62	27.86	28.13	27.54
ARRI	18.73	27.93	28.83	28.57	<b>29.01</b>	28.67	28.22	28.64	28.81	28.03
BSDS	18.69	27.21	28.24	28.00	<b>28.54</b>	28.34	27.73	28.15	28.34	27.38
Avg.	19.10	27.97	28.75	28.57	<b>28.88</b>	28.73	28.26	28.58	28.82	28.10
Noise standard deviation 50										
Image	Noisy	$\ell^{1,1,1}$	$\ell^{2,1,1}$	$\ell^{2,2,1}$	$\ell^{\infty,1,1}$	$\ell^{\infty,2,1}$	$\ell^{\infty,\infty,1}$	$\ell^{2,\infty,1}$	$(S^1, \ell^1)$	$(S^\infty, \ell^1)$
Kodak	14.89	27.30	27.97	27.89	28.06	<b>28.11</b>	27.75	27.94	28.05	27.57
IMAX	15.40	23.97	24.37	24.26	24.34	24.26	23.97	24.16	<b>24.39</b>	24.03
IPOL	15.77	24.04	24.48	24.36	<b>24.55</b>	24.47	24.22	24.37	24.49	24.14
ARRI	14.59	25.48	26.26	26.10	<b>26.47</b>	26.19	25.90	26.19	26.26	25.71
BSDS	14.43	24.68	25.53	25.37	<b>25.77</b>	25.75	25.21	25.50	25.61	24.92
Avg.	15.02	25.09	25.72	25.60	<b>25.84</b>	25.76	25.41	25.63	25.76	25.27

Table 10: For different noise levels, PSNR results on the last image from each dataset.

component of the decomposition as the balancing parameter decreases. Due to the total variation being able to recover sharp discontinuities, the geometry of the input data will be mainly preserved in the cartoon part instead.

There is no unique decomposition of an image into texture and cartoon. A texture seen at close range is just a set of well-distinguished objects. For this reason, it can be kept in the cartoon part for large values of  $\lambda$ , and passes over to the textural part for lower values. The balancing parameter in the algorithm is therefore crucial, and must be chosen by the user. To illustrate how this choice influences visually the results, Figure 17 exhibits the performance of the  $\ell^{1,1,1}$  norm for several values of  $\lambda$ . Note that textural details, such as the tiles on the tablecloth, progressively transfer to the cartoon part as the balancing parameter decreases.

## 8.2 Performance Comparison among Collaborative Norms

We finally display a visual comparison of several CTV methods for cartoon and texture decomposition. Since the contribution of the regularization term to the global energy depends on the choice of the collaborative norm, we estimated the values of  $\lambda$  in such a way that the same BV norms for the grayscale cartoon parts, computed as  $P_i = \frac{1}{C} \sum_k u_{i,k}$ , were obtained

$$\|P\|_{\text{BV}} = \|DP\|_1 = \sum_{i=1}^N \sqrt{((DP)_{i,1})^2 + ((DP)_{i,2})^2}, \quad (19)$$

where  $(DP)_{i,1}$  and  $(DP)_{i,2}$  are computed analogously to (15)-(16). More concretely, we ran the algorithm by using the  $\ell^{1,1,1}$  norm for several  $\lambda$ 's and picked up the value for which the best visual result was obtained. In the remaining regularizations,  $\lambda$  is chosen such that the TV value of the grayscale cartoon parts according to (19) is the same for all regularizations.

Figures 18 and 19 display the cartoon and textural parts, respectively, of an image where almost everything is texture. As a consequence, the cartoon components in Figure 18 only contain smoothed versions of the original image, and all details move to the textural parts in Figure 19. In such cases, large CTV regularizations or, equivalently, small values of the balancing parameter  $\lambda$ , are needed to remove the whole texture. We further observe in Figure 18 that all methods with an  $\ell^\infty$  channel coupling provide cartoon components which turn greyish; see, for instance, the second arch to the left. This fact matches the theoretical analysis provided in [17], where we state that the supremum norm leads to the strongest inter-channel correlation and makes the most prior assumptions. On the other hand,  $\ell^{\infty,1,1}$  and  $\ell^{2,1,1}$  perform best near edges since the contours, such as the upper edge of the roof, in the cartoon parts are straighter and look clearer than in all other cases.

Figures 20 and 21 respectively show the cartoon and textural parts provided by each collaborative TV on a cartoon-like image. Because of the data being almost piecewise-smooth and without texture, a smaller regularization is needed to obtain visually convincing results and, thus, one can choose a large value of  $\lambda$ . Observe that there is no significant differences between the original image and the cartoon parts. Finally, it is worth emphasizing that all collaborative norms using an  $\ell^\infty$  channel coupling are better suited for removing noise from the cartoon components. As a consequence, the associated textural parts contain more oscillations due to noise.

## 9 Conclusions

In this paper, we provided an in-depth analysis of different vectorial total variation regularizations obtained by replacing the usual TV with a *collaborative norm* that penalizes the three-dimensional structure of the discrete gradient of the underlying image. We used pixels, derivatives and color

channels as the corresponding dimensions. We considered several  $\ell^{p,q,r}$  and Schatten  $(S^p, \ell^q)$  norms as penalizations and discussed in detail the numerical implementation of their proximity operators. For the computation of the solution of the minimization problems involving the proposed TV based models, the primal-dual hybrid gradient algorithm was used with adaptive step-size parameters for faster convergence. We also displayed a detailed performance comparison of CTV denoising methods by using images from different databases (thus, images with different inherent properties) and different amounts of noise. In our experiments, we exhibited the superiority of the regularizations based on the  $\ell^{\infty,1,1}$  and  $(S^1, \ell^1)$  norms for a strong suppression of color artifacts. Finally, the proposed CTV-L2 models are used for cartoon and texture decomposition of color images.

## 10 OnLine Demo

An online demo of Algorithm 1 is available at the IPOL web page of this article. The ANSI C source code (documented) used in this online demo is also available from the web page.

The user chooses one of the proposed collaborative norms to be used in the CTV-L2 model. In general, the inputs are a color image and a value of  $\lambda > 0$ . One can upload his/her own image or use any of the images available on the demo page. The algorithm can be run in two different ways:

- Upload a noise-free color image, add Gaussian noise and then denoise it. In this case, the noise standard deviation is also required for creating the noisy image from the uploaded one. The outputs are the ground-truth uploaded image, the noisy image, the denoised image, the difference image between the ground truth and the result, the difference image between the noisy image and the result, the RMSE and the PSNR values of both noisy and denoised images. In the above cases, the difference images are linearly rescaled from  $[-4\sigma, 4\sigma]$  to  $[0, 255]$ , where  $\sigma$  denotes the noise standard deviation.
- Apply directly one of the proposed CTV methods to a noisy or a noise-free color image. In the noisy case, the result is a denoised version of the input data, and the outputs are the noisy uploaded image, the denoised image, and the difference image between the noisy image and the result. In the noise-free case, the result is a cartoon + texture decomposition of the input data, and the outputs are the noise-free uploaded image, the cartoon part (that is, the solution of the minimization of the CTV-L2 energy) and the textural part (computed as the difference image between the uploaded data and the cartoon part). In the above cases, the difference images are linearly rescaled from  $[-20, 20]$  to  $[0, 255]$ .

Finally, let us emphasize that the algorithm tolerance is fixed to  $10^{-4}$  and the maximum number of iterations to 250 for the online demo.

## A Algorithmic Description of Proximity Operators

---

**Algorithm 3:**  $\text{prox}_G(\mathbf{u}, \mathbf{f}, \lambda, \tau)$  – Proximity Operator of Fidelity Term

---

**input** : Proximal argument  $\mathbf{u} \in \mathbb{R}^{N \times 3}$ , noisy vector  $\mathbf{f} \in \mathbb{R}^{N \times 3}$ , trade-off parameter  $\lambda > 0$ , step-size parameter  $\tau > 0$ .

**output:** Proximal vector  $\hat{\mathbf{u}} = \text{prox}_{\tau G}(\mathbf{u}) \in \mathbb{R}^{N \times 3}$ .

**for**  $i \leftarrow 1$  **to**  $N$  **do**

**for**  $k \leftarrow 1$  **to**  $3$  **do**

$\hat{u}_{i,k} \leftarrow \frac{u_{i,k} + \tau \lambda f_{i,k}}{1 + \lambda \tau}$

---

**Algorithm 4:**  $\text{proxL111}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{1,1,1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .  
**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{1,1,1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
  for  $j \leftarrow 1$  to 2 do
    for  $k \leftarrow 1$  to 3 do
       $\hat{q}_{i,j,k} \leftarrow \text{sign}(q_{i,j,k}) \cdot \max(|q_{i,j,k}| - \frac{1}{\sigma}, 0)$ 

```

---



---

**Algorithm 5:**  $\text{proxL211}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{2,1,1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .  
**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{2,1,1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
  for  $j \leftarrow 1$  to 2 do
     $\varrho \leftarrow \sqrt{q_{i,j,1}^2 + q_{i,j,2}^2 + q_{i,j,3}^2}$ 
    if  $\varrho > 10^{-8}$  then  $\mu \leftarrow \frac{1}{\varrho} \cdot \max(\varrho - \frac{1}{\sigma}, 0)$  else  $\mu \leftarrow 0$ 
    for  $k \leftarrow 1$  to 3 do
       $\hat{q}_{i,j,k} \leftarrow \mu \cdot q_{i,j,k}$ 

```

---



---

**Algorithm 6:**  $\text{proxL221}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{2,2,1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .  
**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{2,2,1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
   $\varrho \leftarrow \sqrt{q_{i,1,1}^2 + q_{i,1,2}^2 + q_{i,1,3}^2 + q_{i,2,1}^2 + q_{i,2,2}^2 + q_{i,2,3}^2}$ 
  if  $\varrho > 10^{-8}$  then  $\mu \leftarrow \frac{1}{\varrho} \cdot \max(\varrho - \frac{1}{\sigma}, 0)$  else  $\mu \leftarrow 0$ 
  for  $k \leftarrow 1$  to 3 do
    for  $j \leftarrow 1$  to 2 do
       $\hat{q}_{i,j,k} \leftarrow \mu \cdot q_{i,j,k}$ 

```

---



---

**Algorithm 7:**  $\text{proxLin11}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{\infty,1,1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .  
**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{\infty,1,1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
  for  $j \leftarrow 1$  to 2 do
     $\mathbf{v} \leftarrow (v_1, v_2, v_3)$ 
    for  $k \leftarrow 1$  to 3 do
       $v_k \leftarrow \sigma |q_{i,j,k}|$ 
     $\mathbf{v} \leftarrow \text{L1Projection}(\mathbf{v})$ 
    for  $k \leftarrow 1$  to 3 do
       $\hat{q}_{i,j,k} \leftarrow q_{i,j,k} - \frac{1}{\sigma} \cdot \text{sign}(q_{i,j,k}) \cdot v_k$ 

```

---

---

**Algorithm 8:**  $\text{proxLininf1}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{\infty, \infty, 1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .

**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{\infty, \infty, 1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
   $\mathbf{v} \leftarrow (v_1, v_2, v_3, v_4, v_5, v_6)$ 
  for  $k \leftarrow 1$  to 3 do
    for  $j \leftarrow 1$  to 2 do
       $v_{2k+j-2} \leftarrow \sigma |q_{i,j,k}|$ 
   $\mathbf{v} \leftarrow \text{L1Projection}(\mathbf{v})$ 
  for  $k \leftarrow 1$  to 3 do
    for  $j \leftarrow 1$  to 2 do
       $\hat{q}_{i,j,k} \leftarrow q_{i,j,k} - \frac{1}{\sigma} \cdot \text{sign}(q_{i,j,k}) \cdot v_{2k+j-2}$ 

```

---



---

**Algorithm 9:**  $\text{proxLin21}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{\infty, 2, 1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .

**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{\infty, 2, 1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
  for  $j \leftarrow 1$  to 2 do
     $\mathbf{v}^j \leftarrow (v_1^j, v_2^j, v_3^j)$ 
    for  $k \leftarrow 1$  to 3 do
       $v_k^j \leftarrow \sigma |q_{i,j,k}|$ 
   $(\mathbf{v}^1, \mathbf{v}^2) \leftarrow \text{L12Projection}(\mathbf{v}^1, \mathbf{v}^2)$ 
  for  $k \leftarrow 1$  to 3 do
    for  $j \leftarrow 1$  to 2 do
       $\hat{q}_{i,j,k} \leftarrow q_{i,j,k} - \frac{1}{\sigma} \cdot \text{sign}(q_{i,j,k}) \cdot v_k^j$ 

```

---



---

**Algorithm 10:**  $\text{proxL2inf1}(\mathbf{q}, \sigma)$  – Proximity Operator of  $\ell^{2, \infty, 1}(\text{col}, \text{der}, \text{pix})$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , step-size parameter  $\sigma > 0$ .

**output:** Proximal vector  $\hat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma} \|\cdot\|_{2, \infty, 1}}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

```

for  $i \leftarrow 1$  to  $N$  do
   $\varrho \leftarrow (\varrho_1, \varrho_2, \varrho_3)$ 
   $\mathbf{v} \leftarrow (v_1, v_2, v_3)$ 
  for  $k \leftarrow 1$  to 3 do
     $\varrho_k \leftarrow \sqrt{q_{i,1,k}^2 + q_{i,2,k}^2}$ 
     $v_k \leftarrow \sigma \cdot \varrho_k$ 
   $\mathbf{v} \leftarrow \text{L1Projection}(\mathbf{v})$ 
  for  $k \leftarrow 1$  to 3 do
    if  $\varrho_k > 10^{-8}$  then  $\mu \leftarrow \frac{1}{\varrho_k} \cdot \max(\varrho_k - \frac{1}{\sigma} v_k, 0)$  else  $\mu \leftarrow 0$ 
    for  $j \leftarrow 1$  to 2 do
       $\hat{q}_{i,j,k} \leftarrow \mu \cdot q_{i,j,k}$ 

```

---

---

**Algorithm 11:** proxSchattenL1 ( $p, \mathbf{q}, \sigma$ ) – Proximity Operator of  $(S^p(\text{col}, \text{der}), \ell^1(\text{pix}))$  Norm

---

**input** : Proximal argument  $\mathbf{q} \in \mathbb{R}^{N \times 2 \times 3}$ , index of Schatten norm  $p \in \{1, \infty\}$ , step-size parameter  $\sigma > 0$ .

**output:** Proximal vector  $\widehat{\mathbf{q}} = \text{prox}_{\frac{1}{\sigma}(S^p, \ell^1)}(\mathbf{q}) \in \mathbb{R}^{N \times 2 \times 3}$ .

**for**  $i \leftarrow 1$  **to**  $N$  **do**

$$B \leftarrow \begin{pmatrix} q_{i,1,1} & q_{i,2,1} \\ q_{i,1,2} & q_{i,2,2} \\ q_{i,1,3} & q_{i,2,3} \end{pmatrix}$$

$$M \leftarrow B^T B \in \mathbb{R}^{2 \times 2}$$

$$(T, Z) \leftarrow (M_{11} + M_{22}, M_{11}M_{22} - M_{12}M_{21})$$

$$\text{det} \leftarrow \sqrt{\max\left(\frac{T^2}{4} - Z, 0\right)}$$

$$(eig_1, eig_2) \leftarrow \left(\max\left(\frac{T}{2} + \text{det}, 0\right), \max\left(\frac{T}{2} - \text{det}, 0\right)\right)$$

$$(s_1, s_2) \leftarrow (\sqrt{eig_1}, \sqrt{eig_2})$$

**if**  $|M_{12}| > 10^{-8}$  **then**

$$(\gamma_0, \gamma_1, \gamma_2) \leftarrow (M_{12}, eig_1 - M_{22}, eig_2 - M_{22})$$

$$(\mu_1, \mu_2) \leftarrow \left(\sqrt{\gamma_0^2 + \gamma_1^2}, \sqrt{\gamma_0^2 + \gamma_2^2}\right)$$

$$V \leftarrow \begin{pmatrix} \frac{\gamma_1}{\mu_1} & \frac{\gamma_2}{\mu_2} \\ \frac{\mu_1}{\gamma_0} & \frac{\mu_2}{\gamma_0} \end{pmatrix}$$

**else**

**if**  $M_{11} > M_{22}$  **then**

$$V \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

**else**

$$V \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

**if**  $p = 1$  **then**

$$(\widehat{s}_1, \widehat{s}_2) \leftarrow \left(\max\left(s_1 - \frac{1}{\sigma}, 0\right), \max\left(s_2 - \frac{1}{\sigma}, 0\right)\right)$$

$(S^1(\text{col}, \text{der}), \ell^1(\text{pix}))$  norm

**else if**  $p = \infty$  **then**

$(S^\infty(\text{col}, \text{der}), \ell^1(\text{pix}))$  norm

$$\mathbf{v} \leftarrow (v_1, v_2)$$

$$(v_1, v_2) \leftarrow (\sigma \cdot s_1, \sigma \cdot s_2)$$

$$\mathbf{v} \leftarrow \text{L1Projection}(\mathbf{v})$$

$$(\widehat{s}_1, \widehat{s}_2) \leftarrow \left(s_1 - \frac{1}{\sigma} \cdot v_1, s_2 - \frac{1}{\sigma} \cdot v_2\right)$$

**if**  $s_1 > 10^{-8}$  **then**  $\widehat{s}_1 \leftarrow \frac{\widehat{s}_1}{s_1}$

**if**  $s_2 > 10^{-8}$  **then**  $\widehat{s}_2 \leftarrow \frac{\widehat{s}_2}{s_2}$

$$\widehat{\Sigma}_0^\dagger \leftarrow \begin{pmatrix} \widehat{s}_1 & 0 \\ 0 & \widehat{s}_2 \end{pmatrix}$$

$$\widehat{Z} \leftarrow B V \widehat{\Sigma}_0^\dagger V^T \in \mathbb{R}^{3 \times 2}$$

**for**  $k \leftarrow 1$  **to**  $3$  **do**

**for**  $j \leftarrow 1$  **to**  $2$  **do**

$$\widehat{q}_{i,j,k} \leftarrow \widehat{Z}_{k,j}^T$$

---

**Algorithm 12:** L1projection( $\mathbf{v}$ ) – Projection onto  $\ell^1$  Norm Ball

---

**input** : Input vector  $\mathbf{v} \in \mathbb{R}_{\geq 0}^m$ .  
**output**: Projected vector  $\mathbf{v} = \text{proj}_{\|\cdot\|_1 \leq 1}(\mathbf{v})$ .

$(s, n, \alpha) \leftarrow (\infty, 0, 0)$   
**while**  $s > 1$  **do**  
     $(s, n) \leftarrow (0, 0)$   
    **for**  $i \leftarrow 1$  **to**  $m$  **do**  
         $v_i \leftarrow \max(v_i - \alpha, 0)$   
         $s \leftarrow s + |v_i|$   
        **if**  $v_i \neq 0$  **then**  $n \leftarrow n + 1$   
    **if**  $n > 0$  **then**  $\alpha \leftarrow \frac{1}{n}(s - 1)$  **else**  $s \leftarrow 0$

---



---

**Algorithm 13:** L12projection( $\mathbf{v}^1, \mathbf{v}^2$ ) – Projection onto  $\ell^{1,2}$  Norm Ball

---

**input** : Input vector  $(\mathbf{v}^1, \mathbf{v}^2) \in \mathbb{R}_{\geq 0}^m \times \mathbb{R}_{\geq 0}^m$ .  
**output**: Projected vector  $(\mathbf{v}^1, \mathbf{v}^2) = \text{proj}_{\|\cdot\|_{1,2} \leq 1}(\mathbf{v}^1, \mathbf{v}^2)$ .

$\mathbf{b}^1, \mathbf{b}^2, \mathbf{p}^1, \mathbf{p}^2, \mathbf{z}^1, \mathbf{z}^2 \in \mathbb{R}^m$   
 $(\alpha^1, \alpha^2, \beta^1, \beta^2) \leftarrow (0, 0, 0, 0)$   
**for**  $i \leftarrow 1$  **to**  $m$  **do**  
     $(b_i^1, b_i^2) \leftarrow (0, 0)$   
     $(p_i^1, p_i^2) \leftarrow (v_i^1, v_i^2)$   
     $(\alpha_1, \alpha_2) \leftarrow (\alpha_1 + p_i^1, \alpha_2 + p_i^2)$   
**for**  $n \leftarrow 1$  **to** 100 **do**  
     $(\delta^1, \delta^2) \leftarrow (\alpha^1 + \beta^1, \alpha^2 + \beta^2)$   
     $\Delta \leftarrow \sqrt{(\delta^1)^2 + (\delta^2)^2}$   
    **if**  $\Delta > 1$  **then**  $(\delta^1, \delta^2) \leftarrow \left(\frac{\delta^1}{\Delta}, \frac{\delta^2}{\Delta}\right)$   
    **for**  $i \leftarrow 1$  **to**  $m$  **do**  
         $(z_i^1, z_i^2) \leftarrow (\max(v_i^1 + b_i^1, 0), \max(v_i^2 + b_i^2, 0))$   
        **for**  $i_1 \leftarrow 1$  **to**  $m$  **do**  
             $(\gamma^1, \gamma^2) \leftarrow (0, 0)$   
            **for**  $i_2 \leftarrow 1$  **to**  $m$  **do**  
                **if**  $i_1 = i_2$  **then**  $\alpha \leftarrow 0.4$  **else**  $\alpha \leftarrow -0.1$   
                 $(\gamma^1, \gamma^2) \leftarrow (\gamma^1 + \alpha(\sigma p_{i_2}^1 + \delta^1 - \beta^1 + z_{i_2}^1 - b_{i_2}^1), \gamma^2 + \alpha(\sigma p_{i_2}^2 + \delta^2 - \beta^2 + z_{i_2}^2 - b_{i_2}^2))$   
                 $(v_{i_1}^1, v_{i_1}^2) \leftarrow (\gamma^1, \gamma^2)$   
             $(\alpha^1, \alpha^2) \leftarrow (0, 0)$   
        **for**  $i \leftarrow 1$  **to**  $m$  **do**  
             $(\alpha^1, \alpha^2) \leftarrow (\alpha^1 + v_i^1, \alpha^2 + v_i^2)$   
     $(\beta^1, \beta^2) \leftarrow (\beta^1 + \alpha^1 - \delta^1, \beta^2 + \alpha^2 - \delta^2)$   
    **for**  $i \leftarrow 1$  **to**  $m$  **do**  
         $(b_i^1, b_i^2) \leftarrow (b_i^1 + v_i^1 - z_i^1, b_i^2 + v_i^2 - z_i^2)$

---

## B Figures



Figure 6: Kodak dataset.

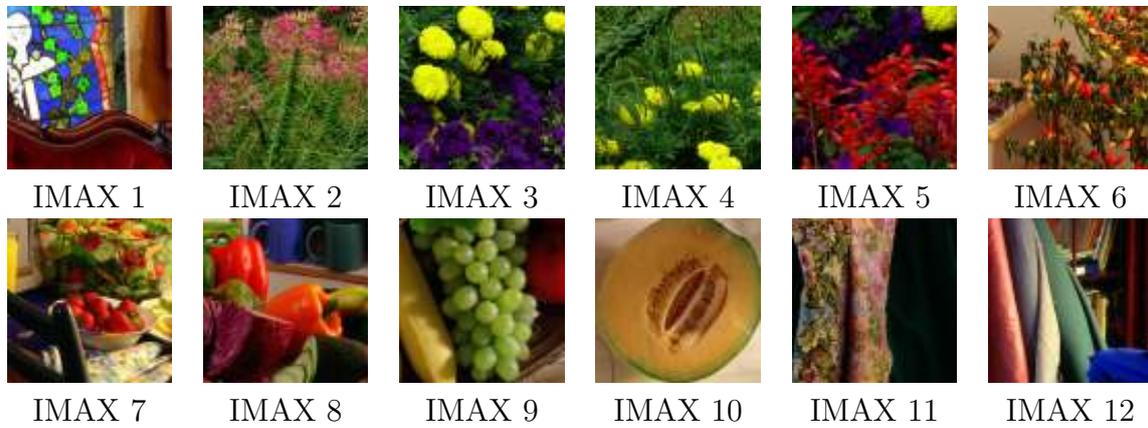


Figure 7: IMAX dataset.

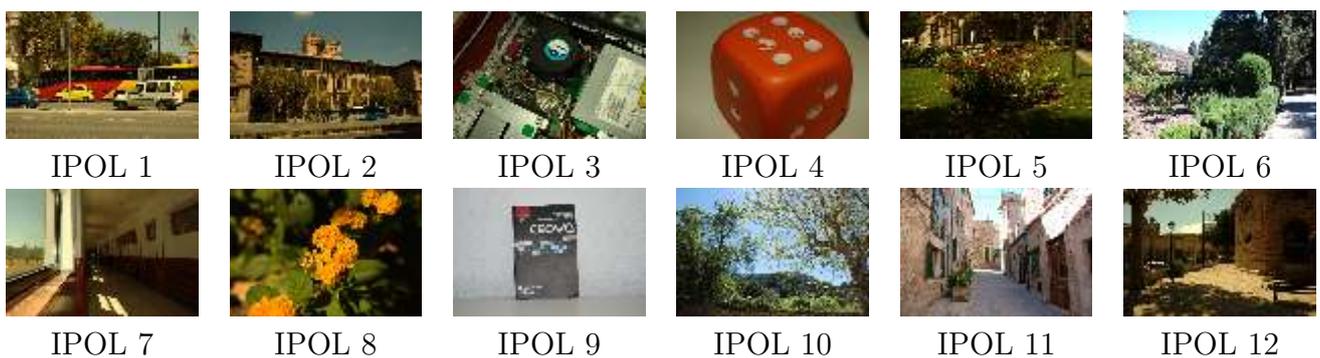


Figure 8: IPOL dataset.



Figure 9: ARRI dataset.



Figure 10: BSDS dataset.



Figure 11: Optimal results on the first Kodak image, which was corrupted with noise of s.d. 25. Note that almost all regularizations are not able to suppress color artifacts around the graffiti or on the window frames. Only the  $\ell^{\infty,1,1}$  norm succeeds in removing most of the noise.

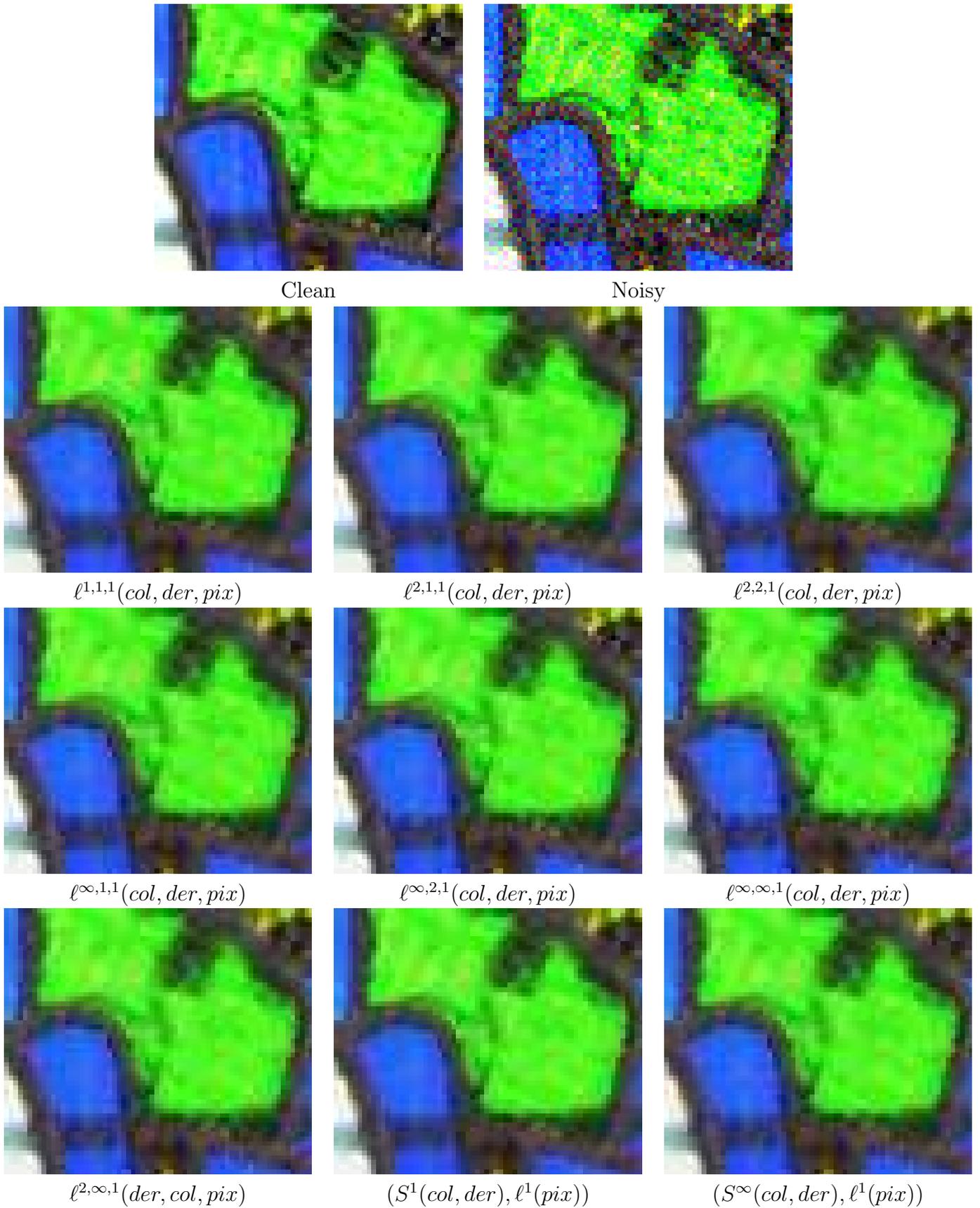


Figure 12: Optimal results on the first IMAX image, which was corrupted with noise of s.d. 25. All methods with an  $\ell^\infty$  channel coupling fail to remove noise not only at edges separating colored regions but also in smooth areas such as the green flowers. The  $(S^1, \ell^1)$  and  $\ell^{2,1,1}$  norms perform the best, closely followed by  $\ell^{1,1,1}$ .

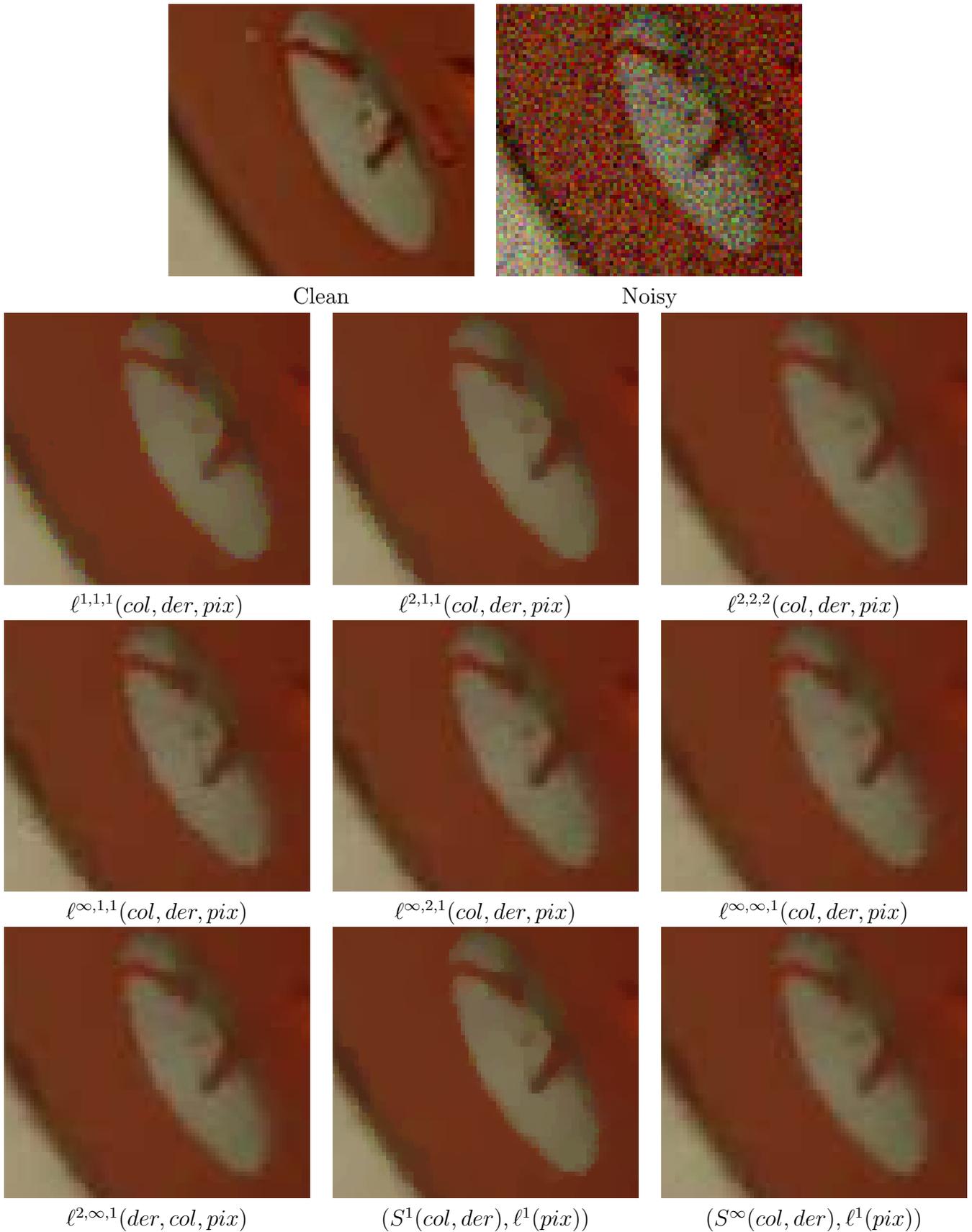


Figure 13: Optimal results on the fourth IPOL image, which was corrupted with noise of s.d. 25. Note that  $\ell^{\infty,\infty,1}$  and  $(S^\infty, \ell^1)$  norms perform the poorest since strong artifacts remain inside and around the white dot. Among the best performances, the  $\ell^{\infty,2,1}$  and  $\ell^{2,\infty,1}$  norms beat the  $\ell^{\infty,1,1}$  norm favoured by the underlying image being piecewise smooth. Finally,  $(S^1, \ell^1)$  is the best denoising method close to edges because of separating saturated colored regions.

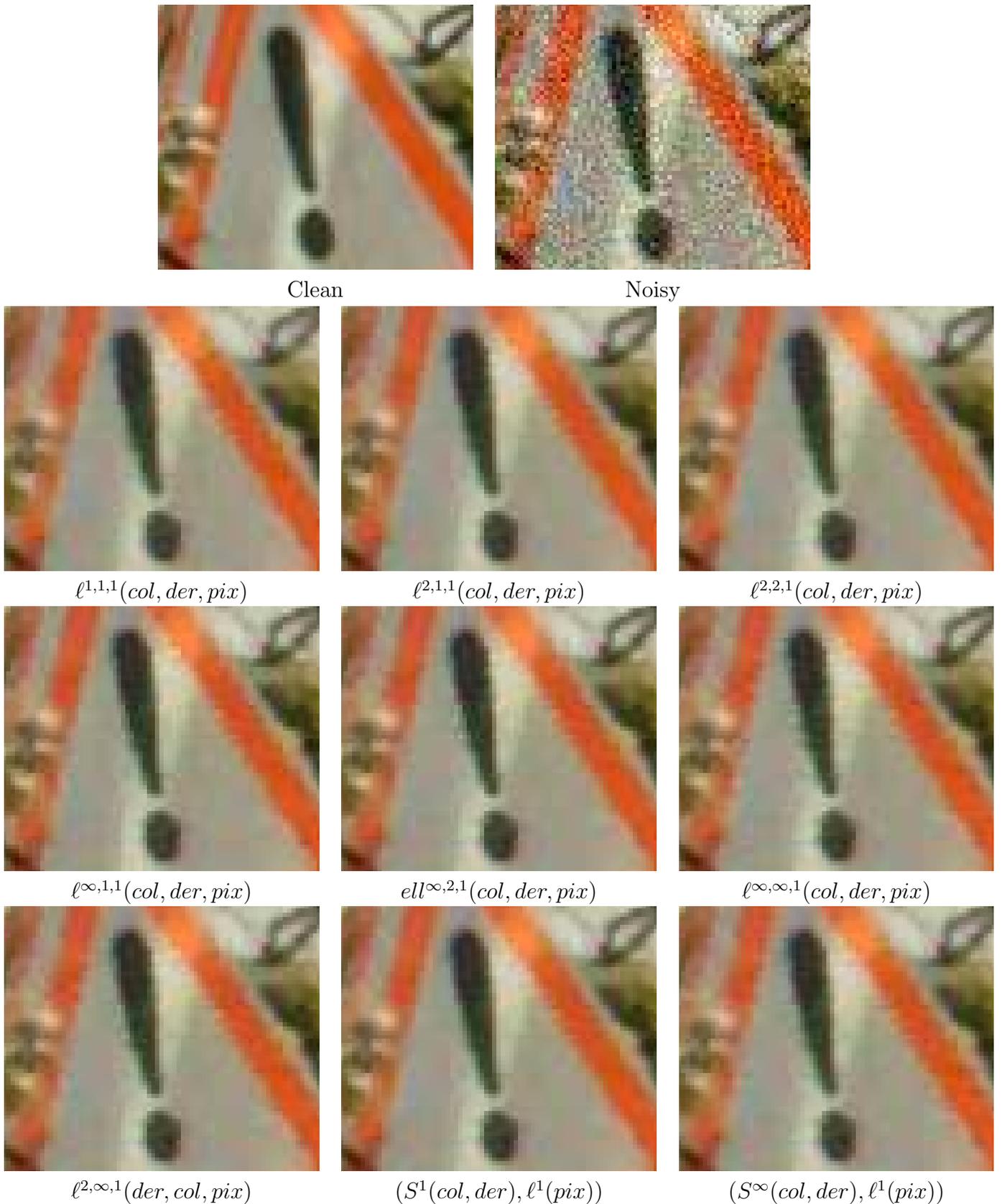
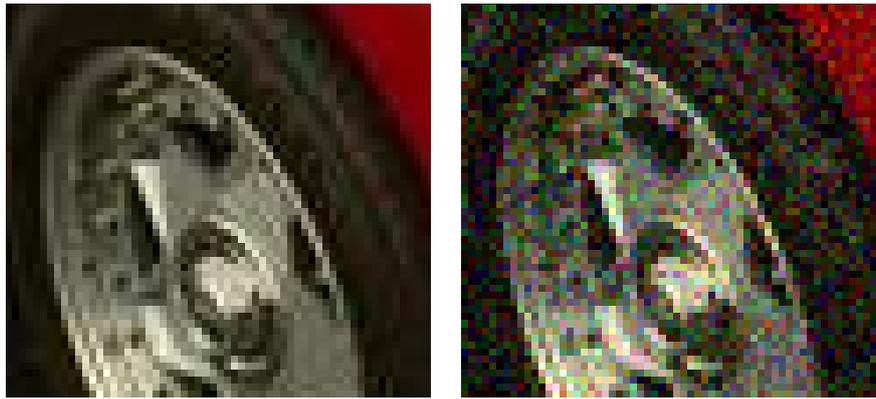


Figure 14: Optimal results on the first ARRI image, which was corrupted with noise of s.d. 25. The staircase effect appears around the warning sign in all denoised images, the  $(S^1, \ell^1)$  norm being the only one that gives a convincing result. Regarding the suppression of noise,  $\ell^{\infty,1,1}$  is the most successful in hiding color artifacts as one can see, for instance, bordering the exclamation mark.



Clean

Noisy



$\ell^{1,1,1}(col, der, pix)$



$\ell^{2,1,1}(col, der, pix)$



$\ell^{2,2,1}(col, der, pix)$



$\ell^{\infty,1,1}(col, der, pix)$



$\ell^{\infty,2,1}(col, der, pix)$



$\ell^{\infty,\infty,1}(col, der, pix)$



$\ell^{2,\infty,1}(der, col, pix)$



$(S^1(col, der), \ell^1(pix))$



$(S^{\infty}(col, der), \ell^1(pix))$

Figure 15: Optimal results on the third BSDS image, which was corrupted with noise of s.d. 25. Note that the  $\ell^{\infty,1,1}$  norm provides the most pleasing result since it actively suppresses color edges between the tire and the wheel rim.



Figure 16: Optimal results on the last Kodak image, which was corrupted with noise of s.d. 50. We observe that strong color artifacts remain on the parrot cheek in all results except for the  $\ell^{\infty,1,1}$  norm. In fact, this method is visually superior to the regularization based on the  $\ell^{\infty,2,1}$  norm which, on the contrary, gave the biggest PSNR value in Table 10.



Barbara image



Cartoon,  $\lambda = 0.075$



Texture,  $\lambda = 0.075$



Cartoon,  $\lambda = 0.05$



Texture,  $\lambda = 0.05$



Cartoon,  $\lambda = 0.025$



Texture,  $\lambda = 0.025$

Figure 17: Cartoon + texture decomposition obtained from the CTV-L2 model (8) with  $\ell^{1,1,1}$  norm. Note, for instance, that the tiles on the tablecloth progressively vanish in the cartoon part when  $\lambda$  decreases, that is, when there is more regularization.



Garden image



$\ell^{1,1,1}(col, der, pix)$



$\ell^{2,1,1}(col, der, pix)$



$\ell^{2,2,1}(col, der, pix)$



$\ell^{\infty,1,1}(col, der, pix)$



$\ell^{\infty,2,1}(col, der, pix)$



$\ell^{\infty,\infty,1}(col, der, pix)$



$\ell^{2,\infty,1}(der, col, pix)$



$(S^1(col, der), \ell^1(pix))$

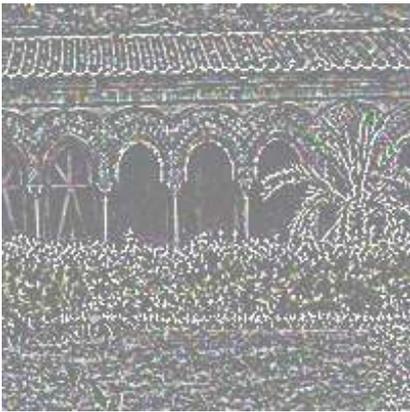


$(S^\infty(col, der), \ell^1(pix))$

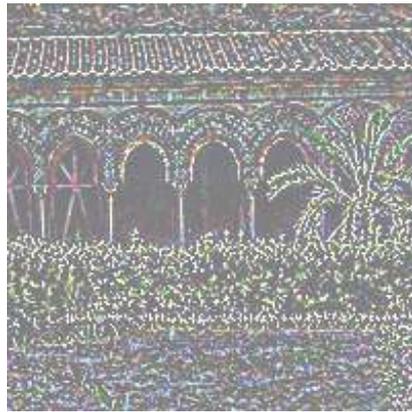
Figure 18: For each collaborative norm, cartoon components of the Garden image which almost consists of high frequencies due to texture. See the text for an analysis of these results.



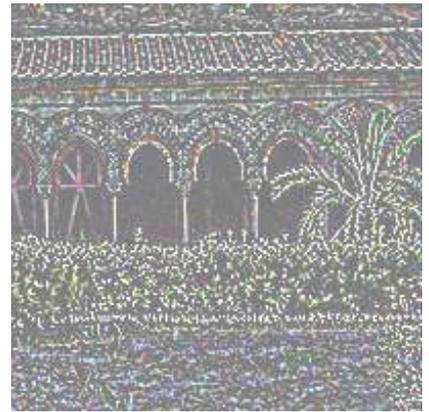
Garden image



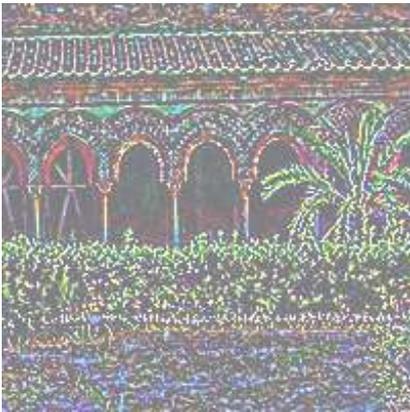
$\ell^{1,1,1}(col, der, pix)$



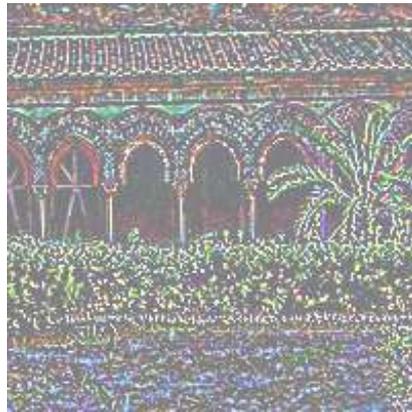
$\ell^{2,1,1}(col, der, pix)$



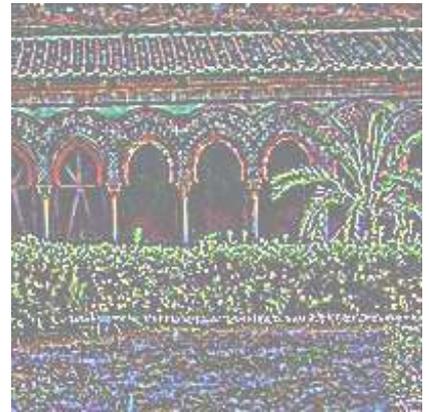
$\ell^{2,2,1}(col, der, pix)$



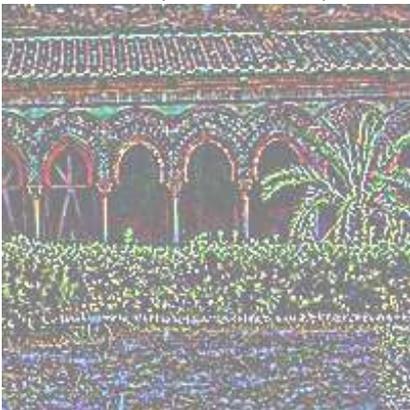
$\ell^{\infty,1,1}(col, der, pix)$



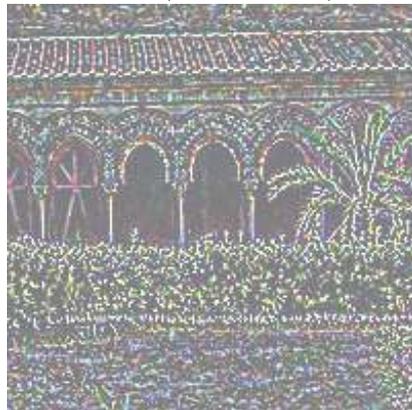
$\ell^{\infty,2,1}(col, der, pix)$



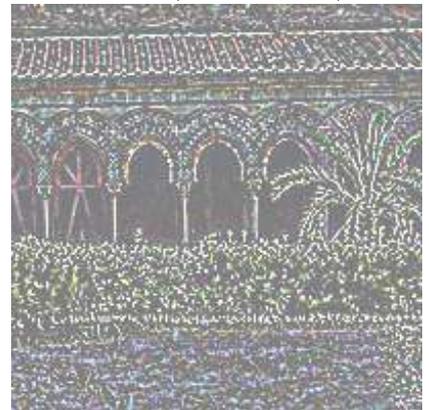
$\ell^{\infty,\infty,1}(col, der, pix)$



$\ell^{2,\infty,1}(der, col, pix)$



$(S^1(col, der), \ell^1(pix))$



$(S^\infty(col, der), \ell^1(pix))$

Figure 19: For each collaborative norm, textural components of the Garden image which almost consists of high frequencies due to texture. See the text for an analysis of these results.



Dolphin image



$\ell^{1,1,1}(col, der, pix)$



$\ell^{2,1,1}(col, der, pix)$



$\ell^{2,2,1}(col, der, pix)$



$\ell^{\infty,1,1}(col, der, pix)$



$\ell^{\infty,2,1}(col, der, pix)$



$\ell^{\infty,\infty,1}(col, der, pix)$



$\ell^{2,\infty,1}(der, col, pix)$



$(S^1(col, der), \ell^1(pix))$



$(S^\infty(col, der), \ell^1(pix))$

Figure 20: For each collaborative norm, cartoon components of the Dolphin image which is already cartoon-like. See the text for an analysis of these results.



Dolphin image



$\ell^{1,1,1}(col, der, pix)$



$\ell^{2,1,1}(col, der, pix)$



$\ell^{2,2,1}(col, der, pix)$



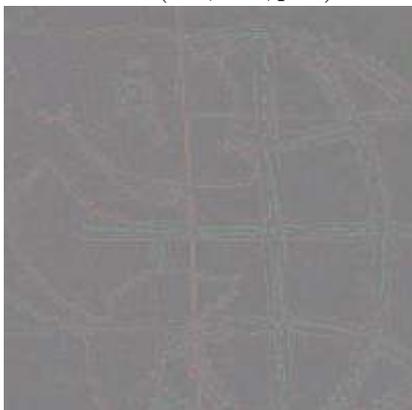
$\ell^{\infty,1,1}(col, der, pix)$



$\ell^{\infty,2,1}(col, der, pix)$



$\ell^{\infty,\infty,1}(col, der, pix)$



$\ell^{2,\infty,1}(der, col, pix)$



$(S^1(col, der), \ell^1(pix))$



$(S^\infty(col, der), \ell^1(pix))$

Figure 21: For each collaborative norm, textural components of the Dolphin image which is already cartoon-like. See the text for an analysis of these results.

## Acknowledgements

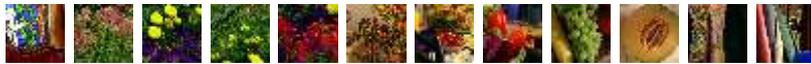
The authors would like to thank Prof. José Luis Lisani from University of Balearic Islands for his time devoted to the implementation of the online demo.

J. Duran and C. Sbert were supported by the Ministerio de Ciencia e Innovación under grant TIN2011-27539, and M. Moeller and D. Cremers were supported by ERC Starting Grant “Convex Vision”. During this work, J. Duran benefited from a fellowship of the Conselleria d’Educació, Cultura i Universitats of the Govern de les Illes Balears for the realization of his Ph.D. thesis, which was selected under an operational program co-financed by the European Social Fund.

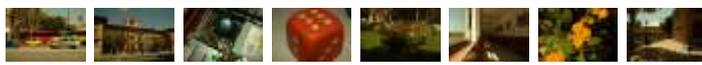
## Image Credits



by Kodak Image Suite.



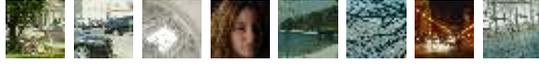
by L. Zhang [24].



by M. Colom, CC-BY-LA.



by A. Buades, CC-BY-LA.



by ARRI [2].



by Berkeley [3, 25].



Barbara standard test image.



J.-M. Morel, CC-BY-LA.

## References

- [1] L. AMBROSIO, N. FUSCO, AND D. PALLARA, *Functions of Bounded Variation and Free Discontinuity Problems*, Oxford Mathematical Monographs, Oxford University Press, 2000. ISBN: 9780198502456.
- [2] S. ANDRIANI, H. BRENDEL, T. SEYBOLD, AND J. GOLDSTONE, *Beyond the Kodak image set: A new reference set of color image sequences*, in IEEE International Conference on Image Processing (ICIP), 2013, pp. 2289–2293. <http://dx.doi.org/10.1109/ICIP.2013.6738472>.
- [3] P. ARBELAEZ, M. MAIRE, C. FOWLKES, AND J. MALIK, *Contour detection and hierarchical image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2011), pp. 898–916. <http://dx.doi.org/10.1109/TPAMI.2010.161>.
- [4] H. ATTOUCH, G. BUTTAZZO, AND G. MICHAILLE, *Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization*, vol. 6 of MPS/SIAM Series on Optimization, SIAM, Philadelphia, PA, 2006. ISBN: 0-89871-600-4.
- [5] A. BECK AND M. TEOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202. <http://dx.doi.org/10.1137/080716542>.

- [6] P. BLOMGREN AND T.F. CHAN, *Color TV: Total variation methods for restoration of vector-valued images*, IEEE Transactions on Image Processing, 7 (1998), pp. 304–309. <http://dx.doi.org/10.1109/83.661180>.
- [7] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Now Publishers, 3 (2011), pp. 1–122. <http://dx.doi.org/10.1561/22000000016>.
- [8] X. BRESSON AND T.F. CHAN, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems and Imaging, 2 (2008), pp. 255–284. <http://dx.doi.org/10.3934/ipi.2008.2.455>.
- [9] A. BUADES, T.M. LE, J.-M. MOREL, AND L.A. VESE, *Fast cartoon + texture image filters*, IEEE Transactions on Image Processing, 19 (2010), pp. 1978–1986. <http://dx.doi.org/10.1109/TIP.2010.2046605>.
- [10] M. BURGER, A. SAWATZKY, AND G. STEIDL, *First order algorithms in variational image processing*. Preprint arXiv:1412.4237, 2014. <http://arxiv.org/abs/1412.4237>.
- [11] A. CHAMBOLLE AND C. DOSSAL, *How to make sure the iterates of FISTA converge*. Preprint hal-01060130v1, 2014. <http://hal.inria.fr/hal-01060130v1>.
- [12] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145. <http://dx.doi.org/10.1007/s10851-010-0251-1>.
- [13] —, *An introduction to continuous optimisation for imaging*, Acta Numerica, 25, to appear (2016).
- [14] M. COLOM AND A. BUADES, *Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image*, Image Processing On Line, 3 (2013), pp. 173–197. <http://dx.doi.org/10.5201/ipol.2013.45>.
- [15] J. DUCHI, S. SHALEV-SCHWARTZ, Y. SINGER, AND T. CHANDRA, *Efficient projections onto the  $\ell^1$ -ball for learning in high dimensions*, in Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, 2008, ACM, pp. 272–279. <http://dx.doi.org/10.1145/1390156.1390191>.
- [16] J. DURAN, M. MOELLER, C. SBERT, AND D. CREMERS, *A novel framework for nonlocal vectorial total variation based on  $\ell^{p,q,r}$ -norms*, in Proceedings of the 10th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), vol. 8932 of Lecture Notes in Computer Science, Hong Kong, 2015, pp. 141–154. [http://dx.doi.org/10.1007/978-3-319-14612-6\\_11](http://dx.doi.org/10.1007/978-3-319-14612-6_11).
- [17] —, *Collaborative total variation: A general framework for vectorial TV models*, SIAM Journal on Imaging Sciences, 9 (2016), pp. 116–151. <http://dx.doi.org/10.1137/15M102873X>.
- [18] E. ESSER, X. ZHANG, AND T. CHAN, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 1015–1046. <http://dx.doi.org/10.1137/09076934X>.
- [19] B. GOLDLUECKE AND D. CREMERS, *An approach to vectorial total variation based on geometric measure theory*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 327–333. <http://dx.doi.org/10.1109/CVPR.2010.5540194>.

- [20] T. GOLDSTEIN, E. ESSER, AND R. BARANIUK, *Adaptive primal-dual hybrid gradient methods for saddle-point problems*. Preprint arXiv:1305.0546, 2013. <http://arxiv.org/abs/1305.0546>.
- [21] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms I, II*, Fundamental Principles of Mathematical Sciences, Springer-Verlag, Berlin, 1993. ISBN: 9780387568508 (I), 9783540568506 (II).
- [22] V.K. IVANOV, *Conditions for well-posedness in the Hadamard sense in spaces of generalized functions*, Siberian Mathematical Journal, 28 (1987), pp. 906–911. <http://dx.doi.org/10.1007/BF00969468>.
- [23] S. LEFKIMMIATIS, A. ROUSSOS, M. UNSER, AND P. MARAGOS, *Convex generalizations of total variation based on the structure tensor with applications to inverse problems*, in Scale Space and Variational Methods in Computer Vision, vol. 7893 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 48–60. [http://dx.doi.org/10.1007/978-3-642-38267-3\\_5](http://dx.doi.org/10.1007/978-3-642-38267-3_5).
- [24] X. LI, B. GUNTURK, AND L. ZHANG, *Image demosaicing: A systematic survey*, in Proceedings of SPIE, vol. 6822, 2008, p. 68221J. <http://dx.doi.org/10.1117/12.766768>.
- [25] D. MARTIN, C. FOWLKES, D. TAL, AND J. MALIK, *A database of human segmented natural images and its application to evaluation segmentation algorithms and measuring ecological statistics*, in 8th IEEE International Conference on Computer Vision (ICCV), vol. 2, 2001, pp. 416–423. <http://dx.doi.org/10.1109/ICCV.2001.937655>.
- [26] Y. MEYER, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: the fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, American Mathematical Society, Providence, RI, 2001. ISBN: 0821829203 9780821829202.
- [27] L. MIRSKY, *Symmetric gauge functions and unitarily invariant norms*, Quarterly Journal of Mathematics, 2 (1960), pp. 50–59. <http://dx.doi.org/10.1093/qmath/11.1.50>.
- [28] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate  $\mathcal{O}(1/k^2)$* , in Soviet Mathematics Doklady, vol. 27, 1983, pp. 372–376.
- [29] M. NIKOLOVA, *Local strong homogeneity of a regularized estimator*, SIAM Journal on Applied Mathematics, 61 (2000), pp. 633–658. <http://dx.doi.org/10.1137/S0036139997327794>.
- [30] Y. OUYANG, Y. CHEN, G. LAN, AND JR. E. PASILIAO, *An accelerated linearized alternating direction method of multipliers*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 644–681. <http://dx.doi.org/10.1137/14095697X>.
- [31] T. POCK, A. CHAMBOLLE, H. BISCHOF, AND D. CREMERS, *A convex relaxation approach for computing minimal partitions*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, 2009, pp. 810–817. <http://dx.doi.org/10.1109/CVPR.2009.5206604>.
- [32] R.T. ROCKAFELLAR, *Convex Analysis*, Princeton Landmarks in Mathematics and Physics, Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original. ISBN: 0-691-08069-0.
- [33] L.I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268. [http://dx.doi.org/10.1016/0167-2789\(92\)90242-F](http://dx.doi.org/10.1016/0167-2789(92)90242-F).

- [34] G. SAPIRO, *Vector-valued active contours*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, pp. 680–685. <http://dx.doi.org/10.1109/CVPR.1996.517146>.
- [35] G. SAPIRO AND D.L. RINGACH, *Anisotropic diffusion of multivalued images with applications to color filtering*, IEEE Transactions on Image Processing, 5 (1996), pp. 1582–1586. <http://dx.doi.org/10.1109/83.541429>.
- [36] L. VESE AND S. OSHER, *Modelling textures with total variation minimization and oscillating patterns in image processing*, Journal of Scientific Computing, 19 (2003), pp. 553–572. <http://dx.doi.org/10.1023/A:1025384832106>.
- [37] M. ZHU AND T. CHAN, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, Tech. Report 08-34, UCLA Cam Report, 2008.