

Convex Relaxation of Vectorial Problems with Coupled Regularization*

Evgeny Strelakovsky[†], Antonin Chambolle[‡], and Daniel Cremers[†]

Abstract. We propose convex relaxations for nonconvex energies on vector-valued functions which are tractable yet as tight as possible. In contrast to existing relaxations, we can handle the combination of nonconvex data terms with coupled regularizers such as l^2 -regularizers. The key idea is to consider a collection of hypersurfaces with a relaxation that takes into account the entire functional rather than separately treating the data term and the regularizers. We provide a theoretical analysis, detail the implementations for different functionals, present run time and memory requirements, and experimentally demonstrate that the coupled l^2 -regularizers give systematic improvements regarding denoising, inpainting, and optical flow estimation.

Key words. optimization, convex relaxation, multilabel, vectorial total variation

AMS subject classifications. 68U10, 49M29, 65K10

DOI. 10.1137/130908348

1. Introduction.

1.1. Vectorial problems. Functional optimization has become an established paradigm for solving a multitude of image analysis problems ranging from image denoising [33] and segmentation [13, 28, 10] to stereo reconstruction [44, 34], optical flow estimation [9], and three-dimensional reconstruction [14].

In the discrete setting with discrete domain and range of the solution functions, energy functionals can be formulated in the framework of Markov random fields (MRFs). Graph cut-based algorithms are frequently employed to compute the minimizers [7]. While submodular energies can be minimized optimally [23, 27], for general problems only approximate solutions can be found, using approaches such as α -expansions [8, 38], linear programming [42], or quadratic pseudo-Boolean optimization [26]. Our approach is based on a spatially continuous representation, which avoids the common drawbacks such as metrication errors and anisotropy of the grid-based approaches.

Nonconvex functionals pose a particular challenge since straightforward algorithms such as gradient descent will generally lead to undesired locally optimal solutions only. In recent years, researchers have made substantial progress regarding algorithms which allow one to compute optimal and near-optimal solutions for certain problem classes. One such class consists of general multilabel problems, where each point of the image domain is to be assigned

*Received by the editors February 1, 2013; accepted for publication (in revised form) November 18, 2013; published electronically February 11, 2014.

<http://www.siam.org/journals/siims/7-1/90834.html>

[†]Department of Computer Science, Technical University Munich, 85748 Garching, Germany (evgeny.strelakovsky@in.tum.de, cremers@tum.de).

[‡]CMAP, École Polytechnique, CNRS, 91128 Palaiseau, France (antonin.chambolle@polytechnique.fr).

a label in a certain optimal way. *Convex relaxation* approaches strive to find a convex lower bound as close as possible to the original functional. Relaxations of different tightness and computational efficiency have been proposed [10, 28, 44, 29, 45, 11]. However, they cannot be applied to vectorial problems with thousands of labels such as optical flow since they are very memory intensive.

Certain nonconvex functionals on scalar-valued functions with continuous range form another class of problems where optimal [35, 34] or near-optimal [1, 11] solutions can be computed. The key idea, sometimes referred to as *functional lifting*, is to consider the domain of values as an additional dimension and show that the optimal solution is a minimal hypersurface, i.e., a codimension-one structure in this higher-dimensional space. This is related to the construction [24] in the discrete setting. After range discretization this yields specific multilabel problems which can be solved using convex relaxation techniques.

Unfortunately, the functional lifting approach is limited to the estimation of scalar-valued functions. As a consequence, the convexification of functionals on vector-valued functions remains an important open problem. Several efforts have been made to generalize the approach to the case of vector-valued functions. For the case of convex data terms and specific convex regularizers (total variation with respect to different norms), Goldluecke, Strelakovski, and Cremers proposed an efficient solution [21]. For the case of possibly nonconvex but separable regularizers Strelakovski, Goldluecke, and Cremers [39] suggested a generalization which amounts to estimating a collection of hypersurfaces; see also the recent journal version [22]. To this end, they use a channelwise lifting strategy for the separable regularizer and an appropriate relaxation of the possibly nonconvex data term.

In this paper, we propose a novel convex relaxation for the estimation of vector-valued functions with nonconvex data terms and convex regularizers. Our work can be interpreted as a generalization of the work in [34] to the vectorial case. It generalizes the approaches of [39] and [21] in the sense that we can handle nonconvex data terms with *coupled* convex regularizers such as isotropic l^2 total variation TV_2 . The key idea is to consider a collection of hypersurfaces with a relaxation which takes into account the entire functional rather than separately treating data term and regularizers in each component as is done in [39].

1.2. Energy minimization. Let $\Omega \subset \mathbb{R}^d$ be a bounded open set with, in practice, $d = 2, 3$. We want to find an appropriate convex representation for functionals of the form

$$(1.1) \quad E(u) = \int_{\Omega} h(x, u, \nabla u) \, dx,$$

where $u \in W^{1,1}(\Omega; \mathbb{R}^k)$, $k \geq 2$. We will also consider the case of *BV*-fields u , with a suitable definition on the jump set.

For the scalar case $k = 1$ and provided that h has suitable convexity properties, such functionals can always be represented in a convex way by functional lifting. The idea is to reformulate the energy in terms of the graph function of u and to consider Cartesian currents associated to the graph [1, 18, 19, 20]. The important contribution of the works [25, 35, 34] is to properly introduce the interaction term $f(x, \nabla u)$ in this program and to suggest a practical implementation. The main advantage of the lifting approach is that it allows one to use nonconvex data terms and also (to some extent) nonconvex regularizers

within a convex optimization framework. On the other hand, it requires one to discretize the range set of u . In dimension $d = 0$, $k = 1$, it is like replacing the problem $\min_u g(u)$ with $\min_i \{g(u_i) \mid 1 \leq i \leq l\}$. It is therefore quite memory intensive and requires suitable hardware to be computationally tractable.

In the vectorial setting $k \geq 2$ things become more complex: in particular, memorywise it is unrealistic to sample the whole range space $[a, b]^k$ (assuming a bounded range). Recent contributions [39, 3] have proposed representing the solutions u by k separate graph functions, one for each channel u_i , to address problems such as optical flow. They then introduced an appropriate convexification of the zero-order data term $g(x, u)$ in this representation. This approach reduces the number of variables from N^k to kN , where N is the number of samples in each channel range. However, the regularizer part is handled in [39, 3] by a channelwise application of the lifting approach. This limits the approach to *separable* interaction terms, which regularize each coordinate u_i independently. Our contribution here is to try to consider more general interaction terms in this setting.

1.3. Contributions. We focus particularly on convex representations which

- are computationally tractable;
- are as tight as possible, that is, as close as possible to the convex envelope of the energy $E(u)$ (in an appropriate representation).

To stay computationally realistic, we therefore choose the representation by separate graph functions (and thus discretizations) for each channel as in [39, 3]. There are restrictions on the type of energies in order for the relaxation to still be exact, i.e., that it coincides with the initial energy for binary graph functions. In particular, we will focus on problems of the form

$$(1.2) \quad \min_u E(u) := \int_{\Omega} f(x, \nabla u(x)) dx + \int_{\Omega} g(x, u(x)) dx$$

for $u \in W^{1,1}(\Omega; \mathbb{R}^k)$, where $f(x, p) : \Omega \times \mathbb{R}^{d \times k} \rightarrow \mathbb{R}_+$ is continuous in x and convex in p (with possibly linear growth, in which case $u \in BV(\Omega; \mathbb{R}^k)$). For the definition of the Sobolev space $W^{1,1}(\Omega; \mathbb{R}^k)$ and the space $BV(\Omega; \mathbb{R}^k)$ of bounded variation functions we refer the reader to [2].

Specifically, we make the following contributions:

- We propose a general convexification strategy for vectorial problems of the form (1.2). The main novelty of our approach is the applicability to possibly nonseparable convex regularizers such as the isotropic TV_{l_2} . For separable regularizers it reduces to the relaxation of [39]. Since the functional is treated as a whole, this provides a natural derivation of the data term relaxation given in [39]. We prove the nontrivial fact that the proposed relaxation is exact.
- Our framework also allows nonconvex data terms $g(x, \cdot)$ in (1.2), which arise in many useful applications such as stereo reconstruction and optical flow. Furthermore, nonconvex data terms can be used together with nonseparable regularizers, which further improves the results, e.g., in denoising applications. This is not possible with previous approaches [21, 39].
- For the important special cases of isotropic total variation TV_{l_2} and its Huber-regularized variant we give a reformulation of the constraint set which allows one to minimize energies with these regularizers as efficiently as in the scalar case.

- We provide extensive implementation details, including memory analysis and notes on the graphics processing unit (GPU) usage. Several experiments demonstrate the advantage of coupled regularizers over the separable ones in applications such as optical flow, denoising, and inpainting. The case of nonconvex data terms with coupling regularizers, which is possible only with our approach, leads to superior results compared to previous approaches.
- We give an explicit formula for the projection onto a parabola, which is needed to implement Huber- TV_{l_2} . This is more robust than using Newton's iterative method as in [34] since the number of iterations depends on the data and increases with increasing regularizer weight λ . We also found the explicit projection to be faster by a factor of 2 or more.

2. Convex relaxation.

2.1. Convexification framework. Following the framework of [34, 22, 1], given $u \in L^1_{\text{loc}}(\Omega; \mathbb{R}^k)$, we consider the function $\mathbf{1}_u := (\mathbf{1}_{u_1}, \dots, \mathbf{1}_{u_k}) \in L^1_{\text{loc}}(\Omega \times \mathbb{R}, \{0, 1\}^k)$, where, for each $i = 1, \dots, k$,

$$(2.1) \quad \mathbf{1}_{u_i}(x, t) = \begin{cases} 1 & \text{if } u_i(x) > t, \\ 0 & \text{else.} \end{cases}$$

Then, we define a convex relaxation of (1.2) on the set $L^1_{\text{loc}}(\Omega \times \mathbb{R}; [0, 1]^k)$. As in [34, 22], it takes the form, if $v \in BV(\Omega \times \mathbb{R}; [0, 1]^k)$,

$$(2.2) \quad \mathcal{E}(v) = \sup \left\{ \sum_{i=1}^k \int_{\Omega \times \mathbb{R}} \phi_i \cdot Dv_i \mid \phi \in C^1(\Omega \times \mathbb{R}; \mathbb{R}^{d \times k}) \cap \mathcal{K} \right\}$$

for some convex set \mathcal{K} . We need this relaxation

- to be exact on characteristics of subgraphs, that is, we want $\mathcal{E}(\mathbf{1}_u) = E(u)$ for any u ;
- to be as “tight” as possible, that is, as close as possible to the convex envelope of the function $v \mapsto E(u)$ if $v = \mathbf{1}_u$ for some u , $+\infty$ else.

If, to simplify, $u \in W^{1,1}(\Omega; \mathbb{R}^k)$, then the terms in (2.2) can be written as [1]

$$(2.3) \quad \sum_{i=1}^k \int_{\Omega \times \mathbb{R}} \phi_i \cdot D\mathbf{1}_{u_i} = \sum_{i=1}^k \int_{\Omega} \phi_i^x(x, u_i(x)) \cdot \nabla u_i(x) - \phi_i^t(x, u_i(x)) dx.$$

As a consequence, a sufficient condition in order to have $\mathcal{E}(\mathbf{1}_u) \leq E(u)$ is that, for any $x \in \Omega$, $t = (t_1, \dots, t_k) \in \mathbb{R}^k$, and $p \in \mathbb{R}^{d \times k}$, the fields ϕ in \mathcal{K} satisfy

$$\sum_{i=1}^k \phi_i^x(x, t_i) \cdot p_i - \phi_i^t(x, t_i) \leq h(x, t, p)$$

or, equivalently,

$$(2.4) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq h^*(x, t, (\phi_i^x(x, t_i))_{i=1}^k),$$

where h^* is the Legendre–Fenchel conjugate of h with respect to the variable p . For $h(x, t, p) = f(x, p) + g(x, t)$ as in (1.2), it boils down to

$$(2.5) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq f^* \left(x, (\phi_i^x(x, t_i))_{i=1}^k \right) - g(x, t)$$

for all $x \in \Omega$ and $t \in \mathbb{R}^k$.

2.2. Concrete example: The vectorial ROF model. Before we continue, let us make this more explicit for a concrete example, namely the vectorial case of the classical Rudin–Osher–Fatemi total variation denoising problem (i.e., the ROF model) [37]. For $u \in C^1(\Omega; [0, 1]^k)$, it is given by

$$(2.6) \quad E_{ROF}(u) = \int_{\Omega} (u - f)^2 + \lambda |\nabla u| dx.$$

We want to rewrite this energy as the supremum of the expression (2.3) with the dual variable ϕ constrained to some appropriately chosen convex set \mathcal{K} . This set should be as large as possible in order for the convex relaxation to be as tight as possible. Yet, how should one choose this set? A necessary condition is that (2.3) should be less than or equal to the ROF-energy (2.6) for all $\phi \in \mathcal{K}$ and all u . This amounts to the inequality

$$(2.7) \quad \int_{\Omega} \sum_{i=1}^k \phi_i^x(x, u_i(x)) \nabla u_i - \lambda |\nabla u| - \sum_{i=1}^k \phi_i^t(x, \nabla u_i) dx \leq \int_{\Omega} (u - f(x))^2 dx \quad \forall \phi \in \mathcal{K}, \forall u.$$

A sufficient condition is the *local* version of this constraint,

$$(2.8) \quad \sum_{i=1}^k \phi_i^x(x, t_i) p_i - \lambda |p| - \sum_{i=1}^k \phi_i^t(x, p_i) \leq (t - f(x))^2 \quad \forall \phi \in \mathcal{K}, \forall t, x, p.$$

Taking the supremum over p , we observe that the first two terms on the left-hand side are equal to

$$(2.9) \quad \sup_p \sum_{i=1}^k \phi_i^x(x, t_i) p_i - \lambda |p| = \begin{cases} 0 & \text{if } \sqrt{\sum_{i=1}^k (\phi_i^x(x, t_i))^2} \leq \lambda, \\ \infty & \text{else.} \end{cases}$$

Therefore, the localized version of constraint (2.7) is equivalent to the two constraints

$$(2.10) \quad \begin{aligned} \sum_{i=1}^k \phi_i^t(x, t_i) &\geq -(t - f(x))^2 && \forall \phi \in \mathcal{K}, \forall t, x, \\ \sqrt{\sum_{i=1}^k (\phi_i^x(x, t_i))^2} &\leq \lambda && \forall \phi \in \mathcal{K}, \forall t, x. \end{aligned}$$

2.3. The scalar case. In the *scalar case* and for “reasonable” Lagrangians h in (1.1) (in particular, continuous and convex in p with at least linear growth), one can check (see [34]) that condition (2.5) allows one to recover tightly the initial energy, in the sense that if \mathcal{K} is the set of smooth vector fields ϕ satisfying (2.4), then minimizing the energy \mathcal{E} defined by (2.2) always solves the original problem. In the simplified case (1.2), one can show that the energy \mathcal{E} has the form

$$\mathcal{E}(v) = \int_{\Omega \times \mathbb{R}} \hat{g}(Dv) - h(x, t) D_t v$$

with $\hat{g}(p^x, p^t) : \mathbb{R}^d \times \mathbb{R} \rightarrow (-\infty, +\infty]$ the convex, one-homogeneous integrand defined by

$$\hat{g}(p^x, p^t) = \begin{cases} |p_t|g(p^x/|p^t|) & \text{if } p^t < 0, \\ g_\infty(p^x) & \text{if } p^t = 0, \\ +\infty & \text{if } p^t > 0. \end{cases}$$

In this notation, $g_\infty(p) := \lim_{t \rightarrow +\infty} g(tp)/t$ is the convex, one-homogeneous “recession function” of g at ∞ (in particular $g_\infty(0) = 0$, and it is $+\infty$ for $p \neq 0$ if g has superlinear growth). After an integration by parts (which is formal here and requires both that h be smooth in t , and, for instance, the domain be replaced with a bounded domain to be rigorously justified),

$$\mathcal{E}(v) \approx \int_{\Omega \times \mathbb{R}} \hat{g}(Dv) + \int_{\Omega \times \mathbb{R}} v(x, t) \partial_t h(x, t) \, dx dt.$$

This energy is always convex, independently of h . To give an example, if $g(p) = |p|$ and $h(x, t) = (t - u_0(x))^2/2$ (in which case (1.2) is the classical ROF denoising problem), then $\hat{g}(p^x, p^t) = |p^x|$ whenever $p^t \leq 0$ ($+\infty$ else), and one has

$$\mathcal{E}(v) \approx \int_{\Omega \times \mathbb{R}} |D_x v| + \int_{\Omega \times \mathbb{R}} v(x, t)(t - u_0(x)) \, dx dt$$

(whenever $v(x, t)$ is nonincreasing in t , and $+\infty$ else). In the case of total variation regularization, it boils down to the approach first proposed in [35] to solve continuous (nonconvex) multilabel problems.

2.4. The general vectorial case. In the vectorial case, though, things are not so simple, and as we already observed, one cannot hope to recover in general an equivalent convex formulation, which remains tractable computationally. Let us now fix arbitrary (finite) bounds a and $b > a$ for the values of the functions u_i (in practice we could choose different intervals $[a_i, b_i]$ for each variable t_i , but the analysis would be strictly identical), and let $\Gamma = [a, b]$: in what follows we will work in $\Omega \times \Gamma^k$. We let

$$(2.11) \quad \mathcal{K} = \left\{ \phi = (\phi_1, \dots, \phi_k) \mid \phi_i = (\phi_i^x, \phi_i^t) \in C^0(\Omega \times \Gamma; \mathbb{R}^d \times \mathbb{R}), \right. \\ \left. (\phi_i)_{i=1}^k \text{ satisfies (2.5)} \ \forall x \in \Omega, t \in \Gamma^k \right\}.$$

We will show the following result.

Proposition 2.1. *For $v \in BV(\Omega \times \Gamma; [0, 1]^k)$ let $\mathcal{E}(v)$ be defined by (2.2). Then if $u \in W^{1,1}(\Omega; \Gamma^k)$, one has $\mathcal{E}(\mathbf{1}_u) = E(u)$.*

This result will be a consequence of the stronger Lemmas 2.2 and 2.3 below, which show the result separately for the data and interaction terms.

We now introduce the convex sets (see [39, 22])

$$(2.12) \quad \mathcal{L} = \left\{ \gamma = (\gamma_1, \dots, \gamma_k) \mid \gamma_i \in C^0(\Omega \times \Gamma), \right. \\ \left. \sum_{i=1}^k \gamma_i(x, t_i) \geq -g(x, t) \quad \forall (x, t) \in \Omega \times \Gamma^k \right\}$$

and

$$(2.13) \quad \mathcal{K}_0 = \left\{ \phi = (\phi_1, \dots, \phi_k) \mid \phi_i = (\phi_i^x, \phi_i^t) \in C^0(\Omega \times \Gamma; \mathbb{R}^d \times \mathbb{R}), \right. \\ \left. \sum_{i=1}^k \phi_i^t(x, t_i) \geq f^* \left(x, (\phi_i^x(x, t_i))_{i=1}^k \right) \quad \forall (x, t) \in \Omega \times \Gamma^k \right\}.$$

Observe in particular that $\mathcal{K}_0 + \mathcal{L} \subseteq \mathcal{K}$. Thus, to prove Proposition 2.1 it is enough to prove it with \mathcal{K} replaced by $\mathcal{K}_0 + \mathcal{L}$ in (2.2).

2.4.1. Exactness for the data term part. The following result is an extension of classical results (in particular, relative to the Monge–Kantorovich duality in optimal transportation problems [41]). A proof is found in [39].

Lemma 2.2. *Let $u \in L^1(\Omega; \Gamma^k)$ and assume that g is bounded, l.s.c. in (x, u) .¹ Then*

$$(2.14) \quad \int_{\Omega} g(x, u(x)) dx = \sup_{\gamma \in \mathcal{L}} \int_{\Omega \times \Gamma} \sum_{i=1}^k \gamma_i \cdot D_{t_i} \mathbf{1}_{u_i}.$$

Here we use the notation $D_{t_i} \mathbf{1}_{u_i}$, which is consistent with (2.2) and makes sense since $\mathbf{1}_{u_i}$ is nondecreasing in the t_i -variable. However, observe that $D_{t_i} \mathbf{1}_{u_i}$ is simply equal to $-\delta_{u_i(x)}$. In particular, the integral on the right-hand side can be equivalently rewritten in the simpler form

$$- \int_{\Omega} \sum_{i=1}^k \gamma_i(x, u_i(x)) dx.$$

Proof. Without loss of generality, we may assume that $0 \leq g \leq K$ on $\Omega \times \Gamma^k$ for some constant $K \in \mathbb{R}$. We first assume that g is uniformly continuous on $\Omega \times \Gamma$. Let $u \in L^1(\Omega; \Gamma^k)$. Let $\varepsilon > 0, r > 0$: by standard covering arguments [17], one can find a disjoint covering $(B_{\alpha})_{\alpha \in \mathbb{N}}$ of almost all Ω , that is, disjoint closed balls $B_{\alpha} = \overline{B(x_{\alpha}, r_{\alpha})}$ with $|\Omega \setminus \bigcup_{\alpha} B(x_{\alpha}, r_{\alpha})| = 0$, with the following properties:

1. $r_{\alpha} \leq r$ for all $\alpha \in \mathbb{N}$.

¹One could consider integrands g which are merely measurable in x and continuous in u , by relaxing the continuity assumption of the fields γ_i .

2. $\int_{B_\alpha} |u(x) - u(x_\alpha)| dx \leq r\varepsilon|B_\alpha|$ for all $\alpha \in \mathbb{N}$.
3. $\int_{B_\alpha} |g(x, u(x)) - g(x_\alpha, u(x_\alpha))| dx \leq \varepsilon|B_\alpha|$ for all $\alpha \in \mathbb{N}$.

We assume that $r > 0$ is chosen in such a way that $|g(x, t) - g(y, s)| \leq \varepsilon$ if $|y - x| \leq r$ and $\max_i |t_i - s_i| \leq r$. Let us choose $N \in \mathbb{N}$ such that $\sum_{\alpha > N} |B_\alpha| < \varepsilon$. Then, for $i = 1, \dots, k$ we let $\gamma_i(x, t) = K$ if $x \in B_\alpha$, $\alpha > N$, while if $x \in B_\alpha$ with $\alpha \leq N$,

$$(2.15) \quad \gamma_i(x, t_i) = -\frac{g(x_\alpha, u(x_\alpha))}{k} \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right) \eta\left(\frac{t_i - u_i(x_\alpha)}{\rho}\right) + K \left(1 - \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right) \eta\left(\frac{t_i - u_i(x_\alpha)}{r}\right)\right) + \frac{\varepsilon}{k},$$

where the cut-off functions $\varphi \in C_c^\infty(B(0, 1); [0, 1])$, $\eta \in C_c^\infty([-1, 1]; [0, 1])$ will be made precise later on.

If $x \in B_\alpha$, $\alpha \leq N$, and $t \in \Gamma^k$ with $|t_i - u_i(x)| \geq r$ for at least one i , then $\sum_i \gamma_i(x, t_i) \geq 0 \geq -g(x, t)$. This is also clear if $x \in B_\alpha$ with $\alpha > N$. Now, if $x \in B_\alpha$, $\alpha \leq N$, and $|t_i - u_i(x)| < r$ for all $i = 1, \dots, k$, then

$$\sum_i \gamma_i(x, t_i) \geq -g(x_\alpha, u(x_\alpha)) + \varepsilon \geq -g(x, t)$$

so that $\gamma \in \mathcal{L}$. On the other hand, we have

$$(2.16) \quad - \int_{\Omega} \sum_{i=1}^k \gamma_i(x, u_i(x)) dx \geq -(kK + \varepsilon) \left(\sum_{\alpha > N} |B_\alpha| \right) - \sum_{\alpha \leq N} \sum_{i=1}^k \int_{B_\alpha} \gamma_i(x, u_i(x)) dx.$$

Now, for $\alpha \leq N$,

$$(2.17) \quad - \int_{B_\alpha} \gamma_i(x, u_i(x)) dx \geq \frac{1}{k} |B_\alpha| (g(x_\alpha, u(x_\alpha)) - \varepsilon) - K \left(1 + \frac{1}{k}\right) \int_{B_\alpha} \left(1 - \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right) \eta\left(\frac{u_i(x) - u_i(x_\alpha)}{r}\right)\right) dx,$$

which we now estimate. Assume that φ was chosen such that

$$\int_{B_1} (1 - \varphi(x)) dx \leq \varepsilon$$

and that $\eta(0) = 1$ and η has a Lipschitz constant $L \geq 1$. Then,

$$\begin{aligned} & \int_{B_\alpha} \left(1 - \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right) \eta\left(\frac{u_i(x) - u_i(x_\alpha)}{r}\right)\right) dx \\ &= \int_{B_\alpha} \left(1 - \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right)\right) dx + \int_{B_\alpha} \varphi\left(\frac{x - x_\alpha}{r_\alpha}\right) \left(1 - \eta\left(\frac{u_i(x) - u_i(x_\alpha)}{r}\right)\right) dx \\ &\leq \varepsilon |B_\alpha| + \frac{L}{r} \int_{B_\alpha} |u_i(x) - u_i(x_\alpha)| dx \leq (1 + L)\varepsilon |B_\alpha|. \end{aligned}$$

Together with (2.17), we deduce that for $\alpha \leq N$,

$$\begin{aligned} - \sum_{i=1}^k \int_{B_\alpha} \gamma_i(x, u_i(x)) dx &\geq |B_\alpha|(g(x_\alpha, u(x_\alpha)) - \varepsilon) - \varepsilon K(k+1)(1+L)|B_\alpha| \\ &\geq \int_{B_\alpha} g(x, u(x)) dx - \varepsilon(2 + K(k+1)(1+L))|B_\alpha| \end{aligned}$$

so that, using (2.16),

$$\begin{aligned} - \int_{\Omega} \sum_{i=1}^k \gamma_i(x, u_i(x)) dx &\geq \int_{\Omega} g(x, u(x)) dx \\ &\quad - ((k+1)K + \varepsilon) \left(\sum_{\alpha > N} |B_\alpha| \right) - \varepsilon(2 + K(k+1)(1+L)) \left(\sum_{\alpha \leq N} |B_\alpha| \right) \\ &\geq \int_{\Omega} g(x, u(x)) dx - (((k+1)K + \varepsilon) + (2 + K(k+1)(1+L))|\Omega|)\varepsilon, \end{aligned}$$

which shows that (2.14) holds when g is uniformly continuous.

Now, if g is only l.s.c. (and bounded), there exist g_n bounded and uniformly continuous such that $\sup_n g_n = g$. If \mathcal{L}_n is the corresponding set for g_n , we have $\mathcal{L}_n \subset \mathcal{L}$ so that

$$\int_{\Omega} g(x, u(x)) dx \geq \sup_{\gamma \in \mathcal{L}} \int_{\Omega \times \Gamma} \sum_{i=1}^k \gamma_i \cdot D_{t_i} \mathbf{1}_{u_i} \geq \int_{\Omega} g_n(x, u(x)) dx$$

and the result follows by sending $n \rightarrow \infty$. \blacksquare

2.4.2. Exactness for the regularizer part. Next, we need the following result. To simplify, let f be minimal and vanishing for $p = 0$, i.e., $f(x, 0) = 0 = \min_p f(x, p)$ for all x . We assume that f is continuous in both variables (x, p) , and that there exists a constant $C > 0$ such that

$$(2.18) \quad f(x, p) \geq C(|p| - 1)$$

for all x and p . This guaranties in particular that if $\int_{\Omega} f(x, Du) < +\infty$, then $u \in BV(\Omega; \mathbb{R}^k)$.

Lemma 2.3. *Let $u \in BV(\Omega; \Gamma^k)$. We have*

$$(2.19) \quad \int_{\Omega} f(x, Du) = \sup_{\phi \in \mathcal{K}^0} \sum_{i=1}^k \int_{\Omega \times \Gamma} \phi_i \cdot D \mathbf{1}_{u_i}.$$

Proof. To simplify, we give an idea of the construction in the case where $u \in W^{1,1}(\Omega; \Gamma^k)$. A precise proof in the general case is discussed in Appendix A. In this case, (2.3) holds, and in particular one deduces that “ \geq ” trivially holds in (2.19). Now, let for all x and $(t_i)_{i=1}^k$

$$(\phi_i^x(x, t_i))_{i=1}^k = \nabla_p f(x, \nabla u(x))$$

and

$$\phi_i^t(x, t_i) = \frac{1}{k} f^*(x, \nabla_p f(x, \nabla u(x))).$$

This field is only measurable; observe that it does not depend on t . It can be smoothed by standard mollification: we consider $\rho \in C_c^\infty(B(0, 1); \mathbb{R}_+)$ with $\int_{B(0, 1)} \rho \, dx = 1$ and let $\rho_\varepsilon(x) = \varepsilon^{-d} \rho(x/\varepsilon)$, and then we let $\phi^\varepsilon = \rho_\varepsilon * (\phi \chi_{\Omega^\varepsilon})$, where $\Omega^\varepsilon = \{x \in \Omega \mid \text{dist}(x, \partial\Omega) > \varepsilon\}$ and the convolution is only in the x -variable. Here we use the fact that $f^*(x, 0) = 0$ so that, in particular, $0 \in \mathcal{K}_0$. This smooth (and compactly supported in x) function might not be in \mathcal{K}_0 , but one can show that it is “close” to \mathcal{K}_0 in some sense. Moreover, using (2.3), we have that

$$(2.20) \quad \lim_{\varepsilon \rightarrow 0} \sum_{i=1}^k \int_{\Omega \times \mathbb{R}} \phi_i^\varepsilon \cdot D\mathbf{1}_{u_i} \, dx = \sum_{i=1}^k \int_{\Omega} \phi_i^x(x) \cdot \nabla u_i(x) - \phi_i^t(x) \, dx = \int_{\Omega} f(x, \nabla u(x)) \, dx,$$

as expected. ■

2.5. Existence of minimizers. In the continuous version (2.2) we seek a solution $v \in BV(\Omega \times \Gamma; [0, 1]^k)$, with $\Gamma = [0, 1]$.

Theorem 2.4. *The problem*

$$(2.21) \quad \min \left\{ \mathcal{E}(v) \mid v \in BV(\Omega \times \Gamma; [0, 1]^k), v_i(x, 0) \equiv 1, v_i(x, 1) \equiv 0 \right\},$$

where \mathcal{E} is defined by (2.2) with \mathcal{K} given by (2.11), has a solution.

Here the trace conditions on v_i are meant in the following sense: the functions are extended on $\Omega \times \mathbb{R}$, with $v_i(x, t) \equiv 0$ if $t \geq 1$ and 1 if $t \leq 0$, and the derivative Dv is then restricted to $\Omega \times \Gamma$. As a consequence, if the inner trace of v is different from 0 at $t = 1$ or 1 at $t = 0$, Dv carries a measure on the corresponding boundary.

Proof. This is straightforward, since by definition (2.2), \mathcal{E} is l.s.c. on BV (with respect to weak convergence), while (2.18) (and the fact we have assumed that v is bounded) yields compactness of minimizing sequences for problem (2.21). ■

3. Special cases. Here we give some examples of regularizers $R(u) = \int_{\Omega} f(x, \nabla u(x)) \, dx$ in (1.2) which can be handled in our framework. For each case we will give the corresponding constraints (2.5) for the set (2.11). We assume that the range set of each channel u_i is $\Gamma := [0, 1]$ for clarity of presentation. The whole theory can, of course, be formulated with general intervals as range sets.

3.1. Separable regularizers. We first consider separable regularizers

$$(3.1) \quad R(u) = \sum_{i=1}^k R_i(u_i), \quad R_i(u_i) = \int_{\Omega} f_i(x, \nabla u_i(x)) \, dx,$$

i.e., $f(x, p) = \sum_{i=1}^k f_i(x, p_i)$ for all $(x, p) \in \Omega \times \mathbb{R}^{d \times k}$, and $f(x, p)$ acts on each u_i independently. Convex relaxation of the functional (1.2) with the restriction to this kind of regularizers has been considered in [39]. The relaxation was obtained by convexifying each term separately, the data term $\int_{\Omega} g(x, u(x)) \, dx$ and the regularizers $R_1(u_1), \dots, R_k(u_k)$. In contrast, our proposed

relaxation of (1.2) considers the functional as a whole and uses a single combined constraint set (2.11). However, for separable regularizers it turns out to be equivalent to relaxing each term separately as in [39], as we will show next.

Define the sets \mathcal{K}_0^i similarly as \mathcal{K}_0 in (2.13) but for the one-dimensional case ($k = 1$):

$$(3.2) \quad \mathcal{K}_0^i = \left\{ \phi = (\phi^x, \phi^t) \in C^0(\Omega \times \Gamma; \mathbb{R}^d \times \mathbb{R}) \mid \right. \\ \left. \phi^t(x, t) \geq f_i^*(x, \phi^x(x, t)) \quad \forall (x, t) \in \Omega \times \Gamma \right\}.$$

Proposition 3.1. *Let the regularizer be separable as in (3.1). Then the relaxation*

$$(3.3) \quad \mathcal{E}(v) = \sup_{\phi \in \mathcal{K}} \sum_{i=1}^k \int_{\Omega \times \Gamma} \phi_i \cdot Dv_i \, dx = \sup_{\phi \in \mathcal{K}} \sum_{i=1}^k \int_{\Omega \times \Gamma} (\phi_i^x D_x v_i + \phi_i^t D_t v_i)$$

with the general set \mathcal{K} in (2.11) is equal to the relaxation of each term separately:

$$(3.4) \quad \mathcal{E}(v) = \sum_{i=1}^k \left(\sup_{\phi_i \in \mathcal{K}_0^i} \int_{\Omega \times \Gamma} \phi_i Dv_i \right) + \left(\sup_{\gamma \in \mathcal{L}} \int_{\Omega \times \Gamma} \sum_{i=1}^k \gamma_i D_t v_i \right).$$

Proof. Let $(\phi^x, \phi^t) \in \mathcal{K}$. The Legendre–Fenchel conjugate of $f(x, p) = \sum_i f_i(x, p_i)$ is given by

$$f^*(x, q) = \sup_{p \in \mathbb{R}^{d \times k}} \sum_{i=1}^k q_i p_i - \sum_{i=1}^k f_i(x, p_i) = \sum_{i=1}^k f_i^*(x, q_i)$$

for all $q \in \mathbb{R}^{d \times k}$. The constraints (2.5) thus become

$$(3.5) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t) + \sum_{i=1}^k f_i^*(x, \phi_i^x(x, t_i)).$$

Define $\gamma_i, \bar{\phi}_i^t \in C^0(\Omega \times \Gamma; \mathbb{R})$ by

$$\bar{\phi}_i^t(x, t_i) := f_i^*(x, \phi_i^x(x, t_i)), \\ \gamma_i := \phi_i^t - \bar{\phi}_i^t$$

for all $(x, t_i) \in \Omega \times \Gamma$. Obviously we have $(\phi_i^x, \bar{\phi}_i^t) \in \mathcal{K}_0^i$ for all i , and by (3.5) also $\gamma \in \mathcal{L}$. On the other hand, if $(\phi_i^x, \bar{\phi}_i^t) \in \mathcal{K}_0^i$ and $\gamma \in \mathcal{L}$, then for $\phi_i^t := \bar{\phi}_i^t + \gamma_i$ we have $(\phi^x, \phi^t) \in \mathcal{K}$. Therefore, (3.4) follows directly from (3.3). ■

As a consequence, in order to arrive at novel relaxations, one has to consider *coupled* regularizers. For this general case, no tractable relaxations have yet been given. In fact, existing relaxations all rely on discretizing the whole k -dimensional label space Γ^k and are thus by no means tractable.

In the following we will give a brief overview of some interesting special cases for separable regularizers (3.1) which were studied in [34]. We will discuss these regularizers in more detail in sections 3.3–3.7, where we introduce the corresponding *coupled* versions.

3.1.1. Total variation with l^1 -coupling. Setting $f_i(x, p_i) := \lambda|p_i|$ with a $\lambda > 0$, we obtain the total variation

$$(3.6) \quad TV_{l^1}(u) = \lambda \sum_{i=1}^k \int_{\Omega} |\nabla u_i| dx.$$

Although this regularizer is a simple way to extend the total variation to vector-valued signals, there is no coupling of the channels. This generally leads to inferior reconstructions, as is demonstrated in Figure 5. The corresponding constraints in (3.2) are

$$(3.7) \quad \phi_i^t(x, t) \geq 0, \quad |\phi_i^x(x, t)| \leq \lambda \quad \forall x \in \Omega, t \in \Gamma, 1 \leq i \leq k.$$

3.1.2. Huber-TV with l^1 -coupling. We now set $f_i(x, p_i) := \lambda h_{\varepsilon}(|p_i|)$ with the Huber function $h_{\varepsilon}(p_i)$, which basically equals $|p_i|$ but smooths out the kink at the origin. We will define it later when we discuss the l^2 -coupled case in section 3.4. This yields the Huber-TV $_{l^1}$ penalization

$$(3.8) \quad \lambda \sum_{i=1}^k \int_{\Omega} h_{\varepsilon}(|\nabla u_i|) dx.$$

This alleviates the staircasing effect caused by TV (i.e., the solutions tend to become piecewise constant in regions where u is almost constant but smooth); however, there is no coupling of the channels at all. The constraints in (3.2) are

$$(3.9) \quad \phi_i^t(x, t) \geq \frac{\varepsilon}{2\lambda} |\phi_i^x(x, t)|^2, \quad |\phi_i^x(x, t)| \leq \lambda \quad \forall x \in \Omega, t \in \Gamma, 1 \leq i \leq k.$$

3.1.3. Lipschitz constraint with l^1 -coupling. Finally, setting $f_i(x, p_i) := \delta_{|p_i| \leq \lambda}$, which is zero for $|p_i| \leq \lambda$ and ∞ otherwise, with a $\lambda > 0$ we obtain the Lipschitz constraint on the gradients of the channels:

$$\sum_{i=1}^k \int_{\Omega} \delta_{|\nabla u_i| \leq \lambda} dx = \delta_{(|\nabla u_i(x)| \leq \lambda \text{ for a.e. } x \in \Omega, 1 \leq i \leq k)}.$$

The growth rate of each channel is constrained by λ individually without any coupling. The constraints in (3.2) are

$$(3.10) \quad \phi_i^t(x, t) \geq \lambda |\phi_i^x(x, t)| \quad \forall x \in \Omega, t \in \Gamma, 1 \leq i \leq k.$$

3.2. Separable data terms. The proposed convex relaxation handles the functional (1.2) as a whole; i.e., both the data term and the regularization term are relaxed simultaneously using a unified constraint set. Consider separable data terms g ,

$$(3.11) \quad D(u) = \sum_{i=1}^k D_i(u_i), \quad D_i(u_i) = \int_{\Omega} g_i(x, u_i(x)) dx,$$

i.e., $g(x, t) = \sum_{i=1}^k g_i(x, t_i)$ for all $(x, t) \in \Omega \times \mathbb{R}^k$. In this case we can show that the overall relaxation is equivalent to relaxing the data term and the regularizer term separately (as was also the case for separable regularizers in the previous section). Furthermore, the data term part decouples into separate relaxations of each channel.

Proposition 3.2. *Let the data term be separable as in (3.11). Then*

$$(3.12) \quad \mathcal{E}(v) = \left(\sup_{\phi \in \mathcal{K}_0} \sum_{i=1}^k \int_{\Omega \times \Gamma} \phi_i \cdot Dv_i \right) + \sum_{i=1}^k \left(\sup_{\gamma_i \in \mathcal{L}_i} \int_{\Omega \times \Gamma} \gamma_i D_t v_i \right).$$

The sets \mathcal{L}_i are defined similarly to \mathcal{L} in (2.12) but for the one-dimensional case:

$$\mathcal{L}_i = \left\{ \gamma \in C^0(\Omega \times \Gamma; \mathbb{R}) \mid \gamma(x, t) \geq -g_i(x, t) \quad \forall (x, t) \in \Omega \times \Gamma \right\}.$$

Proof. The proof basically uses the same construction as the corresponding proof of (3.4) for the case of separable regularizers. The constraints (2.5) read as

$$(3.13) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq - \sum_{i=1}^k g_i(x, t_i) + f^*(x, (\phi_i^x(x, t_i))_{i=1}^k).$$

For $(\phi^x, \phi^t) \in \mathcal{K}$ with the set \mathcal{K} in (2.11), define $\gamma_i, \bar{\phi}_i^t \in C^0(\Omega \times \Gamma; \mathbb{R})$ by

$$\begin{aligned} \gamma_i(x, t_i) &:= -g_i(x, t_i), \\ \bar{\phi}_i^t &:= \phi_i^t - \gamma_i \end{aligned}$$

for all $(x, t) \in \Omega \times \Gamma^k$ and $1 \leq i \leq k$. Then $(\phi^x, \bar{\phi}^t) \in \mathcal{K}_0$ by (3.13), and evidently also $\gamma_i \in \mathcal{L}_i$ for all i . On the other hand, if $(\phi^x, \bar{\phi}^t) \in \mathcal{K}_0$ and $\gamma_i \in \mathcal{L}_i$ for all i , then for $\phi_i^t := \bar{\phi}_i^t + \gamma_i$ we have $(\phi^x, \phi^t) \in \mathcal{K}$. Thus, (3.12) follows from (3.3). ■

Remark. Another case where the overall relaxation is equivalent to relaxing the data term and the regularizer term separately is when the Legendre–Fenchel dual of f has the form

$$(3.14) \quad f^*(x, p) = \sum_{i=1}^k h_i(x, p_i) + \delta_{C(x)}(p)$$

with some convex functions h_i and sets $C(x) \subset \mathbb{R}^{d \times k}$ (the indicator function $\delta_M(p)$ of a set M is defined by 0 for $p \in M$ and by ∞ otherwise). This can be proved analogously as for (3.4) in section 3.1. Specifically, in the case discussed in section 3.1 we have $h_i = f_i^*$ and $C(x) = \mathbb{R}^{d \times k}$. An example of a regularizer which satisfies (3.14) with a nontrivial set $C(x)$ is given by TV_{l^2} , which we will discuss next.

3.3. Total variation with l^2 -coupling. As a first nonseparable regularizer for vectorial signals $u : \Omega \rightarrow \mathbb{R}^k$ we consider the total variation with the l^2 -coupling of the channels. For smooth functions u it is given by

$$(3.15) \quad TV_{l^2}(u) = \lambda \int_{\Omega} \|\nabla u\|_2 dx = \lambda \int_{\Omega} \sqrt{\sum_{i=1}^k |\nabla u_i|^2} dx$$

with a $\lambda > 0$. Basically, it penalizes the Euclidean norm of the gradient. For general $u \in \mathcal{L}^1(\Omega; \mathbb{R}^k)$ it can be defined by its dual representation

$$TV_{l^2}(u) = \sup_{\|\xi\|_2 \leq \lambda} \int_{\Omega} \sum_{i=1}^k u_i \operatorname{div} \xi_i \, dx,$$

where the supremum is taken over all $\xi \in C_c^1(\Omega; \mathbb{R}^{d \times k})$ such that pointwise $\|\xi(x)\|_2 \leq \lambda$ for all $x \in \Omega$. This *coupled* total variation generally leads to higher quality reconstructions in inverse problems than its *separable* counterpart in (3.6), as will be shown in the experiments. Our approach yields the first convex relaxation of this regularizer for vectorial multilabel problems. Furthermore, we will show how to efficiently reformulate the relaxation to obtain roughly the same run time and memory efficiency as in the decoupled case (3.6).

The corresponding function $f : \Omega \times \mathbb{R}^{d \times k} \rightarrow \mathbb{R}$ in (1.2) is $f(x, p) := \lambda \|p\|_2$ with the Legendre–Fenchel convex dual

$$f^*(x, q) = \sup_{p \in \mathbb{R}^{d \times k}} p q - \|p\|_2 = \begin{cases} 0 & \text{if } \|q\|_2 \leq \lambda, \\ \infty & \text{else.} \end{cases}$$

Thus, the constraints (2.5) are given by

$$(3.16) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t),$$

$$(3.17) \quad \sqrt{\sum_{i=1}^k |\phi_i^x(x, t_i)|^2} \leq \lambda$$

for all $(x, t) \in \Omega \times \Gamma^k$.

Constraint decoupling for smoothness part (3.17). For a practical implementation, the range set Γ of each channel u_i must be discretized into a number $n_i \geq 1$ of levels. For each fixed $x \in \Omega$, the second constraint (3.17) then poses $n_1 \cdots n_k$ individual constraints because of $t \in \Gamma^k$. Implementing them requires a large amount of memory for $k \geq 3$ and to some extent for $k \geq 2$. Surprisingly, the special form of the l^2 -coupled TV_{l^2} allows one to decouple (3.17) into only $n_1 + \cdots + n_k$ constraints.

The inequalities (3.17) for all $t \in \Gamma^k$ are equivalent to

$$\sup_{t \in \Gamma^k} \sqrt{\sum_{i=1}^k |\phi_i^x(x, t_i)|^2} = \sqrt{\sum_{i=1}^k \left(\sup_{t_i \in \Gamma} |\phi_i^x(x, t_i)| \right)^2} \leq 1.$$

Defining a new dual variable $a : \Omega \rightarrow \mathbb{R}^k$ by $a_i(x) := \sup_{t_i \in \Gamma} |\phi_i^x(x, t_i)|$ for all i , this shows that the constraints (3.17) can be equivalently written as

$$(3.18) \quad \|a(x)\|_2 \leq \lambda \quad \forall x \in \Omega,$$

$$(3.19) \quad |\phi_i^x(x, t_i)| \leq a_i(x) \quad \forall (x, t_i) \in \Omega \times \Gamma, \quad 1 \leq i \leq k.$$

For each $x \in \Omega$, (3.18) and (3.19) are now only linearly many constraints.

Remark. There is an interesting interpretation of this decoupling: considering the constraints (3.19) without (3.18) for each i , the supremum over ϕ_i^x gives the total variation of u_i , where the contribution at each point $x \in \Omega$ is weighted by $a_i(x)$:

$$(3.20) \quad TV_{a_i}(u_i) = \int_{\Omega} a_i(x) |\nabla u_i| dx.$$

Taking the supremum also over (3.18) means that $TV(u)$ is represented as a weighted sum of $TV_{a_i}(u_i)$:

$$(3.21) \quad TV_2(u) = \sup_{\substack{a: \Omega \rightarrow \mathbb{R}^k, \\ \sum_{i=1}^k |a_i(x)|^2 \leq \lambda}} \sum_{i=1}^k \int_{\Omega} a_i(x) |\nabla u_i| dx.$$

Constraint decoupling for data part (3.16). Similarly to (3.17), after discretization the first constraint (3.16) in its original form gives $n_1 \cdots n_k$ individual constraints for each $x \in \Omega$. However, if the data term is separable as discussed in section 3.2, we can also decouple (3.16) into linearly many constraints.

Proposition 3.3. *Assume that $g(x, t) = \sum_{i=1}^k g_i(x, t_i)$ for all $(x, t) \in \Omega \times \mathbb{R}^k$ with some functions $g_i : \Omega \times \Gamma \rightarrow \mathbb{R}$. Then (3.16) can be replaced by*

$$(3.22) \quad \phi_i^t(x, t_i) \geq -g_i(x, t_i) \quad \forall (x, t_i) \in \Omega \times \Gamma, \quad 1 \leq i \leq k,$$

without altering the supremum in the convex relaxation (2.2).

Proof. First, since the constraints on ϕ^t and on ϕ^x are independent, the relaxation (2.2) reads as

$$\mathcal{E}(v) = \left(\sup_{\phi^x} \int_{\Omega \times \Gamma} \sum_{i=1}^k \phi_i^x D_x v_i \right) + \left(\sup_{\phi^t} \int_{\Omega \times \Gamma} \sum_{i=1}^k \phi_i^t D_t v_i \right)$$

with ϕ^x satisfying (3.17) and ϕ^t satisfying (3.16). For the second term on the right-hand side, regarding the data-term-only constraints (3.16) as the general constraints (2.5) with $f \equiv 0$ and $\phi^x \equiv 0$, we can use (3.12) (as the data term is separable) to get

$$(3.23) \quad \sup_{\phi^t} \int_{\Omega \times \Gamma} \sum_{i=1}^k \phi_i^t D_t v_i = \sum_{i=1}^k \sup_{\gamma_i \in \mathcal{L}_i} \int_{\Omega \times \Gamma} \gamma_i D_t v_i + \sup_{\hat{\phi}^t} \int_{\Omega \times \Gamma} \sum_{i=1}^k \hat{\phi}_i^t D_t v_i$$

with $\hat{\phi}^t$ such that $\sum_{i=1}^k \hat{\phi}_i^t(x, t_i) \geq 0$. The first term on the right-hand side of (3.23) yields the desired constraints (3.22), after renaming γ_i back to ϕ_i^t . It remains to show that the supremum in the second term on the right-hand side of (3.23) is actually zero. But this follows directly from Lemma 2.2. ■

3.4. Huber-TV with l^2 -coupling. Total variation regularization is known to produce solutions exhibiting so-called staircasing effects. In the regions where the solution u is almost

constant or varies very slowly, it may become piecewise constant instead of having a smooth variation. A common solution is to apply the Huber- TV regularization

$$(3.24) \quad R(u) = \lambda \int_{\Omega} h_{\varepsilon}(\|\nabla u\|_2) dx$$

for some small $\varepsilon > 0$, where the Huber function $h_{\varepsilon} : \mathbb{R} \rightarrow \mathbb{R}$ is defined as

$$(3.25) \quad h_{\varepsilon}(z) := \begin{cases} \frac{1}{2\varepsilon} z^2 & \text{if } |z| < \varepsilon, \\ |z| - \frac{\varepsilon}{2} & \text{else.} \end{cases}$$

It smooths out the kink at the origin of $z \mapsto |z|$. The quadratic penalization for near zero ∇u ensures smooth variations in the regions where u is nearly constant, thus avoiding the staircasing effect in these regions. The limiting case $\varepsilon = 0$ yields the usual TV . The advantage in comparison to applying the channelwise Huber regularization (3.8) is that here the quadratic penalization kicks in only if *every* gradient component is small. The coupled Huber- TV_2 is thus nearer to the actual TV_2 which it is approximating.

The function f in (1.2) is $f(x, p) := \lambda h_{\varepsilon}(\|p\|_2)$, with the dual

$$(3.26) \quad f^*(x, q) = \sup_{p \in \mathbb{R}^{d \times k}} \sum_{i=1}^k q_i p_i - \lambda h_{\varepsilon}(\|p\|_2) = \begin{cases} \frac{\varepsilon}{2\lambda} \|q\|_2^2 & \text{if } \|q\|_2 \leq \lambda, \\ \infty & \text{else,} \end{cases}$$

and the constraints in (2.11) become

$$(3.27) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t) + \frac{\varepsilon}{2\lambda} \sum_{i=1}^k |\phi_i^x(x, t_i)|^2,$$

$$(3.28) \quad \sqrt{\sum_{i=1}^k |\phi_i^x(x, t_i)|^2} \leq \lambda$$

for all $(x, t) \in \Omega \times \Gamma^k$.

Constraint decoupling for the smoothness part (3.28). Note that the second constraint (3.28) is exactly the same as (3.17) in the case of TV in section 3.3. The same reduction technique can therefore be applied to decouple this constraint into (3.18) and (3.19).

Constraint decoupling for the data and smoothness part (3.27). As for the first constraint (3.27), for general data terms g we can always separate it into a data-term-only part and a part responsible for regularization. Namely, define $\gamma_i, \bar{\phi}_i^t \in C^0(\Omega \times \Gamma; \mathbb{R})$ by

$$\begin{aligned} \bar{\phi}_i^t(x, t_i) &:= \frac{\varepsilon}{2\lambda} |\phi_i^x(x, t_i)|^2, \\ \gamma_i &:= \phi_i^t - \bar{\phi}_i^t \end{aligned}$$

for all $(x, t_i) \in \Omega \times \Gamma$ and $1 \leq i \leq k$. Then (3.27) is equivalent to

$$(3.29) \quad \phi_i^t = \gamma_i + \bar{\phi}_i^t,$$

$$(3.30) \quad \sum_{i=1}^k \gamma_i(x, t_i) \geq -g(x, t),$$

$$(3.31) \quad \bar{\phi}_i^t(x, t_i) \geq \frac{\varepsilon}{2\lambda} |\phi_i^x(x, t_i)|^2$$

for all $(x, t) \in \Omega \times \Gamma^k$. Thus, Huber- TV_{l_2} penalization requires only linearly many new constraints (3.31) in addition to those of TV_{l_2} . This regularizer is therefore also very efficient, with respect to memory and run time.

If the data term is separable, i.e., $g(x, t) = \sum_i g_i(x, t_i)$, then we can further decouple the data term constraint (3.30) into linearly many constraints (3.22), as has been done for the identical constraint (3.16) in section 3.3. For brevity and to reduce the overall number of variables, we can use (3.29) to combine these data term constraints with (3.31) into just

$$(3.32) \quad \phi_i^t(x, t_i) \geq -g_i(x, t_i) + \frac{\varepsilon}{2\lambda} |\phi_i^x(x, t_i)|^2$$

for all $(x, t_i) \in \Omega \times \Gamma$ and $1 \leq i \leq k$.

3.5. Total variation for general norms. We can also define more general versions of the total variation by choosing other norms $\|\cdot\|$ instead of the Euclidean norm in (3.15) in section 3.3 in which to penalize image gradients ∇u :

$$(3.33) \quad TV_{\|\cdot\|}(u) = \lambda \int_{\Omega} \|\nabla u\| dx.$$

The interaction term here is $f(x, p) = \lambda \|p\|$. The convex dual is given by the indicator function of the corresponding dual norm $\|\cdot\|_*$:

$$(3.34) \quad \begin{aligned} f^*(x, q) &= \sup_{p \in \mathbb{R}^{d \times k}} pq - \lambda \|p\| = \sup_{\substack{p \in \mathbb{R}^{d \times k}, t \geq 0 \\ \|p\|=1}} t(pq - \lambda) \\ &= \sup_{\substack{p \in \mathbb{R}^{d \times k} \\ \|p\|=1}} \delta_{pq \leq \lambda} = \delta_{(\sup_{p \in \mathbb{R}^{d \times k}, \|p\|=1} pq \leq \lambda)} \\ &= \delta_{\|q\|_* \leq \lambda} \end{aligned}$$

with

$$(3.35) \quad \|q\|_* := \sup_{\substack{p \in \mathbb{R}^{d \times k} \\ \|p\| \leq 1}} pq = \sup_{\substack{p \in \mathbb{R}^{d \times k} \\ \|p\|=1}} pq.$$

Constraints (2.5) thus become

$$(3.36) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t),$$

$$(3.37) \quad \left\| (\phi_i^x(x, t_i))_{1 \leq i \leq k} \right\|_* \leq \lambda$$

for all $(x, t) \in \Omega \times \Gamma^k$. While the first constraint (3.36) is the same as (3.16), the second (3.37) is a generalization of (3.17). Note that (3.37) reduces to (3.17) for $\|\cdot\| = \|\cdot\|_2$ since the dual norm is again $\|\cdot\|_2$.

Constraint decoupling. Since the number of constraints in (3.37) is quite large, a practical question is, Which norms allow us to decouple the constraints? Immediate candidates for the generalization of the Euclidean case technique of section 3.4 are the κ -norms:

$$(3.38) \quad \|x\|_\kappa := \sqrt[\kappa]{\sum_{i=1}^k |x_i|^\kappa} \quad \text{for } 1 \leq \kappa < \infty, \quad \|x\|_\infty := \max_{1 \leq i \leq k} |x_i|,$$

defined for $x \in \mathbb{R}^{d \times k}$. The dual norm (3.35) is $\|\cdot\|_* = \|\cdot\|_\zeta$, where $1 \leq \zeta \leq \infty$ is defined by $\frac{1}{\kappa} + \frac{1}{\zeta} = 1$, i.e., $\zeta = \frac{\kappa}{\kappa-1}$. Just as in section 3.4, we can show that (3.37) is equivalent to

$$(3.39) \quad \|a(x)\|_{\frac{\kappa}{\kappa-1}} \leq 1 \quad \forall x \in \Omega,$$

$$(3.40) \quad |\phi_i^x(x, t_i)| \leq a_i(x) \quad \forall (x, t_i) \in \Omega \times \Gamma, \quad 1 \leq i \leq k,$$

introducing additional dual variables $a : \Omega \rightarrow \mathbb{R}^k$.

Natural total variation TV_J for color images. An interesting special case of (3.47) is the vectorial total variation TV_J of Goldluecke, Strelakovski, and Cremers [21]. They showed that it yields best results in inverse problems such as denoising, inpainting, and superresolution in comparison to other possible total variations such as (3.6) and (3.15) (with norms $\|\cdot\|_1$, respectively, $\|\cdot\|_2$). While the initial approach [21] can be used only for convex data terms g , our vectorial multilabel convexification framework extends its applicability to arbitrary data terms.

The corresponding norm in (3.33) is defined for TV_J as the largest singular value of ∇u :

$$(3.41) \quad \|p\| = \|(\sigma_i)_{1 \leq i \leq m}\|_\infty = \max_{1 \leq i \leq m} \sigma_m,$$

where $\sigma_1, \dots, \sigma_m \geq 0$ with $m \leq \min(d, k)$ are the singular values of $p \in \mathbb{R}^{d \times k}$. The dual norm is the nuclear norm of p , which is the sum of the singular values:

$$(3.42) \quad \|p\|_* = \|(\sigma_i)_{1 \leq i \leq m}\|_1 = \sum_{i=1}^m \sigma_m.$$

There is no immediate way to decouple the arising constraints (3.37). Thus this regularizer is more costly in terms of memory and run time for nonconvex data terms than TV_{l_2} and TV_{l_1} .

3.6. Huber-TV for general norms. Just as for TV_{l_2} in section 3.4, one can consider Huber-TV regularization with general norms. The staircasing effects are then eliminated, while the desired properties of the respective TV are still preserved. The case (3.24) generalizes to

$$(3.43) \quad R(u) = \lambda \int_{\Omega} h_\varepsilon(\|\nabla u\|) dx$$

with a general norm $\|\cdot\|$ and the Huber function h_ε in (3.25). The convexification is a straightforward generalization of section 3.4: the function f in (1.2) is $f(x, p) := \lambda h_\varepsilon(\|p\|)$, and (3.26) becomes

$$(3.44) \quad f^*(x, q) = \begin{cases} \frac{\varepsilon}{2\lambda} \|q\|_*^2 & \text{if } \|q\|_* \leq \lambda, \\ \infty & \text{else} \end{cases}$$

with the dual norm (3.35). The constraints in (2.11) are now

$$(3.45) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t) + \frac{\varepsilon}{2\lambda} \left\| (\phi_i^x(x, t_i))_{1 \leq i \leq k} \right\|_*^2,$$

$$(3.46) \quad \left\| (\phi_i^x(x, t_i))_{1 \leq i \leq k} \right\|_* \leq \lambda$$

for all $(x, t) \in \Omega \times \Gamma^k$. As an example, one can consider the Huber regularization of the natural TV for color images in section 3.5.

3.7. Lipschitz constraint with l^2 -coupling. In some applications the rate of growth of u is bounded a priori by a constant. To enforce this, we can consider the following regularizer:

$$(3.47) \quad R(u) = \int_{\Omega} \delta_{\|\nabla u\|_2 \leq \lambda} dx.$$

The interaction term here is $f(x, p) = \delta_{\|p\|_2 \leq \lambda}$ with the dual

$$(3.48) \quad f^*(x, q) = \sup_{p \in \mathbb{R}^{d \times k}} pq - \delta_{\|p\|_2 \leq \lambda} = \sup_{\|p\|_2 \leq \lambda} pq = \lambda \|q\|_2.$$

The constraints (2.5) become

$$(3.49) \quad \sum_{i=1}^k \phi_i^t(x, t_i) \geq -g(x, t) + C \sqrt{\sum_{i=1}^k |\phi_i^x(x, t_i)|^2}$$

for all $(x, t) \in \Omega \times \Gamma^k$. The constraints (3.49) cannot be easily decoupled, making the l^2 -coupled Lipschitz constraint a costly regularizer.

4. Implementation.

4.1. Discretization. We discretize the image domain into a rectangular pixel grid, which we again denote by Ω . For each $1 \leq i \leq k$ we also discretize the range set Γ of $u_i : \Omega \rightarrow \Gamma$ into a number $n_i \geq 1$ of levels $\frac{0}{n_i-1}, \dots, \frac{n_i-1}{n_i-1}$ and write $\Gamma_i := \{0, \dots, n_i - 1\}$. The range discretization is necessary since the relaxed energy (2.2) is defined on the space $\Omega \times \Gamma$. We set $\Lambda := \Gamma_1 \times \dots \times \Gamma_k$ for the set of all labels. The discretized variables v , ϕ , and g are represented by their node values

$$(4.1) \quad \begin{aligned} v_i \left(x, \frac{j}{n_i-1} \right) &= v_i^j(x) \in \mathbb{R}, \\ \phi_i^x \left(x, \frac{j}{n_i-1} \right) &= \phi_i^{x,j}(x) \in \mathbb{R}^d, \\ \phi_i^t \left(x, \frac{j}{n_i-1} \right) &= \phi_i^{t,j}(x) \in \mathbb{R}, \\ g \left(x, \left(\frac{j_1}{n_1-1}, \dots, \frac{j_k}{n_k-1} \right) \right) &= g^{(j_1, \dots, j_k)}(x) \in \mathbb{R} \end{aligned}$$

for all pixels $x \in \Omega$, $0 \leq j < n_i$, $0 \leq i \leq k$, and $(j_1, \dots, j_k) \in \Lambda$. The discretized energy becomes

$$(4.2) \quad \min_{v \in \mathcal{D}} \mathcal{E}(v), \quad \mathcal{E}(v) = \max_{\phi \in \mathcal{K}} \sum_{x \in \Omega} \sum_{i=1}^k \sum_{0 \leq j < n_i} \left(\frac{1}{n_i-1} \phi_i^{x,j}(x) \nabla_x v_i^j(x) + \phi_i^{t,j}(x) \partial_{t,i}^+ v_i^j(x) \right).$$

The factors $\frac{1}{n_i-1}$ before ϕ^x arise from the discretization of the integrals $\int_{\Gamma} dt_i$. They are not present before ϕ^t since they are canceled through the discretization of $\partial_{t_i} v_i$.

We use forward differences with Neumann boundary conditions for the spatial gradient ∇_x . For example, $\partial_{x_1} f_{x_1, x_2} = f_{x_1+1, x_2} - f_{x_1, x_2}$ if $(x_1 + 1, x_2) \in \Omega$ and 0 else, and we define the divergence by $\text{div} := -\nabla^T$, the negative adjoint operator, in order for a discrete form of partial integration to hold. For the t -derivative $\partial_{t_i}^+$ of the i th range space Γ_i , we use forward differences with zero boundary condition: $\partial_{t_i}^+ v_i^j = v_i^{j+1} - v_i^j$ if $j + 1 < n_i$ and $-v_i^j$ else. This way, in (4.2) we implicitly use $v_i^{n_i}(x) = 0$. The negative adjoint t -derivative $\partial_{t,i}^- = -(\partial_{t,i}^+)^T$ is then given by backward differences: $\partial_{t,i}^- f_t = f_t - f_{t-1}$ if $t > 0$ and f_t if $t = 0$.

We are looking for minimizers v which lie in the convex set

$$(4.3) \quad \mathcal{D} = \left\{ v = (v_i)_{1 \leq i \leq k} \mid v_i : \Omega \rightarrow [0, 1]^{n_i}, v_i^0(x) = 1 \quad \forall x \in \Omega, 1 \leq i \leq k \right\}.$$

Of the two boundary conditions $v_i(x, 0) = 1$ and $v_i(x, 1) = 0$, only the first one is imposed explicitly in (4.3). The second is encoded implicitly through the discretization of $\partial_{t,i}^+$ as described above. The monotonicity constraint on v_i , i.e., that $v_i^j(x)$ is nonincreasing in j , is not included in \mathcal{D} since it is already implicitly implied by the constraint set (2.11): the dual variable ϕ^t may be arbitrarily large, and therefore the supremum in (4.2) is finite only if $\partial_{t,i}^+ v_i^j \leq 0$, i.e., if v_i is nonincreasing.

The discretized set \mathcal{K} in (2.11) is

$$(4.4) \quad \mathcal{K} = \left\{ \phi = (\phi_i)_{1 \leq i \leq k} \mid \phi_i = (\phi_i^x, \phi_i^t) : \Omega \times \Gamma_i \rightarrow \mathbb{R}^d \times \mathbb{R}, \right. \\ \left. \sum_{i=1}^k \phi_i^{t, j_i}(x) \geq -g^j(x) + f^*(x, (\phi_i^{x, j_i}(x, t_i))_{i=1}^k) \quad \forall x \in \Omega, j \in \Lambda \right\}.$$

The set \mathcal{K} depends on the employed regularizer function f , as well as on the corresponding strategies to decouple the constraints. Implementation of these constraints is detailed later in section 4.2.

Optimality of solutions. Because of the convex relaxation of the range of graph functions v_i from $\{0, 1\}$ to $[0, 1]$, the computed solution v of (4.2) may be nonbinary. Therefore, at the end we need to project the result back to the space of binary functions. One possible solution is to threshold at $\frac{1}{2}$; i.e.,

$$(4.5) \quad (v_{\text{bin}})_i^j(x) = \begin{cases} 1 & \text{if } j \leq j_0 \\ 0 & \text{if } j > j_0 \end{cases} \quad \text{with } j_0 := \arg \max_{\substack{0 \leq j < n_i, \\ v_i^j(x) \geq \frac{1}{2}}} j$$

for every channel $1 \leq i \leq k$. From this we then construct a solution u_{bin} by (2.1). Though this solution is not necessarily optimal for the initial problem (1.2), we can give an energy bound to estimate how far v_{bin} is from a true solution u^* :

$$(4.6) \quad E(u_{\text{bin}}) - E(u^*) \leq \mathcal{E}(v_{\text{bin}}) - \mathcal{E}(v).$$

In our experiments this bound was around 3% for the separable regularizers such as TV_{l^1} , and around 6% for the coupled ones such as TV_{l^2} . This shows that our approach is able to provide optimal or near-optimal solutions.

For candidate solutions u of the initial problem (1.2) it is not required that the values lie in a fixed discretized set, as is the case for u_{bin} above. Therefore, when computing the actual end result for (1.2) we use interpolated thresholding:

$$(4.7) \quad u_{\text{res},i}(x) := \frac{j_0 + s - \frac{1}{2}}{n_i - 1} \quad \text{with} \quad s := \frac{v_i^{j_0}(x) - \frac{1}{2}}{v_i^{j_0}(x) - v_i^{j_0+1}(x)}$$

and j_0 given by (4.5). We observed that this generally yields a higher quality solution, with $E(u_{\text{res}}) < E(u_{\text{bin}})$.

4.2. Numerical algorithm. To solve the saddle-point problem (4.2) we use the recent fast primal-dual algorithm [12]. Basically, it computes gradient descent steps in the primals v and gradient ascent steps in the duals ϕ , and projects back onto the constraint sets after each iteration. The algorithm can be massively parallelized on GPUs. Since there will be many more dual variables than primal ones, to save memory we suggest using the version of the algorithm where the “bar” copies are introduced for the primals rather than for the duals.

To set the time steps automatically, we use the convenient preconditioning scheme of [32] for the primal-dual algorithm of [12]. In fact, we use an extension of this scheme where all primal time steps given by [32] are multiplied by τ and all dual time steps by $\frac{1}{\tau}$, where $\tau > 0$ is a fixed constant. The convergence proof of [32] can be easily extended to this case. Depending on the problem at hand, choosing τ appropriately may significantly speed up convergence. On typical problems, compared to $\tau = 1$ we observed a speed up by around a factor of 2–5, and sometimes even more. We used $\tau = 10$ in all our experiments, except for inpainting, where $\tau = 1000$.

The projection of the main primal variable v is straightforward and can be done by simple clipping of the values $v_i^j(x)$ to $[0, 1]$. For ϕ the projection is more involved since \mathcal{K} contains many nonlocal constraints. Our strategy is therefore to implement these constraints using the method of Lagrange multipliers or of convex dualization, depending on the regularizer.

4.2.1. Total variation with l^2 -coupling. The continuous version has the constraints (3.18) and (3.19) for the smoothness part, as well as (3.16), respectively, (3.22) for the data term part. We dualize the constraints (3.19) using convex dualization (for $x \in \mathbb{R}^d$, $y \in \mathbb{R}$):

$$(4.8) \quad \delta_{|x| \leq y} = \sup_{\substack{\alpha \in \mathbb{R}^d, \beta \in \mathbb{R}, \\ |\alpha| \leq \beta}} \alpha x - \beta y.$$

Applying this, (3.19) is equivalent to adding the terms $-\delta_{|\phi^x| \leq a}$, or

$$(4.9) \quad \inf_{\alpha, \beta} \sum_{x \in \Omega} \sum_{i=1}^k \sum_{0 \leq j < n_i} -\alpha_i^j(x) \phi_i^{x,j}(x) + \beta_i^j(x) a_i(x),$$

to the energy, with the constraints $|\alpha_i^j(x)| \leq \beta_i^j(x)$ for all x , i , and j . The optimization is then also performed over these additional dual variables α and β , with $\alpha_i^j(x) \in \mathbb{R}^d$ and $\beta_i^j(x) \in \mathbb{R}$

for all x , i , and j . This way, for a only the constraints (3.18) remain. The corresponding projection is easily done by clipping the absolute value of a . The projection of α, β can also be easily computed; see section B.1.

For the data term part, we have the constraints (3.22) if g is separable, and (3.16) for the general case. The projection onto (3.22) is straightforward. For general data terms, there are two strategies for handling (3.16), depending on available memory.

As a first approach, we dualize every constraint of (3.16) by adding specific additional energy terms. We use the convex dualization (for $x, q \in \mathbb{R}$)

$$(4.10) \quad \delta_{x \geq q} = \sup_{\mu \leq 0} \mu(x - q)$$

and add the terms $-\delta_{\sum_i \phi_i^t \geq -g}$ to the energy, i.e.,

$$(4.11) \quad \inf_{\mu} \sum_{x \in \Omega} \sum_{j \in \Lambda} -\mu^j(x) \left(\sum_{i=1}^k \phi_i^{t,j_i}(x) + g^j(x) \right)$$

with the constraints $\mu^j(x) \leq 0$ for all $x \in \Omega$ and $j \in \Lambda$. The prox-operator for μ is local for every $x \in \Omega$ and $j \in \Lambda$:

$$(4.12) \quad \arg \min_{\mu \leq 0} \frac{(\mu - \mu^0)^2}{2\tau} - \mu g^j(x)$$

for some $\tau > 0$ and $\mu^0 \in \mathbb{R}$. The solution can be easily computed giving $\mu = \min(0, \mu^0 + \tau g^j(x))$. Dualization (4.11) introduces additional dual variables μ into the *global* energy. Since these are $|\Omega| \prod_{i=1}^n n_i$ individual variables, this approach is quite costly memorywise. In comparison, the number of all other variables scales only as $|\Omega| \sum_{i=1}^n n_i$.

The second approach is to solve the projection

$$(4.13) \quad \arg \min_{\phi^t \text{ s.t. (3.16)}} \sum_{x \in \Omega} \sum_{i=1}^k \sum_{0 \leq j < n_i} \left(\phi_i^{t,j}(x) - (\phi_i^{t,j}(x))^0 \right)^2$$

directly as a subproblem. This does not require any additional variables, but the projection must then be performed after each iteration of the primal-dual algorithm. Thus, this trades off a reduction of memory requirements for an increased run time (many times over). The projection (4.13) can be solved by introducing variables μ and terms (4.11) to the *local* energy (4.13), with sup instead of inf and with μ instead of $-\mu$. This can be done sequentially pixel for pixel, requiring only $\prod_{i=1}^k n_i$ additional variables. To accelerate this process, one can process chunks of $N_c \geq 1$ pixels in parallel, where $N_c \leq |\Omega|$ is chosen as large as possible to fit into the available memory.

We employ the same primal-dual algorithm [12] for the projection subproblem (4.13). Specifically, because of the quadratic terms in ϕ^t we can use Algorithm 2 of [12], which has an accelerated $\mathcal{O}(1/N_{\text{iter}}^2)$ convergence rate. Since ϕ^t does not change much between two outer iterations, a small number of inner iterations can be chosen, e.g., $N_{\text{iter}} = 10$.

4.2.2. Huber-TV with l^2 -coupling. Here we have the constraints (3.28), as well as (3.32) for separable data terms g and (3.29)–(3.31) for general g . Implementation of (3.32) is done exactly as for the TV case above. For separable data terms, one has to project onto the remaining constraint (3.32). This is a projection onto a parabola. Its computation leads to a cubic equation which can be solved in closed form. This is detailed in section B.2.

For general data terms, by (3.29) ϕ^t is replaced by two independent dual variables γ_i and $\bar{\phi}_i$. The constraints (3.30) are implemented in exactly the same way as for the case of TV above. Finally, the projection onto (3.31) is again the projection onto a parabola and can be done quickly and in closed form.

4.2.3. Total variation for general norms. For special cases of κ -norms $\|\cdot\|$ (3.38), we have the simplified constraints (3.39) and (3.40) together with (3.36). Implementation for this case is the same as previously for TV with the Euclidean norm, again using the dualization (4.9). The only difference is that (3.18) is now replaced by (3.39); i.e., a is constrained in the $\frac{\kappa}{\kappa-1}$ -norm instead of the 2-norm. The most interesting cases for κ are 1, 2, and ∞ . The projection onto the corresponding dual ball (3.39) can be done in a straightforward way for each of these cases. Moreover, because of (3.40) the a_i are always nonnegative, so (3.39) can be replaced by the more simple inequality $\sum_{i=1}^k a_i \leq 1$ for $\kappa = \infty$.

For general norms we have the constraints (3.37). In general they cannot be decoupled to linearly many constraints. To implement (3.37) we can dualize every constraint using the convex dualization (for $x \in \mathbb{R}^{d \times k}$)

$$(4.14) \quad \delta_{\|x\|_* \leq \lambda} = \sup_{\eta \in \mathbb{R}^{d \times k}} \eta x - \lambda \|\eta\|.$$

We add the terms $-\delta_{\|\phi^x\|_* \leq \lambda}$ to the energy, i.e.,

$$(4.15) \quad \inf_{\eta} \sum_{x \in \Omega} \sum_{j \in \Lambda} \left\{ - \sum_{i=1}^k \eta_i^j(x) \phi_i^{x, j_i}(x) + \lambda \|\eta^j(x)\| \right\}.$$

This introduces additional primal variables $\eta^j(x) \in \mathbb{R}^{d \times k}$ for every $x \in \Omega$ and $j \in \Lambda$ into the overall optimization. The prox-operator is local for each $x \in \Omega$ and $j \in \Lambda$:

$$(4.16) \quad \arg \min_{\eta \in \mathbb{R}^{d \times k}} \frac{(\eta - \eta^0)^2}{2\tau} + \lambda \|\eta\|.$$

Using Moreau's identity [36], the solution is given by

$$(4.17) \quad \eta = \eta^0 - \pi_{\|\cdot\|_* \leq \tau\lambda}(\eta^0),$$

where $\pi_{\|\cdot\|_* \leq \tau\lambda}$ is the projection onto the scaled dual unit ball $\{x \in \mathbb{R}^{d \times k} \mid \|x\|_* \leq \tau\lambda\}$. In the case of TV_J , the dual norm $\|\cdot\|_*$ is the nuclear norm (3.42). We refer the reader to [21] for a detailed description of how to perform the corresponding projection.

The dualization (4.15) requires a considerable number of additional variables, namely $kd|\Omega| \prod_{i=1}^k n_i$ for η . The same discussion about memory reduction applies here as previously with (4.11) for the case of nonseparable data terms. If there is enough available memory, we can dualize every constraint of (3.37). Otherwise we can handle them locally by a subproblem, computing a projection after every outer iteration of the primal-dual algorithm. Again, chunks of pixels can be processed in parallel to accelerate the process.

4.2.4. General regularizers. In the general case we have the constraints (2.5). These are $n_1 \cdots n_k$ constraints, after the discretization of the range spaces, for each pixel $x \in \Omega$. Beside the *TV* and *Huber-TV* cases, in general they cannot be decoupled into linearly many constraints. To implement the general case, we can dualize every constraint using the convex dualization (for $x \in \mathbb{R}^{d \times k}$, $y \in \mathbb{R}$)

$$(4.18) \quad \delta_{y \geq f^*(x)} = \sup_{\eta \in \mathbb{R}^{d \times k}, \mu \in \mathbb{R}} (\mu y + \eta x) - \delta_{y \geq f^*(x)}^*(\eta, \mu).$$

We add the terms $-\delta_{\sum_i \phi_i^t + g \geq f^*(\phi^x)}$ to the energy, i.e.,

$$(4.19) \quad \inf_{\eta, \mu} \sum_{x \in \Omega} \sum_{j \in \Lambda} \left\{ -\mu^j(x) \left(\sum_{i=1}^k \phi_i^{t, j_i}(x) + g^j(x) \right) - \sum_{i=1}^k \eta_i^j(x) \phi_i^{x, j_i}(x) + \delta_{y \geq f^*(x)}^*(\eta^j(x), \mu^j(x)) \right\}.$$

This introduces additional primal variables $\eta^j(x) \in \mathbb{R}^{d \times k}$ and $\mu^j(x) \in \mathbb{R}$ for every $x \in \Omega$ and $j \in \Lambda$. Note that both (4.11) and (4.15) are special cases of the above general dualization (4.19).

The prox-operator is local for each $x \in \Omega$ and $j \in \Lambda$:

$$(4.20) \quad \arg \min_{\eta \in \mathbb{R}^{d \times k}, \mu \in \mathbb{R}} \frac{(\eta - \eta^0)^2}{2\tau} + \frac{(\mu - \mu^0)^2}{2\tau} - \mu g^j(x) + \delta_{y \geq f^*(x)}^*(\eta, \mu).$$

Define $\hat{\mu}^0 := \mu^0 + \tau g^j(x)$. Using Moreau's identity [36] again, the solution is then

$$(4.21) \quad (\eta, \mu) = (\eta^0, \hat{\mu}^0) - \tau \pi_{y \geq f^*(x)} \left(\frac{\eta^0}{\tau}, \frac{\hat{\mu}^0}{\tau} \right),$$

where $\pi_{y \geq f^*(x)}(x^0, y^0)$ is the projection onto $\{(x, y) \in \mathbb{R}^{d \times k} \times \mathbb{R} \mid y \geq f^*(x)\}$.

Note that the advantage of the implementation framework (4.19) is that it is the same independent of the interaction terms f , i.e., of the regularizer. The only place where f enters the computation is the projection (4.21). Once the framework is implemented, it can be easily adapted for different regularizers by merely replacing the projection.

As previously with total variation for general norms, the dualization (4.19) also requires many additional variables. The same memory reduction strategy can be applied, at the expense of a higher run time.

4.2.5. Huber-TV for the spectral norm. The regularizer (3.47) is implemented using the general scheme presented above. By formula (3.48), in (4.21) we need to project onto

$$(4.22) \quad \left\{ (x, y) \in \mathbb{R}^{d \times k} \times \mathbb{R} \mid y \geq \frac{\varepsilon}{2\lambda} \|x\|_*^2, \|x\|_* \leq \lambda \right\}$$

with the nuclear norm $\|\cdot\|_*$ in (3.42). Writing out the norm of x in terms of the singular values, this projection can be done in closed form. The common case of two-dimensional color images, i.e., $d = 2$ and $k = 3$, is detailed in section B.3.

4.2.6. Lipschitz regularizer with l^2 -coupling. The Lipschitz regularizer (3.47) is also implemented by the general scheme. By (3.48), one has to project onto

$$\{(x, y) \mid y \geq \lambda \|x\|_2\}.$$

This projection can be done in closed form as detailed in section B.1.

4.3. Memory requirements. Here we give a summary of the overall memory required to implement the proposed convexification approach with coupling regularizers such as TV and Huber- TV . Depending on the regularizer and on the separability of the data term there are basically two kinds of variables and constants. The first kind, such as the basic variables v and ϕ , requires a *linearly* scaling amount of memory in terms of image size and the discretization levels, $\mathcal{O}(|\Omega| \sum_{i=1}^k n_i)$, and is thus relatively cheap to store. For the second kind, e.g., for variables η and μ , the memory scales *exponentially* as $\mathcal{O}(|\Omega| \prod_{i=1}^k n_i)$.

Table 1

Required number of floating point numbers for energy (4.2). Memory is proportional to these amounts with 4 bytes per float.

Variable or constant	Floating point numbers
v_i^j	$ \Omega \sum n_i$
ϕ_i^j	$(d+1) \Omega \sum n_i$
g_i^j (if g separable)	$ \Omega \sum n_i$
g^j (if g nonseparable)	$ \Omega \prod n_i$

Table 2

Additional float numbers depending on the employed regularizer. Overall memory for isotropic TV_{l^2} and Huber- TV_{l^2} scales linearly with the range discretization for separable data terms. It is also in the same range as for the case of nonisotropic TV_{l^1} and Huber- TV_{l^1} .

Regularizer	Additional variables	Floating point numbers
TV_{l^1} /Huber- TV_{l^1}	μ^j (if g nonseparable)	$ \Omega \prod n_i$
TV_{l^2} /Huber- TV_{l^2} / TV_{l^κ} /Huber- TV_{l^κ}	α_i^j β_i^j γ_i^j (if g nonseparable, for Huber) μ^j (if g nonseparable)	$kd \Omega \sum n_i$ $ \Omega \sum n_i$ $ \Omega \sum n_i$ $ \Omega \prod n_i$
TV_J / $TV_{\ \cdot\ }$	η_i^j μ^j (if g nonseparable)	$kd \Omega \prod n_i$ $ \Omega \prod n_i$
Huber- TV_J / Lipschitz $_{l^2}$ /general	η_i^j μ^j	$kd \Omega \prod n_i$ $ \Omega \prod n_i$

The number of floating point numbers, and thus the amounts of memory, required by the variables and constants appearing in the energy (4.2) are shown in Table 1. Additional memory is needed to decouple and dualize the nonlocal constraints in the set \mathcal{K} in (2.11) depending on the type of employed regularizer, as shown in Table 2. First, observe that in case of separable data terms, the overall memory for TV and Huber- TV with l^2 -coupling scales *linearly*. Also, it is always in the same range as in the uncoupled l^1 -case, no matter whether the data term is separable or not. Thus, for TV and Huber- TV regularized optimization

problems with nonconvex data terms, the proposed framework offers the advantage of *channel coupling* at nearly the same costs as without any coupling.

For general data terms, as well as for general regularizers, additional memory of size proportional to $|\Omega| \prod n_i$ is needed if we use the global dualization in each case. This strategy should be used whenever possible as this gives the best run times. If there is not enough memory, as described in section 4.2 we can revert to *local* projection subproblems instead. The memory is then $\mathcal{O}(N_C \prod n_i)$, and N_C is chosen appropriately as big as possible for the overall problem to fit on the GPU. This approach, however, comes at a high penalty in terms of run time (larger by factor of 10).

Another costly constant is the data term g^j (if it is nonseparable). During local projections it is advisable to store the whole array on the GPU. If this is not possible, g can be stored on the CPU side, and one can copy the required parts for the currently processed chunk of pixels to the GPU. Since CPU-to-GPU memory copies are rather slow, this increases the run time by a factor of 5–10. Thus, if the data term has a simple structure, one should compute it on the fly as needed.

Finally, the whole approach can also be parallelized on multiple GPUs. To do this, one subdivides the image domain Ω into equal parts (e.g., horizontal stripes) and lets each GPU perform the described computations for one such part. After each iteration, the overlap regions need to be copied between the GPUs, which comes at virtually no cost since the overlap regions are rather small. Beside decreasing the run time, another compelling advantage of a multi-GPU setting is the much higher overall amount of memory available, allowing one to solve larger problems. This is especially interesting, e.g., for optical flow computation, as will be detailed in section 5.1.

5. Experiments. In the following we demonstrate the usage of the proposed convex relaxations on several vectorial imaging problems. All algorithms were parallelized using the CUDA framework on three NVIDIA GTX 680 GPUs. The number of iterations for the primal-dual algorithm was chosen appropriately so that the solution remained visually stable and did not change anymore, which is usually the case after 1000–5000 iterations.

5.1. Optical flow. The task of optical flow estimation, or image matching, is to find point correspondences between two images. Given two color images $I_1 : \Omega_1 \rightarrow \mathbb{R}^3$, $I_2 : \Omega_2 \rightarrow \mathbb{R}^3$ which show the same scene, but taken from different viewpoints or at different times, one seeks a function $u : \Omega_1 \rightarrow \mathbb{R}^2$ such that $I_1(x) = I_2(x + u(x))$. In practice, this relation will not be satisfiable exactly, and therefore one seeks u as a minimizer of the energy functional

$$(5.1) \quad E(u) = \int_{\Omega_1} g(x, u(x)) dx + TV(u),$$

where the data term is given, e.g., by color differences

$$(5.2) \quad g(x, t) = \|I_1(x) - I_2(x + t)\|_2.$$

One can also sum up the image differences in a small window around x , e.g., with a 1-pixel radius. A regularization term, such as $TV(u)$ here, is needed to ensure a spatial coherence of the flow, as the local estimates by minimizing $g(x, \cdot)$ pointwise in each point x may differ

considerably. The *separable* TV_{l_1} regularizer is often employed in optical flow computations. With our framework, it becomes possible to also use the *coupled* TV_{l_2} , which is rotationally invariant and thus a more favorable choice. Note that some literature uses the term “optical flow” to refer to a *linearized* version of the model (5.1), which is then already convex. However, the linearization is only a technical means of coping with nonconvexity and does not describe the image matching problem correctly, e.g., for large scale flow.

Comparison of coupled TV_{l_2} with uncoupled TV_{l_1} . Figure 1 shows an optical flow computation on a real world image taken from the Middlebury optical flow dataset [4].² There is no need for much smoothing in this case; therefore the results for TV_{l_1} and TV_{l_2} are visually almost the same. The 640×480 image with a 50×50 label space requires 68.16 seconds for TV_{l_1} and 71.04 seconds for TV_{l_2} . The most time is spent for the convexification scheme (4.11) of the nonseparable data term (5.2), while the updates due to TV_{l_1}/TV_{l_2} are negligible. GPU memory required for TV_{l_1} and TV_{l_2} is 6918 MB, respectively 7275 MB.

A comparison with the ground truth optical flow is shown in Figure 2. In terms of common error measures, the TV_{l_2} regularization yields a more accurate flow than TV_{l_1} for the same data term. In this example, for the 584×388 images we used a 22×22 label space.

The difference between TV_{l_1} and TV_{l_2} becomes more apparent for larger weights λ . Since TV_{l_1} is not rotationally invariant, the same optical flow may be penalized differently if the underlying images—and with them the flow—are rotated by a certain angle. This is stated in the following proposition.

Proposition 5.1 (TV_{l_1} is not rotationally invariant). *Set $\Omega = \{x \in \mathbb{R}^2 \mid |x| < 1\}$. For fixed angles $\theta \in \mathbb{R}$ define the vector field $v_\theta : \Omega \rightarrow \mathbb{R}^2$ by*

$$(5.3) \quad v_\theta(x) = \begin{cases} e_\theta & \text{if } |x| \leq r, \\ 0 & \text{else} \end{cases}$$

for all $x \in \Omega$ with $e_\theta := (\cos \theta, \sin \theta)$. Then

$$(5.4) \quad TV_{l_1}(v_\theta) = 2\pi r (|\cos \theta| + |\sin \theta|).$$

Proof. This follows directly from the representation

$$\begin{aligned} TV_{l_1}(v_\theta) &= TV((v_\theta)_1) + TV((v_\theta)_2) = TV(\chi_{B_r(0)}) |(e_\theta)_1| + TV(\chi_{B_r(0)}) |(e_\theta)_2| \\ &= 2\pi r (|\cos \theta| + |\sin \theta|), \end{aligned}$$

together with the property of TV that $TV(\chi_A) = \mathcal{H}^{d-1}(A)$ for rectifiable sets $A \subset \Omega$. ■

In contrast, the l^2 -coupled TV_{l_2} -regularizer is rotationally invariant. For the above vector field we would get $TV_{l_2}(v_\theta) = 2\pi r$, which is the same for all θ . This difference is demonstrated in Figure 3. The two circles are moving in essentially the same way, but in different directions. The background motion outside the circles will be nearly zero, as the texture ensures that the zero vector field will have the smallest data term. Since TV_{l_1} penalizes diagonal motions more (by factor $\sqrt{2}$ due to (5.4)) than the ones parallel to the axes, choosing the weight λ in a certain range may result in unequal treatment of the two basically identical motions. This is

²See <http://vision.middlebury.edu/flow>.

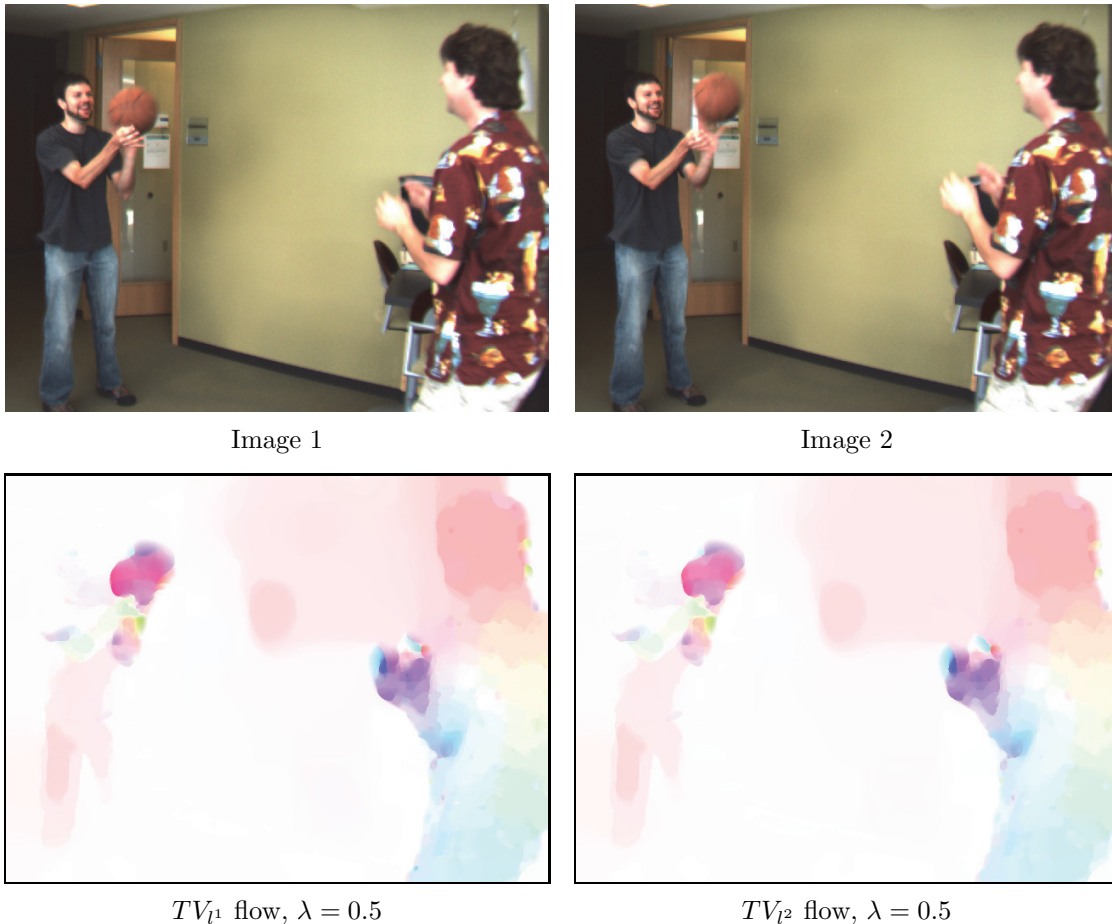


Figure 1. Optic flow. “Basketball” image pair from the Middlebury dataset [4]. If there is no need for much smoothing, as here, the results with TV_{l_1} and TV_{l_2} are visually similar. The difference is more pronounced for larger weights; see Figure 3. Images 640×480 using 50×50 labels.

not the case for TV_{l_2} . For every value of λ the two motions are recognized at the same time. The run time for the 160×120 image with an 80×80 label space is 22.05 seconds for TV_{l_1} and 22.45 seconds for TV_{l_2} , and memory requirements are 1047 MB, respectively, 1080 MB.

Comparison with locally convergent approaches. In the case of a nonseparable data term, such as in the optical flow example, the proposed approach is computationally expensive in terms of memory and run time. In view of this, a natural question is whether this complexity is really necessary and whether one can use *locally* convergent methods instead. An advantage of these methods is that they work in the original image domain and thus require much less memory than the convexification approach.

Their main disadvantage is, of course, their local optimality, which means that only certain flows can be reliably detected. Recent local approaches for optical flow estimation, such as those in [40] and [43], employ some kind of convex approximation to cope with the nonconvexity of the original problem. For instance, the data term is usually linearized around the current solution. Since this is valid only for small motions, a heuristic coarse-to-fine scheme needs to be employed in order to also recognize large scale motions: starting with downscaled

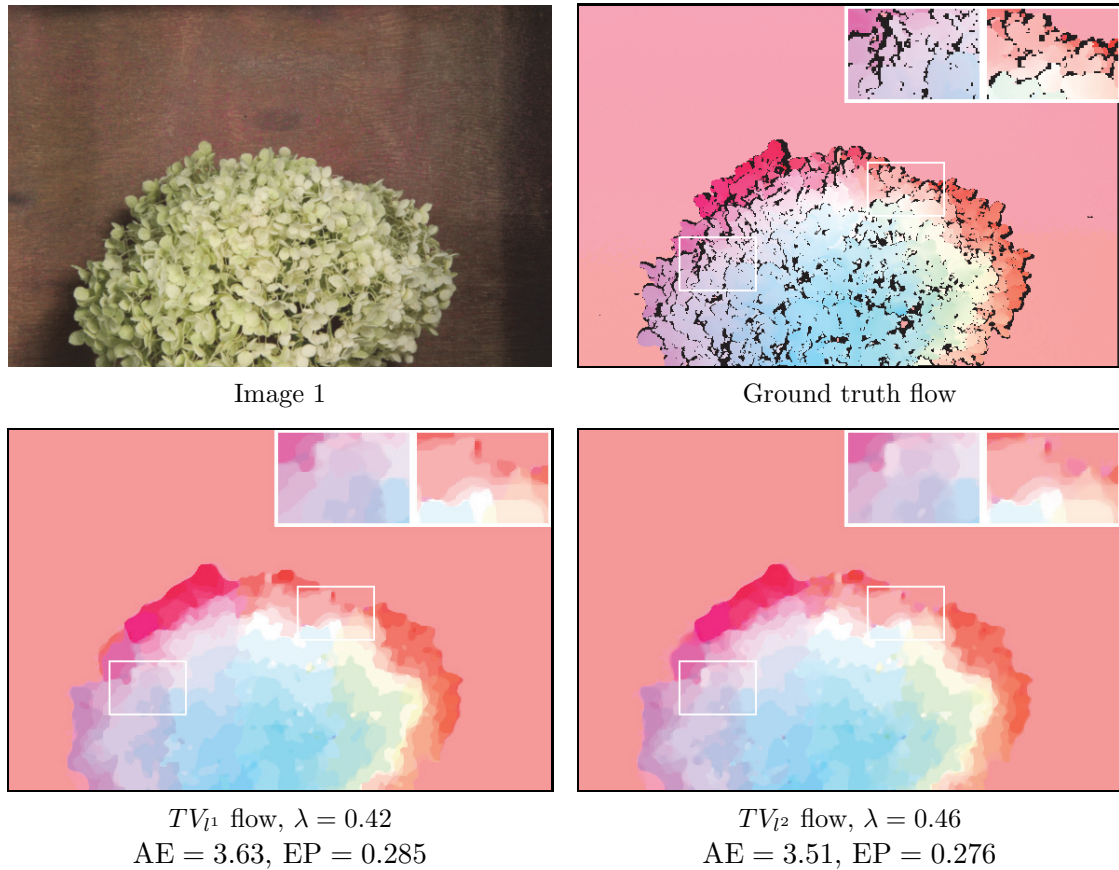


Figure 2. Optic flow for the “Hydrangea” image pair from the Middlebury dataset [4]. Comparing to ground truth, the flow using TV_{l2} has smaller error measures (average angular error (AE), average endpoint error (EP)) than with TV_{l1} . Image size 584×388 using 22×22 labels. The run time for TV_{l1} and TV_{l2} is 13.95, respectively 15.19, seconds, requiring 1182, respectively 1299, MB of memory.

images the optical flow is transferred to and iteratively refined on the next higher resolution.

As a result, these methods generally cannot detect large scale motion of small scale objects. This is demonstrated in Figure 4: the ball on the left side of the images is not recognized for any choice of the parameters. In contrast, our convex approach handles the original, nonlinearized data term and avoids this limitation. We used the publicly available source code for [40],³ respectively binaries for [43],⁴ for the experiments. The CPU run times for [40, 43] (no GPU versions are available) are comparable with the GPU run times for the convexification.

On the other hand, for small scale motion, state-of-the-art local methods tend to outperform the convexification method in terms of flow quality; see the comparison in [22] for the case of TV_{l1} . However, this comes at the cost of increased complexity: they usually employ highly engineered data terms and features such as handling of occlusions and illumination changes, combined in a multistage optimization framework. In contrast, we use a very simple

³See <http://ps.is.tuebingen.mpg.de/person/black>.

⁴See <http://www.cse.cuhk.edu.hk/~leojia/projects/flow>.

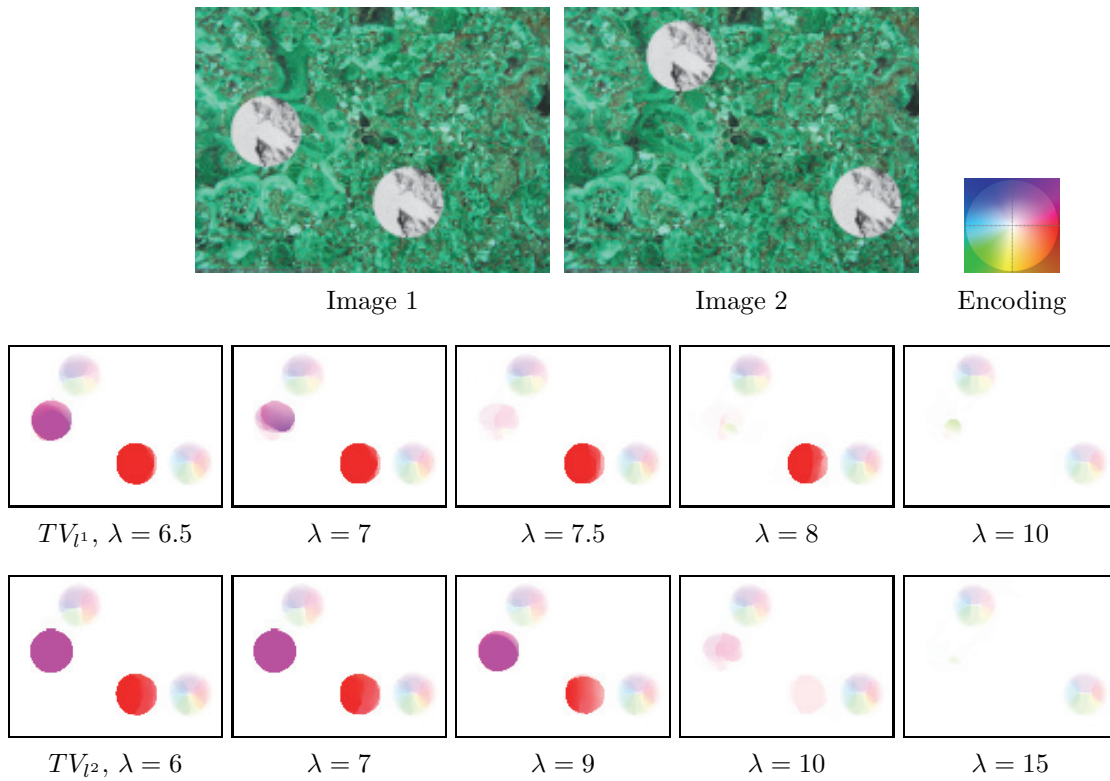


Figure 3. Rotational invariance. The synthetic image pair consists of two identical circles, traveling along the x -axis and diagonally. Although the two motions are equal except for direction, TV_{l^1} prefers grid aligned motions. The diagonal one disappears first when the regularizer weight λ increases. In contrast, the rotationally invariant TV_{l^2} handles the two motions equally. The right and top artifacts arise because of occlusions, which are not modeled by the simple data term (5.2). 160×120 images using 80×80 labels.

data term since the focus of this paper lies in the regularizer part. With more elaborate data terms the convexification method is likely to achieve competitive results, which is left for future work.

5.2. Color denoising. A natural application of our framework is to apply the coupled regularizers to denoise color images $f : \Omega \rightarrow \mathbb{R}^3$. We seek an image $u : \Omega \rightarrow \mathbb{R}^3$ minimizing

$$(5.5) \quad E(u) = \int_{\Omega} g(x, u(x)) dx + TV(u),$$

where TV is either TV_{l^2} or TV_{l^1} . Note that our framework allows possibly *nonconvex* data terms g . Specifically, we choose the truncated quadratic differences, or the truncated linear ones:

$$(5.6) \quad g_2(x, t) = \sum_{i=1}^3 \min \{T, (t_i - f_i(x))^2\}, \quad g_1(x, t) = \sum_{i=1}^3 \min \{T, |t_i - f_i(x)|\}$$

with a fixed threshold $0 < T \leq 1$. Since g is separable in each case, the model allows for a fast implementation with only linearly many constraints, as described in section 3.3. Note that the

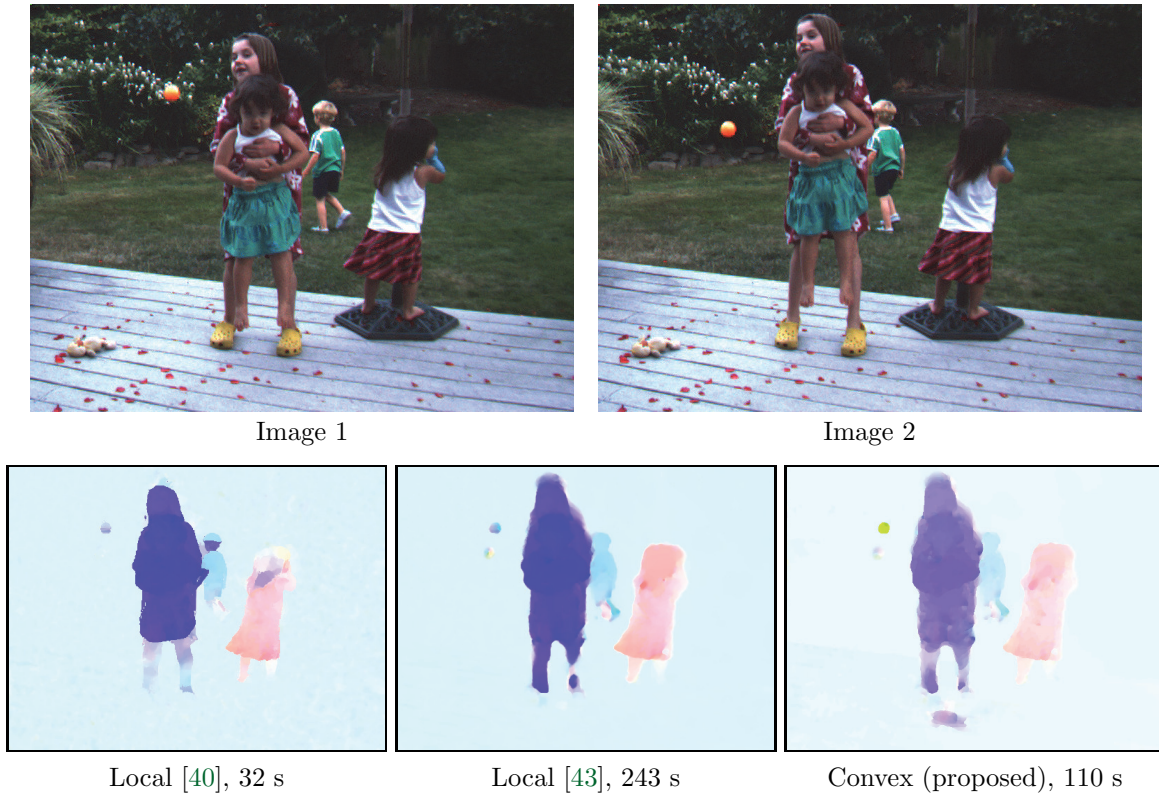


Figure 4. Comparison with local approaches, optical flow between frames 10 and 12 of the “Backyard” sequence from the Middlebury dataset [4]. Due to linearization and coarse-to-fine schemes, local approaches usually fail to correctly recognize large motion of small scale objects, such as the orange ball (moving downwards, green color in the flow encoding in Figure 3). In contrast, convex approaches, such as the proposed approach, can use the original nonconvex data term and are able to recognize such motion. Image size 544×408 using 75×75 labels. Run time for local methods is on the CPU (no GPU versions are available), for convexification on the GPU.

data term is convex for $T = 1$, and nonconvex for $T < 1$. More precisely, e.g., for g_2 , it becomes nonconvex once T has an effect on g_2 , i.e., for $T < \max_{x \in \Omega, 1 \leq i \leq k} \max(f_i(x), 1 - f_i(x))^2$.

To compare the denoising capabilities of TV_{l_2} and TV_{l_1} , we perform an experiment where we add a certain amount of noise to several given clean images, and then we try to recover these images by means of TV denoising. We compute the maximal achievable peak signal-to-noise ratios (PSNRs), defined by

$$(5.7) \quad PSNR(u, f) = 10 \log_{10} \frac{3}{\sum_{x \in \Omega} \|u(x) - f(x)\|_2^2},$$

where we take the Euclidean distance of the RGB values $u(x), f(x) \in \mathbb{R}^3$. Higher PSNR values indicate a better quality of the reconstruction u .

As seen in Figure 5 in the case of Gaussian noise and a truncated quadratic data term g_2 in (5.6), TV_{l_2} systematically leads to a better denoising quality, which is indicated by the higher PSNR values. We have chosen the weights λ to maximize these values in each case. The best denoising results are achieved using the *coupling* TV_{l_2} -regularizer in combination

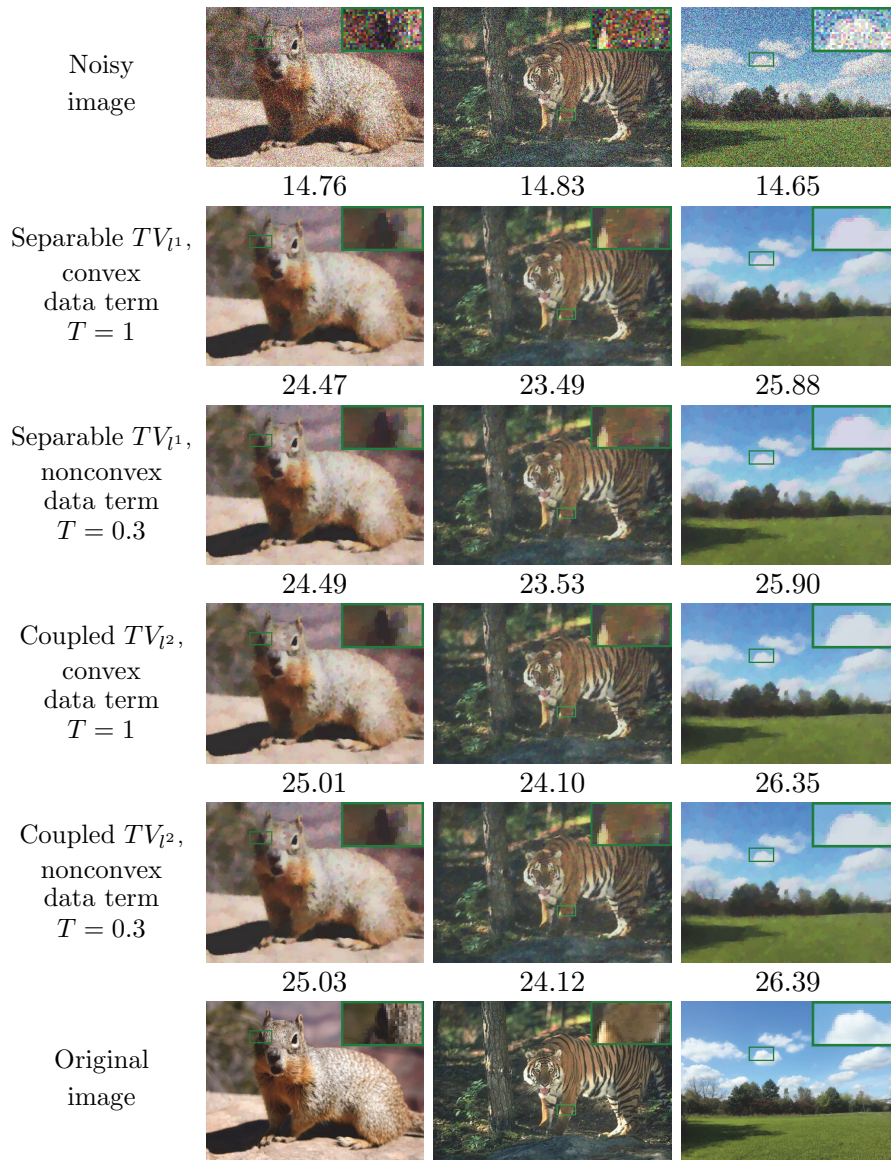





Figure 5. Denoising. Input images were degraded by additive channelwise Gaussian noise with standard deviation $\sigma = 0.2$. For each image and regularizer, the optimal weight λ was chosen manually to maximize the PSNR value. The coupled TV_{l_2} -regularizer leads to systematically higher PSNR values and thus denoising quality. Using a nonconvex data term, g_2 in (5.6), reconstructions of a higher quality can be achieved. Label space $32 \times 32 \times 32$.

with the *nonconvex* truncated data term with $T = 0.3$. Note that minimizing energies with nonconvex data terms *and* coupled regularizers becomes possible only with our proposed convex relaxation framework. Approach [39] allows nonconvex data terms, but the regularizer must be separable, and [21] allows coupling regularizers, but the data term must be convex. The run times are independent of the data term and are listed in Table 3.

Figure 6 further compares the denoising capabilities of the different regularizers for a fixed

Table 3

Run times (s) for Figure 5. Label space $32 \times 32 \times 32$.

Image	TV_{l_1}	TV_{l_2}
 368×270	2.04	2.85
 481×321	3.51	5.16
 326×244	2.01	2.79

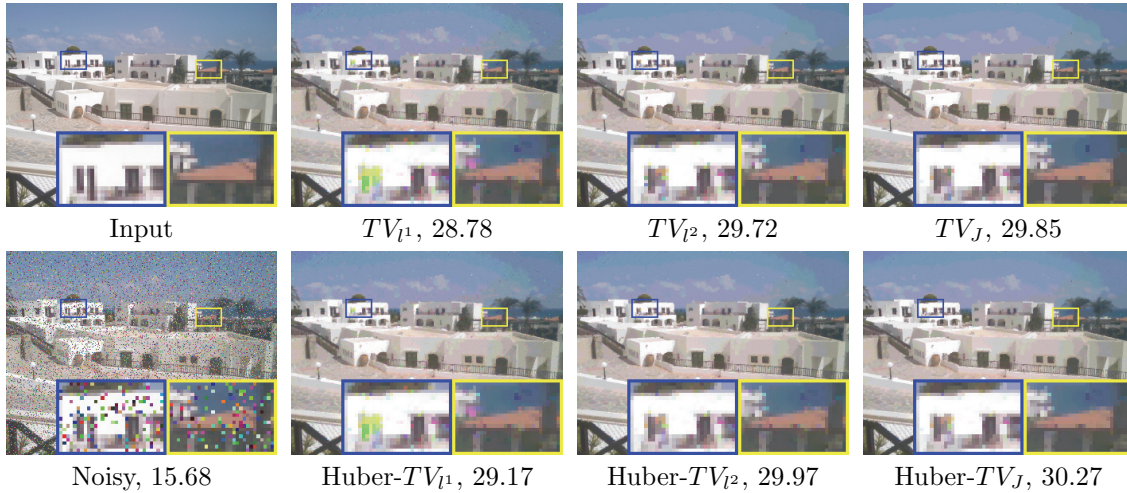


Figure 6. Regularizer comparison. Input image has been degraded by impulse noise (20% of pixels set to random values). For each regularizer, the denoising model with a nonconvex data term (g_1 in (5.6), $T = 0.2$) was solved with manually chosen optimal parameters λ and ε . The coupled regularizers TV_{l_2} and TV_J outperform the separable TV_{l_1} in denoising quality in terms of the PSNR values. The Huber versions consistently improve the result. Label space $16 \times 16 \times 16$.

data term. In the input image 20% of all pixels were set to random RGB values. For this case we use the robust truncated linear data term g_1 in (5.6) with $T = 0.2$. The coupling regularizers TV_{l_2} and TV_J become possible only with our approach and lead to better denoising results than with the separable regularizer TV_{l_1} . While TV_J yields the best results, it is also the most costly regularizer memorywise. Switching to the Huber-regularized versions of the total variations has a positive effect on the denoising quality in each case. This is because it allows a smoother variation of the solution, which is then able to more closely resemble the natural image. The best result is achieved with Huber- TV_J , which, however, requires the general costly implementation scheme (4.19). The run times for this experiment, as well as the GPU memory requirements, are given in Table 4.

Table 4

Run times (s) and required GPU memory (MB) for Figure 6. Image resolution 341×256 , label space $16 \times 16 \times 16$.

	TV_{l_1}	Huber- TV_{l_1}	TV_{l_2}	Huber- TV_{l_2}	TV_J	Huber- TV_J
Run time	1.04	1.07	1.42	1.58	126	329
Memory	131	131	181	181	8314	9710

5.3. Color inpainting. Another useful application is inpainting. When a color image $f : \Omega \rightarrow \mathbb{R}^3$ has some missing parts in an area $A \subset \Omega$, the task is to inpaint the missing colors in A using the available information from $\Omega \setminus A$. This can be formulated as an energy minimization problem

$$(5.8) \quad E(u) = \int_{\Omega} g(x, u(x)) \, dx + R(u).$$

The data term ensures that $u = f$ in $\Omega \setminus A$ and lets u vary freely in A . The resulting inpainted values in A will then be interpolated from the surrounding values in a way such that the regularizer value $R(u)$ is minimal. We set

$$(5.9) \quad g(x, t) = \sum_{i=1}^k \begin{cases} \min \{T, (t_i - f_i(x))^2\} & \text{if } x \in \Omega \setminus A, \\ 0 & \text{if } x \in A. \end{cases}$$

We use $R = TV_{l_1}$ and $R = TV_{l_2}$ with a very small weight $\lambda = 0.001$ so that the given values in $\Omega \setminus A$ remain unchanged.

Figure 7 shows the removal of text on top of the image ($T = 0.3$). Comparing the obtained inpainted values with the known true values, we can compute the corresponding PSNR values. The coupled TV_{l_2} achieves a slightly higher value than the separable TV_{l_1} . For the 512×384 image using $32 \times 32 \times 32$ labels, the run time for TV_{l_1} is 16.75 seconds, while that for TV_{l_2} is 24.33 seconds.

6. Conclusion. We introduced convex relaxations for nonconvex variational models on vector-valued functions which are computationally tractable and in a certain sense as tight as possible. In contrast to existing relaxations of vectorial multilabel problems, we can handle the combination of nonconvex data terms with coupled regularizers such as TV_{l_2} and Huber- TV_{l_2} . The key idea is to consider a collection of hyperplanes with a relaxation that takes into account the entire functional rather than treating data term and regularizers separately. We provided a theoretical analysis, detailed the implementations for different functionals, and presented run time and memory requirements. In particular, for the isotropic TV_{l_2} and Huber- TV_{l_2} regularizers we proposed efficient equivalent constraint reformulations. This allows one to account for channel coupling with negligible overhead in memory and run time compared to the uncoupled versions of these regularizers. In numerous experiments on denoising, optical flow, and inpainting, we experimentally demonstrated that coupled l^2 -regularizers give systematic improvements regarding rotational invariance and quantitative performance.

Appendix A. Proof of Lemma 2.3. Here we give a proof of Lemma 2.3 in case u is a function with bounded variation.

Proof. When $u \in BV(\Omega; \Gamma^k)$, its derivative Du has an absolutely continuous part with respect to the Lebesgue measure (traditionally denoted by $\nabla u(x) \, dx$) and a singular part $D^s u$ (consisting of a ‘‘Cantor’’ part and a ‘‘Jump’’ part; see [2]). Then, the integral in (2.19) is classically defined [16] as

$$\int_{\Omega} f(x, Du) = \int_{\Omega} f(x, \nabla u(x)) \, dx + \int_{\Omega} f_{\infty}(x, \nu_u(x)) |D^s u|(x),$$

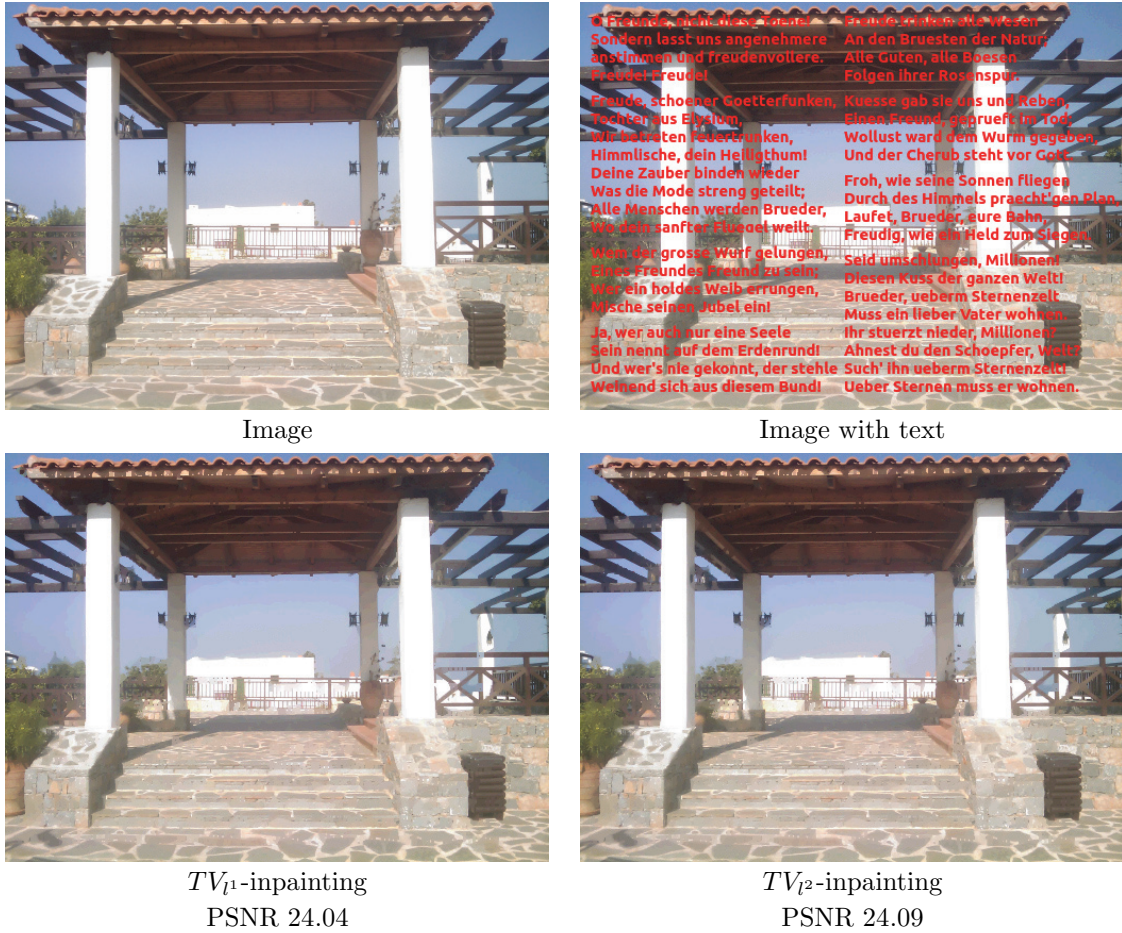


Figure 7. *Inpainting. Text has been placed on top of the input image (top row). This text is to be removed through inpainting, i.e., by filling in suitable colors which are consistent with the surrounding pixels. The inpainted image values are computed by minimizing their TV_{l_1} or TV_{l_2} energy (bottom row). The coupled TV_{l_2} achieves a slightly higher PSNR value than TV_{l_1} , compared with the known solution. Label space $32 \times 32 \times 32$.*

where $\nu_u(x) = \frac{D^s u}{|D^s u|}$ is the (unit) Radon–Besicovitch derivative of the measure $D^s u$ with respect to its total variation, and $f_\infty(x, p) = \lim_{t \rightarrow \infty} f(x, tp)/t$ is the convex, one-homogeneous recession function of f at ∞ . See [16, 5] for details. Observe that since we have assumed $f(x, 0) = \min_p f(x, p) = 0$, we have $f(x, sp)/s \leq f(x, tp)/t$ if $s \leq t$, so that

$$f_\infty(x, p) = \sup_{t>0} \frac{1}{t} f(x, tp)$$

(and f_∞ is l.s.c. as a supremum of l.s.c. functions).

We will admit the following fact: the function $u \mapsto \int_\Omega f(x, Du)$ is l.s.c. in $L^1(\Omega; \mathbb{R}^k)$, that is, if $u_n \rightarrow u$ in L^1 , then

$$\int_\Omega f(x, Du) \leq \liminf_{n \rightarrow \infty} \int_\Omega f(x, Du_n);$$

see, for instance, [15, 6, 5]. Here the (semi)continuity of f is important, whereas in the $W^{1,1}$ case, f could be merely measurable in the first variable.

We first remark that we may assume that there exists $L > 0$ with

$$(A.1) \quad f(x, p) \leq L|p|.$$

Indeed, it is always possible to replace $f(x, p)$ with the inf-convolution $f_L(x, p) := \min_q f(x, q) + L|p - q|$, which satisfies (A.1) (since $f(x, 0) = 0$) and is such that the union for $L > 0$ of the corresponding sets \mathcal{K}_0 is the set \mathcal{K}_0 of f . The conclusion easily follows.

Next, we also assume that f is uniformly continuous in x (for p bounded), but it is a bit more complicated to explain why. The idea is to define, for $\lambda > 0$,

$$f_\lambda(x, p) = \min_{y \in \Omega} \lambda|y - x||p| + f(y, p),$$

which satisfies

$$|f_\lambda(x, p) - f_\lambda(y, p)| \leq \lambda|x - y||p|$$

and $\sup_{\lambda > 0} f_\lambda(x, p) = f(x, p)$. However, this new f is not convex in p , and one must show that its convex envelope $f_\lambda^{**}(x, p)$ (with respect to the second variable) also enjoys these properties. Given any $x, y \in \Omega$, $p \in \mathbb{R}^{d \times k}$, and $\varepsilon > 0$, since $\dim \mathbb{R}^{d \times k} = dk$, there exists $(\theta_i, p_i)_{i=1}^{dk} \subset (\mathbb{R} \times \mathbb{R}^{d \times k})^{dk}$ with $\sum_i \theta_i p_i = p$, $\sum_i \theta_i = 1$, $\theta_i \geq 0$, and such that $f_\lambda^{**}(y, p) \geq \sum_i \theta_i f_\lambda(y, p_i) - \varepsilon$. Hence, using (2.18) and (A.1) (which are trivially still satisfied by f_λ),

$$\begin{aligned} f_\lambda^{**}(x, p) - f_\lambda^{**}(y, p) + \varepsilon &\leq \lambda|x - y| \sum_i \theta_i |p_i| \\ &\leq \lambda|x - y| \sum_i \theta_i \left(1 + \frac{f_\lambda(y, p_i)}{C}\right) \\ &\leq \lambda|x - y| \left(1 + \frac{f_\lambda^{**}(y, p) + \varepsilon}{C}\right) \leq \lambda|x - y| \left(1 + \frac{L}{C}|p| + \frac{\varepsilon}{C}\right) \end{aligned}$$

so that, letting $\varepsilon \rightarrow 0$, we see that f_λ^{**} satisfies

$$(A.2) \quad |f_\lambda^{**}(x, p) - f_\lambda^{**}(y, p)| \leq C'|x - y|(1 + |p|)$$

for some constant C' .

Then, we must check that the convex, l.s.c. functions $\sup_\lambda f_\lambda^{**}(x, p)$ and $f(x, p)$ are the same. If not (thanks to the separation theorem), there exist $a \in \mathbb{R}^{d \times k}$, $b' < b \in \mathbb{R}$, and a point (x, p) such that $f_\lambda^{**}(x, p) \leq a : p + b'$ for all λ , while

$$(A.3) \quad a : q + b \leq f(x, q)$$

for all $q \in \mathbb{R}^{d \times k}$, where “:” denotes the scalar product of matrices. This means that one can find $(\theta_i^\lambda, p_i^\lambda)_{i=0}^{dk} \subset (\mathbb{R} \times \mathbb{R}^{d \times k})^{dk}$ with $\lim_{\lambda \rightarrow \infty} \sum_i \theta_i^\lambda p_i^\lambda = p$ and

$$(A.4) \quad \sum_i \theta_i^\lambda f_\lambda(x, p_i^\lambda) \leq a : p + b'',$$

where $b'' = (b + b')/2$. Up to a subsequence, one has $\theta_i^\lambda \rightarrow \bar{\theta}_i$, and either $p_i^\lambda \rightarrow \bar{p}_i$ (if $i \in I \subset \{0, \dots, dk\}$) or $t_i^\lambda = |p_i^\lambda| \rightarrow \infty$ and $\xi_i^\lambda = p_i^\lambda/t_i^\lambda \rightarrow \bar{\xi}_i$, a unit vector (when $i \notin I$). In the latter case, one introduces y_i^λ such that

$$f_\lambda(x, p_i^\lambda) = \lambda|x - y_i^\lambda||p_i^\lambda| + f(y_i^\lambda, p_i^\lambda),$$

and one observes that, given any $t > 0$, if λ is large enough so that $t_i^\lambda > t$,

$$f_\lambda(x, p_i^\lambda) \geq f\left(y_i^\lambda, t \frac{t_i^\lambda}{t} \xi_i^\lambda\right) \geq \frac{t_i^\lambda}{t} f(y_i^\lambda, t \xi_i^\lambda).$$

Hence, denoting $\rho_i^\lambda = \theta_i^\lambda t_i^\lambda$ for $i \notin I$, by (A.4) one has

$$\sum_{i \in I} \theta_i^\lambda f_\lambda(x, p_i^\lambda) + \sum_{i \notin I} \rho_i^\lambda \frac{1}{t} f(y_i^\lambda, t \xi_i^\lambda) \leq a : p + b'',$$

and since ρ_i^λ must therefore be bounded from above, we may also assume that it converges to some $\bar{\rho}_i$ for each $i \notin I$ (in particular, we must have $\theta_i = 0$ in these cases). Since f is l.s.c., in the limit we obtain

$$\sum_{i \in I} \bar{\theta}_i f(x, \bar{p}_i) + \sum_{i \notin I} \bar{\rho}_i \frac{1}{t} f(x, t \bar{\xi}_i) \leq a : p + b''$$

for any $t > 0$, and it follows from (A.3) that

$$a : \left(\sum_{i \in I} \bar{\theta}_i \bar{p}_i + \sum_{i \notin I} \bar{\rho}_i \bar{\xi}_i \right) + b \left(\sum_{i \in I} \bar{\theta}_i + \sum_{i \notin I} \frac{\bar{\rho}_i}{t} \right) \leq a : p + b''.$$

Sending $t \rightarrow \infty$ and observing that $\sum_{i \in I} \bar{\theta}_i \bar{p}_i + \sum_{i \notin I} \bar{\rho}_i \bar{\xi}_i = p$, we obtain

$$a : p + b \sum_{i \in I} \bar{\theta}_i = a : p + b \leq a : p + b'',$$

where we have used $\theta_i = 0$ if $i \notin I$. This is a contradiction, since $b'' < b$.

It follows that $\sup_\lambda f_\lambda^{**}(x, p) = f(x, p)$, and we easily deduce that also $\sup_\lambda (f_\lambda^{**})_\infty(x, p) = f_\infty(x, p)$. Therefore, for any u with bounded variation, $\sup_\lambda \int_\Omega f_\lambda^{**}(x, Du) = \int_\Omega f(x, Du)$. Moreover the set $\mathcal{K}_0(\lambda)$ of f_λ^{**} is clearly a subset of \mathcal{K}_0 , so that if Lemma 2.3 holds for f_λ^{**} , it will also hold for f .

We observe eventually that it is not restrictive to assume that f is smooth in the p variable (a mollification of $(f(x, p) - \varepsilon)^+$ will provide a smooth, convex function below f , enjoying the same properties as f , and arbitrarily close).

To sum up, we are reduced to the case where f is convex, L -Lipschitz (satisfying (A.1)), and smooth in p , and, moreover, has the spatial regularity (A.2).

Given $u \in BV(\Omega; \Gamma^k)$, we fix $\delta > 0$ and choose a subset $\Omega' \subset\subset \Omega$ such that $\int_\Omega f(x, Du) < \int_{\Omega'} f(x, Du) + \delta$. We let ρ be a symmetric mollifier (convolution kernel) and let $u^\varepsilon = \rho_\varepsilon * u$,

which is well defined in Ω' if $\varepsilon > 0$ is small enough. Moreover, by lower semicontinuity, $\int_{\Omega'} f(x, Du) \leq \liminf_{\varepsilon \rightarrow 0} \int_{\Omega'} f(x, \nabla u^\varepsilon(x)) dx$ so that if ε is small enough,

$$(A.5) \quad \int_{\Omega'} f(x, \nabla u^\varepsilon(x)) dx > \int_{\Omega} f(x, Du) - \delta.$$

Since this new function u^ε is smooth, we can let for all x (we drop the dependence on $(t_i)_{i=1}^k$; i.e., ϕ is defined as the same value for every t)

$$(\phi_i^x(x))_{i=1}^k = \nabla_p f(x, \nabla u^\varepsilon(x))$$

(which is continuous, since by using the convexity one can check that $\nabla_p f$ is continuous) and for each i

$$\phi_i^t(x) = \frac{1}{k} f^*(x, \nabla_p f(x, \nabla u^\varepsilon(x))).$$

The Legendre–Fenchel identity [36]

$$(A.6) \quad f(x, \nabla u^\varepsilon(x)) + f^*(x, \nabla_p f(x, \nabla u^\varepsilon(x))) = (\nabla u^\varepsilon(x)) : (\nabla_p f(x, \nabla u^\varepsilon(x)))$$

(denoting by “:” the scalar product of matrices) shows that also ϕ_i^t is continuous. By definition, using (A.6) again and (A.5),

$$\begin{aligned} \sum_{i=1}^k \int_{\Omega'} \phi_i \cdot D\mathbf{1}_{u_i^\varepsilon} &= \sum_{i=1}^k \int_{\Omega'} \phi_i^x(x) \cdot \nabla u_i^\varepsilon(x) - \phi_i^t(x) dx \\ &= \int_{\Omega'} f(x, \nabla u^\varepsilon(x)) dx > \int_{\Omega} f(x, Du) - \delta. \end{aligned}$$

It is enough to observe, now, that (extending ϕ by the value 0 in $\Omega \setminus \Omega'$)

$$\begin{aligned} \int_{\Omega'} \phi_i^x(x) \cdot \nabla u_i^\varepsilon(x) dx &= \int_{\Omega'} \phi_i^x(x) \cdot \left(\int_{\Omega} \rho_\varepsilon(x-y) Du_i(y) \right) dx \\ &= \int_{\Omega} \left(\int_{\Omega'} \phi_i^x(x) \rho_\varepsilon(x-y) dx \right) \cdot Du_i(y) = \int_{\Omega} (\rho_\varepsilon * \phi_i^x)(y) \cdot Du_i(y) \end{aligned}$$

for each i , while, in the same way, $\int_{\Omega'} \phi_i^t(x) dx = \int_{\Omega} \rho_\varepsilon * \phi_i^t(y) dy$ for each i . We deduce that

$$\sum_{i=1}^k \int_{\Omega} (\rho_\varepsilon * \phi_i) \cdot D\mathbf{1}_{u_i} \geq \int_{\Omega} f(x, Du) - \delta,$$

and Lemma 2.3 follows if we show that $\rho_\varepsilon * \phi$ is in (or close to) \mathcal{K}_0 . For any $y \in \Omega$, $p \in \mathbb{R}^{d \times k}$, one has (including in $\Omega \setminus \Omega'$ where ϕ has been defined as 0) using (A.6) and then (A.2)

$$\sum_{i=1}^k \phi_i^t(y) \geq \sum_{i=1}^k \phi_i^x(y) \cdot p_i - f(y, p) \geq \sum_{i=1}^k \phi_i^x(y) \cdot p_i - f(x, p) - C'|x-y|(1+|p|)$$

so that, thanks also to (2.18),

$$\begin{aligned} \sum_{i=1}^k (\rho_\varepsilon * \phi_i^t)(x) &\geq \sum_{i=1}^k (\rho_\varepsilon * \phi_i^x)(x) \cdot p_i - f(x, p) - C'\varepsilon(1 + |p|) \\ &\geq \sum_{i=1}^k (\rho_\varepsilon * \phi_i^x)(x) \cdot p_i - \left(1 + \varepsilon \frac{C'}{C}\right) f(x, p) - 2\varepsilon C'. \end{aligned}$$

Hence, the field $\tilde{\phi}$, defined by $\tilde{\phi}_i^x = (\rho_\varepsilon * \phi_i^x)/(1 + \varepsilon C'/C)$ and $\tilde{\phi}_i^t = (\rho_\varepsilon * \phi_i^t + 2\varepsilon C'/k)/(1 + \varepsilon C'/C)$ for each i , is an element of \mathcal{K}_0 , which is such that

$$\begin{aligned} \sup_{\psi \in \mathcal{K}_0} \sum_{i=1}^k \int_{\Omega} \psi_i \cdot D\mathbf{1}_{u_i} &\geq \sum_{i=1}^k \int_{\Omega} \tilde{\phi}_i \cdot D\mathbf{1}_{u_i} \\ &\geq \left(\int_{\Omega} f(x, Du) - \delta - 2\varepsilon C'|\Omega'| \right) \left(1 + \varepsilon \frac{C'}{C}\right)^{-1}. \end{aligned}$$

We can then send ε and then δ to zero to complete the proof. \blacksquare

Appendix B. Projections.

B.1. Projection onto cones $y \geq \alpha\|x\|_2$. Let $\alpha \geq 0$. For $x_0 \in \mathbb{R}^d$ and $y_0 \in \mathbb{R}$ consider the projection

$$(B.1) \quad \arg \min_{\substack{x \in \mathbb{R}^d, y \in \mathbb{R}, \\ y \geq \alpha\|x\|_2}} \frac{(x - x_0)^2}{2} + \frac{(y - y_0)^2}{2}.$$

If already $y_0 \geq \alpha\|x_0\|_2$, the solution is $(x, y) = (x_0, y_0)$. Otherwise set

$$(B.2) \quad v := \max\left(0, \frac{\|x_0\|_2 + \alpha y_0}{1 + \alpha^2}\right).$$

The solution is then given by

$$(B.3) \quad x = \begin{cases} v \frac{x_0}{\|x_0\|_2} & \text{if } x_0 \neq 0 \\ 0 & \text{else} \end{cases}, \quad y = \alpha\|x\|_2.$$

Proof. First, for $y_0 \geq \alpha\|x_0\|_2$ the projection is obviously $(x, y) = (x_0, y_0)$. Otherwise, we set $x = t\omega$, $t \geq 0$, $\omega \in \mathbb{R}^d$ with $|\omega| = 1$. For fixed t the expression $(x - x_0)^2 = (t\omega - x_0)^2$ is minimized for $\omega = \frac{x_0}{\|x_0\|_2}$ if $x_0 \neq 0$ and with arbitrary ω else. Since $(t\omega - x_0)^2 = (t - |x_0|)^2$, the solution (t, y) is given by the projection $(t, y) = \pi_{y \geq \alpha t}(\|x_0\|_2, y_0)$. This projection can be easily computed by projecting onto the line $y = \alpha t$, respectively, on the point $(0, 0)$, depending on whether $(\|x_0\|_2, y_0)$ is above or below the corresponding orthogonal line through the origin. \blacksquare

B.2. Projection onto parabolas $y \geq \alpha\|x\|_2^2$. Let $\alpha > 0$. For $x_0 \in \mathbb{R}^d$ and $y_0 \in \mathbb{R}$ consider the projection onto a parabola:

$$(B.4) \quad \arg \min_{\substack{x \in \mathbb{R}^d, y \in \mathbb{R}, \\ y \geq \alpha\|x\|_2^2}} \frac{(x - x_0)^2}{2} + \frac{(y - y_0)^2}{2}.$$

If already $y_0 \geq \alpha\|x_0\|_2^2$, the solution is $(x, y) = (x_0, y_0)$. Otherwise, with $a := 2\alpha\|x_0\|_2$, $b := \frac{2}{3}(1 - 2\alpha y_0)$, and $d := a^2 + b^3$ set

$$(B.5) \quad v := \begin{cases} c - \frac{b}{c} & \text{with } c = \sqrt[3]{a + \sqrt{d}} & \text{if } d \geq 0, \\ 2\sqrt{-b} \cos\left(\frac{1}{3} \arccos \frac{a}{\sqrt{-b^3}}\right) & \text{if } d < 0. \end{cases}$$

If $c = 0$ in the first case, set $v := 0$. The solution is then given by

$$(B.6) \quad x = \begin{cases} \frac{v}{2\alpha} \frac{x_0}{\|x_0\|_2} & \text{if } x_0 \neq 0 \\ 0 & \text{else} \end{cases}, \quad y = \alpha\|x\|_2^2.$$

Remark. In the case $d < 0$ it always holds that $\frac{a}{\sqrt{-b^3}} \in [0, 1]$. To ensure this also numerically, one should compute d by $d = (a - \sqrt{-b^3})(a + \sqrt{-b^3})$ for $b < 0$.

Proof. First, for $y_0 \geq \alpha\|x_0\|_2^2$ the projection is obviously $(x, y) = (x_0, y_0)$. Otherwise, we dualize the parabola constraint using $\delta_{z \geq 0} = \sup_{\lambda \geq 0} -\lambda z$ (for $z \in \mathbb{R}$):

$$(B.7) \quad \min_{x \in \mathbb{R}^d, y \in \mathbb{R}} \max_{\lambda \geq 0} \frac{(x - x_0)^2}{2} + \frac{(y - y_0)^2}{2} - \lambda(y - \alpha\|x\|_2^2).$$

Since this expression is convex in x, y and concave in λ , we can interchange the ordering of min and max. The inner minimization problem in x and y is then easily solved, giving the following necessary representation, with a certain $\lambda \geq 0$:

$$(B.8) \quad x = \frac{x_0}{1 + 2\alpha\lambda}, \quad y = y_0 + \lambda.$$

For instance, x has the same direction as x_0 , so only the norm of x is unknown. The solution must also necessarily satisfy $y = \alpha\|x\|_2^2$. Plugging this into the second equation of (B.8), as well as the expression for λ obtained from the first equation by taking the norms, we obtain the cubic equation

$$(B.9) \quad 2\alpha^2\|x\|_2^3 + (1 - 2\alpha y_0)\|x\|_2 - \|x_0\|_2 = 0.$$

Set $a := 2\alpha\|x_0\|_2$, $b := \frac{2}{3}(1 - 2\alpha y_0)$, and $t := 2\alpha\|x\|_2$. Then (B.9) becomes

$$(B.10) \quad t^3 + 3bt - 2a = 0.$$

Since the derivative $3t^2 + 3b$ of the left-hand side is monotonically increasing for $t \geq 0$, the t we are looking for is the unique nonnegative solution of (B.10) for $x_0 \neq 0$ (so that $a > 0$). This cubic equation can be solved using the method of elementary hyperbolic/trigonometric function identities [30], yielding the claimed solution. The second case in (B.5) corresponds to “ x_2 ” in equation (23) of [30].

For $x_0 = 0$, because of the assumed inequality $y_0 < \alpha\|x_0\|_2^2 = 0$ we have the first case in (B.5), which leads to the correct solution $(x, y) = (0, 0)$. ■

B.3. Projection for Huber-TV_J. Let $\alpha > 0$ and $\lambda > 0$. For $x_0 \in \mathbb{R}^{d \times k}$ and $y_0 \in \mathbb{R}$ consider the projection

$$(B.11) \quad \arg \min_{\substack{x \in \mathbb{R}^{d \times k}, y \in \mathbb{R}, \\ y \geq \alpha \|x\|_*^2, \|x\|_* \leq \lambda}} \frac{(x - x_0)^2}{2} + \frac{(y - y_0)^2}{2}$$

with the nuclear norm $\|\cdot\|_*$ in (3.42). Here we will consider only the case of two-dimensional color images, i.e., $d = 2$ and $k = 3$. Let $x_0 = U_0 \Sigma_0 V_0^{-1}$ be the singular value decomposition of x_0 , with some $U_0 \in \text{SO}(2)$ and $V_0 \in \text{SO}(3)$, and with the matrix $\Sigma_0 \in \mathbb{R}^{2 \times 3}$, $\Sigma_0 = \text{diag}(\sigma_1^0, \sigma_2^0)$, containing the singular values $\sigma_1^0 \geq \sigma_2^0 \geq 0$. Using the definition (3.42), the projection (B.11) can be reformulated in terms of the singular values σ_1, σ_2 of x :

$$(B.12) \quad \arg \min_{\substack{x \in \mathbb{R}^{d \times k}, y \in \mathbb{R}, \\ y \geq \alpha(\sigma_1 + \sigma_2)^2, \sigma_1 + \sigma_2 \leq \lambda}} \sum_{i=1}^2 \frac{(\sigma_i - \sigma_i^0)^2}{2} + \frac{(y - y_0)^2}{2}.$$

Having a (σ_1, σ_2, y) which solves (B.12), the solution (x, y) of (B.11) can be obtained by $x = U_0 \text{diag}(\sigma_1, \sigma_2) V_0^{-1}$. Finally, the solution (σ_1, σ_2, y) of (B.12) is $(\sigma_1^0, \sigma_2^0, y^0)$ if this point already satisfies the constraints, and otherwise, with $\delta := \sigma_1^0 - \sigma_2^0$, is given by

$$(B.13) \quad \begin{cases} (\lambda, 0, \max(\alpha\lambda^2, y_0)) & \text{if } \delta \geq \lambda \text{ and } y_0 \geq \alpha\lambda^2 - \frac{\sigma_1^0 - \lambda}{2\alpha\lambda}, \\ \left(\frac{\lambda + \delta}{2}, \frac{\lambda - \delta}{2}, \max(\alpha\lambda^2, y_0)\right) & \text{if } \delta \leq \lambda \text{ and } y_0 \geq \alpha\lambda^2 - \frac{\sigma_1^0 + \sigma_2^0 - \lambda}{4\alpha\lambda}, \\ \left(\frac{\hat{x}}{\sqrt{2}} + \frac{\delta}{2}, \frac{\hat{x}}{\sqrt{2}} - \frac{\delta}{2}, \hat{y}\right) \\ \quad \text{with } (\hat{x}, \hat{y}) := \pi_{y \geq 2\alpha x^2} \left(\frac{\sigma_1^0 + \sigma_2^0}{\sqrt{2}}, y_0\right) & \text{if } \delta \leq \lambda \text{ and } y_0 \geq \alpha\delta^2 - \frac{\sigma_2^0}{2\alpha\delta}, \\ (\hat{x}, 0, \hat{y}) \\ \quad \text{with } (\hat{x}, \hat{y}) := \pi_{y \geq \alpha x^2}(\sigma_1^0, y_0) & \text{otherwise.} \end{cases}$$

Proof. First, let us establish the equivalence of (B.11) and (B.12). For candidate solutions (x, y) of (B.11), let $x = U \Sigma V^{-1}$ with $U \in \text{SO}(2)$, $V \in \text{SO}(3)$, $\Sigma \in \mathbb{R}^{2 \times 3}$, $\Sigma = \text{diag}(\sigma_1, \sigma_2)$ be the singular value decomposition of x . By Mirsky's inequality [31] we have $(x - x_0)^2 \geq \sum_{i=1}^2 (\sigma_i - \sigma_i^0)^2$ with equality if $U = U_0$, $V = V_0$. Thus, an optimal x for (B.11) will have the form $U_0 \Sigma V_0^{-1}$, where the singular values σ_1, σ_2 satisfy the constraints of (B.12).

The derivation of (B.13) is rather lengthy, but straightforward. The third case corresponds to the projection onto the paraboloid segment $y = \alpha(\sigma_1 + \sigma_2)^2$, $\sigma_1, \sigma_2 \geq 0$, $\sigma_1 + \sigma_2 \leq \lambda$, and is active when there is an outer surface normal passing through the point $(\sigma_1^0, \sigma_2^0, y^0)$. The fourth case is the projection onto the parabola line $y = \alpha(\sigma_1 + \sigma_2)^2 = \alpha\sigma_1^2$, $0 \leq \sigma_1 \leq \lambda$, $\sigma_2 = 0$, active when there is an orthogonal line on it passing through the given point. The second case projects onto the plane segment $y \geq \alpha(\sigma_1 + \sigma_2)^2 = \alpha\lambda^2$, $\sigma_1, \sigma_2 \geq 0$, $\sigma_1 + \sigma_2 = \lambda$, and the first case projects onto the line $y \geq \alpha(\sigma_1 + \sigma_2)^2 = \alpha\lambda^2$, $\sigma_1 = \lambda$, $\sigma_2 = 0$. ■

REFERENCES

- [1] G. ALBERTI, G. BOUCHITTÉ, AND G. DAL MASO, *The calibration method for the Mumford-Shah functional and free-discontinuity problems*, Calc. Var. Partial Differential Equations, 3 (2003), pp. 299–333.
- [2] L. AMBROSIO, N. FUSCO, AND D. PALLARA, *Functions of Bounded Variation and Free Discontinuity Problems*, Oxford Math. Monogr., The Clarendon Press, Oxford University Press, New York, 2000.
- [3] E. BAE, J. YUAN, AND X.-C. TAI, *Global minimization for continuous multiphase partitioning problems using a dual approach*, Int. J. Comput. Vis., 92 (2011), pp. 112–129.
- [4] S. BAKER, D. SCHARSTEIN, J. P. LEWIS, S. ROTH, M. J. BLACK, AND R. SZELISKI, *A database and evaluation methodology for optical flow*, Int. J. Comput. Vis., 92 (2011), pp. 1–31.
- [5] G. BOUCHITTÉ AND M. VALADIER, *Integral representation of convex functionals on a space of measures*, J. Funct. Anal., 80 (1988), pp. 398–420.
- [6] G. BOUCHITTÉ AND M. VALADIER, *Multifonctions s.c.i. et régularisée s.c.i. essentielle*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 6 (1989), pp. 123–149.
- [7] Y. BOYKOV AND V. KOLMOGOROV, *Computing geodesics and minimal surfaces via graph cuts*, in Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, pp. 26–33.
- [8] Y. BOYKOV, O. VEKSLER, AND R. ZABIH, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2001), pp. 1222–1239.
- [9] T. BROX, A. BRUHN, N. PAPENBERG, AND J. WEICKERT, *High accuracy optical flow estimation based on a theory for warping*, in Proceedings of the 8th European Conference on Computer Vision, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 25–36.
- [10] A. CHAMBOLLE, D. CREMERS, AND T. POCK, *A Convex Approach for Computing Minimal Partitions*, Technical report TR-2008-05, Department of Computer Science, University of Bonn, Bonn, Germany, 2008.
- [11] A. CHAMBOLLE, D. CREMERS, AND T. POCK, *A convex approach to minimal partitions*, SIAM J. Imaging Sci., 5 (2012), pp. 1113–1158.
- [12] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
- [13] T. F. CHAN, S. ESEDOĞLU, AND M. NIKOLOVA, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66 (2006), pp. 1632–1648.
- [14] D. CREMERS AND K. KOLEV, *Multiview stereo and silhouette consistency via convex functionals over convex domains*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1161–1174.
- [15] G. DAL MASO, *Integral representation on $BV(\Omega)$ of Γ -limits of variational integrals*, Manuscripta Math., 30 (1979), pp. 387–416.
- [16] F. DEMENGEL AND R. TEMAM, *Convex functions of a measure and applications*, Indiana Univ. Math. J., 33 (1984), pp. 673–709.
- [17] L. C. EVANS AND R. F. GARIEPY, *Measure Theory and Fine Properties of Functions*, Stud. Adv. Math., CRC Press, Boca Raton, FL, 1992.
- [18] H. FEDERER, *Geometric Measure Theory*, Springer-Verlag, New York, 1969.
- [19] H. FEDERER, *Real flat chains, cochains and variational problems*, Indiana Univ. Math. J., 24 (1974), pp. 351–407.
- [20] M. GIAQUINTA, G. MODICA, AND J. SOUČEK, *Cartesian Currents in the Calculus of Variations. I. Cartesian Currents*, Springer-Verlag, Berlin, 1998.
- [21] B. GOLDLUECKE, E. STREKALOVSKIY, AND D. CREMERS, *The natural total variation which arises from geometric measure theory*, SIAM J. Imaging Sci., 5 (2012), pp. 537–563.
- [22] B. GOLDLUECKE, E. STREKALOVSKIY, AND D. CREMERS, *Tight convex relaxations for vector-valued labeling*, SIAM J. Imaging Sci., 6 (2013), pp. 1626–1664.
- [23] D. GREIG, B. PORTEOUS, AND A. SEHEULT, *Exact maximum a posteriori estimation for binary images*, J. Roy. Statist. Soc. Ser. B, 51 (1989), pp. 271–279.
- [24] H. ISHIKAWA, *Exact optimization for Markov random fields with convex priors*, IEEE Trans. Pattern Anal. Mach. Intell., 25 (2003), pp. 1333–1336.
- [25] H. ISHIKAWA AND D. GEIGER, *Segmentation by grouping junctions*, in Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998, pp. 125–131.
- [26] V. KOLMOGOROV AND C. ROTHER, *Minimizing non-submodular functions with graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 1274–1279.

- [27] V. KOLMOGOROV AND R. ZABIH, *What energy functions can be minimized via graph cuts?*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 147–159.
- [28] J. LELLMANN, F. BECKER, AND C. SCHNÖRR, *Convex optimization for multi-class image labeling with a novel family of total variation based regularizers*, in Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 646–653.
- [29] J. LELLMANN AND C. SCHNÖRR, *Continuous multiclass labeling approaches and algorithms*, SIAM J. Imaging Sci., 4 (2011), pp. 1049–1096.
- [30] J. P. MCKELVEY, *Simple transcendental expressions for the roots of cubic equations*, Amer. J. Phys., 52 (1984), pp. 269–270.
- [31] L. MIRSKY, *Symmetric gauge functions and unitarily invariant norms*, Quart. J. Math. Oxford Ser. (2), 11 (1960), pp. 50–59.
- [32] T. POCK AND A. CHAMBOLLE, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in Proceedings of the 2011 IEEE International Conference on Computer Vision, 2011, pp. 1762–1769.
- [33] T. POCK, D. CREMERS, H. BISCHOF, AND A. CHAMBOLLE, *An algorithm for minimizing the piecewise smooth Mumford-Shah functional*, in Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, 2009.
- [34] T. POCK, D. CREMERS, H. BISCHOF, AND A. CHAMBOLLE, *Global solutions of variational models with convex regularization*, SIAM J. Imaging Sci., 3 (2010), pp. 1122–1145.
- [35] T. POCK, T. SCHOENEMANN, G. GRABER, H. BISCHOF, AND D. CREMERS, *A convex formulation of continuous multi-label problems*, in Computer Vision—ECCV 2008, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 792–805.
- [36] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1996.
- [37] L. I. RUDIN, S. J. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [38] D. SCHLESINGER AND B. FLACH, *Transforming an Arbitrary MinSum Problem into a Binary One*, Technical report, Department of Computer Science, Dresden University of Technology, Dresden, Germany, 2006.
- [39] E. STREKALOVSKIY, B. GOLDLUECKE, AND D. CREMERS, *Tight convex relaxations for vector-valued labeling problems*, in Proceedings of the 2011 IEEE International Conference on Computer Vision, 2011.
- [40] D. SUN, S. ROTH, AND M. J. BLACK, *Secrets of optical flow estimation and their principles*, in Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2432–2439.
- [41] C. VILLANI, *Topics in Optimal Transportation*, Grad. Stud. Math. 58, AMS, Providence, RI, 2003.
- [42] M. WAINWRIGHT, T. JAAKKOLA, AND A. WILLSKY, *Map estimation via agreement on trees: Message-passing and linear programming*, IEEE Trans. Inform. Theory, 51 (2005), pp. 3697–3717.
- [43] L. XU, J. JIA, AND Y. MATSUSHITA, *Motion detail preserving optical flow estimation*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 1744–1757.
- [44] C. ZACH, D. GALLUP, J.-M. FRAHM, AND M. NIETHAMMER, *Fast global labeling for real-time stereo using multiple plane sweeps*, in Vision, Modeling and Visualization, Aka GmbH, Berlin, 2008, pp. 243–252.
- [45] C. ZACH, C. HÄNE, AND M. POLLEFEYS, *What is optimized in tight convex relaxations for multi-label problems?*, in Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1664–1671.