A Cutting-Plane Method for Sublabel-Accurate Relaxation of Problems with Product Label Spaces

Zhenzhang Ye, Bjoern Haefner, Yvain Quéau, Thomas Möllenhoff, Daniel Cremers

A Cutting-Plane Method for Sublabel-Accurate Relaxation of Problems with Product Label Spaces

Zhenzhang Ye[1*], Bjoern Haefner[1],
Yvain Quéau[2], Thomas Möllenhoff[3] and Daniel Cremers[1]

[1]Department of Informatics, Technical University of Munich, Garching, Germany.
[2]Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France.
[3]Center for AI Project, RIKEN, Tokyo, Japan.

*Corresponding author(s). E-mail(s): zhenzhang.ye@tum.de;
Contributing authors: bjoern.haefner@tum.de; yvain.queau@ensicaen.fr;
thomas.moellenhoff@riken.jp; cremers@tum.de;

Abstract

Many problems in imaging and low-level vision can be formulated as nonconvex variational problems. A promising class of approaches to tackle such problems are convex relaxation methods, which consider a lifting of the energy functional to a higher-dimensional space. However, they come with increased memory requirements due to the lifting. The present paper is an extended version of the earlier conference paper by Ye et al. (2021) which combined two recent approaches to make lifting more scalable: product-space relaxation and sublabel-accurate discretization. Furthermore, it is shown that a simple cutting-plane method can be used to solve the resulting semi-infinite optimization problem. This journal version extends the previous conference work with additional experiments, a more detailed outline of the complete algorithm and a user-friendly introduction to functional lifting methods.

Keywords: Variational Methods, Manifold-Valued Problems, Convex Relaxation, Global Optimization

1 Introduction

In this paper, we present a convex optimization framework for total-variation regularized problems of the following form:

$$\inf_{u:\Omega\to\Gamma} \int_\Omega c(x, u_1(x), \dots, u_k(x))\,\mathrm{d}x + \sum_{i=1}^{k} \lambda_i \mathrm{TV}(u_i). \tag{1}$$

The set $\Gamma = \{(\gamma_1, \dots, \gamma_k) \in \mathbf{R}^N : \gamma_i \in \Gamma_i, i = 1 \dots k\}$ is defined by k individual submanifolds $\Gamma_i \subset \mathbf{R}^{N_i}$ with $N = N_1 + \dots + N_k$. The individual Γ_i are required to be bounded subsets of $\mathbf{R}^{N_i}$.

Since the focus of this paper are imaging applications we assume $\Omega \subset \mathbf{R}^2$ to be a rectangular domain but the approach is easily generalized to higher dimensional or non-rectangular domains.

We make no special assumptions on the cost $c : \Omega \times \Gamma \to \mathbf{R}_{\geq 0}$ in (1) and allow it to be a general non-negative nonconvex function. This turns (1) into an overall nonconvex optimization problem, which can be challenging to solve using standard gradient-based methods. Moreover, we do not assume that we are able to compute gradients, projections or proximal operators of the cost function $c(x, u(x))$.

Our approach only requires function evaluations. This allows us to consider degenerate costs that are out of reach for gradient-based approaches.

The regularizer in (1) is a separable *total variation regularization* $\mathrm{TV}(u_i)$ on the individual components $u_i : \Omega \to \mathbf{R}^{N_i}$ weighted by a parameter $\lambda_i > 0$. The total variation (TV) (Chambolle et al., 2010; Rudin et al., 1992) encourages a spatially smooth but edge-preserving solution. It can be defined as the following equation:

$$\begin{aligned}\mathrm{TV}(u_i) &:= \sup_{\substack{p:\Omega\to\mathbf{R}^{N_i\times 2}\\ \|p(x)\|_*\le 1}} \int_\Omega \langle \mathrm{Div}_x\, p(x), u_i(x)\rangle \,\mathrm{d}x \quad (2)\\ &= \int_\Omega \|\nabla u_i(x)\| \,\mathrm{d}x,\end{aligned}$$

where by $\nabla u_i(x) \in \mathbf{R}^{N_i\times 2}$ we denote the Jacobian matrix and by $\|\cdot\|_*$ the dual norm of $\|\cdot\|$. The last equality in (2) holds for sufficiently smooth u_i.

The convex relaxation approach we use in this paper works for general convex and nonconvex regularizers which depend on the Jacobian ∇u_i, see Möllenhoff and Cremers (2019); Pock et al. (2009, 2010); Vogt et al. (2020). However, the main focus of this paper is an efficient implementation of the data cost c, and therefore we consider only the separable total variation (2).

Motivation and Applications.

To motivate Problem (1), let us consider some practical applications in low-level vision and imaging. One example is the variational estimation of *optical flow* between two RGB images $I_1, I_2 : \Omega \to \mathbf{R}^3$, see Horn and Schunck (1981). In that case, $\Gamma_1 = \Gamma_2 = [a, b] \subset \mathbf{R}$ models the displacement between the two images and the cost function is given by a photometric error.

Often, Γ_i is a curved manifold, see, e.g., the applications presented by Lellmann, Strekalovskiy, et al. (2013); Weinmann et al. (2014). Examples include $\Gamma_i = \mathbb{S}^2$ for normal field processing (Lellmann, Strekalovskiy, et al., 2013), $\mathrm{SO}(3)$ for motion estimation (Görlitz et al., 2019) or the circle $\mathbb{S}^1$ for processing of cyclic data (Cremers & Strekalovskiy, 2013; Steinke et al., 2010).

Many real-wolrd applications requires to estimate multiple quantities in a joint fashion. This naturally leads to the formulation of product space which is considered in (1), where $\Gamma = \Gamma_1 \times \ldots \times \Gamma_k$. In this case, each Γ_i models one quantity of interest that one aims to estimate.

A prominent approach to address joint optimization problems of this form are alternating procedures such as expectation maximization (Dempster et al., 1977), block-coordinate descent and alternating direction-type methods (Boyd et al., 2011). There, the idea is to estimates a single quantity while holding the other ones fixed. However, these approaches usually depend on a good initialization and are easy to get stuck in a very poor local optima. Therefore, the goal of this paper is to instead consider a convex relaxation of Problem (1). The relaxed problem can then be solved to global optimality with standard proximal methods such as the primal dual algorithm (Pock & Chambolle, 2011). These methods are usually well parallelizable. Thus, they can be efficiently implemented on GPUs, allowing to solve large-scale problems in a reasonable time.

Contributions.

The main difficulty with convex approaches to (1) is the large memory requirements which are inherent to a lifted problem formulation which renders the problem convex. In order to improve the memory-efficiency of relaxations, two disparate ideas have been considered in previous work: sublabel-accurate liftings (Möllenhoff et al., 2016) and product-space relaxations (Goldluecke et al., 2013). In this paper, we combine both approaches and present a sublabel-accurate implementation of Goldluecke et al. (2013). Unlike previous liftings (Möllenhoff et al., 2016; Möllenhoff & Cremers, 2017; Vogt et al., 2020), our approach does not require epigraphical projections and can therefore be applied in a black-box fashion, requiring only evaluations of the cost c.

Our main contribution is a simple way to implement the resulting semi-infinite optimization problem with a cutting-plane method. Moreover, we show that using this method, we can achieve a lower energy than the product-space lifting (Goldluecke et al., 2013) on optical flow estimation and manifold-valued denoising problems.

This journal paper is an extended version of the conference paper (Ye et al., 2021). In particular, we offer the following contributions over the conference version:

- We have added additional background and explanations on the basics of functional lifting to Section 2 and Section 3.
- In Section 4 we added the detailed update equations for the primal-dual algorithm.
- We added additional figures and explanations to Section 5 to illustrate and provide an intuition of our algorithm on a simple example.
- We added additional experiments on optical flow and manifold-valued image denoising to Section 5 and evaluated our method on a larger set of images.

Overview of the paper.

After this introduction, we provide in Section 2 an introduction to functional lifting methods for Problem (1), review existing works and explain our contributions relatively to them. We summarize the convex relaxation for (1) and its discretization in Section 3. The proposed cutting-plane method and sampling strategy which we use to implement the discretized relaxation are presented in Section 4. In Section 5 we evaluate our method on a toy problem and several real-world imaging applications. Our conclusions are eventually drawn in Section 6.

2 An introduction to functional lifting

Let us first consider a simplified version of Problem (1) where Ω consists only of a single point, i.e., the nonconvex minimization of one data term:

$$\min_{\gamma \in \Gamma} \; c(\gamma_1, \dots, \gamma_k). \tag{3}$$

A well-known approach to the global optimization of (3) is a *lifting* or *stochastic relaxation* procedure, which has been considered in diverse fields such as polynomial optimization (Lasserre, 2000), continuous Markov random fields (Bauermeister et al., 2021; Fix & Agarwal, 2014; Peng et al., 2011), variational methods (Pock et al., 2008), and black-box optimization (de Boer et al., 2005; Ollivier et al., 2017; Schaul, 2011). The idea is to relax the search space in (3) from $\gamma \in \Gamma$ to *probability distributions* $\mathbf{u} \in \mathcal{P}(\Gamma)$ and solve[1]

$$\min_{\mathbf{u} \in \mathcal{P}(\Gamma)} \int_{\Gamma} c(\gamma_1, \dots, \gamma_k) \, \mathrm{d}\mathbf{u}(\gamma_1, \dots, \gamma_k). \tag{4}$$

Due to linearity of the integral wrt. $\mathbf{u}$ and convexity of the relaxed search space, this is a convex problem for arbitrary cost c. Moreover, the minimizers of (4) concentrate at the optima of c and can hence be identified with solutions to (3). However, if Γ is a continuum, this problem is infinite-dimensional and therefore challenging.

We illustrate the conceptual difference between the formulation (3) and (4) on a one-dimensional example in Figure 1.

Discrete/traditional multilabeling.

In the context of Markov random fields (Ishikawa, 2003; Kappes et al., 2013) and multilabel optimization (Chambolle et al., 2012; Lellmann et al., 2009; Lellmann & Schnörr, 2011; Zach et al., 2008) one typically discretizes Γ into a finite set of points (called the *labels*) $\Gamma = \{\mathbf{v}_1, \dots, \mathbf{v}_\ell\}$. This turns (4) into a finite-dimensional linear program $\min_{\mathbf{u} \in \Delta^\ell} \langle c', \mathbf{u} \rangle$ where $c' \in \mathbf{R}^\ell_{\geq 0}$ denotes the label cost and $\Delta^\ell \subset \mathbf{R}^\ell$ is the $(\ell - 1)$-dimensional unit simplex. If we evaluate the cost at the labels, this program *upper bounds* the continuous problem (3), since instead of all possible solutions, one considers a restricted subset determined by the labels. Since the solution will be attained at one of the labels, typically a fine meshing is needed. Similar to black-box and zero-order optimization methods, this strategy suffers from the *curse of dimensionality*. When each Γ_i is discretized into ℓ labels, the overall number is ℓ^k which quickly becomes intractable since many labels are required for a smooth solution. This limits the method to be applied on more practical problems. Additionally, for pairwise or regularizing terms, often a large number of dual constraints has to be implemented. In that context, the work from Lellmann, Lellmann, et al. (2013) considers a constraint pruning strategy as an offline-preprocessing.

[1] $\mathcal{P}(\Gamma)$ is the set of nonnegative Radon measures on Γ with total mass $\mathbf{u}(\Gamma) = 1$.

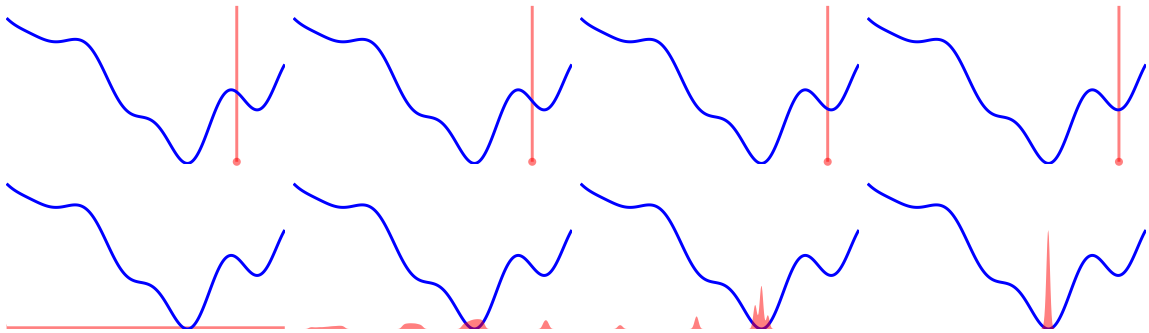


Fig. 1 Traditional optimization vs. optimization over a space of probability distributions. In the top row, going from left to right, we illustrate a traditional gradient-based local optimization method on a nonconvex problem of the form (3). Given a bad initialization, the algorithm might get stuck in a poor local optimum. The bottom row illustrates an optimization procedure on the relaxed problem (4). Due to convexity of the objective and search space of probability measures, the solution concentrates at a Dirac distribution centered around a global optimum.

Sublabel-accurate multilabeling.

The discrete-continuous MRF (Fix & Agarwal, 2014; Zach, 2013; Zach & Kohli, 2012) and lifting methods (Laude et al., 2016; Möllenhoff et al., 2016; Möllenhoff & Cremers, 2017) attempt to find a more label-efficient convex formulation. These approaches can be understood through duality (Fix & Agarwal, 2014; Möllenhoff & Cremers, 2017). Applied to (3), the idea is to replace the cost $c : \Gamma \to \mathbf{R}$ with a dual variable $\mathbf{q} : \Gamma \to \mathbf{R}$:

$$\min_{\mathbf{u} \in \mathcal{P}(\Gamma)} \sup_{\mathbf{q}:\Gamma \to \mathbf{R}} \int_{\Gamma} \mathbf{q}(\gamma_1, \dots, \gamma_k) \, \mathrm{d}\mathbf{u}(\gamma_1, \dots, \gamma_k),$$
$$\text{s.t. } \mathbf{q}(\gamma) \leq c(\gamma) \quad \text{for all } \gamma \in \Gamma. \tag{5}$$

The inner supremum in the formulation (5) *maximizes* the lower-bound $\mathbf{q}$. Additionally, if the dual variable is sufficiently expressive, this problem is actually equivalent to (4).

Approximating $\mathbf{q}$, for example with piecewise linear functions on Γ, one arrives at a *lower-bound* to the nonconvex problem (3). It has been observed in a recent series of works (Laude et al., 2016; Möllenhoff et al., 2016; Möllenhoff & Cremers, 2017; Vogt et al., 2020; Zach & Kohli, 2012) that piecewise linear dual variables can lead to smooth solutions even when $\mathbf{q}$ (and therefore also $\mathbf{u}$) is defined on a rather coarse mesh. As remarked by Fix and Agarwal (2014); Laude et al. (2016); Möllenhoff et al. (2016), for an *affine* dual variable this strategy corresponds to minimizing the convex envelope of the cost, $\min_{\gamma \in \Gamma} c^{**}(\gamma)$, where c^{**} denotes the Fenchel biconjugate of c.

The implementation of the constraints in (5) can be challenging even in the case of piecewise-linear $\mathbf{q}$. This is partly due to the fact that Problem (5) is a semi-infinite optimization problem (Blankenship & Falk, 1976), i.e., an optimization problem with *infinitely many constraints*. The works (Möllenhoff et al., 2016; Zach & Kohli, 2012) implement the constraints via projections onto the epigraph of the (restricted) conjugate function of the cost within a proximal optimization framework. Such projections are only available in closed form for some choices of c and expensive to compute if the dimension is larger than one (Laude et al., 2016). This limits the applicability in a "plug-and-play" fashion.

Product-space liftings.

The product-space lifting approach (Goldluecke et al., 2013) attempts to overcome the aforementioned exponential memory requirements of labeling methods in an orthogonal way to the sublabel-based methods. The main idea is to exploit the product-space structure in (1) and optimize over k marginal distributions of the probability measure $\mathbf{u} \in \mathcal{P}(\Gamma)$, which we denote by $\mathbf{u}_i \in \mathcal{P}(\Gamma_i)$. Applying Goldluecke et al. (2013) to the single data term (3) one arrives at the following relaxation:

$$\min_{\{\mathbf{u}_i \in \mathcal{P}(\Gamma_i)\}} \sup_{\{\mathbf{q}_i:\Gamma_i \to \mathbf{R}\}} \sum_{i=1}^{k} \int_{\Gamma_i} \mathbf{q}_i(\gamma_i) \, \mathrm{d}\mathbf{u}_i(\gamma_i)$$
$$\text{s.t. } \sum_{i=1}^{k} \mathbf{q}_i(\gamma_i) \leq c(\gamma) \quad \text{for all } \gamma \in \Gamma. \tag{6}$$

Since one only has to discretize the individual Γ_i this substantially reduces the memory requirements from $\mathcal{O}(\ell^N)$ to $\mathcal{O}(\sum_{i=1}^k \ell^{N_i})$. While at first glance it seems that the curse of dimensionality is lifted, the difficulty is moved to the dual, where we still have a large (or even infinite) number of constraints. A global implementation of the constraints with Lagrange multipliers as proposed in Goldluecke et al. (2013) again leads to the same exponential dependency on the dimension.

As a side note, readers familiar with optimal transport may notice that the supremum in (6) is a multi-marginal transportation problem (Carlier, 2003; Villani, 2008) with transportation cost c. This view is mentioned by Bach (2019) where relaxations of form (6) are analyzed under submodularity assumptions.

In summary, the sublabel-accurate lifting methods, discrete-continuous MRFs (Möllenhoff et al., 2016; Zach & Kohli, 2012) and product-space liftings (Goldluecke et al., 2013) all share a common difficulty: *implementation of an exponential or even infinite number of constraints on the dual variables.*

Summary of our contribution.

Our main contribution is a simple way to implement the dual constraints in an online fashion with a random sampling strategy which we present in Section 4. This allows a black-box implementation, which only requires an *evaluation* of the cost c and no epigraphical projection operations as in Möllenhoff et al. (2016); Zach and Kohli (2012). Moreover, the sampling approach allows us to propose and implement a sublabel-accurate variant of the product-space relaxation (Goldluecke et al., 2013) which we describe in the following section.

3 Product-space relaxation

Our starting point is the convex relaxation of (1) presented in Goldluecke et al. (2013); Strekalovskiy et al. (2014). In these works, $\Gamma_i \subset \mathbf{R}$ is chosen to be an interval. We first denote the Lagrangian as:

$$\mathcal{L}(\{\mathbf{u}_i\},\{\mathbf{q}_i\},\{\mathbf{p}_i\}) = \sum_{i=1}^k \int_\Omega \int_{\Gamma_i} \mathbf{q}_i(x,\gamma_i) \tag{7}$$

$$- \operatorname{Div}_x \mathbf{p}_i(x,\gamma_i)\ \mathrm{d}\mathbf{u}_i^x(\gamma_i)\,\mathrm{d}x. \tag{8}$$

Following Vogt et al. (2020) we consider a generalization to manifolds $\Gamma_i \subset \mathbf{R}^{N_i}$ which leads us to the following relaxation:

$$\min_{\{\mathbf{u}_i:\Omega\to\mathcal{P}(\Gamma_i)\}} \sup_{\substack{\{\mathbf{q}_i:\Omega\times\Gamma_i\to\mathbf{R}\}\\ \{\mathbf{p}_i:\Omega\times\Gamma_i\to\mathbf{R}^2\}}} \mathcal{L}(\{\mathbf{u}_i\},\{\mathbf{q}_i\},\{\mathbf{p}_i\}),$$

$$\text{s.t.}\quad \|P_{T_{\gamma_i}} \nabla_{\gamma_i} \mathbf{p}_i(x,\gamma_i)\|_* \le \lambda_i,\ \forall i,x,\gamma_i, \tag{9}$$

$$\sum_{i=1}^k \mathbf{q}_i(x,\gamma_i) \le c(x,\gamma),\ \forall x,\gamma \tag{10}$$

This cost function appears similar to (6) explained in the previous section, but there are two differences. First, we now have marginal distributions $\mathbf{u}_i(x)$ for every $x \in \Omega$ since we do not consider only a single data term anymore. The notation $\mathrm{d}\mathbf{u}_i^x$ in (8) denotes the integration against the probability measure $\mathbf{u}_i(x) \in \mathcal{P}(\Gamma_i)$. The variables $\mathbf{q}_i$ play the same role as in (6) and lower-bound the cost under constraint (10). The second difference is the introduction of additional dual variables $\mathbf{p}_i$ and the term $-\operatorname{Div}_x \mathbf{p}_i$ in (8). Together with the constraint (9), this can be shown to implement the total variation regularization as in Lellmann, Strekalovskiy, et al. (2013); Vogt et al. (2020). Following Vogt et al. (2020), the derivative $\nabla_{\gamma_i}\mathbf{p}_i(x,\gamma_i)$ in (9) denotes the $(N_i\times 2)$-dimensional Jacobian considered in the Euclidean sense and $P_{T_{\gamma_i}}$ the projection onto the tangent space of Γ_i at the point γ_i.

To get an intuition on the total variation regularization, we use a concrete example to illustrate how (8) and (9) implement it. Consider the case when the labeling variable $\mathbf{u}_i^x = \delta_{u(x)}$ is given as a Dirac measure at every point x. As a concrete example, we consider $k = 1$ for simplicity. The term in (8) then simplifies to

$$\int_\Omega -\operatorname{Div}_x \mathbf{p}(x,u(x))\ \mathrm{d}x \tag{11}$$

$$= \int_\Omega \langle \nabla_\gamma \mathbf{p}(x,u(x)), \nabla u(x)\rangle\ \mathrm{d}x, \tag{12}$$

which follows by applying the chain-rule and the fact that $\mathbf{p}$ has compact support. Finally, taking a point-wise supremum over $\mathbf{p}$ inside the integral in (12) under the dual-norm constraint (9) gives us the total variation of u: $\int_\Omega \|\nabla u(x)\|\ \mathrm{d}x$, which is the same as defined in (2).

3.1 Finite-element discretization

We approximate the infinite-dimensional problem (8) by restricting $\mathbf{u}_i$, $\mathbf{p}_i$ and $\mathbf{q}_i$ to be piecewise functions on a discrete meshing of $\Omega \times \Gamma_i$. The considered discretization is a standard finite-element approach and largely follows the work from Vogt et al. (2020). Unlike the forward-differences considered in Vogt et al. (2020) we use lowest-order Raviart-Thomas elements (see, e.g., Caillaud and Chambolle (2020), Section 5) in Ω, which are specifically tailored towards the considered total variation regularization.

Discrete mesh.

We approximate each d_i-dimensional manifold $\Gamma_i \subset \mathbf{R}^{N_i}$ with a simplicial manifold Γ_i^h, given by the union of a collection of d_i-dimensional simplices $\mathcal{T}_i$. We denote the number of vertices ("labels") in the triangulation of Γ_i as ℓ_i. The set of labels is denoted by $\mathcal{L}_i = \{\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,\ell_i}\}$. As assumed, $\Omega \subset \mathbf{R}^2$ is a rectangle which we split into a set of faces $\mathcal{F}$ of edge-length h_x with edge set $\mathcal{E}$. The number of faces and edges are denoted by $F = |\mathcal{F}|$, $E = |\mathcal{E}|$.

Data term and the $\mathbf{u}_i$, $\mathbf{q}_i$ variables.

We assume the cost $c : \Omega \times \Gamma \to \mathbf{R}_{\geq 0}$ is constant in $x \in \Omega$ on each face and denote its value as $c(x(f), \gamma)$ for $f \in \mathcal{F}$, where $x(f) \in \Omega$ denotes the midpoint of the face f. Similarly, we also assume the variables $\mathbf{u}_i$ and $\mathbf{q}_i$ to be constant in $x \in \Omega$ on each face but continuous piecewise linear functions in γ_i. They are represented by coefficient functions $\mathbf{u}_i^h, \mathbf{q}_i^h \in \mathbf{R}^{F \cdot \ell_i}$, i.e., we specify the values on the labels and linearly interpolate inbetween. This is done by the interpolation operator $\mathbf{W}_{i,f,\gamma_i} : \mathbf{R}^{F \cdot \ell_i} \to \mathbf{R}$ which given an index $1 \leq i \leq k$, face f, and (continuous) label position $\gamma_i \in \Gamma_i$ computes the function value based on barycentric coordinates: $\mathbf{W}_{i,f,\gamma_i} \mathbf{u}_i^h = \mathbf{u}_i(x(f), \gamma_i)$. Note that after discretization, $\mathbf{u}_i$ is only defined on Γ_i^h but we can uniquely associate to each $\gamma_i \in \Gamma_i^h$ a point on Γ_i.

Divergence and $\mathbf{p}_i$ variables.

Our variable $\mathbf{p}_i$ is represented by coefficients $\mathbf{p}_i^h \in \mathbf{R}^{E \cdot \ell_i}$ which live on the edges in Ω and the labels in Γ_i. The vector $\mathbf{p}_i(x, \gamma_i) \in \mathbf{R}^2$ is obtained by linearly interpolating the coefficients on the vertical and horizontal edges of the face and using the interpolated coefficients to evaluate the piecewise-linear function on Γ_i^h. Under this approximation, the discrete divergence $\mathrm{Div}_x^h : \mathbf{R}^{E \cdot \ell_i} \to \mathbf{R}^{F \cdot \ell_i}$ is given by $(\mathrm{Div}_x^h \mathbf{p}_i^h)(f) = \left(\mathbf{p}_i^h(e_r) + \mathbf{p}_i^h(e_t) - \mathbf{p}_i^h(e_l) - \mathbf{p}_i^h(e_b)\right)/h_x$ where e_r, e_t, e_l, e_b are the right, top, left and bottom edges of f, respectively.

Total variation constraint.

Computing the operator $P_{T_{\gamma_i}} \nabla_{\gamma_i}$ is largely inspired by Vogt et al. (2020), Section 2.2. It is implemented by a linear map $\mathbf{D}_{i,f,\alpha,t} : \mathbf{R}^{E \cdot \ell_i} \to \mathbf{R}^{d_i \times 2}$. Here, $f \in \mathcal{F}$ and $\alpha \in [0,1]^2$ correspond to a point $x \in \Omega$ while $t \in \mathcal{T}_i$ is the simplex containing the point corresponding to $\gamma_i \in \Gamma_i$. First, the operator computes coefficients in $\mathbf{R}^{\ell_i}$ of two piecewise-linear functions on the manifold by linearly interpolating the values on the edges based on the face index $f \in \mathcal{F}$ and $\alpha \in [0,1]^2$. For each function, the derivative in simplex $t \in \mathcal{T}_i$ on the triangulated manifold is given by the gradient of an affine extension. Projecting the resulting vector onto the d_i-dimensional tangent space for both functions leads to a $d_i \times 2$-matrix which approximates $P_{T_{\gamma_i}} \nabla_{\gamma_i} \mathbf{p}_i(x, \gamma_i)$.

Final discretized problem.

Plugging our discretized $\mathbf{u}_i$, $\mathbf{q}_i$, $\mathbf{p}_i$ into (8), we arrive at the following finite-dimensional optimization problem:

$$\min_{\{\mathbf{u}_i^h \in \mathbf{R}^{F \cdot \ell_i}\}} \max_{\substack{\{\mathbf{p}_i^h \in \mathbf{R}^{E \cdot \ell_i}\}, \\ \{\mathbf{q}_i^h \in \mathbf{R}^{F \cdot \ell_i}\}}} h_x^2 \cdot \sum_{i=1}^{k} \langle \mathbf{u}_i^h, \mathbf{q}_i^h - \mathrm{Div}_x^h \mathbf{p}_i^h \rangle + \sum_{f \in \mathcal{F}} \mathbf{i}\{\mathbf{u}_i^h(f) \in \Delta^{\ell_i}\}, \tag{13}$$

$$\text{s.t.} \quad \|\mathbf{D}_{i,f,\alpha,t} \mathbf{p}_i^h\|_* \leq \lambda_i, \quad \forall i \in [k], f \in \mathcal{F}, \alpha \in \{0,1\}^2, t \in \mathcal{T}_i, \tag{14}$$

$$\sum_{i=1}^{k} \mathbf{W}_{i,f,\gamma_i} \mathbf{q}_i^h \leq c\left(x(f), \gamma\right), \forall f \in \mathcal{F}, \gamma \in \Gamma, \tag{15}$$

where $\mathbf{i}\{\cdot\}$ is the indicator function. In our applications, we found that it is sufficient to enforce the constraint (14) at the corners of each face which corresponds to choosing $\alpha \in \{0,1\}^2$. Apart from the infinitely many constraints in (15), this is a finite-dimensional convex-concave saddle-point problem and can be tackled by many numerical optimization algorithms.

3.2 Solution recovery

Before presenting in the next section how we propose to implement the constraints (15), we briefly discuss how a primal solution $\{\mathbf{u}_i^h\}$ of the above problem is turned into an approximate solution to (1). To that end, we follow Lellmann, Strekalovskiy, et al. (2013); Vogt et al. (2020) and compute the Riemannian center of mass via an iteration $\tau = 1, \dots, T$:

$$V_j^\tau = \log_{u_i^\tau}(\mathbf{v}_{i,j}), \tag{16}$$

$$v^\tau = \sum_{j=1}^{\ell_i} \mathbf{u}_i^h(f,j) V_j^\tau, \tag{17}$$

$$\mathbf{u}_i^{\tau+1} = \exp_{u_i^\tau}(v^\tau). \tag{18}$$

Here, $u_i^0 \in \Gamma_i$ is initialized by the label with the highest probability according to $\mathbf{u}_i^h(f,\cdot)$. $\log_{u_i^\tau}$ and $\exp_{u_i^\tau}$ denote the logarithmic and exponential mapping between Γ_i^h and its tangent space at $u_i^\tau \in \Gamma_i$, which are both available in closed-form for the manifolds we consider here. In our case $T = 20$ was enough to reach convergence. For flat manifolds, $T = 1$ is enough, as both mappings boil down to the identity and (18) computes a weighted Euclidean mean.

In general, there is no theory which shows that $u^T(x) = (u_1^T(x), \dots, u_k^T(x))$ from (18) is a global minimizer of (1). Tightness of the relaxation in the special case $k = 1$ and $\Gamma \subset \mathbf{R}$ is shown in Pock et al. (2010). For higher dimensional Γ, the tightness of related relaxations is ongoing research; see Ghoussoub et al. (2021) for results on the Dirichlet energy. By computing a-posteriori optimality gaps, solutions of (8) were shown to be typically near the global optimum of Problem (1); see, e.g., Goldluecke et al. (2013).

4 Implementation of the constraints

Though the optimization variables in (13) are finite-dimensional, the energy is still difficult to optimize because of the infinitely many constraints in (15).

Before we present our approach, let us first describe what we refer to as the *baseline method* for the remainder of this paper. As the baseline approach, we consider the direct solution of (13) where we implemented the constraints only at the label/discretization points $\mathcal{L}_1 \times \dots \times \mathcal{L}_k$ via Lagrange multipliers (this strategy is also employed by the global variant of the product-space approach (Goldluecke et al., 2013)). This baseline actually corresponds to a single outer iteration N_{it} of the proposed Algorithm 1, with a large number M_{it} of inner iterations.

We aim for a framework that allows for solving a better approximation of (15) than the baseline while being of similar memory complexity. To this end, Algorithm 1 alternates the following two steps:

1) Sampling. Based on the current solution we prune previously considered but feasible constraints and sample a new subset of the infinitely many constraints in (15). From all the current sampled constraints, we consider the most violated constraints for each face, add one sample at the current solution and discard the rest.

2) Solving the subsampled problem. Considering the current finite subset of constraints, we solve Problem (13) using a primal-dual method (Chambolle & Pock, 2011) with diagonal preconditioning (Pock & Chambolle, 2011). Both constraints (14) and (15) are implemented using Lagrange multipliers.

These two phases are performed alternatingly, with the aim to eventually approach the solution of the continuous problem (13). In practice, a fixed number of outer iterations N_{it} is set. While we do not prove convergence of the overall algorithm, convergence results for related procedures exist; see, e.g., Blankenship and Falk (1976), Theorem 2.4.

The detailed algorithm is explained in Algorithm 1. The cost matrix $\mathbf{C}$ is constructed by evaluating $c(x(f), \gamma)$ at proposed samples $\mathcal{S}_f$. We denote ξ and ν as the Lagrange multipliers. The Lagrange multiplier ξ is initialized by a warm-start strategy, i.e. ξ^{it} keeps same if we have the same proposed sample from previous outer iteraion. The prox of a function g with step size τ is defined as:

$$\operatorname{prox}_{\tau g}(x) = \arg\min_y \frac{1}{2\tau}\|x - y\| + g(y) \tag{19}$$

Our constraint sampling strategy is detailed in Algorithm 2. For each face in $\mathcal{F}$, it generates a finite set of "sublabels" $\mathcal{S}_f \subset \Gamma$ at which we implement the constraints (15). Next, we provide the motivation behind each line in the algorithm.

Algorithm 1 Proposed algorithm for problem (1).

Input: $c : \Omega \times \Gamma \to \mathbf{R}$, $\lambda_i > 0$, $N_{it} > 0$, $M_{it} > 0$, $n > 0$, $\delta > 0$, $r > 0$.

1: $\mathbf{u}_i^{h,0} = \mathbf{1}/\ell_i$, $\mathbf{q}_i^{h,0} = \mathbf{0}$, $\mathbf{p}_i^{h,0} = \mathbf{0}$.
2: $\mathcal{S}_f^0 = \mathcal{L}_1 \times \ldots \times \mathcal{L}_k$.
3: **for** $it = 0$ to N_{it} **do**
4: Construct interpolation matrix $\mathbf{W}_i^{it}$ and cost matrix $\mathbf{C}^{it}$ based on $\mathcal{S}_f^{it}$ from Algorithm 2.
5: Initialize ξ_i^{it} with the warm-start strategy.
6: Construct the diagonal preconditioners $\mathbf{T}_u^{it}$, $\mathbf{T}_\nu^{it}$, $\mathbf{T}_\xi^{it}$, $\mathbf{\Sigma}_p^{it}$ and $\mathbf{\Sigma}_q^{it}$ by Pock and Chambolle (2011).
7: **for** $j = 0$ to M_{it} **do**
8: $\mathbf{p}_i^{h,j+1} = \mathbf{p}_i^{h,j} + \mathbf{\Sigma}_p^{it}(-h_x^2 \cdot (\mathrm{Div}_x^h)^T \mathbf{u}_i^{h,j} + \mathbf{D}_{i,\alpha,t}^T \nu_i^j)$
9: $\mathbf{q}_i^{h,j+1} = \mathbf{q}_i^{h,j} + \mathbf{\Sigma}_q^{it}(h_x^2 \cdot \mathbf{u}_i^{h,j} - (\mathbf{W}_i^{it})^T \xi_i^j)$
10: $\bar{\mathbf{p}}_i^{h,j} = 2\mathbf{p}_i^{h,j+1} - \mathbf{p}_i^{h,j}$
11: $\bar{\mathbf{q}}_i^{h,j} = 2\mathbf{q}_i^{h,j+1} - \mathbf{q}_i^{h,j}$
12: $\mathbf{u}_i^{h,j+1} = \mathrm{prox}_{\cdot \in \Delta^{\ell_i}}(\mathbf{u}_i^{h,j} - \mathbf{T}_u^{it}(\bar{\mathbf{q}}_i^{h,j} - \mathrm{Div}_x^h \bar{\mathbf{p}}_i^{h,j}))$
13: $\nu_i^{j+1} = \mathrm{prox}_{\lambda \mathbf{T}_\nu^{it} \|\cdot\|_2}(\nu_i^j - \mathbf{T}_\nu^{it} \mathbf{D}_{i,v,t} \bar{\mathbf{p}}_i^{h,j})$
14: $\xi_i^{j+1} = \mathrm{prox}_{\cdot \geq 0}(\xi_i^j - \mathbf{T}_\xi^{it}(\mathbf{C}^{it} - \sum_i^k \mathbf{W}_i^{it} \bar{\mathbf{q}}_i^{h,j}))$
15: **end for**
16: Get sampled $\mathcal{S}_f^{it+1}$ for each face by Algorithm 2.
17: **end for**

Algorithm 2 Sampling strategy at face $f \in \mathcal{F}$.

Input: Solution $u = (u_1, \ldots, u_k)$ at face f, sublabel-set $\mathcal{S}_f$, n, δ, r.

/* global exploration */
1: $\mathcal{S}'_f \leftarrow$ uniformSample(Γ, n)
/* local exploration around solution */
2: $\mathcal{S}'_f \leftarrow \mathcal{S}'_f \cup$ localPerturb(u, δ, n)
/* remove feasible constraints */
3: $\mathcal{S}_f \leftarrow \{\gamma \in \mathcal{S}_f : \sum_{i=1}^k \mathbf{q}_i(f, \gamma) > c(f, \gamma)\}$
/* add the most violated r samples */
4: $\mathcal{S}_f \leftarrow$ top-k($\mathcal{S}'_f$, r) $\cup\, \mathcal{S}_f$
/* have one sample at current solution */
5: $\mathcal{S}_f \leftarrow \mathcal{S}_f \cup \{u\}$.
6: Return $\mathcal{S}_f$

Random uniform sampling (Line 1).

To have a global view of the cost function, we consider a uniform sampling on the label space Γ. The parameter $n > 0$ determines the number of the samples for each face.

Local perturbation around the mean (Line 2).

Besides the global information, we apply local perturbation around the current solution $\mathbf{u}$. In case the current solution is close to the optimal one, this strategy allows us to refine it with these samples. The parameter $\delta > 0$ determines the size of the local neighbourhood. In our experiments, we always used a Gaussian perturbation with $\delta = 0.1$.

Pruning strategy (Lines 3-4).

Most samples from previous iterations are discarded because the corresponding constraints are already satisfied. We prune all current feasible constraints as in Blankenship and Falk (1976). Similarly, the two random sampling strategies (Lines 1 and 2) might return some samples for which the constraints are already fulfilled. Therefore, we only consider the samples with violated constraints and pick the r most violated from them. This pruning strategy is essential for a memory efficient implementation as shown later.

Sampling at u (Line 5).

Finally, we add one sample which is exactly at the current solution $u \in \Gamma$ to have at least one guaranteed sample per face. In the next section, we illustrate the behavior of Algorithm 1 on a toy problem, and evaluate its performance on real-world imaging problems.

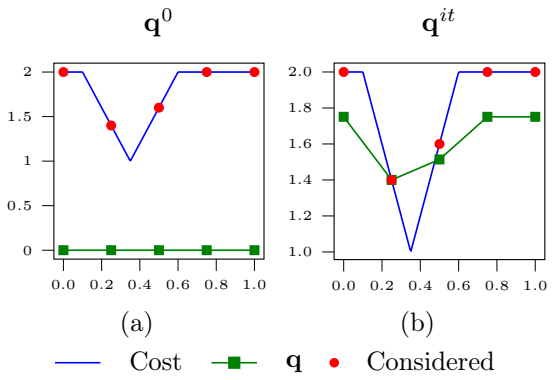


Fig. 2 Illustration of the baseline algorithm. (a) Five samples (red dots) from the labels are considered (b) The dual variable $\mathbf{q}$ satisfies the constraints on the samples. However, $\mathbf{q}$ is not globally optimal as it violates the constraint on the optimal solution ("green > blue").

5 Numerical validation

Our approach and the baseline are implemented in PyTorch. Code for reproducing the following experiments can be found here: https://github.com/zhenzhangye/sublabel_meets_product_space. Note that that one of the runtime bottlenecks of our sampling strategy is creating the samples and picking the most violated r as shown in Algorithm 2. Additionally, the sparse matrix operations and the PDHG updates can be more efficiently implemented in CUDA, as all PDHG updates can be executed in a single CUDA kernel, compared to PyTorch's multiple kernel calls. Therefore, a specialized implementation as in Goldluecke et al. (2013) will allow the method to scale by factor $10 - 100\times$ in favor of runtime.

5.1 Illustration of the Algorithm

First of all, we consider a simplistic minimization problem on a single nonconvex data term:

$$c(u) = \min \begin{cases} -4u + 2.4, & u \in [0.1, 0.35), \\ 4u - 0.4, & u \in [0.35, 0.6], \\ 2, & \text{otherwise}, \end{cases} \quad (20)$$

with 5 labels to illustrate the behavior of both, the baseline algorithm and our sampling strategies.

Figure 2 depicts the baseline's behavior. While it only evaluates the energy on the labels, five samples are considered as illustrated by the red dots in Figure 2 (a). Figure 2 (b) shows the dual variable $\mathbf{q}_i^h$ after it iterations. Since the algorithm is maximizing $\mathbf{q}_i^h$, the green and red dots should overlay (e.g. the second label) when it converges. Despite the convergence, the resulting $\mathbf{q}^h$ violates the constraints significantly close to the optimal solution ("green > blue"). Therefore, to attain the global optimal solution, the baseline approach needs more labels which requires more memory.

The motivation of random uniform sampling (Line 1, Algorithm 2) and local perturbation around the mean (Line 2, Algorithm 2) in our sampling strategy is intuitively clear. However, as demonstrated later in the experiment of a truncated quadratic energy, sampling at u (Line 5, Algorithm 2) is critical for the stability of our method. A comparison in Figure 3 helps to show the necessity of this strategy. We ran two experiments with the identical settings except for the sampling at u. After a given number of iterations, the dual variable $\mathbf{q}^h$ is approximately optimal, as indicated in Figure 3 (a). Our pruning strategy (Line 3, Algorithm 2) removes all of the proposed samples since all of them satisfy the constraints. As a result, the subproblem becomes unconstrainted on that dual variable $\mathbf{q}^h$ and its update has no significance, Figure 3 (b). To solve this problem, we propose to always at least have one sample at u even when $\mathbf{q}^h$ is nearly optimal, cf. Figure 3 (c). As illustrated in Figure 3 (d), this can avoid the degeneration of $\mathbf{q}^h$ as it is still constrained.

Finally, the complete sampling strategy is illustrated in Figure 4. As shown in Figure 4 (a), the primal-dual method can obtain the optimal $\mathbf{q}^h$ for the sampled subproblem. Our sampling strategy can provide necessary samples and prune the feasible ones, cf. Figure 4 (b). These few but meaningful samples lead the $\mathbf{q}^h$ to achieve global optimality, cf. Figure 4 (c).

5.2 Truncated Quadratic Energy

In this section, we study the numerical effect of each line in Algorithm 2. We evaluate our method on the truncated quadratic energy $c(x, u(x)) = \min\{(u(x) - f(x))^2, \nu\}$. where $f : \Omega \to \mathbf{R}$ is the input data as show in Figure 5. For this specific experiment, we generate a 64×64 gray image degraded with Gaussian noise of standard deviation $\sigma = 0.05$ and 5% salt-and-pepper noise. The parameters are chosen as $\nu = 0.025$, $\lambda = 0.25$, $N_{it} = 10$, $M_{it} = 200$, $n = 10$ and $r = 1$. To reduce

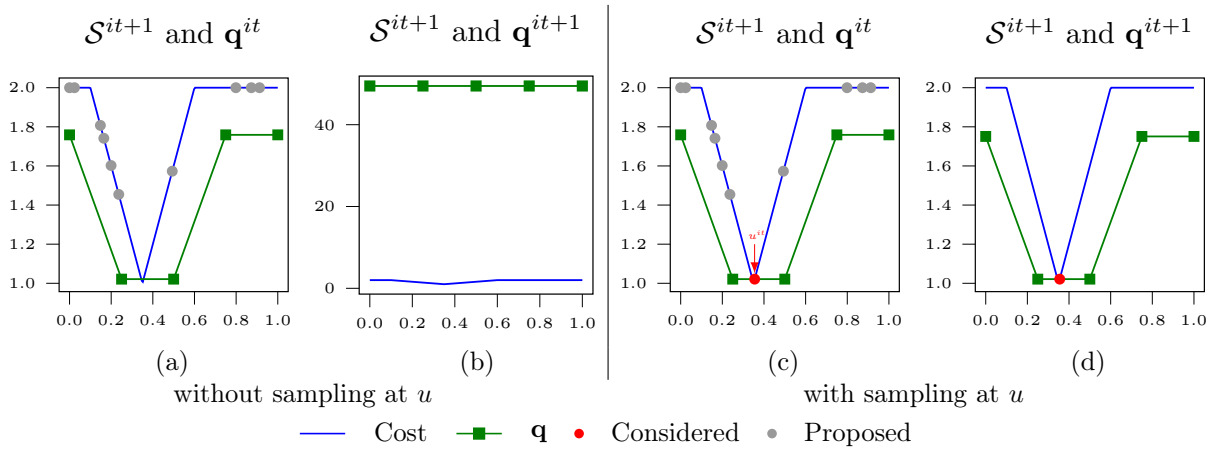


Fig. 3 Benefit of sampling at u (Line 5, Algorithm 2). (a) Because all of the proposed samples (gray dots) fullfil the constraint, they are all pruned (Line 3, Algorithm 2). (b) The updated $\mathbf{q}$ deteriorates because the subproblem is unconstrainted on it. (c) At least one sample is taken (namely u^{it}, red dot). (d) This sample constraints $\mathbf{q}$ and thus prevents a degenerate solution.

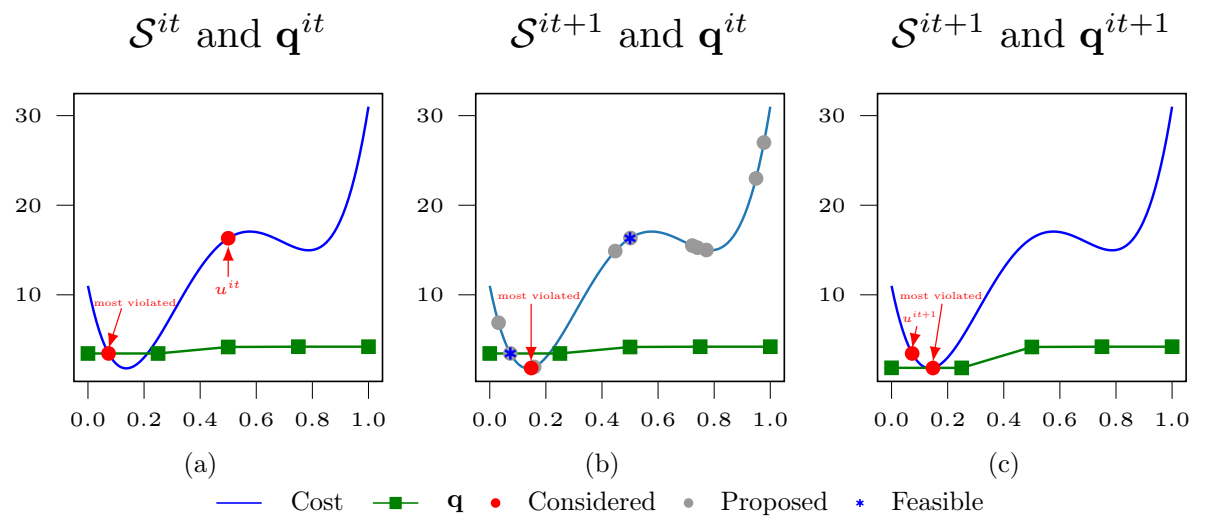


Fig. 4 Illustration of our proposed sampling strategies. (a) Two samples (red dots) are considered leading to the shown optimal dual variable $\mathbf{q}$ after running primal-dual iterations. (b) The two samples are pruned because the constraints are feasible. Several random samples are proposed (gray dots) and only one of them is picked (red dot). (c) One more sample on u^{it} is added and the $\mathbf{q}$ is refined.

the effect of randomness, we run each algorithm 20 times and report mean and standard deviation of the final energy for different number of labels in Table 1. We want to emphasize that more labels have benefits for both baseline and our algorithm. Nevertheless, the proposed approach can reach lower energies with the same number of labels and similar memory requirements.

As can be seen in this table, adding uniform sampling and picking the most violated constraint per face (Lines 1 and 4 of Algorithm 2) already decreases the final energy significantly. We also consider local exploration around the current solution (Line 2), which helps to find better energies

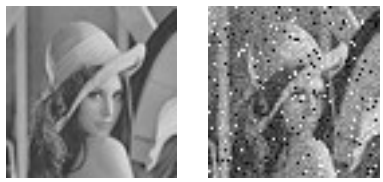


Fig. 5 (a) The original 64×64 grayscale image. (b) Degraded with Gaussian noise of standard deviation $\sigma = 0.05$ and 5% salt-and-pepper noise.

	Labels	Baseline	+Line 1&4	+Line 2	+Line 3	+Line 5
Energy	3	4589 (±0.00)	2305 (±3.73)	2291 (±3.6)	8585 (±130.4)	2051 (±10.7)
Time [s]		8.98	22.77	23.22	23.22	23.33
Mem. [Mb]		11.21	13.94	15.53	11.65	12.05
Energy	7	2582 (±0.00)	2020 (±2.68)	2012 (±1.3)	7209 (±116.7)	1969 (±3.6)
Time [s]		74.13	16.02	16.61	15.56	18.38
Mem. [Mb]		28.35	32.96	33.49	28.356	28.68
Energy	13	2029 (±0.00)	1935 (±1.14)	1926 (±0.7)	5976 (±75.7)	1901 (±3.7)
Time [s]		183.80	37.65	38.84	38.29	38.22
Mem. [Mb]		52.85	60.55	60.94	54.35	54.73

Table 1 Ablation study indicating the effect of individual lines in Algorithm 2. Numbers in parentheses indicate the standard deviation across 20 runs.

at the expense of higher memory requirements. The pruning strategy (Line 3) circumvents this memory issue, however the energy deteriorates dramatically because some faces could end up having no samples after pruning. Therefore, keeping the current solution as a sample (Line 5) per face prevents the energy from degrading. Including all these sampling strategies, the proposed method can achieve the best energy and runtime, at comparable memory usage to the baseline method.

We further illustrate the comparison on the number of iterations and time between the baseline and our proposed method in Figure 6. Due to the replacement on the samples, we have a peak right after each sampling phase. The energy however converges immediately, leading to an overall decreasing trend.

Additionally, we compare our method to the baseline on a more practical dataset CBSD from Martin et al. (2001). This dataset contains 68 images and noisy ones with additive white Gaussian noise (5% in this experiment). The number of labels is 7 for both methods. 5K iterations are performed on the baseline method, while we set $N_{it} = 10$ and $M_{it} = 500$ to get a fair comparison. The other parameters are chosen as $\lambda = 0.25$, $n = 50$ and $r = 1$. The results are shown in Figure 7. Our method outpeforms the baseline among all the images regarding both energy and peak signal-to-noise ratio (PSNR).

5.3 Manifold-value Denoising

To show the flexibility of our algorithm, we next evaluate it on a manifold-valued denoising problem in HSV color space. The hue component of this space is a circle, i.e., $\Gamma_1 = \mathbb{S}^1$, $\Gamma_2, \Gamma_3 = [0, 1]$.

The data term of this experiment is still a truncated quadratic energy, where for the hue component the distance is taken on the circle $\mathbb{S}^1$. The input images (Baker et al., 2011; Martin et al., 2001) are degraded with the same setting as above.

Both the baseline and our method are implemented with 7 labels. First of all, we evaluate the impact of the most violated r samples. As shown in Figure 8, The maximum difference of energy and memory is only 0.8% and 0.06%, respectively, which can be considered almost constant wrt. r. Therefore, we pick $r = 5$. To get an equal number of total iterations, 30K iterations are performed on the baseline, while we set $N_{it} = 100$ outer iterations with $M_{it} = 300$ inner primal-dual steps for our method. Other parameters are chosen as $\lambda = 0.015$ and $n = 30$. As shown in Figure 9, our method can achieve a lower energy than the baseline. Qualitatively, since our method implements the constraints not only at the labels but also inbetween, there is less bias.

5.4 Optical flow

Given two input images I_1, I_2, we compute the optical flow $u : \Omega \to \mathbf{R}^2$ for the label space $\Gamma = [a, b]^2$. We use a simple ℓ_1-norm for the data term, i.e. $c(x, u(x)) = \|I_2(x) - I_1(x + u(x))\|_1$ and set the regularization weight as $\lambda = 0.04$. The baseline approach runs for 50K iterations, while we set $N_{it} = 50$ and $M_{it} = 1000$ for a fair comparison. Additionally, the parameters are chosen as $n = 20$ and $r = 1$ in Algorithm 2.

We consider three methods for this experiment: the baseline, the method from Laude et al. (2016) and ours. The method from Laude et al. (2016) approximates the dataterm in a piecewise convex manner and requires a specific epigraph projection for the dataterm. Though it can attain lower energy, our approach requires less memory and tackle any cost function in a simple black-box fashion. Table 2 summarizes the quantitative results obtained on the Middleburry dataset (Baker et al.,

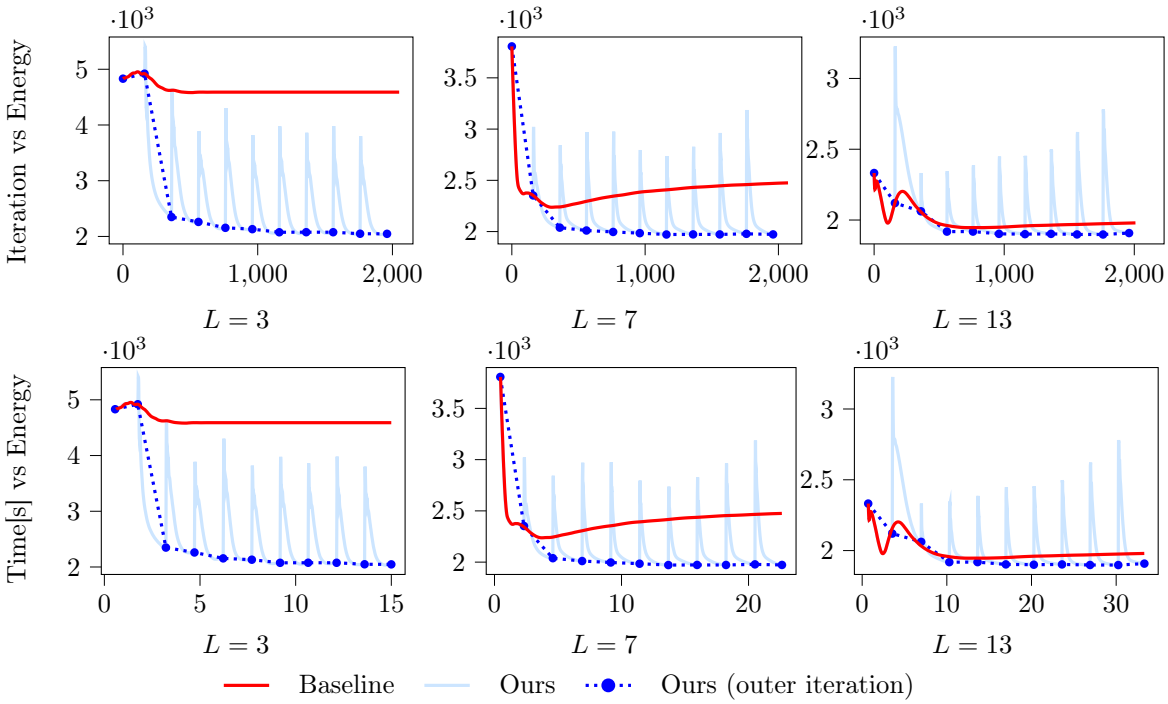


Fig. 6 Comparison between the baseline and our approach on a 64×64 grayscale image shown in Figure 5, degraded with Gaussian and salt-and-pepper noise. Our approach finds lower energies in fewer iterations and less time than the baseline, which implements the constraints only at the label points.

#Labels	3	7	11	15	19
Rel. Energy	50.91%(±6.54%)	64.15%(±9.83%)	73.68%(±11.17%)	79.40%(±9.70%)	80.69%(±11.23%)
Rel. Memory	99.84%(±0.75%)	100.02%(±0.07%)	99.99%(±0.04%)	99.99%(±0.03%)	99.98%(±0.04%)
Rel. aep	91.94%(±7.79%)	99.92%(±3.17%)	98.67%(±1.63%)	99.43%(±1.78%)	100.71%(±1.16%)
Rel. aae	82.34%(±8.68%)	95.19%(±5.74%)	94.93%(±6.15%)	96.55%(±5.85%)	100.86%(±3.97%)

Table 2 We compute the optical flow on the Middlebury dataset Baker et al. (2011) using our method and the baseline for a varying amount of labels. Given an equal number of labels/memory, our sampling strategy performs favorably to an implementation of the constraints at the labels. The relative numbers of energy, memory, average endpoint error (aep) and average angular error (aae) are calculated as "mean($\frac{\text{Ours}}{\text{Baseline}}$)" across all 8 datasets. The number in the parentheses resemble the standard deviation. The detailed table with all results can be found in the appendix.

2011), while the detailed absolute numbers can be found in the appendix. This table shows how our approach performs relatively to the baseline, i.e. "mean($\frac{\text{Ours}}{\text{Baseline}}$)", e.g. for three labels our energy is 50.91% of the baseline energy for all 8 datasets, while using 99.84% of the baseline's memory and having an average end point error (aep) and average angular error (aae) of 91.94% and 82.34% of the baseline error metrics, respectively. To enable qualitative comparison, we visualize in Figure 10 the results on two of the datasets. The remaining qualitative results on the Middlebury data set (Baker et al., 2011) are shown in the appendix. Our method outperforms the baseline approach regarding energy under the same number of labels and requires the same amount of memory. Because Laude et al. (2016) uses a tighter relxation on the label space, they can achieve lower energy with a smaller number of labels. However, it runs out of memory easily while the proposed method scales better wrt. memory consumption.

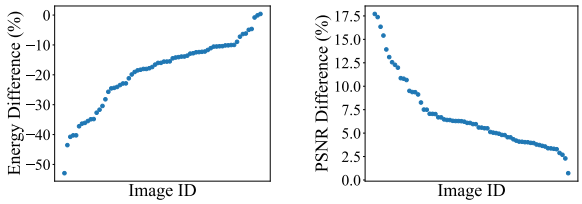
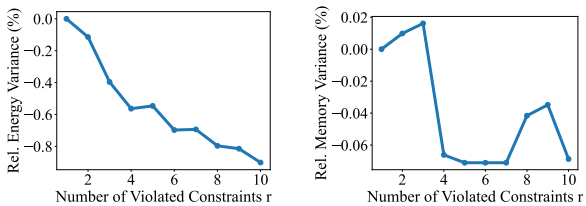


Fig. 7 Quantitative results on the CBSD dataset (Martin et al., 2001). We ordered both results from the best to the worst. The difference is calculated using the formula $\frac{\text{ours - baseline}}{\text{baseline}}$. It is clear that our method consistently outperforms the baseline across all images.



(a) (b)

Fig. 8 Comparison between different number of most violated constraints on the energy and memory: (a) the change of the energy with $r = 1$ as the basis; (b) the change of memory with $r = 1$ as the basis. It can be observed that both the energy and memory vary very little (0.8% and 0.06%, respectively) regarding different r.

6 Conclusion and Limitations

In this paper we made functional lifting methods more scalable by combining two advances, namely product-space relaxations (Goldluecke et al., 2013) and sublabel-accurate discretizations (Möllenhoff & Cremers, 2017; Vogt et al., 2020). This combination is enabled by adapting a cutting-plane method from semi-infinite programming (Blankenship & Falk, 1976). This allows an implementation of sublabel-accurate methods without difficult epigraphical projections.

Moreover, our approach makes sublabel-accurate functional-lifting methods applicable to any cost function in a simple black-box fashion. In experiments, we demonstrate the effectiveness of the approach over a baseline based on the product-space relaxation (Goldluecke et al., 2013) and provided a proof-of-concept experiment showcasing the method in the manifold-valued setting.

Future work will concentrate on applying and adapting the presented framework to solve large inverse problems in computer vision with multiple data terms, different regularizers and several manifold-valued optimization variables in a joint fashion. However, it is not obvious if our presented cutting plane approach is easily applicable for such large problems or if novel ideas have to be pursued.

Acknowledgements This work was supported by the ERC Advanced Grant SIMULACRON.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Bach, F. (2019). Submodular functions: from discrete to continuous domains. *Math. Progam.*, *175*(1), 419–459.

Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R. (2011). A database and evaluation methodology for optical flow. *Int. J. Comput. Vis. (IJCV)*, *92*(1), 1–31.

Bauermeister, H., Laude, E., Möllenhoff, T., Moeller, M., Cremers, D. (2021). *Lifting the convex conjugate in lagrangian relaxations: A tractable approach for continuous Markov random fields.* arXiv:2107.06028.

Blankenship, J.W., & Falk, J.E. (1976). Infinitely constrained optimization problems. *J. Optim. Theory Appl.*, *19*(2), 261–281.

Boyd, S.P., Parikh, N., Chu, E., Peleato, B., Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, *3*(1), 1–122.

Caillaud, C., & Chambolle, A. (2020). *Error estimates for finite differences approximations of the total variation.* preprint hal-02539136.

Carlier, G. (2003). On a class of multidimensional optimal transportation problems. *J. Convex*

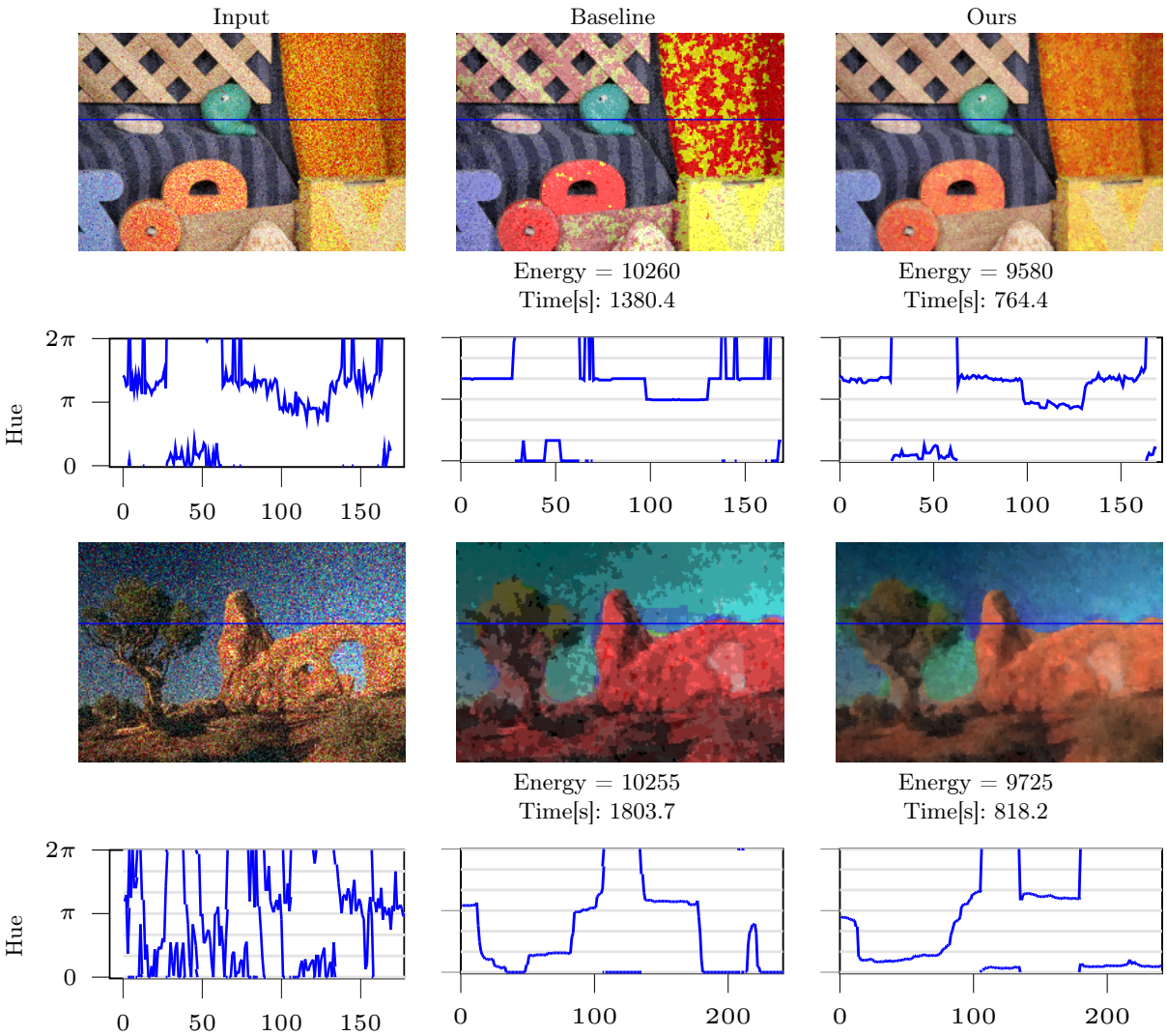


Fig. 9 Denoising of images (Baker et al., 2011; Martin et al., 2001) in HSV color space ($\Gamma_1 = \mathbb{S}^1$, $\Gamma_2 = \Gamma_3 = [0, 1]$) using our method and the baseline with 7 labels. The hue component plot demonstrates that our approach is able to handle the manifold setting where the jump from 2π to 0 is permitted. Since our approach implements the constraints adaptively inbetween the labels (gray lines) it reaches a lower energy with less label bias.

Anal., *10*(2), 517–530.

Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T. (2010). An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, *9*(263-340), 227.

Chambolle, A., Cremers, D., Pock, T. (2012). A convex approach to minimal partitions. *SIAM J. Imaging Sci.*, *5*(4), 1113–1158.

Chambolle, A., & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, *40*, 120–145.

Cremers, D., & Strekalovskiy, E. (2013). Total cyclic variation and generalizations. *J. Math. Imaging Vis.*, *47*(3), 258–277.

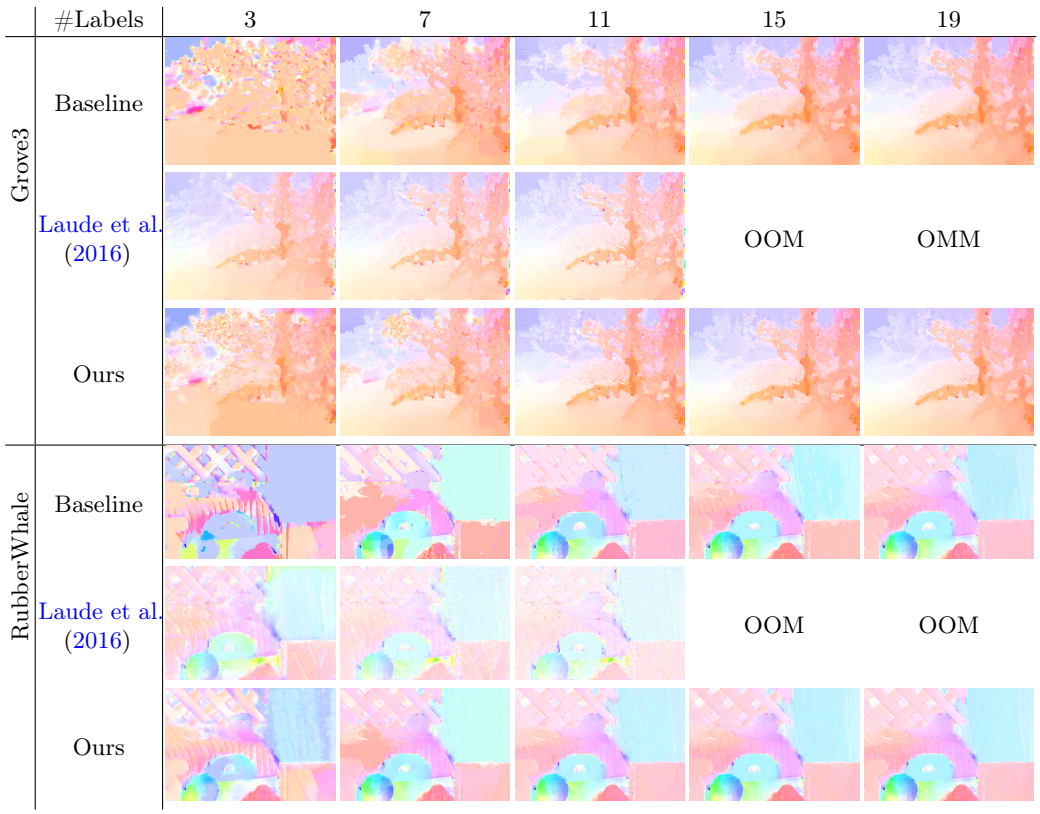


Fig. 10 Visualization of the optical flow on Grove3 and RubberWhale from the Middleburry dataset (Baker et al., 2011), using the baseline, the method from Laude et al. (2016) and our method for a varying amount of labels. OOM stands for out of memory. More qualitative results can be found in the appendix.

de Boer, P., Kroese, D.P., Mannor, S., Rubinstein, R.Y. (2005). A tutorial on the cross-entropy method. *Ann. Oper. Res.*, *134*(1), 19–67.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Fix, A., & Agarwal, S. (2014). Duality and the continuous graphical model. *European Conference on Computer Vision (ECCV)*.

Ghoussoub, N., Kim, Y.-H., Lavenant, H., Palmer, A.Z. (2021). Hidden convexity in a problem of nonlinear elasticity. *SIAM J. Math. Anal.*, *53*(1), 1070–1087.

Goldluecke, B., Strekalovskiy, E., Cremers, D. (2013). Tight convex relaxations for vector-valued labeling. *SIAM J. Imaging Sci.*, *6*(3), 1626–1664.

Görlitz, A., Geiping, J., Kolb, A. (2019). Piecewise rigid scene flow with implicit motion segmentation. *International Conference on Intelligent Robots and Systems (IROS)*.

Horn, B.K., & Schunck, B.G. (1981). Determining optical flow. *Artificial Intelligence*, *17*(1-3), 185–203.

Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, *25*(10), 1333-1336.

Kappes, J., Andres, B., Hamprecht, F., Schnorr, C., Nowozin, S., Batra, D., ... others (2013). A comparative study of modern inference techniques for discrete energy minimization problems. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Lasserre, J.-B. (2000). Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, *11*(3), 796–817.

Laude, E., Möllenhoff, T., Moeller, M., Lellmann, J., Cremers, D. (2016). Sublabel-accurate convex relaxation of vectorial multilabel energies. *European Conference on Computer Vision (ECCV).*

Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C. (2009). Convex multi-class image labeling by simplex-constrained total variation. *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)* (pp. 150–162).

Lellmann, J., Lellmann, B., Widmann, F., Schnörr, C. (2013). Discrete and continuous models for partitioning problems. *Int. J. Comput. Vis. (IJCV)*, *104*(3), 241–269.

Lellmann, J., & Schnörr, C. (2011). Continuous multiclass labeling approaches and algorithms. *SIAM J. Imaging Sci.*, *4*(4), 1049–1096.

Lellmann, J., Strekalovskiy, E., Koetter, S., Cremers, D. (2013). Total variation regularization for functions with values in a manifold. *International Conference on Computer Vision (ICCV).*

Martin, D., Fowlkes, C., Tal, D., Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *International Conference on Computer Vision (ICCV)* (Vol. 2, pp. 416–423).

Möllenhoff, T., Laude, E., Moeller, M., Lellmann, J., Cremers, D. (2016). Sublabel-accurate relaxation of nonconvex energies. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Möllenhoff, T., & Cremers, D. (2017). Sublabel-accurate discretization of nonconvex free-discontinuity problems. *International Conference on Computer Vision (ICCV).*

Möllenhoff, T., & Cremers, D. (2019). Lifting vectorial variational problems: a natural formulation based on geometric measure theory and discrete exterior calculus. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Ollivier, Y., Arnold, L., Auger, A., Hansen, N. (2017). Information-geometric optimization algorithms: A unifying picture via invariance principles. *J. Mach. Learn. Res.*, *18*, 18:1–18:65.

Peng, J., Hazan, T., McAllester, D., Urtasun, R. (2011). Convex max-product algorithms for continuous MRFs with applications to protein folding. *International Conference on Machine Learning (ICML).*

Pock, T., & Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. *International Conference on Computer Vision (ICCV).*

Pock, T., Cremers, D., Bischof, H., Chambolle, A. (2009). An algorithm for minimizing the piecewise smooth Mumford-Shah functional. *International Conference on Computer Vision (ICCV).*

Pock, T., Cremers, D., Bischof, H., Chambolle, A. (2010). Global solutions of variational models with convex regularization. *SIAM J. Imaging Sci.*, *3*(4), 1122–1145.

Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D. (2008). A convex formulation of continuous multi-label problems. *European*

Conference on Computer Vision (ECCV).

Rudin, L.I., Osher, S., Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, *60*(1-4), 259–268.

Schaul, T. (2011). *Studies in continuous black-box optimization* (Unpublished doctoral dissertation). Technische Universität München.

Steinke, F., Hein, M., Schölkopf, B. (2010). Nonparametric regression between general Riemannian manifolds. *SIAM J. Imaging Sci.*, *3*(3), 527–563.

Strekalovskiy, E., Chambolle, A., Cremers, D. (2014). Convex relaxation of vectorial problems with coupled regularization. *SIAM J. Imaging Sci.*, *7*(1), 294–336.

Villani, C. (2008). *Optimal transport: Old and new.* Springer.

Vogt, T., Strekalovskiy, E., Cremers, D., Lellmann, J. (2020). Lifting methods for manifold-valued variational problems. *Handbook of variational methods for nonlinear geometric data.* Springer.

Weinmann, A., Demaret, L., Storath, M. (2014). Total variation regularization for manifold-valued data. *SIAM J. Imaging Sci.*, *7*(4), 2226–2257.

Ye, Z., Haefner, B., Quéau, Y., Möllenhoff, T., Cremers, D. (2021). Sublabel-accurate multilabeling meets product label spaces. *DAGM German Conference on Pattern Recognition (GCPR).*

Zach, C. (2013). Dual decomposition for joint discrete-continuous optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS).*

Zach, C., Gallup, D., Frahm, J.-M., Niethammer, M. (2008). Fast global labeling for real-time stereo using multiple plane sweeps. *Proceedings of the Vision, Modeling and Visualization Workshop (VMV).*

Zach, C., & Kohli, P. (2012). A convex discrete-continuous approach for Markov random fields. *European Conference on Computer Vision (ECCV).*

A Additional Optical Flow Results

Table 3 shows the energy and memory requirement as well as the average endpoint error (aep) and average angular error (aae) for the baseline, the method from Laude et al. (2016) and our approach across different number of labels for all 8 used Middlebury datasets (Baker et al., 2011). Figure 11 and Figure 12 visualize the remaining optical flow results on the Middlebury dataset (Baker et al., 2011) using the baseline, the method from Laude et al. (2016) and ours for a varying amount of labels. Note that the approach from Laude et al. (2016) leverages a tighter discretization on the label space and is implemented on CUDA. Though their approach achieves better energy under fewer labels, ours has a better scability.

#Labels		3			11			19		
		Baseline	Laude et al. (2016)	Ours	Baseline	Laude et al. (2016)	Ours	Baseline	Laude et al. (2016)	Ours
Dimetrodon	Energy	8326.07	3233.34	4789.68	4607.09	2805	3982.92	4161.21		3886.04
	Memory [Mb]	244.45	573	245.43	1795.51	10296	1795.76	4583.72		4582.42
	Time [s]	83.86	470.3	904.6	967.3	2274.7	2881.9	1375.1	OOM	3883.3
	aep [px]	1.39	1.15	1.31	1.08	1.15	1.08	1.07		1.08
	aae [°]	0.57	0.45	0.49	0.34	0.48	0.34	0.33		0.34
Grove2	Energy	24919.9	9815.9	10873.84	10344.75	8798.9	7679.19	8600.58		7340.10
	Memory [Mb]	330.25	786	331.43	2426.97	13941	2426.33	6220.34		6220.34
	Time [s]	294.7	606.1	1195.2	911.2	3501.2	3801.2	1375.1	OOM	3883.3.4
	aep [px]	1.74	1.41	1.88	1.77	1.64	1.74	1.80		1.78
	aae [°]	0.48	0.26	0.46	0.37	0.34	0.35	0.38		0.37
Grove3	Energy	56730.6	14789	31917.5	25943.1	14039	12891.6	16402.7		10451.8
	Memory [Mb]	334.9	783	334.9	2427.3	13926	2427.3	6220.6		6213.1
	Time [s]	998.1	579.9	1195.6	923.3	3926.8	3809.3	1245.9	OOM	3919.3
	aep [px]	3.08	1.97	2.58	2.19	2.11	2.09	2.09		2.07
	aae [°]	0.79	0.28	0.65	0.37	0.33	0.31	0.31		0.30
Hydrangea	Energy	36943.87	11598	17622.77	9857.40	5634	6829.89	7793.58		6416.15
	Memory [Mb]	244.27	586	245.29	1795.06	10320	1795.59	4582.12		4582.42
	Time [s]	850.2	394.3	904.0	381.0	2824.7	2890.6	443.6	OOM	2713.9
	aep [px]	3.01	2.56	2.62	1.89	1.67	1.87	1.78		1.81
	aae [°]	0.85	0.65	0.63	0.26	0.27	0.25	0.23		0.24
RubberWhale	Energy	13142.73	4526	6069.95	6375.56	3948	5198.42	5502.00		5020.85
	Memory [Mb]	246.38	576	246.52	1796.18	10303	1795.76	4583.00		4582.27
	Time [s]	170.8	442.9	905.8	806.9	2810.6	2881.8S	1091.6	OOM	2704.9
	aep [px]	0.94	0.76	0.81	0.71	0.72	0.70	0.68		0.69
	aae [°]	0.60	0.45	0.48	0.39	0.42	0.38	0.37		0.37
Urban2	Energy	31765.19	10153	18200.39	11083.50	9562	8775.62	7694.50		6523.03
	Memory [Mb]	332.54	760	333.05	2426.33	13863	2426.33	6220.34		6220.34
	Time [s]	1303.9	543.9	1196.6	876.7	3410.7	3806.6	938.9	OOM	3926.3
	aep [px]	5.79	6.01	5.38	4.62	5.91	4.51	4.24		4.26
	aae [°]	0.85	0.36	0.72	0.32	0.39	0.28	0.23		0.22
Urban3	Energy	25991.05	7901	14622.53	10966.07	6696	7809.14	10303.19		6587.87
	Memory [Mb]	332.63	763	328.53	2426.97	13848	2426.79	6220.34		6220.34
	Time [s]	1302.7	677.8	1194.8	786.3	3904.7	3805.6	1019.3	OOM	3925.1
	aep [px]	4.83	3.91	4.63	4.10	4.30	4.10	4.06		4.11
	aae [°]	0.68	0.32	0.60	0.34	0.37	0.34	0.31		0.32
Venus	Energy	24589.15	8203	10429.92	10475.71	4151	8155.96	9775.23		7897.65
	Memory [Mb]	173.16	404	170.64	1263.71	7247	1262.51	3231.28		3230.04
	Time [s]	369.7	256.8	669.3	396.5	1957.8	2135.2	573.1	OOM	2026.8
	aep [px]	2.63	2.28	2.30	2.08	2.21	2.09	2.08		2.12
	aae [°]	0.74	0.45	0.50	0.32	0.42	0.32	0.31		0.33

Table 3 We compute the optical flow on the Middlebury dataset (Baker et al., 2011) using the baseline, the method from Laude et al. (2016) and ours for a varying amount of labels. We denote the out of memory error as OOM. Given an equal number of labels/memory, our sampling strategy performs favorably to an implementation of the constraints at the labels comparing to the baseline. Additionally, our method obtains a better scalibilty than the one from Laude et al. (2016).

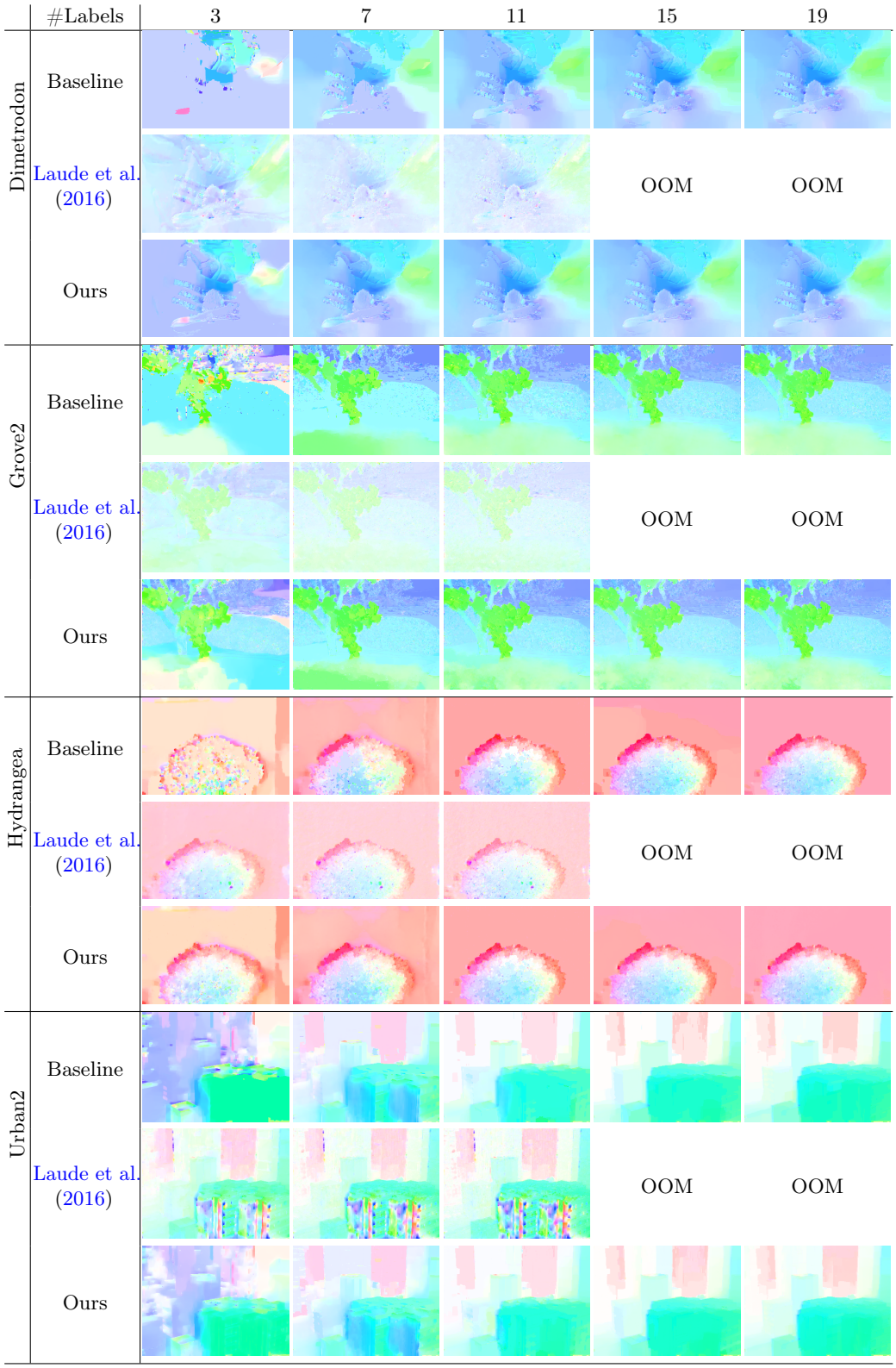


Fig. 11 Part I: We visualize the optical flow on the Middlebury dataset (Baker et al., 2011) using baseline, the method from Laude et al. (2016) and ours for a varying amount of labels for qualitative inspection. OOM stands for out of memory.

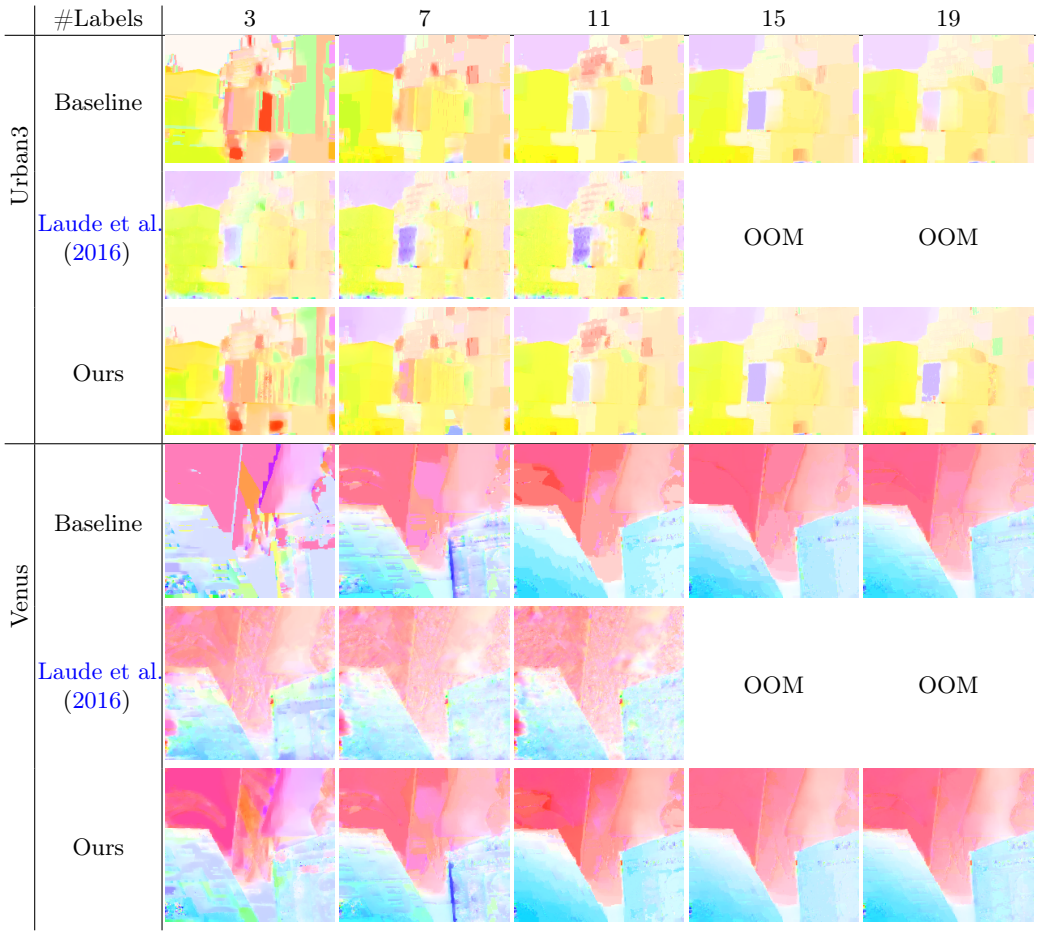


Fig. 12 Part II: We visualize the optical flow on the Middlebury dataset (Baker et al., 2011) using baseline, the method from Laude et al. (2016) and ours for a varying amount of labels for qualitative inspection. OOM stands for out of memory.