Noname manuscript No.
(will be inserted by the editor)

Tight Convex Relaxations for Vector-Valued Labeling

Bastian Goldluecke · Evgeny Strekalovskiy · Daniel Cremers

Received: date / Accepted: date

Abstract Multi-label problems are of fundamental importance in computer vision and image analysis. Yet, finding global minima of the associated energies is typically a hard computational challenge. Recently, progress has been made by reverting to spatially continuous formulations of respective problems and solving the arising convex relaxation globally. In practice this leads to solutions which are either optimal or within an a posteriori bound of the optimum. Unfortunately, in previous methods, both run time and memory requirements scale linearly in the total number of labels, making them very inefficient and often inapplicable for problems with higher dimensional label spaces.

In this paper, we propose a reduction technique for the case that the label space is a continuous product space, and introduce proper regularizers. The resulting convex relaxation requires orders of magnitude less memory and computation time than previously, which enables us to apply it to large-scale problems like optic flow, stereo with occlusion detection, segmentation into a very large number of regions, and joint denoising and local noise estimation. Despite the drastic gain in performance, we do not arrive at less accurate solutions than the original relaxation. Using the novel method, we can for the first time efficiently compute solutions to the optic flow functional which are within provable bounds (typically 5%) of the global optimum.

B. Goldluecke
Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
Tel.: (+49) 89 / 289-17781
E-mail: bastian.goldluecke@in.tum.de

E. Strekalovskiy
Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
Tel.: (+49) 89 / 289-17750
E-mail: evgeny.strekalovskiy@in.tum.de

D. Cremers
Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
Tel.: (+49) 89 / 289-17755
E-mail: daniel.cremers@in.tum.de

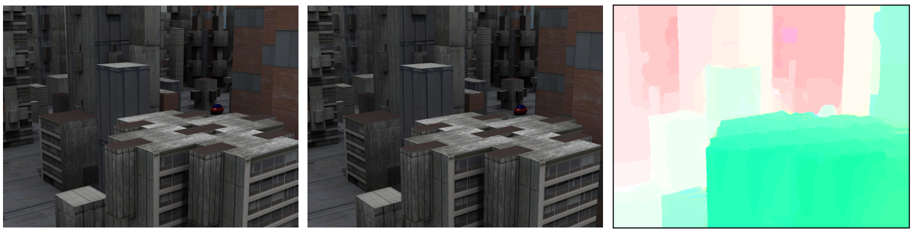


Fig. 1: *The proposed relaxation method can approximate the solution to multi-labeling problems with a huge number of possible labels by globally solving a convex relaxation model. This example shows two images and the optic flow field between them, where flow vectors were assigned from a possible set of* 50×50 *vectors, with truncated linear distance as a regularizer. The problem has so many different labels that a solution cannot be computed by alternative relaxation methods on current hardware.*

1 Introduction

1.1 The Multi-labeling Problem

Recently, there has been a surge of research activity on convex relaxation techniques for energy minimization in computer vision. Particular efforts were directed towards binary and multilabel problems, as they lie at the heart of fundamental problems like segmentation [19,18,7,29], stereo [22], 3D reconstruction [9], Mumford-Shah denoising [21] and optic flow [11].

The aim is to assign to each point x in an image domain $\Omega \subset \mathbb{R}^n$ a *label* from a set $\Gamma \subset \mathbb{R}^d$. Assigning the label $\gamma \in \Gamma$ to x is associated with the *cost* $c^\gamma(x) = c(x, \gamma) \in \mathbb{R}$. In computer vision applications, the local costs usually denote how well a given labeling fits some observed data. They can be arbitrarily sophisticated, for instance derived from statistical models or complicated local matching scores, our only assumptions being that the cost functions c^γ lie in the Hilbert space of square integrable functions $\mathcal{L}^2(\Omega)$. Aside from minimizing the local costs, we want the optimal assignment to exhibit a certain regularity. We enforce this requirement by penalizing each possible labeling $\boldsymbol{u} : \Omega \to \Gamma$ with a *regularization* or *prior term* $J(\boldsymbol{u}) \in \mathbb{R}$. This prior reflects our knowledge about which label configurations are a priori more likely, and typically enforces a form of spatial coherence.

Finding a labeling $\boldsymbol{u} : \Omega \to \Gamma$ which minimizes the sum of data term and regularizer, i.e.

$$\underset{\boldsymbol{u} \in \mathcal{L}^2(\Omega, \Gamma)}{\operatorname{argmin}} \left\{ J(\boldsymbol{u}) + \int_\Omega c(x, \boldsymbol{u}(x)) \, \mathrm{d}x \right\} \tag{1}$$

is a hard computational challenge as the overall energy is not convex. For some cases, good results may be obtained by local minimization, starting from a good initialization, possibly further improved by coarse-to-fine strategies commonly employed in optical flow estimation. Yet, such methods cannot guarantee any form of quality of the result and performance typically depends on data, on initialization and on the choice of algorithmic minimization scheme (number of levels in the coarse-to-fine hierarchy, number of iterations per level, etc.). The goal of this paper is to develop solutions to such problems which do not depend on initialization and which lie within a computable bound of the global optimum.

1.2 Contribution: Product label spaces

In this work, we discuss label spaces which can be written as a product of a finite number d of spaces, $\Gamma = \Lambda_1 \times \cdots \times \Lambda_d$. The central idea is as follows. Assume that the spaces Λ_k are discrete or have been discretized, and let N_k be the number of elements in Λ_k. Then the total number of labels is $N = N_1 \cdot \ldots \cdot N_d$. In previous relaxations for the multi-label problem, this means that we need to optimize over a number of N binary indicator functions, which can be rather large in practical problems. In order to make problems of this form feasible to solve, we present a reduction method which only requires $N_1 + \cdots + N_d$ binary functions. As a consequence, both memory and computation time grow linearly (rather than exponentially) in the number of dimensions.

We will show that with this novel reduction technique, it is possible to efficiently solve convex relaxations to multi-label problems which are far too large to approach with existing techniques. A prototypical example is optic flow, where the total number of labels is typically around 32^2 for practical problems. In that case we only require 64 indicator functions instead of 1024. However, the proposed method applies to a much larger class of labeling problems. This reduction in variable size not only allows for substantially higher resolution of the label space, but it also gives rise to a drastic speedup.

The present paper is a significantly revised and extended version of our original conference paper [11]. Compared to this early version, we make a number of important additional contributions:

- The regularizer in [11] was based on the relaxation in [18] for multilabel problems with a discrete set of labels, which is known to be less tight than the relaxation introduced in [7] for continuous label spaces. In contrast, we propose in this paper a general framework for convex relaxations of multilabel problems, which is based on a *continuous, multi-dimensional label space* and the calibration method detailed in [1]. It allows to use additional tight regularizers in in the case of product spaces, with arbitrary choice of regularizer for each label dimension.
- We establish that the previous regularization based on Euclidean representations of the label distance can be recovered as a special case from the discretization of our framework. Interestingly, from this new point of view we also obtain a relaxation which is provably tighter.
- The relaxation of the data term in [11] was suboptimal in that it introduces an unwanted trivial solution when relaxing from binary to continuous labels, which had to be avoided by an additional smoothing degrading the quality of solutions. In this paper, we propose a novel convex relaxation of the data term which does not suffer from these problems.
- The new framework yields solutions which are provably closer to the global optimum. It allows using exact solvers without the need for approximations, which also leads to faster computation times compared to the one we originally proposed in [11].
- Finally, we reworked all experiments and introduce adaptive smoothing as an interesting novel image processing application for multidimensional label spaces.

2 Related work

2.1 Discrete approaches

It is well known that in the fully discrete setting, the minimization problem (1) is equivalent to maximizing a Bayesian posterior probability, where the prior probability gives rise to the regularizer [27]. The problem can be stated in the framework of Markov Random Fields [14] and discretized using a graph representation, where the nodes denote discrete pixel locations and the edges encode the energy functional [4].

Fast combinatorial minimization methods based on graph cuts can then be employed to search for a minimizer. In the case that the label space is binary and the regularizer submodular, a global solution of (1) can be found by computing a minimum cut [12,17]. For multi-label problems, one can approximate a solution for example by solving a sequence of binary problems (α-expansions) [5,24], linear programming relaxations [28] or quadratic pseudo-boolean optimization [16]. Exact solutions to multi-label problems can only be found in some special cases, notably [13], where a cut in a multi-layered graph is computed in polynomial time to find a global optimum. The construction is restricted to convex interaction terms with respect to a linearly ordered label set. In [25,26] the problem of image registration is formulated as an MRF labeling problem, which is minimized via LP relaxation. The authors present a decoupling strategy for the displacement components which is related to ours, albeit only applicable in the discrete case. It allows a simplification of the graph and consequently larger numbers of labels. The problem of large label spaces is also tackled in [10], where the authors compute optical flow from an MRF labeling problem using a lower dimensional parametric description for the displacements.

However, in many important scenarios the label space can not be ordered, or a non-convex regularizer is more desirable to better preserve discontinuities in the solution. Even for relatively simple non-convex regularizers like the Potts distance, the resulting combinatorial problem is NP-hard [5]. In this paper, we work in the fully continuous setting, avoiding typical problems of graph-based discretization like anisotropy and metrication errors [15].

2.2 Continuous approaches

Continuous approaches deal with the multi-label problem by *convex relaxation*. The original non-convex energy is replaced with a convex lower bound, which can be minimized globally. We automatically get a bound on the solution and know how far we are from the global optimum. How good the bound is depends on the *tightness* of the relaxation, i.e. how close the new energy is to the old one. While it sometimes can be possible to even achieve global optimality using this class of methods [19,22], there is no relaxation known which leads to globally optimal solutions of the general problem.

As in the discrete setting, it is possible to solve the two-label problem in a globally optimal way by minimizing a continuous convex energy and subsequent thresholding [19]. Our framework for regularization is based on the calibration or lifting idea for the Mumford-Shah functional, which was analyzed in depth in [2,1]. The idea is that the instead of optimizing for the original labeling function, one instead uses the characteristic function of its epigraph (called the subgraph in [1]). Thus, one ends up with a relaxation of the original problem in terms of these characteristic functions,

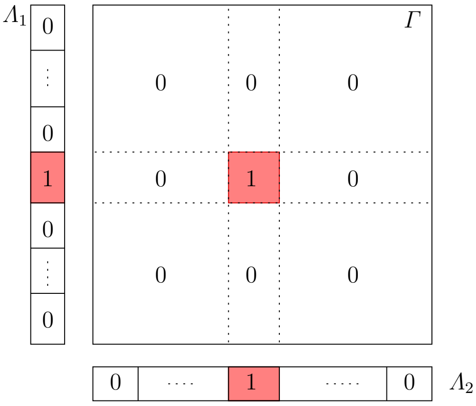


Fig. 2: *The central idea of the reduction technique is that if a single indicator function in the product space Γ takes the value* 1*, then this is equivalent to setting an indicator function in each of the factors Λ_j. The memory reduction stems from the fact that there are much more labels in Γ than in all the factors Λ_j combined.*

which is convex. The question is whether the solution of the relaxation corresponds to a solution of the original problem. In [23, 22], it was shown that one can achieve a globally optimal solution for the special case of convex interaction terms and a linearly ordered set of labels. Their construction can be viewed as a continuous version of [13].

For the general multi-label case, however, there is no relaxation known which leads to globally optimal solutions of the discrete problem. Relaxations of different tightness have been proposed in [18, 7, 29]. They all have in common that they are very memory intensive if the number of labels becomes larger, which makes it impossible to use them for scenarios with thousands of labels, like for example optic flow. Currently the most tight relaxation for the regularizer can be found in [20] based on the lifting framework. We use this form of relaxation in the present paper, while our previous conference publication [11] was based on the slightly more transparent, but less tight formulation introduced in [29] and further generalized in [18]. However, we use a different set of relaxation variables, which enables us to reveal the version of regularization in [18, 11] as another special case of the lifting framework.

3 Multi-dimensional Label Spaces

3.1 Discrete product label spaces

From now on we assume that the space of labels is a product of a finite number d of spaces, $\Gamma = \Lambda_1 \times \cdots \times \Lambda_d$. In order to give a more visual explanation of the main idea behind our work, we first discuss the *discrete* case, where $|\Lambda_k| = N_k \in \mathbb{N}$.

The convex relaxation introduced in [18, 29] works as follows. Instead of looking for a labeling $\boldsymbol{u} : \Omega \to \Gamma$ directly, we associate each label γ with a binary indicator function $u^\gamma \in \mathcal{L}^2(\Omega, \{0, 1\})$, where $u^\gamma(x) = 1$ if and only if $\boldsymbol{u}(x) = \gamma$. To make sure that a unique label is assigned to each point, only one of the indicator functions can have the value one. We can model this restriction by viewing $\boldsymbol{u}$ as a function mapping into the set of corners $\mathit{\Delta}$ of the N-simplex:

$$\boldsymbol{u} \in \mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta}) \text{ with } \mathit{\Delta} = \left\{ \boldsymbol{x} \in \{0,1\}^N \;:\; \sum_{j=1}^{N} x_j = 1 \right\}. \tag{2}$$

Obviously, we can identify $\boldsymbol{u}$ with the vector $(u^\gamma)_{\gamma \in \mathit{\Gamma}}$ of indicator functions. Let $\langle \cdot, \cdot \rangle$ denote the inner product on the Hilbert space $\mathcal{L}^2(\mathit{\Omega})$, then problem (1) can thus be written in the equivalent form

$$\underset{\boldsymbol{u} \in \mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta})}{\operatorname{argmin}} \left\{ J(\boldsymbol{u}) + \sum_{\gamma \in \mathit{\Gamma}} \left\langle u^\gamma, c^\gamma \right\rangle \right\}, \tag{3}$$

where we use bold face notation $\boldsymbol{u}$ for vectors $(u^\gamma)_{\gamma \in \mathit{\Gamma}}$ indexed by elements in $\mathit{\Gamma}$. We use the same symbol J to also denote the regularizer on the reduced space. Its definition requires careful consideration, and will be discussed in detail later.

The central idea of the paper is the following. The full discrete label space $\mathit{\Gamma}$ has $N = N_1 \cdot ... \cdot N_d$ elements, which means that it requires N indicator functions to represent a labeling, one for each label. We will show that it suffices to use $N_1 + ... + N_d$ indicator functions, which is a considerable reduction in problem dimensionality, thus computation time and memory requirements. We achieve this by replacing the indicator functions on the product $\mathit{\Gamma}$ by indicator functions on the components $\mathit{\Lambda}_k$.

To this end, we associate to each label $\lambda \in \mathit{\Lambda}_k, 1 \leq k \leq d$ an indicator function v_k^λ. In each component k, only one of the indicator functions can be set. Thus, the vector $\boldsymbol{v}_k = (v_k^\lambda)_{\lambda \in \mathit{\Lambda}_k}$ which consists of N_k binary functions can be viewed as a mapping into the corners of the simplex $\mathit{\Delta}_k$,

$$\mathit{\Delta}_k = \left\{ \boldsymbol{x} \in \{0,1\}^{N_k} \;:\; \sum_{j=1}^{N_k} x_j = 1 \right\}. \tag{4}$$

In particular, the reduced set of indicator functions $\boldsymbol{v} = (v_k^\lambda)_{1 \leq k \leq d, \gamma \in \mathit{\Lambda}_k}$ can be seen as a map $\mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta}_\times)$ with

$$\mathit{\Delta}_\times = \mathit{\Delta}_1 \times ... \times \mathit{\Delta}_d \subset \mathbb{R}^{N_1 + ... + N_d}. \tag{5}$$

Note that an element $\boldsymbol{v} \in \mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta}_\times)$ consists indeed of exactly $N_1 + ... + N_d$ binary functions.

The following proposition illuminates the relationship between the original space of indicator functions $\mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta})$ and the reduced space of indicator functions $\mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta}_\times)$, which is easy to understand visually, see Fig. 2.

Proposition 1 *A bijection $\boldsymbol{v} \mapsto \boldsymbol{u}$ from $\mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta}_\times)$ onto $\mathcal{L}^2(\mathit{\Omega}, \mathit{\Delta})$ is defined by setting*

$$u^\gamma := v_1^{\gamma_1} \cdot ... \cdot v_d^{\gamma_d}, \tag{6}$$

for all $\gamma = (\gamma_1, ..., \gamma_d) \in \mathit{\Gamma}$.

Proof In order to proof the proposition, we show that the mapping induces a point-wise bijection from $\mathit{\Delta}_\times$ onto $\mathit{\Delta}$. We first show it is onto: for $\boldsymbol{u}(x)$ in $\mathit{\Delta}$, there exists exactly one $\gamma \in \mathit{\Gamma}$ with $u^\gamma(x) = 1$. Set $v_k^\lambda(x) = 1$ if $\lambda = \gamma_k$, and $v_k^\lambda(x) = 0$ otherwise. Then equation (6) is satisfied as desired, see Fig. 2. To see that the map is one-to-one, we just count the elements in $\mathit{\Delta}_\times$. Since $\mathit{\Delta}_k$ contains N_k elements, the number of elements in $\mathit{\Delta}_\times$ is $N_1 \cdot ... \cdot N_d = N$, the same as in $\mathit{\Delta}$. □

With this reduced function space, another equivalent formulation to (1) and (3) is

$$\underset{\boldsymbol{v}\in\mathcal{L}^2(\Omega,\Delta_\times)}{\operatorname{argmin}} \left\{ J(\boldsymbol{v}) + \sum_{\gamma\in\Gamma} \left\langle v_1^{\gamma_1} \cdot \ldots \cdot v_d^{\gamma_d}, c^\gamma \right\rangle \right\}. \tag{7}$$

Note that while we have reduced the dimensionality of the problem considerably, we have introduced another difficulty: the data term is not convex anymore, since it contains a product of the components. Thus, in the relaxation, we need to take additional care to make the final problem again convex.

3.2 Continuous label spaces and relaxation framework

We now turn to the more general case that each factor Λ_k is an interval in $\mathbb{R}$, which means that we deal with a continuous label space with an infinite number of labels. In this situation, one is also interested in a number of continuous regularizers, which can not be modeled satisfyingly on a discrete label space. As in the discrete case, the regularizers are usually not convex and require a relaxation.

In the context of continuous labeling problems where the label range is an interval, a central idea is *functional lifting*, which is a variant of the calibration method [1]. Here, one works with characteristic functions describing the hypograph instead of the labeling function itself, an idea that was further refined and applied to a variety of image processing problems in a number of subsequent works [7,20,21,22]. We are going to translate this framework to the case of a product label space. With the regularizer, we can restrict ourselves to the case that it can be decomposed into the sum of regularizers on each component. However, for the data term this is not possible since the cost function usually cannot be decomposed in a similar way. Therefore, we need to define a relaxation framework in which we still can express arbitrary cost functions.

Let us first consider a single component $u_k : \Omega \to \Lambda_k$ of the full labeling function $\boldsymbol{u}$. The characteristic function of its hypograph is defined on $\Omega \times \Lambda_k$ as

$$1_{\operatorname{hyp}(u_k)}(x,\lambda) = \begin{cases} 1, & \text{if } \lambda \le u_k(x) \\ 0, & \text{else.} \end{cases} \tag{8}$$

In [1,21,22], the labeling problem is reformulated in terms of new unknowns which correspond to these characteristic functions. The reason is equation (14), which we discuss later and which allows to give a convex reformulation of the regularizer in terms of the new unknowns. This allows to obtain a globally optimal solution in the new variables, which often is at least close to and sometimes equal to the solution of the original non-convex problem.

In our case, however, we need different variables in order to be able to simultaneously formulate a convex relaxation of the data term. We work with the *indicator functions* denoting if a specific label λ is set at a point $x \in \Omega$, related to a labeling $\boldsymbol{u}$ by

$$v_k(x,\lambda) = \delta(u_k(x) - \lambda). \tag{9}$$

Note that the new unknowns are actually distributions on the higher dimensional space $\Omega \times \Lambda_k$, which however will reduce to regular functions after discretization. They

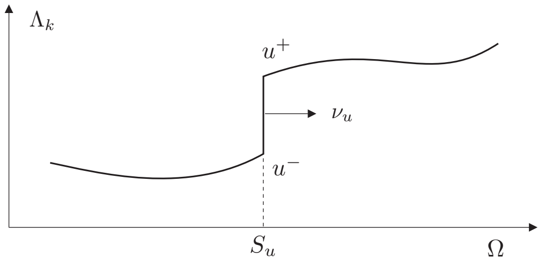


Fig. 3: *A special function of bounded variation u has an approximate gradient everywhere except on a nullset S_u, where the values jump from u^- to u^+. The normal ν_u denotes the direction of the jump from small to large values.*

serve as a generalization of the discrete label indicator functions $v_k^\lambda \in \mathcal{L}^2(\Omega, \Delta_k)$ to the continuous case, in particular they satisfy the relations

$$\int_{\Lambda_k} v_k(x,\lambda)\,\mathrm{d}\lambda = 1, \int_{\Lambda_k} \lambda\, v_k(x,\lambda)\,\mathrm{d}\lambda = u_k(x), \tag{10}$$

which mimic the discrete case with sums replaced by integrals. Intuitively, this means that for each fixed $x \in \Omega$, $v_k(x,\cdot)$ has a total mass of 1 and is concentrated on the label $u_k(x) \in \Lambda_k$.

We will reformulate the labeling problem in terms of the new variables $\boldsymbol{v}$ in section 4. Some things have to be kept in mind, however. Since the new variables are distributions in the continuous case, we cannot formulate a well-defined minimization problem without first reducing them to $\mathcal{L}^2$-functions . This means that before writing down the actual minimization problem we want to solve in the new variables, we have to introduce a discretization of the label space. This is not a major drawback. First, note that the definition of the continuous regularizers in section 5 does not require the discretization, which means that we are still dealing correctly with the continuous case. Furthermore, a discretization of the continuous label space into a finite number of labels is also necessary in other previous work which employs the lifting idea [22,21] when it comes to the actual implementation.

3.3 Regularization

We consider a general *separable* regularizer of the form

$$J(\boldsymbol{u}) = \sum_{k=1}^{d} J_k(u_k), \tag{11}$$

which means that J acts on the components of $\boldsymbol{u}$ independently. In order to define the regularizer, we require some technical preliminaries. Recall [3] that for functions u_k in the space $\mathcal{SBV}(\Omega)$ of special functions of bounded variation, the distributional derivative Du_k can be decomposed as

$$Du_k = \nabla u_k\ \mathrm{d}x + (u_k^+ - u_k^-)\nu_{u_k}\ \mathrm{d}\mathcal{H}^{n-1} \llcorner S_{u_k} \tag{12}$$

into a differentiable part and a jump part, see Fig. 3. Here, S_{u_k} is the $(n-1)$-dimensional jump set of u_k, where the values jump from u_k^- to u_k^+, ν_{u_k} is the normal to S_{u_k} oriented towards the the u_k^+ side, and ∇u_k is the approximate gradient of u_k. The measure $\mathcal{H}^{n-1} \llcorner S_{u_k}$ is the $(n-1)$-dimensional Hausdorff measure restricted to the set S_{u_k}. We refer to [3] for a comprehensive introduction to functions of bounded variation.

Making use of this decomposition, we can introduce the framework for regularization. We consider regularizers for problem (1) of the form (11), with

$$J_k(u_k) = \int_{\Omega \setminus S_{u_k}} h_k(x, u_k(x), \nabla u_k(x))\, \mathrm{d}x + \int_{S_{u_k}} d_k\big(s, u_k^-(s), u_k^+(s)\big)\, \mathrm{d}\mathcal{H}^{n-1}(s), \quad (13)$$

with functions $h_k : \Omega \times \Lambda_k \times \mathbb{R}^n \to \mathbb{R}$ and $d_k : \Omega \times \Lambda_k \times \Lambda_k \to \mathbb{R}$. The functions h_k and d_k have to satisfy the following conditions:

1. $h_k(x, \lambda, p)$ is convex in p for fixed x, λ.
2. $d_k(x, \cdot, \cdot)$ is a metric on Λ_k for fixed x.

The interesting task, of course, is to identify suitable choices of h_k and d_k, and to interpret what the choice means in practice. We will turn to this in section 5. Before we can explore the possible regularizers, however, we need to introduce a convex relaxation of the general regularizer (13) in section 4.

3.4 Notation conventions

Because the label space is multi-dimensional, the notation requires multiple indices and is slightly more complex. Throughout this work, we keep the following conventions to keep it as clear as possible. The index $k = 1, \ldots, d$ enumerating the factors of the product space is always written as a subscript. Indices which are Greek letters always enumerate labels, where γ, χ are labels in the full product space Γ with components $\gamma_k, \chi_k \in \Lambda_k$. Greek letters λ, μ denote labels in one of the factors Λ_k. If the label space is discrete or has been discretized, the label is written as a superscript to the indicator functions v_k^λ. In the case of a continuous label space, the indicator functions v_k live on $\Omega \times \Lambda_k$, thus the label appears as an argument of the function $v_k(x, \lambda)$.

4 Convex relaxation

The minimization problem (3) which we want to solve is not convex: neither is the energy a convex function nor is the domain of minimization a convex set. Thus, the task of finding a global minimizer is in general computationally infeasible. We therefore propose a *convex relaxation*. This means that instead of minimizing the original functional, we minimize a convex one (ideally, the exact convex envelope) over the convex hull of the original domain.

The relaxation is defined in terms of the new variables v_k defined in (9). After obtaining a solution $\hat{\boldsymbol{v}}$, the question remains of whether the solution corresponds to a function $\hat{\boldsymbol{u}}$ which solves the original problem. In general, this is not correct, but we can compute a projection $\Pi(\hat{\boldsymbol{v}})$ onto the original problem domain and obtain an optimality bound. Indeed, the energy of the optimal solution $\hat{\boldsymbol{u}}$ must lie somewhere between the energies of $\hat{\boldsymbol{v}}$ and $\Pi(\hat{\boldsymbol{v}})$, as $\hat{\boldsymbol{v}}$ minimizes the relaxation and $\Pi(\hat{\boldsymbol{v}})$ lies in the original problem domain in which $\hat{\boldsymbol{u}}$ is a minimizer.

In the following subsection we will introduce first a convex relaxation of the regularizer, which is based on the calibration method - however, our variables are different from the ones used in previous work, which requires a slight reformulation. Thereafter, we present the new convex relaxation of the data term and show how it is an improvement over the one presented in the original conference paper [11].

4.1 Convex relaxation of the regularizer

Our first goal is to give a new representation of the regularizer defined in (13). While it is not convex in the labeling $\boldsymbol{u}$, we will obtain a representation which is convex in the new variables v_k defined in (9). We do this by making use of the calibration or lifting technique described in detail in [1]. Lemma 3.9 in [1] states that under the previous assumptions on h_k and d_k, the regularizer J_k for each component can be represented as

$$J_k(u_k) = \sup_{\boldsymbol{\phi} \in K} \left\{ \int_{\Omega \times \Lambda_k} \boldsymbol{\phi}^1 \cdot \nabla_x 1_{\mathrm{hyp}(u_k)} + \phi^2 \, \partial_\lambda 1_{\mathrm{hyp}(u_k)} \, \mathrm{d}(x, \lambda) \right\} \tag{14}$$

with the convex set

$$\begin{aligned} K = \Big\{ \boldsymbol{\phi} = & (\boldsymbol{\phi}^1, \phi^2) : \Omega \times \Lambda_k \to \mathbb{R}^n \times \mathbb{R} \text{ such that for all } x \in \Omega \text{ and } \lambda, \mu \in \Lambda_k, \\ & \phi^2(x,\lambda) \geq h_k^*(x, \lambda, \boldsymbol{\phi}^1(x,\lambda)) \text{ and } \left| \int_\lambda^\mu \boldsymbol{\phi}^1(x,s) \, \mathrm{d}s \right| \leq d_k(x,\lambda,\mu) \Big\}. \end{aligned} \tag{15}$$

Note that (14) is a *convex* representation of the regularizer in terms of the characteristic functions $1_{\mathrm{hyp}(u_k)}$ of the hypograph of u_k, see equation (8). However, what we want is a convex representation in terms of our new unknowns v_k. We give this reformulation in the following theorem.

Theorem 1 *Let J_k be of the form (13), and the indicator functions v_k defined as in (9). Then*

$$J_k(u_k) = \sup_{(\boldsymbol{p}, b) \in C_k} \left\{ \int_{\Omega \times \Lambda_k} (-\mathrm{div}(\boldsymbol{p}) - b) \, v_k \, \mathrm{d}(x, \lambda) \right\}, \tag{16}$$

with the convex set

$$\begin{aligned} C_k = \Big\{ & (\boldsymbol{p}, b) : \Omega \times \Lambda_k \to \mathbb{R}^n \times \mathbb{R} \textit{ such that for all } x \in \Omega \textit{ and } \lambda, \mu \in \Lambda_k, \\ & b(x,\lambda) \geq h_k^*\big(x, \lambda, \partial_\lambda \boldsymbol{p}(x,\lambda)\big), \\ & |\boldsymbol{p}(x,\lambda) - \boldsymbol{p}(x,\mu)|_2 \leq d_k(x,\lambda,\mu) \Big\}. \end{aligned} \tag{17}$$

Above, $h_k^*(x, \lambda, q)$ denotes the convex conjugate of $h_k(x, \lambda, p)$ with respect to p.

Proof Since derivatives of indicator functions do not exist in ordinary sense, the integral in (14) is meant to be a convenient notation for

$$\int_{\Omega \times \Lambda_k} (\boldsymbol{\phi}^1, \phi^2) \cdot \nu_{\Gamma_{u_k}} \, \mathrm{d}\mathcal{H}^n(x, \lambda) \tag{18}$$

where

$$\mathit{\Gamma}_{u_k} := \Big\{(x, u(x)) \,\big|\, x \in \mathit{\Omega} \setminus S_{u_k}\Big\} \cup \Big\{(x, s) \,\big|\, x \in S_{u_k},\ s \in [u_k^-, u_k^+]\Big\} \tag{19}$$

is the extended graph of u, and $\nu_{\mathit{\Gamma}_{u_k}}$ is the normal on $\mathit{\Gamma}_{u_k}$ pointing "downwards". Intuitively, $\nabla 1_{\mathrm{hyp}(u_k)}$ in (14) is nonzero only on $\mathit{\Gamma}_{u_k}$, and equals $\nu_{\mathit{\Gamma}_{u_k}}$ up to a delta function factor. For a fixed $\boldsymbol{\phi}$ denote the integral (18) by $J_{\boldsymbol{\phi}}$. It is equal to [1, lemma 2.8]

$$\begin{aligned} J_{\boldsymbol{\phi}} = &\int_{\mathit{\Omega}\setminus S_{u_k}} \Big(\boldsymbol{\phi}^1(x, u_k) \cdot \nabla u_k - \phi^2(x, u_k)\Big)\, \mathrm{d}x \\ &+ \int_{S_{u_k}} \left(\int_{u_k^-}^{u_k^+} \boldsymbol{\phi}^1(x, s)\, \mathrm{d}s\right) \cdot \nu_{u_k}\ \mathrm{d}\mathcal{H}^{n-1}(x). \end{aligned} \tag{20}$$

Define $\boldsymbol{p} : \mathit{\Omega} \times \mathit{\Lambda}_k \to \mathbb{R}^n$ and $b : \mathit{\Omega} \times \mathit{\Lambda}_k \to \mathbb{R}$ by

$$\boldsymbol{p}(x, \lambda) := \int_{\lambda_0}^{\lambda} \boldsymbol{\phi}^1(x, s)\, \mathrm{d}s, \qquad b(x, \lambda) := \phi^2(x, \lambda) \tag{21}$$

for some $\lambda_0 \in \mathit{\Lambda}_k$. With these new variables we have

$$\begin{aligned} J_{\boldsymbol{\phi}} = &\int_{\mathit{\Omega}\setminus S_{u_k}} \Big(\partial_\lambda \boldsymbol{p}(x, u_k) \cdot \nabla u_k - b(x, u_k)\Big)\, \mathrm{d}x \\ &+ \int_{S_{u_k}} \Big(\boldsymbol{p}(x, u_k^+) - \boldsymbol{p}(x, u_k^-)\Big) \cdot \nu_{u_k}\ \mathrm{d}\mathcal{H}^{n-1}(x). \end{aligned} \tag{22}$$

By the divergence theorem,

$$\begin{aligned} \int_{\mathit{\Omega}\setminus S_{u_k}} \operatorname{div}\big(\boldsymbol{p}(x, u_k)\big)\, \mathrm{d}x = &\int_{S_{u_k}} \Big(\boldsymbol{p}(x, u_k^+) \cdot (-\nu_{u_k}) + \boldsymbol{p}(x, u_k^-) \cdot \nu_{u_k}\Big)\, \mathrm{d}\mathcal{H}^{n-1}(x) \\ &+ \int_{\partial\mathit{\Omega}} \boldsymbol{p}(x, u_k) \cdot \nu_{\partial\mathit{\Omega}}\ \mathrm{d}\mathcal{H}^{n-1}(x). \end{aligned} \tag{23}$$

In the integrand of the first integral on the right hand side there are two addends for each point of S_{u_k}, because the integration on the left hand side is performed on *both* sides of S_{u_k}. The outer normal for the u_k^- side is ν_{u_k} by definition, and for the u_k^+ side it is just the opposite. The last integral on the right hand side is zero because $\boldsymbol{\phi}$ and therefore also $\boldsymbol{p}$ has compact support in $\mathit{\Omega}$. Using (23) in (22) we obtain

$$J_{\boldsymbol{\phi}} = \int_{\mathit{\Omega}\setminus S_{u_k}} \Big(\partial_\lambda \boldsymbol{p}(x, u_k) \cdot \nabla u_k - b(x, u_k)\Big)\, \mathrm{d}x - \int_{\mathit{\Omega}\setminus S_{u_k}} \operatorname{div}\big(\boldsymbol{p}(x, u_k)\big)\, \mathrm{d}x \tag{24}$$

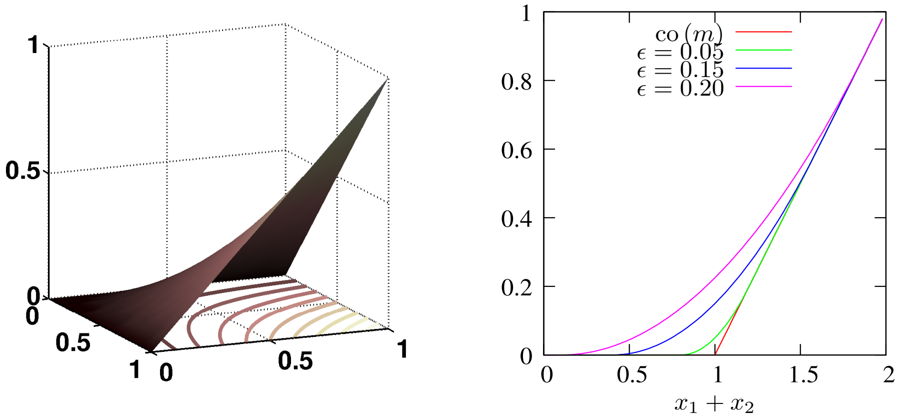
By the chain rule,

$$\operatorname{div}\big(\boldsymbol{p}(x, u_k)\big) = (\operatorname{div}\boldsymbol{p})(x, u_k) + \partial_\lambda \boldsymbol{p}(x, u_k) \cdot \nabla u_k. \tag{25}$$

Thus, the expression (24) simplifies to

$$J_{\boldsymbol{\phi}} = \int_{\mathit{\Omega}\setminus S_{u_k}} \big(-(\operatorname{div}\boldsymbol{p})(x, u_k) - b(x, u_k)\big)\, \mathrm{d}x = \int_{\mathit{\Omega}\times\mathit{\Lambda}_k} (-\operatorname{div}\boldsymbol{p} - b)\, \boldsymbol{v}_k\ \mathrm{d}(x, \lambda). \tag{26}$$

The last equality is simply the definition of how the distribution $\boldsymbol{v}_k(x, \lambda) = \delta(u_k - \lambda)$, defined for $u_k \in \mathcal{SBV}(\mathit{\Omega})$, acts on functions. Now, the claim of the proposition follows directly from (14) and (26). □



(a) Product function $m(x_1, x_2) = x_1 x_2$

(b) Convex envelope $\operatorname{co}(m)$ and mollified versions for different ϵ

Fig. 4: *Product function and its mollified convex envelope for the case* $d = 2$.

Note that similarly to the discrete version of the indicator functions, the discrete version of the set C_k in (17) consists of tuples $(\boldsymbol{p}^\lambda, b^\lambda)_{\lambda \in \Lambda_k}$ of functions. Taking a closer look at equation (16), we can see that the right hand side is a convex functional in the new variables v_k. Thus, we have achieved our goal and can turn towards finding a similar relaxation of the data term.

4.2 Convex relaxation of the data term

In this subsection, we deal with the non-convexity of the data term in (7),

$$E_{\text{data}}(\boldsymbol{v}) = \sum_{\gamma \in \Gamma} \left\langle v_1^{\gamma_1} \cdot \ldots \cdot v_d^{\gamma_d}, c^\gamma \right\rangle . \tag{27}$$

Specifically, we show two different ways how it can be replaced with a convex function which has the same binary minimizers with equal energy. We first describe the convexification idea from the original conference paper [11] in the discrete case with a label space of dimension $d = 2$. While it leads to a working relaxation, it has certain shortcomings, the main problem being that an unwanted constant solution has to be avoided by additional smoothing when moving on from binary to continuous functions. These shortcomings will be remedied by a new relaxation technique which we explain thereafter. Note that for the data term, we already work in the setting of a discretized label space. While it is, up to a point, possible to give a well-defined theoretical justification of the relaxation for the continuous case, the associated trouble and loss of clarity is not worth the small theoretical gain.

Discrete two-dimensional case. In [11], we suggested to replace the multiplication function $m(v_1^{\gamma_1}, \ldots, v_d^{\gamma_d}) := v_1^{\gamma_1} \cdot \ldots \cdot v_d^{\gamma_d}$ with its convex envelope $\operatorname{co}(m)$. Analyzing the epigraph of m, see Fig. 4(a), shows that

$$\mathrm{co}\,(m)\,(x_1, ..., x_d) = \begin{cases} 1 & \text{if } x_1 = ... = x_d = 1, \\ 0 & \text{if any } x_k = 0. \end{cases} \tag{28}$$

This means that if in the functional, m is replaced by the convex function $\mathrm{co}\,(m)$, we retain the same binary solutions, as the function values on binary input are the same.

We lose nothing on first glance, but on second glance, we forfeited differentiability of the data term, since $\mathrm{co}\,(m)$ is not a smooth function anymore. Furthermore, the new function we obtain is not the correct convex envelope of the full data term, only for the constituting addends. The particular problem this leads to is that for the constant function $\hat{\boldsymbol{v}}$ defined by

$$\hat{v}_k^\lambda(x) := 1/N_k \tag{29}$$

the energy of the data term and hence the total energy is zero.

In [11], this problem was circumvented by an additional mollification of the convex envelope. We replaced $\mathrm{co}\,(m)$ again by a mollified function $\mathrm{co}\,(m)_\epsilon$, where $\epsilon > 0$ is a small constant. We illustrate this for the case $d = 2$, where one can easily write down the functions explicitly. In this case, the convex envelope of multiplication is

$$\mathrm{co}\,(m)\,(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 + x_2 \leq 1 \\ x_1 + x_2 - 1 & \text{otherwise.} \end{cases} \tag{30}$$

This is a piecewise linear function of the sum of the arguments, i.e symmetric in x_1 and x_2, see Fig. 4(b). We smoothen the kink by replacing $\mathrm{co}\,(m)$ with

$$\mathrm{co}\,(m)_\epsilon\,(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 + x_2 \leq 1 - 4\epsilon \\ \frac{1}{16\epsilon}(x_1 + x_2 - (1 - 4\epsilon))^2 & \text{if } 1 - 4\epsilon < x_1 + x_2 < 1 + 4\epsilon \\ 1 & \text{if } x_1 + x_2 \geq 1 + 4\epsilon \end{cases} \tag{31}$$

This function does not satisfy the envelope condition (28) exactly, but only fulfills the less tight

$$\mathrm{co}\,(m)_\epsilon\,(x_1, \ldots, x_d) \begin{cases} = 1 & \text{if } x_1 = \cdots = x_d = 1, \\ \leq \epsilon & \text{if any } x_j = 0. \end{cases} \tag{32}$$

Notably, the data term energy of the constant trivial minimizer (29) is now $\epsilon \sum_\gamma c^\gamma$, which means that the relaxation of the data term leads to the correct pointwise solution with energy $\min_\gamma(c^\gamma)$ if $\epsilon > \min_\gamma(c^\gamma)/\sum_\gamma c^\gamma$. Since the condition must be satisfied for each point $x \in \Omega$, it is best to set ϵ point-wise to the minimal possible value. However, the choice of mollified envelope is suboptimal since it is just an approximation to the correct envelope and distorts the original problem. Thus, we are now going to propose a novel relaxation of the data term which avoids this problem altogether and is easier to deal with in higher dimensional label spaces.

New tighter relaxation for general d-dimensional case. In this paragraph, we describe our new relaxation of the data term. It it much tighter and does not suffer from the described drawbacks of the relaxation in [11]. The new relaxation of $E_{\text{data}}(\boldsymbol{v})$ is one of the main additional contributions of this paper. It is defined as

$$R_{\text{data}}(\boldsymbol{v}) := \sup_{\boldsymbol{q}\in\mathcal{Q}} \left\{ \int_\Omega \sum_{\gamma_1\in\Lambda_1} q_1^{\gamma_1} v_1^{\gamma_1} + \ldots + \sum_{\gamma_d\in\Lambda_d} q_d^{\gamma_d} v_d^{\gamma_d} \,\mathrm{d}x \right\}. \tag{33}$$

The additional dual variables $\boldsymbol{q} = (\boldsymbol{q}_k)_{k=1..d}$ range over the convex set

$$\begin{aligned} \mathcal{Q} := \big\{ (\boldsymbol{q}_k : \Lambda_k \to \mathbb{R})_{k=1..d} \text{ such that for all } \gamma \in \Gamma, \\ q_1^{\gamma_1} + \ldots + q_d^{\gamma_d} \le c^\gamma \big\}. \end{aligned} \tag{34}$$

We first establish that the relaxation coincides with the original energy for binary functions.

Proposition 2 *Let $\boldsymbol{v} \in \mathcal{L}^2(\Omega, \Delta_\times)$ be a binary function representing the label $\gamma(x) \in \Gamma$ in each point $x \in \Omega$. Then*

$$R_{data}(\boldsymbol{v}) = \int_\Omega c(x, \gamma(x))\,\mathrm{d}x = E_{data}(\boldsymbol{v}). \tag{35}$$

Proof Since in each point, $\gamma(x)$ is the label indicated by $\boldsymbol{v}(x)$, we have $v_k^{\gamma_k} = 1$ pointwise. Thus for all $\boldsymbol{q} \in \mathcal{Q}$,

$$\sum_{k=1}^d \sum_{\lambda\in\Lambda_k} q_k^\lambda v_k^\lambda = \sum_{k=1}^d q_k^{\gamma_k} v_k^{\gamma_k} = \sum_{k=1}^d q_k^{\gamma_k} \le c^\gamma. \tag{36}$$

This shows that at least $R(\boldsymbol{v}) \le \int_\Omega c(x, \gamma(x))\,\mathrm{d}x$. To prove equality, we use Lagrange multipliers to write the constraints in (34) as additional energy terms:

$$\begin{aligned} R_{\text{data}}(\boldsymbol{v}) &= \sup_{\boldsymbol{q}\in\mathcal{Q}} \sum_{k=1}^d \sum_{\lambda\in\Lambda_k} q_k^\lambda v_k^\lambda \\ &= \sup_{\boldsymbol{q}} \inf_{\boldsymbol{\mu}^{\widehat{\gamma}}\ge 0} \sum_{k=1}^d \sum_{\lambda\in\Lambda_k} q_k^\lambda v_k^\lambda - \sum_{\widehat{\gamma}\in\Gamma} \boldsymbol{\mu}^{\widehat{\gamma}} \big(q_1^{\widehat{\gamma}_1} + \ldots + q_d^{\widehat{\gamma}_d} - c^{\widehat{\gamma}} \big) \\ &= \inf_{\boldsymbol{\mu}^{\widehat{\gamma}}\ge 0} \sum_{\widehat{\gamma}\in\Gamma} \boldsymbol{\mu}^{\widehat{\gamma}} c^{\widehat{\gamma}} + \sup_{\boldsymbol{q}} \sum_{k=1}^d \sum_{\lambda\in\Lambda_k} q_k^\lambda \Bigg(v_k^\lambda - \sum_{\widehat{\gamma}\in\Gamma:\, \widehat{\gamma}_k=\lambda} \boldsymbol{\mu}^{\widehat{\gamma}} \Bigg), \end{aligned} \tag{37}$$

interchanging the ordering of $\sup_{\boldsymbol{q}}$ and $\inf_\mu$. Evaluating the supremum over $\boldsymbol{q}$ leads to constraints on the variables $\boldsymbol{\mu}^{\widehat{\gamma}}$ and we obtain

$$R_{\text{data}}(\boldsymbol{v}) = \inf_{\boldsymbol{\mu}^{\widehat{\gamma}}\ge 0} \sum_{\widehat{\gamma}\in\Gamma} \boldsymbol{\mu}^{\widehat{\gamma}} c^{\widehat{\gamma}} \tag{38}$$

with $\boldsymbol{\mu}^{\widehat{\gamma}}$ such that additionally

$$\sum_{\widehat{\gamma}\in\Gamma:\, \widehat{\gamma}_k=\lambda} \boldsymbol{\mu}^{\widehat{\gamma}} = v_k^\lambda \tag{39}$$

for all $1 \leq k \leq d$ and $\lambda \in \Lambda_k$. First, for any fixed k and $\lambda \neq \gamma_k$, we have $v_k^\lambda = 0$. Since $\boldsymbol{\mu}^{\widehat{\gamma}} \geq 0$, (39) then gives $\boldsymbol{\mu}^{\widehat{\gamma}} = 0$ for all $\widehat{\gamma}$ with $\widehat{\gamma}_k \neq \gamma_k$. Thus, $\boldsymbol{\mu}^{\widehat{\gamma}} = 0$ for all $\widehat{\gamma} \neq \gamma$. Next, plug $\lambda = \gamma_k$ for some k into (39). Since any other addend $\boldsymbol{\mu}^{\widehat{\gamma}}$ is zero, the sum is just μ^γ, while the right hand side is $v_k^{\gamma_k} = 1$.

Therefore, the constraints (39) ensure that $\boldsymbol{\mu}^{\widehat{\gamma}} = 0$ for all $\widehat{\gamma} \neq \gamma$ and $\mu^\gamma = 1$, so (38) gives $R_{\mathrm{data}}(\boldsymbol{v}) = c^\gamma$. □

In addition, one can prove the following theorem, which shows that the relaxation of the data term has the correct pointwise minimizers, in contrast to the one proposed in [11]. This means that no smoothing is necessary and an exact minimization algorithm can be employed to obtain solutions.

Theorem 2 *Let $\hat{\boldsymbol{v}} \in \mathcal{L}^2(\Omega, \Delta_\times)$ be a binary minimizer of E_{data}. Then $\hat{\boldsymbol{v}}$ is also a minimizer of the relaxation,*

$$\hat{\boldsymbol{v}} \in \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{L}^2(\Omega, \mathrm{co}(\Delta_\times))} \{R_{data}(\boldsymbol{v})\}\,. \tag{40}$$

In particular, $E_{data}(\hat{\boldsymbol{v}}) = R_{data}(\hat{\boldsymbol{v}}) = \int_\Omega \hat{c}\,\mathrm{d}x$ with $\hat{c} := \inf_{\gamma \in \Gamma}(c^\gamma)$ pointwise.

Proof Let $\boldsymbol{v} \in \mathcal{L}^2(\Omega, \mathrm{co}\,(\Delta_\times))$ be arbitrary, and set $q_k^\lambda := \hat{c}/d$. Then

$$\sum_{k=1}^{d} \sum_{\lambda \in \Lambda_k} q_k^\lambda v_k^\lambda = \sum_{k=1}^{d} \frac{\hat{c}}{d} \sum_{\lambda \in \Lambda_k} v_k^\lambda = d\frac{\hat{c}}{d} = \hat{c}, \tag{41}$$

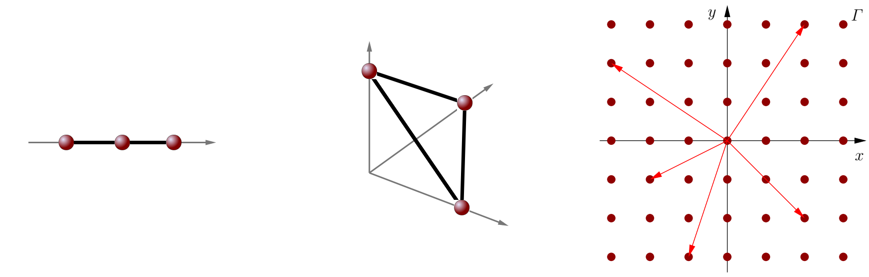
and $\sum_k q_k^{\gamma_k} = \hat{c} \leq c^\gamma$ for all γ, so $\boldsymbol{q} \in \mathcal{Q}$. This shows that $R(\boldsymbol{v}) \geq \hat{c}$, which is the minimum of E_{data} for binary functions. □

5 Multilabel Regularizers

In this section, we will explore suitable choices of the regularizer, and how they fit within the proposed framework. In particular, we will see how our model can be specialized to the case of discrete label spaces where the label distance has a Euclidean representation. This special case was discussed in [18,11], and we will see that our framework leads to a tighter relaxation for this case. We will also discuss additional continuous regularizers which become possible based on the lifting framework discussed in the last section. These were introduced in the previous works [7,21,22] when the unknowns were the characteristic functions of the hypographs of u_k. We show how we can accommodate them to depend on the new unknowns. Notably, in each dimension of the label space its own type of regularization can be chosen, in particular discrete and continuous regularizers can be mixed freely.

5.1 Discrete label space and its Euclidean representation

We first consider the special case of a discrete label space Λ_k. Thus, we need to define a regularizer $J_k : \mathcal{L}^2(\Omega, \mathrm{co}\,(\Delta_k)) \to \mathbb{R}$ for functions with values in the convex hull of the simplex Δ_k. We first present the construction used in [18,11], and then show how we can embed it into our more general framework.



(a) Ordered embedding (b) Potts embedding (c) Optic flow embedding

Fig. 5: *Different embeddings for a label space. In an ordered embedding, all labels are mapped onto a line, while for the Potts model, every label is mapped onto a different unit vector. For optical flow, each label is already a vector in* $\mathbb{R}^2$*, so a sensible embedding is given by the identity.*

We assume that the metric d_k has a *Euclidean* representation. This means that each label $\lambda \in \Delta_k$ shall be *represented* by an M_k-dimensional vector $\boldsymbol{a}_k^\lambda \in \mathbb{R}^{M_k}$, and the distance d_k is defined as the Euclidean distance between the representations,

$$d_k(\lambda,\mu) = \left|\boldsymbol{a}_k^\lambda - \boldsymbol{a}_k^\mu\right|_2 \text{ for all } \lambda,\mu \in \Delta_k \,. \tag{42}$$

The goal in the construction of J_k is that the higher the distance between labels and the longer the jump set, the higher shall be the penalty imposed by J_k. To make this idea precise, we introduce the linear mappings $A_k : \mathrm{co}\,(\Delta_k) \to \mathbb{R}^{M_k}$ which map labels onto their representations,

$$A_k(\lambda) = \boldsymbol{a}_k^\lambda \text{ for all } \lambda \in \Delta_k \,. \tag{43}$$

When the labels are enumerated, then in matrix notation, the vectors $\boldsymbol{a}_k^\lambda$ become exactly the columns of A_k, which shows the existence of this map. It turns out that a regularizer with desirable properties can be defined by

$$J_k^A(\boldsymbol{v}_k) := \mathrm{TV_v}(A_k \boldsymbol{v}_k)\,, \tag{44}$$

where

$$\mathrm{TV_v}(\boldsymbol{f}) := \int_\Omega \sqrt{\sum_{i=1}^m |\nabla f_i|_2^2}\,\mathrm{d}x \tag{45}$$

denotes the vectorial total variation for functions $\boldsymbol{f} : \Omega \to \mathbb{R}^m$ taking values in a real vector space of dimension m. The following theorem was proved in [18] and shows why the above definition makes sense.

Theorem 3 *The regularizer* J_k^A *defined in* (44) *has the following properties:*

1. J_k^A *is convex and positively homogeneous on* $\mathcal{L}^2(\Omega, \mathrm{co}\,(\Delta_k))$.
2. $J_k^A(\boldsymbol{v}_k) = 0$ *for any constant labeling* $\boldsymbol{v}_k$.

3. *If $S \subset \Omega$ has finite perimeter* $\mathrm{Per}(S)$*, then for all labels* $\lambda, \mu \in \Lambda_k$*,*

$$J_k^A(\lambda 1_S + \mu 1_{S^c}) = d_k(\lambda, \mu)\,\mathrm{Per}(S)\,, \tag{46}$$

i.e. a change in labels is penalized proportional to the distance between the labels and the perimeter of the interface.

For the sake of simplicity, we only give the main examples for distances with Euclidean representations. More general classes of distances on the labels can also be used, see [18].

- The case of ordered labels, where the embedding follows the natural ordering of $\lambda, \mu \in \mathbb{R}$, Fig. 5(a), for example by setting simply $\boldsymbol{a}_k^\lambda = \lambda$. If $d = 1$, then this case can be solved in a globally optimal way using the lifting method [22].
- The Potts or uniform distance, where $d_k(\lambda, \mu) = 1$ if and only if $\lambda = \mu$, and zero otherwise. This distance function can be achieved by setting $\boldsymbol{a}_k^\lambda = \frac{1}{\sqrt{2}}\boldsymbol{e}^\lambda$, where $(\boldsymbol{e}^\lambda)_{\lambda \in \Lambda_k}$ is an orthonormal basis in $\mathbb{R}^{N_k}$, see Fig. 5(b). All changes between labels are penalized equally.
- Another typical case is that the $\boldsymbol{a}_k^\lambda$ denote feature vectors or actual geometric points, for which $|\cdot|_2$ is a natural distance. For example, in the case of optic flow, each label corresponds to a flow vector in $\mathbb{R}^2$, see Fig. 5(c). The representations $\boldsymbol{a}_1^\lambda, \boldsymbol{a}_2^\mu$ are just real numbers, denoting the possible components of the flow vectors in x and y-direction, respectively. The Euclidean distance is a sensible distance on the components to regularize the flow field, corresponding to the regularizer of the TV-L^1 functional in [30]. Optic flow (and other geometric kinds of labels) would however more naturally be modeled with a continuous label space using one of the continuous regularizers in the later subsections.

5.2 New relaxation for the discrete label space

We will now show how to formulate the regularizer J_k^A defined above in the new more general framework. While the previous formulation (44) already yields a relaxation to non-binary functions $\boldsymbol{v}$, we will see that our framework results in a provably tighter one.

Taking a look at theorem 3, we see that the regularizer must penalize the length of the jump set weighted by the label distance. Thus, our general regularizer in (13) must reduce to

$$J_k(u_k) = \int_{S_{u_k}} d_k\left(u_k^-, u_k^+\right) \mathrm{d}\mathcal{H}^{n-1}. \tag{47}$$

where d_k is the same metric as used above in the representation (42). We can see that in order to reduce the general form to the one above, we must enforce a piecewise constant labeling, since the approximate gradient ∇u_k must be constant zero outside the jump set. Applying theorem 1 we can find a convex representation of J_k in terms of the variables $\boldsymbol{v}$, which we formulate in the following proposition in its discretized form.

Proposition 3 *A convex representation of* (47) *in terms of the variables* $\boldsymbol{v}$ *is given by*

$$J_k(u_k) = \sup_{\boldsymbol{p} \in C_k} \left\{ \sum_{\lambda \in \Lambda_k} \int_\Omega v_k^\lambda \,\mathrm{div}\left(\boldsymbol{p}^\lambda\right) \mathrm{d}x \right\}, \tag{48}$$

with

$$C_k = \left\{ \boldsymbol{p} : \Omega \times \Lambda_k \to \mathbb{R}^n \; : \; \left|\boldsymbol{p}^\lambda - \boldsymbol{p}^\mu\right|_2 \leq d_k(\lambda,\mu) \textit{ for all } \lambda,\mu \in \Lambda_k \right\}. \tag{49}$$

Proof We can enforce a piecewise constant labeling u_k, if we enforce the approximate gradient ∇u_k to be constant zero. In (13), this can be achieved by setting $h_k(x, u_k(x), \nabla u_k(x)) = c\,|\nabla u_k|$ with a constant $c > 0$, and then letting $c \to \infty$ to enforce $\nabla u_k \equiv 0$ on $\Omega \setminus S_{u_k}$. Inserting the convex conjugate $h_k^*(x,\lambda,q) = \delta_{\{|q| \leq c\}}$, we find that the conditions in (17) now reduce to

$$b^\lambda \geq 0, \quad \left|\partial_\lambda \boldsymbol{p}^\lambda\right|_2 \leq c, \left|\boldsymbol{p}^\lambda - \boldsymbol{p}^\mu\right|_2 \leq d_k(\lambda,\mu). \tag{50}$$

The supremum over $b^\lambda \geq 0$ is easily eliminated from (16) since $v_k^\lambda \geq 0$, i.e. $-b^\lambda v_k^\lambda \leq 0$ with 0 being the maximum possible value. The second constraint in (50) follows from the third if we choose $c \geq \max_{\lambda > \mu} \frac{d_k(\lambda,\mu)}{|\lambda-\mu|}$. Thus we arrive at (48) with the set C_k as claimed in the proposition. □

We can now establish the relationship between our framework and the regularizer J_k^A derived from a representation of the labels, and show that ours is more tight.

Proposition 4 *Let the regularizer J_k be defined by the relaxation on the right hand side in equation (48). Then for all $\boldsymbol{v}_k \in \mathcal{L}^2(\Omega, \mathrm{co}\,(\Delta_k))$,*

$$J_k(\boldsymbol{v}_k) \geq \mathrm{TV_v}(A_k \boldsymbol{v}_k) = J_k^A(\boldsymbol{v}_k). \tag{51}$$

Equality holds if $\boldsymbol{v}_k$ is binary.

Proof The claim follows from our general formulation (48) with a special choice of the dual variables $\boldsymbol{p}$ together with additional relaxations of the equations in C_k. The special form for $\boldsymbol{p}^\lambda$ we choose is

$$\boldsymbol{p}^\lambda = \sum_{i=1}^{M_k} \boldsymbol{a}_{k,i}^\lambda \boldsymbol{q}_i, \tag{52}$$

with $\boldsymbol{q} : \Omega \times \{1,\dots,M_k\} \to \mathbb{R}^n$ such that $|\boldsymbol{q}|_2 \leq 1$ and the vectors $\boldsymbol{a}_k^\lambda \in \mathbb{R}^{M_k}$ which define the Euclidean representation of d_k, see equation (42). This is only a subset of possible $\boldsymbol{p} \in C_k$ in proposition 4. The constraint on $\boldsymbol{p}$ in (48) is satisfied, since by the Cauchy-Schwarz inequality and the definition of the representation,

$$\begin{aligned} \left|\boldsymbol{p}^\lambda - \boldsymbol{p}^\mu\right|_2 &= \left| \sum_{i=1}^{M_k} (\boldsymbol{a}_{k,i}^\lambda - \boldsymbol{a}_{k,i}^\mu)\boldsymbol{q}_i \right|_2 \\ &\leq \sqrt{\sum_{i=1}^{M_k} (\boldsymbol{a}_{k,i}^\lambda - \boldsymbol{a}_{k,i}^\mu)^2} \cdot \sqrt{\sum_{i=1}^{M_k} |\boldsymbol{q}_i|_2^2} \\ &= \left|A_k \boldsymbol{e}^\lambda - A_k \boldsymbol{e}^\mu\right|_2 \left|\boldsymbol{q}\right|_2 \leq d_k(\lambda,\mu). \end{aligned} \tag{53}$$

Plugging (52) into (48) we obtain the desired result

$$\begin{aligned} J_k(\boldsymbol{v}_k) &\geq \sup_{|\boldsymbol{q}|_2 \leq 1} \left\{ \sum_{\lambda \in \Lambda_k} \int_\Omega \left(\sum_{i=1}^{M_k} \boldsymbol{a}_{k,i}^\lambda \boldsymbol{q}_i \right) \cdot \nabla v_k^\lambda \, \mathrm{d}x \right\} \\ &= \sup_{|\boldsymbol{q}|_2 \leq 1} \left\{ \int_\Omega \sum_{i=1}^{M_k} \boldsymbol{q}_i \cdot \nabla \left(\sum_{\lambda \in \Lambda_k} \boldsymbol{a}_{k,i}^\lambda v_k^\lambda \right) \, \mathrm{d}x \right\} \\ &= \sup_{|\boldsymbol{q}|_2 \leq 1} \left\{ \int_\Omega \sum_{i=1}^{M_k} \boldsymbol{q}_i \cdot \nabla (A_k \boldsymbol{v}_k)_i \, \mathrm{d}x \right\} \\ &= \mathrm{TV}_{\mathrm{v}}(A_k \boldsymbol{v}_k). \end{aligned} \tag{54}$$

The inequality in the first step is a consequence of choosing the special form of $\boldsymbol{p}$'s, thus reducing the set over which the supremum is taken. □

The right hand side of inequality (51) is exactly the previous regularizer used in [11,18]. This implies that for binary functions, the regularizers coincide, which can already be seen from representation (47), see theorem 3. However, if we perform the relaxation to functions taking values between 0 and 1, inequality (51) implies that the new relaxation is more tight, leading to solutions closer to the global optimum.

We will show in the remainder of the section that in addition to handling the discrete case better, our method also can handle continuous regularizers which penalize a *smooth variation* of the labels. This is not possible with the piecewise constant approach of [18,11] which uses vectorial total variation. For instance, our formulation is capable of representing more sophisticated regularizers such as Huber-TV and the piecewise smooth Mumford-Shah functional, as we will show in the following paragraphs. For the regularizers presented in the remainder of this section, relaxations have previously been proposed for the case of a one-dimensional label space in [7,21,22]. However, the framework presented here is more general and allows to combine them freely in the different label dimensions.

5.3 Huber-TV

The TV regularization is known to produce staircasing effects in the reconstruction, i.e. the solution will be piecewise constant. While this is natural in case of a discrete label space, for continuous label spaces it impedes smooth variations of the solution. A remedy for this is replacing the norm $|\nabla u_k|_2$ of the gradient by the Huber function

$$|\nabla u_k|_\alpha := \begin{cases} \frac{1}{2\alpha} |\nabla u_k|_2^2, & \text{if } |u_k|_2 \leq \alpha \\ |\nabla u_k|_2 - \frac{\alpha}{2}, & \text{else.} \end{cases} \tag{55}$$

which smooths out the kink at the origin. The Huber-TV regularizer is then defined by

$$J_k(u_k) = \int_\Omega |\nabla u_k|_\alpha \, \mathrm{d}x. \tag{56}$$

The special case $\alpha = 0$ leads to the usual TV regularization. Theorem (1) gives a convex representation for J_k, see also [22]. The constraint set in (17) is found to be

$$C_k = \Big\{ (\boldsymbol{p}, b) : \Omega \times \Lambda_k \to \mathbb{R}^n \times \mathbb{R} \text{ such that for all } \lambda \in \Lambda_k, \quad b^\lambda \geq \frac{\alpha}{2} \left| \partial_\lambda \boldsymbol{p}^\lambda \right|_2^2, \quad \left| \partial_\lambda \boldsymbol{p}^\lambda \right|_2 \leq 1 \Big\}. \tag{57}$$

5.4 Piecewise smooth Mumford-Shah

The celebrated Mumford-Shah regularizer [1, 21]

$$J_k(u_k) = \int_{\Omega \setminus S_{u_k}} \frac{1}{2\alpha} |\nabla u_k|_2^2 \, \mathrm{d}x \; + \; \nu \mathcal{H}^{n-1}(S_{u_k}) \tag{58}$$

allows to estimate a denoised image u_k which is *piecewise smooth*. Parameter ν can be used to easily control the total length of the jump set S_{u_k}. Bigger values of ν lead to a smaller jump set, i.e. the solution is smooth on wider subregions of Ω. The constraint set in the convex representation of theorem 1 becomes

$$C_k = \Big\{ (\boldsymbol{p}, b) : \Omega \times \Lambda_k \to \mathbb{R}^n \times \mathbb{R} \text{ such that for all } \lambda, \mu \in \Lambda_k, \quad b^\lambda \geq \frac{\alpha}{2} \left| \partial_\lambda \boldsymbol{p}^\lambda \right|_2^2, \quad \left| \boldsymbol{p}^\lambda - \boldsymbol{p}^\mu \right|_2 \leq \nu \Big\}. \tag{59}$$

The limiting case $\alpha = 0$ gives the piecewise constant Mumford-Shah regularizer, which can also be obtained from proposition 3 setting $d_k(\lambda, \mu) = \nu$ for all $\lambda \neq \mu$.

5.5 Truncated linear

For many applications, it is useful to penalize the difference between two label values λ and μ only up to a certain threshold, reasoning that once they are that different, it does not matter anymore how different exactly they are. This means that if $|\lambda - \mu|$ becomes greater than a certain value t, jumps from λ to μ are still penalized only by the constant t. Using linear penalization for small values this leads to the robust *truncated linear* regularizer [7]

$$J_k(u_k) = \int_{\Omega \setminus S_{u_k}} |\nabla u_k|_2 \, \mathrm{d}x \; + \int_{S_{u_k}} \min\left(t, \left|u_k^+ - u_k^-\right|\right) \, \mathrm{d}\mathcal{H}^{n-1}(s). \tag{60}$$

The constraint set for this case is

$$C_k = \Big\{ (\boldsymbol{p}, b) : \Omega \times \Lambda_k \to \mathbb{R}^n \times \mathbb{R} \text{ such that for all } \lambda, \mu \in \Lambda_k, \quad \left| \partial_\lambda \boldsymbol{p}^\lambda \right|_2 \leq 1, \quad \left| \boldsymbol{p}^\lambda - \boldsymbol{p}^\mu \right|_2 \leq t, \quad b = 0 \Big\}. \tag{61}$$

The second constraint needs to be imposed only if $|\lambda - \mu| \geq t$, since otherwise it is already implied by the first constraint.

6 Implementation

6.1 Final relaxation to a convex problem

In order to transform the multilabel problem into the final form which we are going to solve, we formulate it in terms of the indicator functions v_k^λ on the discretized label space using the representation (16) for the regularizer and the relaxation (33) of the data term. Discretization of the label space is necessary now to arrive at a well-posed problem. Let us briefly summarize and review the objects we are dealing with in the final problem. The minimizer we are looking for is a vector $\boldsymbol{v} = (\boldsymbol{v}_k)$ of functions $\boldsymbol{v}_k \in \mathcal{L}^2(\Omega, \mathrm{co}\,(\Delta_k))$, which means that we are looking for a minimizer in a convex set $\mathcal{D}$,

$$\begin{split} \boldsymbol{v} \in \mathcal{D} := \big\{ \boldsymbol{v} \in \mathcal{L}^2(\Omega, \mathbb{R}^{N_1+\ldots+N_d}) \text{ such that for all } x \in \Omega,\ \boldsymbol{v}(x) \in \mathrm{co}\,(\Delta_\times)\,, \\ \text{with } \Delta_\times = \Delta_1 \times \ldots \times \Delta_d \big\}. \end{split} \tag{62}$$

Let us now turn to the regularizer, which is defined via the relaxation in theorem 1. The key ingredients are the convex sets C_k which depend on the kind of regularization we want to use - possible options were detailed in the last section. Let $\mathcal{C} := C_1 \times \ldots \times C_d$ denote the convex set of all dual variables, and define the linear operator $K : \mathcal{C} \to \mathcal{L}^2(\Omega, \mathbb{R}^{N_1+\ldots+N_d})$ via

$$K(\boldsymbol{p}, \boldsymbol{b}) := \left(-\mathrm{div}(\boldsymbol{p}_k^\lambda) - b_k^\lambda \right)_{k=1..d, \lambda \in \Lambda_k}. \tag{63}$$

Then theorem 1 in fact shows that the regularizer can be written in terms of $\boldsymbol{v}$ in the simple form

$$J(\boldsymbol{v}) = \sup_{(\boldsymbol{p},\boldsymbol{b}) \in \mathcal{C}} \left\{ \langle K(\boldsymbol{p}, \boldsymbol{b}), \boldsymbol{v} \rangle \right\}, \tag{64}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{L}^2(\Omega, \mathbb{R}^{N_1+\ldots+N_d})$. The fully relaxed problem we are going to solve can now be written as

$$\underset{\boldsymbol{v} \in \mathcal{D}}{\mathrm{argmin}} \left\{ J(\boldsymbol{v}) + R_{\mathrm{data}}(\boldsymbol{v}) \right\}, \tag{65}$$

using the relaxation R_{data} of the data term defined in (33). We will show in the next subsection that a solution always exists, and describe a numerical algorithm to find one afterwards.

Note that because of the relaxation, the solution might not be binary. If it already has values in Δ_k, we have found the global optimum of the original problem (1), otherwise we have to project the result back to the smaller set of binary valued functions. For this, let $\hat{\boldsymbol{v}}$ be a minimizer of the final relaxation (65). Thus, the functions $\hat{v}_k^\lambda$ might have values in between 0 and 1. In order to obtain a feasible solution to the original problem (1), we just project back to the space of allowed functions. The function $\hat{\boldsymbol{u}} \in \mathcal{L}^2(\Omega, \Gamma)$ closest to $\hat{\boldsymbol{v}}$ is given by setting

$$\hat{\boldsymbol{u}}(x) = \underset{\gamma \in \Gamma}{\mathrm{argmax}} \left\{ \hat{v}_1^{\gamma_1}(x) \cdot \ldots \cdot \hat{v}_d^{\gamma_d}(x) \right\}, \tag{66}$$

i.e. we choose the label where the combined indicator functions have the highest value.

We cannot guarantee that the solution $\hat{\boldsymbol{u}}$ is indeed a global optimum of the original problem (1), since there is nothing equivalent to the thresholding theorem [19] known for this kind of relaxation. However, we still can give a bound how close we are to the global optimum. Indeed, the energy of the optimal solution of (1) must lie somewhere between the energies of $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{u}}$, as previously explained.

6.2 Existence of solutions

Regarding existence of solutions to the final problem (65), one can prove the following proposition.

Proposition 5 *Problem* (65) *always has a minimizer* $\hat{\boldsymbol{v}} \in \mathcal{D}$, *which is in general not unique.*

Proof Both J and R_{data} are support functionals of convex sets in the Hilbert space $\mathcal{L} := \mathcal{L}^2(\Omega, \mathbb{R}^{N_1+\ldots+N_d})$: equation (64) shows that the regularizer J is the support functional of $K\mathcal{C}$, while we can see from definition (33) that the data term R_{data} is the support functional of $\mathcal{Q}$. It follows that both J and R_{data} are lower semi-continuous and convex on $\mathcal{L}$. The set $\mathcal{D}$ is closed, thus its indicator function $\delta_{\mathcal{D}}$ is also convex and closed, furthermore $\delta_{\mathcal{D}}$ is coercive since $\mathcal{D}$ is bounded. From the above, it follows that the functional

$$\boldsymbol{v} \mapsto J(\boldsymbol{v}) + R_{\text{data}}(\boldsymbol{v}) + \delta_{\mathcal{D}}(\boldsymbol{v}) \tag{67}$$

is closed and coercive. Since being closed is equivalent to being lower semi-continuous in the Hilbert space topology of $\mathcal{L}$, these properties imply the existence of a minimizer in $\mathcal{L}$, see theorems 3.2.5 and 3.3.3 in [3], which must necessarily lie in $\mathcal{D}$. Since neither functional is strictly convex, the solution is usually not unique. □

6.3 Numerical method

Using the representation (64) for J, and the definition (33) for the relaxation R_{data}, we can transform the final formulation (65) of the multilabel problem into the saddle point problem

$$\min_{\boldsymbol{v}\in\mathcal{D}} \max_{\substack{(\boldsymbol{p},\boldsymbol{b})\in\mathcal{C}\\ \boldsymbol{q}\in\mathcal{Q}}} \left\{ \langle K(\boldsymbol{p},\boldsymbol{b}) + \boldsymbol{q}, \boldsymbol{v}\rangle \right\}. \tag{68}$$

We minimize the energy (68) with a recent general fast primal-dual algorithm in [8], which is designed for this type of problem. The algorithm is essentially a gradient descent in $\boldsymbol{v}$ and gradient ascent in $\boldsymbol{p}$, $\boldsymbol{b}$ and $\boldsymbol{q}$, with a subsequent application of proximation operators, which act as generalized reprojections. In our case, these are just the usual orthogonal projection onto the constraint sets $\mathcal{D}$, $\mathcal{C}$ and $\mathcal{Q}$. Since they are defined by numerous non-local constraints, a direct projection is quite costly. Therefore, we implement as many constraints as possible using Lagrange multipliers.

First, the simplex constraint $\boldsymbol{v} \in \mathcal{D}$, i.e. $v_k \in \text{co}\,(\Delta_k)$ with Δ_k in (4) for $1 \leq k \leq d$, is enforced by adding the Lagrange multiplier terms

$$\sup_{\boldsymbol{\sigma}} \sum_{k=1}^{d} \int_{\Omega} \sigma_k \left(\sum_{\lambda\in\Lambda_k} v_k^\lambda - 1 \right) \mathrm{d}x \tag{69}$$

to the energy (68), optimizing over $\boldsymbol{\sigma} : \Omega \to \mathbb{R}^d$ in addition to the other variables. This leaves just the simple condition $\boldsymbol{v} \geq 0$ for the indicator variables $\boldsymbol{v}$.

Next, we enforce the constraints $(\boldsymbol{p}, \boldsymbol{b}) \in \mathcal{C}$ on the dual variables of the regularizer by introducing new variables

$$\boldsymbol{d}_k^\lambda = \partial_\lambda \boldsymbol{p}_k^\lambda \qquad \text{or} \qquad \boldsymbol{d}_k^{\lambda\mu} = \boldsymbol{p}_k^\lambda - \boldsymbol{p}_k^\mu, \tag{70}$$

depending on the kind of constraints in C_k. Corresponding to these, we add Lagrange multiplier terms

$$\inf_{\boldsymbol{\eta}} \left\langle \boldsymbol{\eta}, \partial_\gamma \boldsymbol{p}_k^\lambda - \boldsymbol{d}_k^\lambda \right\rangle \qquad \text{or} \qquad \inf_{\boldsymbol{\eta}} \left\langle \boldsymbol{\eta}, \boldsymbol{p}_k^\lambda - \boldsymbol{p}_k^\mu - \boldsymbol{d}_k^{\lambda\mu} \right\rangle \tag{71}$$

to the energy to enforce the equalities (70). Instead of computing the projection of $(\boldsymbol{p}, \boldsymbol{b})$ in each step, we can then instead perform the projection of the new variables $(\boldsymbol{d}_k, \boldsymbol{b}_k)$ on a corresponding constraint set. The advantage is that it decouples into independent projections of $\boldsymbol{d}_k^\lambda$ or $\boldsymbol{d}_k^{\lambda\mu}$ and b_k^λ onto simple convex sets, which are easy to implement. Alternatively, constraints of the form $\left|\boldsymbol{p}_k^\lambda - \boldsymbol{p}_k^\mu\right|_2 \le m$ can be enforced using convex duality, by simply adding the terms

$$\inf_{\boldsymbol{\eta}} \left\langle \boldsymbol{\eta}, \boldsymbol{p}_k^\lambda - \boldsymbol{p}_k^\mu \right\rangle + m \left|\boldsymbol{\eta}\right|_2 \tag{72}$$

to the energy instead of (71). We used this way in our implementation, as it turns out to be much faster in practice. The optimization (68) is now performed over $\boldsymbol{v}$, $\boldsymbol{p}$, $\boldsymbol{b}$, $\boldsymbol{q}$, $\boldsymbol{\sigma}$ and $\boldsymbol{d}, \boldsymbol{\eta}$.

Finally, the projection of a $\tilde{\boldsymbol{q}}$ onto $\mathcal{Q}$ consists of solving

$$\operatorname*{argmin}_{\boldsymbol{q} \in \mathcal{Q}} \left\{ \sum_{k, \lambda \in \Lambda_k} \frac{1}{2} \left(q_k^\lambda - \tilde{q}_k^\lambda\right)^2 \right\} \tag{73}$$

pointwise for each $x \in \Omega$. The number of constraints in $\mathcal{Q}$, as defined in (34), equals the total number of labels in the product space. Unfortunately, implementing these constraints by adding Lagrange multipliers to the *global* problem (68), i.e. for each $x \in \Omega$, is not possible for larger problems since it requires too many dual variables to be memory efficient. Thus, for larger problems, the projection needs to be computed explicitly after each iteration, which increases run time, see Fig. 6. To make sure that $\boldsymbol{q}$ lies in $\mathcal{Q}$ we add the Lagrange multiplier terms

$$\sup_{\boldsymbol{\mu} \ge 0} \left\{ \sum_{\gamma \in \Gamma} \mu^\gamma \left(q_1^{\gamma_1} + \ldots + q_d^{\gamma_d} - c^\gamma\right) \right\} \tag{74}$$

to the local energy (73). This results in another saddle point problem to be optimized over now unconstrained $\boldsymbol{q}$ and $\boldsymbol{\mu} \ge 0$, which we again solve using the algorithm in [8]. Specifically, since the $\boldsymbol{q}$-only terms are uniformly convex, we use the algorithm 2 of [8] with the accelerated $\mathcal{O}(\frac{1}{N^2})$ convergence rate. Note that for each $x \in \Omega$ we thus need $\mathcal{O}(N_1 \cdots N_d)$ memory to solve the local problem (73), which at first *seems* to contradict our statement to substantially reduce the overall memory requirements to $\mathcal{O}((N_1 + \ldots + N_d)|\Omega|)$. However, the problems (73) are *independent* of each other for different $x \in \Omega$ and can be solved in chunks of $\mathcal{O}\big((N_1 + \ldots + N_d)|\Omega|/N_1 \cdots N_d\big)$ points x in parallel. Since there is only a small change in the variables $\boldsymbol{q}$ per outer iteration, only a small number of inner iterations is required. In our experiments, we used 10 inner iterations.

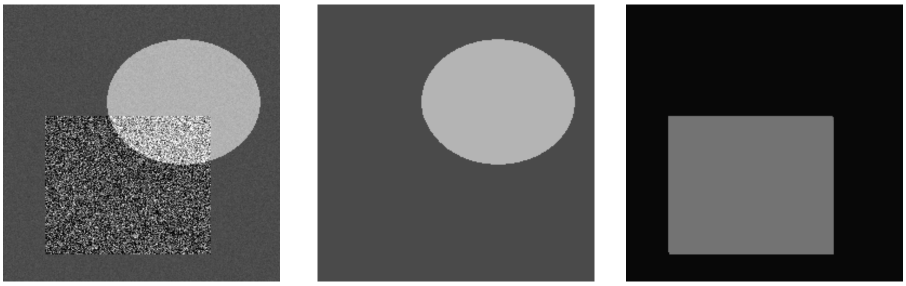
# of Pixels	# Labels	Memory [Mb]		Run time [s]	
$P = P_x \times P_y$	$N_1 \times N_2$	Previous	Proposed (g/p)	Previous	Proposed (g/p)
320 × 240	8 × 8	112	112 / 102	196	26 / 140
320 × 240	16 × 16	450	337 / 168	*	80 / 488
320 × 240	32 × 32	1800	1124 / 330	*	215 / 1953
320 × 240	50 × 50	4394	2548 / 504	*	950 / 5188
320 × 240	64 × 64	7200	4050 / 657	-	1100 / 8090
640 × 480	8 × 8	448	521 / 413	789	102 / 560
640 × 480	16 × 16	1800	1351 / 676	*	295 / 1945
640 × 480	32 × 32	7200	4502 / 1327	-	1290 / 7795
640 × 480	50 × 50	17578	10197 / 2017	-	- / 32887
640 × 480	64 × 64	28800	16202 / 2627	-	- / 48583

Fig. 6: *The table shows the total amount of memory required for the implementations of the previous and proposed methods depending on the size of the problem. For the proposed method, the projection* (73) *of the data term dual variables can be implemented either globally (g), or slower but more memory efficient as a sub-problem of the proximation operator (p), here using $N_1/5$ chunks. Also shown is the total run time for 5000 iterations, which usually suffices for convergence. Numbers in red indicate a memory requirement larger than what fits on the largest currently available CUDA capable devices (6 GB). Failures marked with a "$*$" are due to another limitation: the shared memory is only sufficient to store the temporary variables for the simplex projection up until dimension 128. Note that the proposed framework can still handle all problem sizes above.*

7 Experiments

We demonstrate the correctness and usability of our method on several examples. Different regularizers are used in the examples. In the cases where the regularizer can be simulated with the previous relaxation [11], we compared the resulting optimality bounds. On average, our bounds were approximately three times better ($3-5\%$ with the proposed framework compared to $10-15\%$ with the previous relaxation). All experiments were performed with a parallel CUDA implementation running on a nVidia GTX 480 GPU, respectively on a TESLA C2070 for larger problems.

When the domain Ω is discretized into P pixels, the primal and dual variables are represented as matrices. For (68), we have to store $P \cdot (N_1 + ... + N_d)$ floating point numbers for the primal variables $\boldsymbol{v}$, and $P(n+2) \cdot (N_1 + ... + N_d)$ floating point numbers for the dual variables $\boldsymbol{p}$, $\boldsymbol{b}$ and $\boldsymbol{q}$. Depending on whether the projection (73) is implemented globally or as a sub-problem of the proximation, as described in



Input, spatially varying noise　　Denoised image and reconstructed standard deviation

Fig. 7: *The algorithm allows to jointly recover the unknown standard deviation σ of the noise as well as the intensity of a denoised image by solving a single optimization problem. Ground truth: Within rectangle Gaussian noise with standard deviation $\sigma = 0.25$, outside $\sigma = 0.02$; image intensity within ellipsoid $u = 0.7$, outside $u = 0.3$. Image resolution is 256×256 using 32×32 labels. Computation time is 4.4 minutes.*

the previous section, we need $P \cdot N_1 \cdots N_d$, respectively $n_c \cdot N_1 \cdots N_d$ floating point numbers for the auxiliary variables $\boldsymbol{\mu}$ in (74). The number of "chunks" n_c can be chosen appropriately depending on available memory, e.g. $n_c = Pa \cdot (N_1 + \ldots + N_d)/N_1 \cdots N_d$ with some constant $a > 0$. Finally, e.g. for the TV or Huber-TV regularizer with the constraint set (57), the auxiliary variables $\boldsymbol{d}_k^\lambda$ in (70) are stored using $Pn \cdot (N_1 + \ldots + N_d)$ floating point numbers.

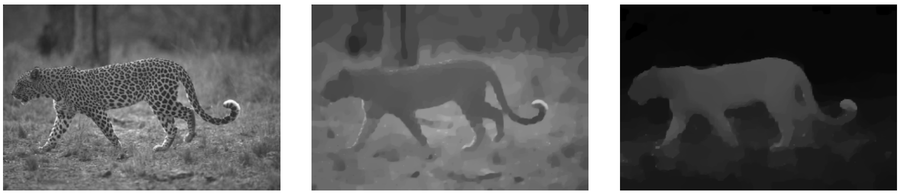
In contrast, without using our reduction scheme, the memory requirements grow proportional to $N_1 \cdots N_d$ instead of only $N_1 + \ldots + N_d$. For the algorithm [8], we need space for dual variables and two times the primal variables in total, so we end up with the total values shown in Fig. 6. Thus, problems with large number of labels can only be handled with the proposed reduction technique.

7.1 Adaptive denoising

As a novel application of a multi-dimensional label space, we present adaptive denoising, where we *jointly* estimate a noise level and a denoised image by solving a single minimization problem. Note that here we require the continuous label space to represent the image intensity range. The Mumford-Shah energy can be interpreted as a denoising model which yields the maximum a posteriori estimate for the original image under the assumption that the input image f was distorted with Gaussian noise of standard deviation σ. If this standard deviation is itself viewed as an unknown which varies over the image, the label space becomes two-dimensional, with one dimension representing the unknown intensity u of the original image, the second dimension representing the unknown standard deviation σ of the noise. The data term of the energy can then be written as [6]

$$\int_\Omega \frac{(u-f)^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2)\,\mathrm{d}x. \tag{75}$$

Results of the optimization can be observed in Fig. 7 and Fig. 8. For the regularizer, we used piecewise constant Mumford-Shah for both σ and u in Fig. 7, and piecewise



Input, textured object

Simultaneous piecewise smooth approximation of intensity (left) and standard deviation (right)

Fig. 8: *A piecewise smooth image approximation of both intensity and noise standard deviation using* (75) *and the Mumford-Shah regularizer for both u and σ. This model allows to separate textured objects in a natural way by jointly estimating the mean and standard deviation of image intensities. The amount of smoothing is stronger in region of larger standard deviation. Image resolution is* 320×214 *using* 32×32 *labels, leading to a run time of* 10.3 *minutes.*

smooth Mumford-Shah in Fig. 8. In the real world example Fig. 8, the solution can be interpreted as a uniformly smooth approximation, where all regions attain a similar smoothness level regardless of the amount of texture in the input.

7.2 Depth and Occlusion map

In this test, we simultaneously compute a depth map and an occlusion map for a stereo pair of two color input images $I_L, I_R : \Omega \to \mathbb{R}^3$. The occlusion map shall be a binary map denoting whether a pixel in the left image has a matching pixel in the right image. Thus, the space of labels is two-dimensional with Λ_1 consisting of the disparity values and a binary Λ_2 for the occlusion map. We use the a TV smoothness penalty on the disparity values. A Potts regularizer is imposed for the occlusion map. The distance on the label space thus becomes

$$d(\gamma, \chi) = s_1 \left|\gamma_1 - \chi_1\right| + s_2 \left|\gamma_2 - \chi_2\right| , \tag{76}$$

with suitable weights $s_1, s_2 > 0$. We penalize an occluded pixel with a constant cost $c_{occ} > 0$, which corresponds to a threshold for the similarity measure above which we believe that a pixel is not matched correctly anymore. The cost associated with a label γ at $(x, y) \in \Omega$ is then defined as

$$c^{\gamma}(x, y) = \begin{cases} c_{occ} & \text{if } \gamma_2 = 1, \\ \left|I_L(x, y) - I_R(x - \lambda_1, y)\right|_2 & \text{otherwise.} \end{cases} \tag{77}$$

The result for the "Moebius" test pair from the Middlebury benchmark is shown in Fig. 9. The input image resolution was scaled to 640×512, requiring 128 disparity labels, which resulted in a total memory consumption which was slightly too big for previous methods, but still in reach of the proposed algorithm. Total computation time required was 1170 seconds.

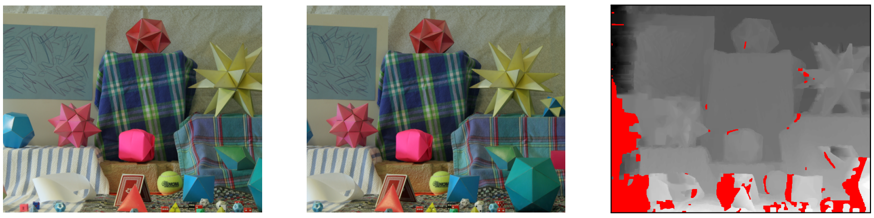


Fig. 9: *The proposed method can be employed to simultaneously optimize for a displacement and an occlusion map. This problem is also too large to be solved by alternative relaxation methods on current GPUs. From left to right: Left and right input image I_L and I_R, and computed disparity and occlusion map; red areas denote occluded pixels.*

7.3 Optic Flow

In this experiment, we compute optic flow between two color input images $I_0, I_1 : \Omega \to \mathbb{R}^3$ taken at two different time instants. The space of labels is again two-dimensional, with $\Lambda_1 = \Lambda_2$ denoting the possible components of flow vectors in x and y-direction, respectively. We regularize both directions with either TV or a truncated linear penalty on the component distance, i.e.

$$d(\gamma, \chi) = s \min(t, |\gamma_1 - \chi_1|) + s \min(t, |\gamma_2 - \chi_2|) \,, \tag{78}$$

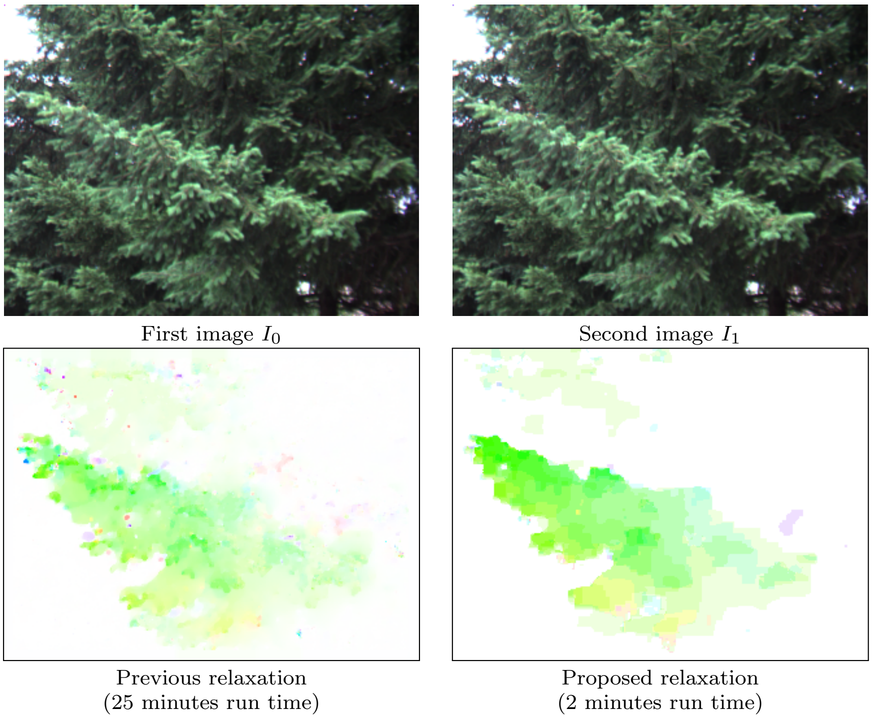
with a suitable manually chosen weight $s > 0$ and threshold $t > 0$. Note that we can provide a tight relaxation of the exact penalizer, which was only coarsely approximated in the previous approaches [11,18]. The cost function just compares pointwise pixel colors in the images, i.e.

$$c^{\gamma}(x, y) = |I_0(x, y) - I_1(x + \gamma_1, y + \gamma_2)|_2 \,. \tag{79}$$

Results can be observed in Figs. 1, 10, 11 and 12. See Fig. 11 for the color code of the flow vectors. In all examples, the number of labels is so high that this problem is currently impossible to solve with previous convex relaxation techniques by a large margin, see Fig. 6. Compared to the relaxation proposed in the original conference publication [11], total computation time was reduced dramatically, see Fig. 10. Due to the global optimization of a convex energy, we can successfully capture large displacements without having to implement a coarse-to-fine scheme, see Fig. 11. A comparison of the energies of the continuous and discretized solution shows that we are within 5% of the global optimum for all examples.

8 Conclusion

We have introduced a continuous convex relaxation for multi-label problems where the label space is a product space. Such labeling problems are plentiful in computer vision. The proposed reduction method improves on previous methods in that it requires orders of magnitude less memory and computation time, while retaining the advantages: a very flexible choice of regularizer on the label space, a globally optimal solution of the relaxed problem and an efficient parallel GPU implementation with guaranteed convergence.



First image I_0

Second image I_1

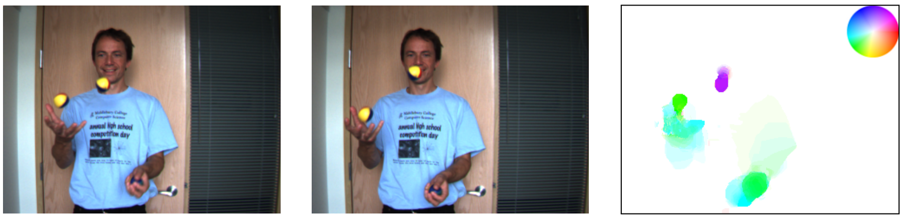
Previous relaxation (25 minutes run time)

Proposed relaxation (2 minutes run time)

Fig. 10: *Optical flow fields with* $32{\times}32$ *labels computed on an image with resolution* 320×240 *using TV regularization. With the new relaxation of the regularizers, we achieve optimality bounds which are on average three times lower than with previous relaxations from [11, 18]. Since the scaling of the regularity term is not directly comparable, we chose optimal parameters for both algorithms manually. The large time difference results from a narrow constraint on the time step for [18].*

Compared to the original conference publication [11], we presented a much tighter relaxation for the products in the data term, which avoids the problem of a trivial pointwise solution and therefore eliminates the need for additional smoothing. Furthermore, we improved upon the regularization by combining the advantages of the efficient multi-dimensional relaxation with the tight relaxation of the regularizers in [7]. The new framework also allows to formulate more general continuous regularizers on multi-dimensional label spaces and thus solve a more general class of problems efficiently. For example, we can explicitly encourage the solution to be smooth in certain regions, and can represent Huber-TV and truncated linear regularization by an exact and tight relaxation. The regularizers can be arbitrarily mixed, in the sense that each dimension of the label space can have its own type of regularity.

Because of the reduced memory requirements, we can successfully handle specific problems with very large number of labels, which could not be done with previous convex relaxation techniques. Among other examples we presented a convex relaxation for the optic flow functional with truncated linear penalizer on the distance between the flow vectors. To our knowledge, this is the first relaxation for this functional which can be optimized globally and efficiently.

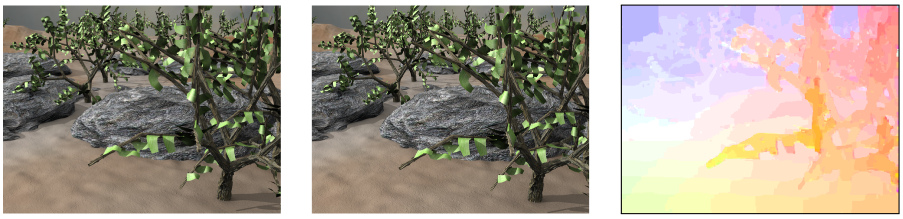


First image I_0 Second image I_1 Flow field and color code

Fig. 11: *When employed for optic flow, the proposed method can successfully capture large displacements without the need for coarse-to-fine approaches, since a global optimization is performed over all labels. In contrast to existing methods, our solution is within a known bound of the global optimum.*

References

1. Albert, G., Bouchitt, G., Maso, G.D.: The calibration method for the Mumford-Shah functional and free-discontinuity problems. Calculus of Variations and Partial Differential Equations **16**(3), 299–333 (2002) 3, 4, 7, 10, 11, 20
2. Alberti, G., Bouchitté, G., Maso, G.D.: The calibration method for the Mumford-Shah functional. C. R. Acad. Sci. Paris Sr. I Math. **329 (1999)**(3), 249–254 (1999) 4
3. Attouch, H., Buttazzo, G., Michaille, G.: Variational Analysis in Sobolev and BV Spaces. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (2006) 8, 9, 22
4. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: IEEE International Conference on Computer Vision (ICCV), pp. 26–33 (2003) 4
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001) 4
6. Brox, T., Cremers, D.: On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional. International Journal of Computer Vision **84**, 184–193 (2009) 25
7. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Tech. Rep. TR-2008-05, Dept. of Computer Science, University of Bonn (2008) 2, 3, 5, 7, 15, 19, 20, 28
8. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. preprint (2010) 22, 23, 25
9. Cremers, D., Kolev, K.: Multiview stereo and silhouette consistency via convex functionals over convex domains. IEEE Transactions on Pattern Analysis and (2010) 2
10. Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., Navab, N.: Optical flow estimation with uncertainties through dynamic MRFs. In: Proc. International Conference on Computer Vision and Pattern Recognition (2008) 4
11. Goldluecke, B., Cremers, D.: Convex relaxation for multilabel problems with product label spaces. In: Proc. European Conference on Computer Vision (2010) 2, 3, 5, 10, 12, 13, 14, 15, 19, 24, 27, 28
12. Greig, D., Porteous, B., Seheult, A.: Exact maximum a posteriori estimation for binary images. J. Royal Statistics Soc. **51**(Series B), 271–279 (1989) 4
13. Ishikawa, H.: Exact optimization for Markov random fields with convex priors. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10), 1333–1336 (2003) 4, 5
14. Kindermann, R., Snell, J.: Markov Random Fields and Their Applications. American Mathematical Society (1980) 4
15. Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., Cremers, D.: An experimental comparison of discrete and continuous shape optimization methods. In: European Conference on Computer Vision (ECCV), pp. 332–345 (2008) 4
16. Kolmogorov, V., Rother, C.: Minimizing non-submodular functions with graph cuts - a review. IEEE Transactions on Pattern Analysis and **29**(7), 1274–1279 (2007) 4
17. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004) 4



First image I_0 Second image I_1 Flow field

Fig. 12: *Example with a larger image resolution of* 640×480 *pixels, which requires* 32×32 *labels. Regularizer is the total variation in each component. Computation time is* 21.6 *minutes.*

18. Lellmann, J., , Becker, F., Schnörr, C.: Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In: IEEE International Conference on Computer Vision (ICCV) (2009) 2, 3, 5, 15, 16, 17, 19, 27, 28
19. Nikolova, M., Esedoglu, S., T.Chan: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal of Applied Mathematics **66**(5), 1632–1648 (2006) 2, 4, 21
20. Pock, T., Chambolle, A., Bischof, H., Cremers, D.: A convex relaxation approach for computing minimal partitions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 810–817 (2009) 5, 7
21. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In: Proc. International Conference on Computer Vision (2009) 2, 7, 8, 15, 19, 20
22. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: Global solutions of variational models with convex regularization. SIAM Journal on Imaging Sciences (2010) 2, 4, 5, 7, 8, 15, 17, 19, 20
23. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: European Conference on Computer Vision (ECCV), pp. 792–805 (2008) 5
24. Schlesinger, D., Flach, B.: Transforming an arbitrary min-sum problem into a binary one. Tech. rep., Dresden University of Technology (2006) 4
25. Shekhovtsov, A., Garcia-Arteaga, J., Werner, T.: A discrete search method for multimodal non-rigid image registration. In: Proceedings of the 2008 IEEE CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (2008) 4
26. Shekhovtsov, A., Kovtun, I., Hlavac, V.: Efficient MRF deformation model for non-rigid image matching. Computer Vision and Image Understanding **112**, 91–99 (2008) 4
27. Szeliski, R.: Bayesian modeling of uncertainty in low-level vision. International Journal of Computer Vision **5**(3), 271–301 (1990) 4
28. Wainwright, M., Jaakkola, T., Willsky, A.: Map estimation via agreement on trees: message-passing and linear programming. IEEE Trans. Inf. Theory **51**(11), 3697–3717 (2005) 4
29. Zach, C., Gallup, D., Frahm, J., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vision, Modeling and Visualization, pp. 243–252 (2009) 2, 5
30. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime $TV - L^1$ optical flow. In: Pattern Recognition (Proc. DAGM), pp. 214–223 (2007) 17