

MRF Optimization with Separable Convex Prior on Partially Ordered Labels

Csaba Domokos^{1,2} ^{*}, Frank R. Schmidt^{1,2}, and Daniel Cremers¹

¹ Technical University of Munich, Garching bei München, Germany
`cremers@tum.de`

² Bosch Center for Artificial Intelligence, Renningen, Germany
`{csaba.domokos, frank.r.schmidt}@de.bosch.com`

Abstract. Solving a multi-labeling problem with a convex penalty can be achieved in polynomial time if the label set is totally ordered. In this paper we propose a generalization to partially ordered sets. To this end, we assume that the label set is the Cartesian product of totally ordered sets and the convex prior is separable. For this setting we introduce a general combinatorial optimization framework that provides an approximate solution. More specifically, we first construct a graph whose minimal cut provides a lower bound to our energy. The result of this relaxation is then used to get a feasible solution via classical move-making cuts. To speed up the optimization, we propose an efficient coarse-to-fine approach over the label space. We demonstrate the proposed framework through extensive experiments for optical flow estimation.

Keywords: Multi-labeling problem, Poset, Sub-modular relaxation

1 Introduction

Many computer vision problems like stereo matching [1–3], semantic image segmentation [4] or optical flow estimation [5, 6] can be formulated as a *multi-labeling problem*. For a set of variables \mathcal{V} and a finite label set \mathcal{L} , a mapping $f: \mathcal{V} \rightarrow \mathcal{L}$ is called a *multi-labeling*. The multi-labeling problem aims to find a multi-labeling f that minimizes an energy $E(f)$. In general, this problem is known to be NP-hard, moreover, there is no algorithm that can approximate this general energy minimization with an approximation ratio better than some exponential function in the input size [7]. Nevertheless, by making some assumptions the multi-labeling problem becomes tractable [8–10].

In this paper we address the problem of solving a multi-labeling problem. In order to find an optimal multi-labeling $f: \mathcal{V} \rightarrow \mathcal{L}$ we want to minimize an energy of the form

$$E(f) = \sum_{i \in \mathcal{V}} E_i(f_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(f_i, f_j), \quad (1)$$

^{*} This work was done while the author was an Alexander von Humboldt Fellow at the Technical University of Munich

where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes the pairwise dependencies of different variables. The energies $E_i: \mathcal{L} \rightarrow \mathbb{R}$ and $E_{ij}: \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_0^+$ describe the data fidelity terms and pairwise smoothness terms, respectively. While the data term E_i for all $i \in \mathcal{V}$ can be chosen arbitrarily, the smoothness terms E_{ij} are of the following form

$$E_{ij}(f_i, f_j) = w_{ij} \cdot d(f_i, f_j) \quad \text{for all } (i, j) \in \mathcal{E} . \quad (2)$$

The energy (1) corresponds to a Markov random field (MRF) formulation [8] over an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $P(f) \sim \exp(-E(f))$. Here, $w_{ij} \geq 0$ depends on the input data and $d: \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_0^+$ is a *metric* on \mathcal{L} . Under these mild restrictions, it is known that (1) can be minimized globally in polynomial time if $|\mathcal{L}| = 2$ [11] or if \mathcal{L} is the totally ordered set $\{1, \dots, \ell\}$ and there is an even, convex function $g: \mathbb{R} \rightarrow \mathbb{R}_0^+$ such that $d(f_i, f_j) = g(f_i - f_j)$ [8].

In this paper we focus on a more general setting of partially ordered label sets \mathcal{L} . In particular, we assume that $\mathcal{L} = \mathcal{L}_1 \times \dots \times \mathcal{L}_k$ can be written as the Cartesian product of k different totally ordered label sets. In addition, we assume that the function d that penalizes different labels for interacting pixels (*i.e.* for all $(i, j) \in \mathcal{E}$) has the form $d(f_i, f_j) = g(f_i - f_j)$, where g is an even, separable convex function, *i.e.* a sum of regularizers for each dimension of the label space.

The rest of this paper is organized as follows. We give a short overview of the related work in Section 1.1. In Section 2 we introduce the theoretical background of partially ordered sets, a.k.a. posets. The two main contributions of the paper can be summarized as follows:

- We propose a combinatorial optimization framework, which can be applied for minimizing energies defined on poset labelings. Namely, we show a general graph construction (see Section 2.2), whose minimal cut provides a lower bound to our energy. This relaxation is exploited to get a feasible solution by making use of classical move-making cuts [1]. The proposed graph construction can handle arbitrary data costs and separable convex smoothness costs.
- We also propose an efficient coarse-to-fine strategy in the label space (see Section 3), which effectively reduces the possible search space and results in a considerable speed-up of the algorithm.

As an illustration of the proposed optimization scheme we consider the problem of optical flow estimation. Comprehensive experiments in Section 4 show that the proposed method provides competitive results with other combinatorial optimization algorithms at reduced complexity. Section 5 concludes the paper.

1.1 Related Work

Partially ordered label sets are very common in several computer vision applications like optical flow estimation, image registration, stereo exposure fusion, *etc.*, where the label set \mathcal{L} is the Cartesian product of totally ordered sets.

Schekhovtsov *et al.* [12] proposed an MRF model for image registration, where the deformation is described by a coupled field of discrete x - and y -displacements of pixels. The model consists of two layers of variables. The inter-layer interaction is used to encode the data term, and the intra-layer interactions encode pairwise (smoothness) constraints for neighboring pixels. This model leads to a simpler relaxation to which the sequential tree-reweighted message passing (TRW-S) algorithm [2] is applied. Chen and Koltun [6] addressed the problem of optical flow estimation, where the classical Horn-Schunck objective [13] is minimized over a regular grid by making use of the TRW-S algorithm [2]. Another discrete optimization approach was presented in [5] for optical flow estimation. The authors formulated the problem as a discrete inference and applied a block coordinate descent method, which iteratively optimizes all image rows and columns via dynamic programming.

Kohli *et al.* [14] considered the problem of optimizing multi-label pairwise MRFs. The multi-label MRF model is first converted into an equivalent binary MRF and then it is relaxed, which can be efficiently solved using a maximum flow algorithm [11]. The solution provides a partially optimal labeling of the binary variables, which is transferred to the multi-label problem. A detailed review for minimizing functions with both sub-modular³ and non-submodular terms can be found in [15], referred to as the QPBO method (quadratic pseudo-Boolean optimization). The output of QPBO, however, is a *partial labeling*, which means there is a special label that is interpreted as “unknown”.

Goldstein *et al.* [16] presented a general variational functional lifting technique for minimizing vector-valued problems. This technique allows to find global minimizers for optical flow. The authors consider total-variation as regularizer. In contrast to our approach, L_2^2 penalty cannot be considered in [16]. A continuous convex relaxation for multi-label problems was proposed in [17] for the case when the label space is a continuous product space and the regularizer is separable. Through the relaxed problem, various problems like optic flow, stereo matching and segmentation can be solved within provable bounds of the global optimum. This approach allows a very general class of continuous regularizers on multi-dimensional label spaces. The regularizers can be arbitrarily mixed, in the sense that each dimension of the label space can have its own type of regularity. We note that, in contrast to continuous relaxations, we focus on combinatorial optimization approaches in this paper.

2 Energy Minimization on Posets

In the following, we address the problem of minimizing (1) if \mathcal{L} is a partially ordered set (or poset). In Section 2.1 we provide a short introduction to posets [18] and explain their difficulties in an energy minimization framework. In Section 2.2 we show how to design a sub-modular energy that is a relaxed version of (1). In particular, we will show in Section 2.3 how to efficiently minimize this lifted

³ A set function $f: 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is called sub-modular, if for any pair of subsets $A, B \subset \mathcal{V}$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ is satisfied

energy by finding a minimal cut in a graph and how to employ a heuristic projection scheme in order to find a feasible solution of the original energy.

2.1 Posets, Lower Level Sets and Lower Ideals

A partially ordered set is a set \mathcal{L} together with a relation that stores for any pair of elements $\alpha, \beta \in \mathcal{L}$ whether the statement $\alpha \leq \beta$ is true or not.

Definition 1 (Poset). *Given a set \mathcal{L} and a relation \leq on \mathcal{L} . We call (\mathcal{L}, \leq) a partially ordered set or poset if the following conditions are satisfied for all $\alpha, \beta, \gamma \in \mathcal{L}$*

$$\begin{aligned} \alpha &\leq \alpha && \text{(Reflexivity)} \\ \alpha \leq \beta, \beta \leq \alpha &\Rightarrow \alpha = \beta && \text{(Antisymmetry)} \\ \alpha \leq \beta, \beta \leq \gamma &\Rightarrow \alpha \leq \gamma && \text{(Transitivity)} \end{aligned}$$

(\mathcal{L}, \leq) is called a totally ordered set if, for any pair $\alpha, \beta \in \mathcal{L}$ the statement $\alpha \leq \beta$ or $\beta \leq \alpha$ is true.

The main difference between posets and totally ordered sets is that there may be two different elements $\alpha, \beta \in \mathcal{L}$ in a poset for which we cannot decide whether one element is larger than the other. From now on we use the notation $\alpha < \beta$ iff $\alpha \leq \beta$ and $\alpha \neq \beta$ holds. The easiest way to create a poset is to take the Cartesian product of two or more totally ordered sets.

Lemma 1 (Cartesian Product). *Let (\mathcal{L}_1, \leq_1) and (\mathcal{L}_2, \leq_2) be two totally ordered sets. The Cartesian product $\mathcal{L} := \mathcal{L}_1 \times \mathcal{L}_2$ becomes a poset (\mathcal{L}, \leq) via*

$$(\alpha_1, \alpha_2) \leq (\beta_1, \beta_2) :\Leftrightarrow (\alpha_1 \leq_1 \beta_1) \wedge (\alpha_2 \leq_2 \beta_2) .$$

Proof. Follows directly from the definition of posets.

A common way to visualize the internal structure of a poset is to consider its *Hasse diagram*.

Definition 2 (Hasse Diagram). *Let (\mathcal{L}, \leq) be a finite poset. Then, the Hasse diagram of \mathcal{L} is a directed graph $\mathcal{H} = (\mathcal{L}, \mathcal{E}_{\mathcal{L}})$ with the vertex set \mathcal{L} and the edge set*

$$\mathcal{E}_{\mathcal{L}} := \{(\beta, \alpha) \in \mathcal{L} \times \mathcal{L} \mid \alpha < \beta, \forall \gamma \in \mathcal{L}: \neg(\alpha < \gamma < \beta)\} .$$

For the totally ordered set $\mathcal{L} = \{1, \dots, \ell\}$, the Hasse diagram has exactly $\ell - 1$ edges. These edges are of the form $(\alpha + 1, \alpha)$. Thus, the Hasse diagram of a totally ordered set is always a chain. If \mathcal{L} is a poset on the other hand, the Hasse diagram becomes a DAG (directed acyclic graph) (see Figure 1).

Of particular interest for the next section is the set of lower ideals.

Definition 3. *For each $\alpha \in \mathcal{L}$, we refer to the set*

$$[\alpha] := \{\beta \in \mathcal{L} \mid \beta \leq \alpha\}$$

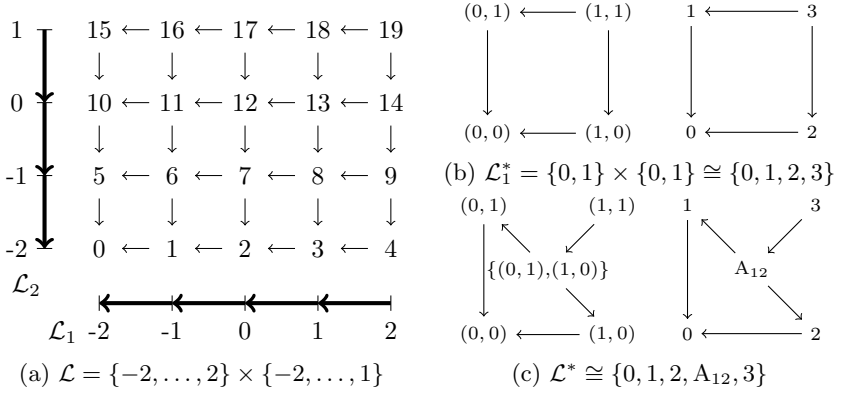


Fig. 1. Hasse diagrams. (a) Hasse diagram for the poset $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2$, where $\mathcal{L}_1 = \{-2, \dots, 2\}$ and $\mathcal{L}_2 = \{-2, \dots, 1\}$ are totally ordered sets. The Hasse diagrams of \mathcal{L}_1 and \mathcal{L}_2 are chains. (b) Two isomorphic Hasse diagrams for $\mathcal{L} = \{0, 1\} \times \{0, 1\} \cong \mathcal{L}_1^*$. (c) Two isomorphic Hasse diagrams for $\mathcal{L}^* = \mathcal{L}_1^* \cup \{(1, 0)\} \cup \{(0, 1)\} \cong \{0, 1, 2, 3, A_{12}\}$

as its lower level set. Further, we call a subset $I \subset \mathcal{L}$ a lower ideal if the following holds

$$\alpha \in I \Rightarrow [\alpha] \subset I.$$

We denote the set of all lower ideals as $\mathcal{L}^* \subset 2^{\mathcal{L}}$ and the set of all lower level sets as $\mathcal{L}_1^* \subset \mathcal{L}^*$.

In fact, every element of \mathcal{L}^* can be represented as the union of elements included in \mathcal{L}_1^* . In other words, a lower ideal $L \in \mathcal{L}^*$ is a set that accumulates lower level sets, that is

$$L = \bigcup_{\alpha \in L} [\alpha].$$

Note that, by construction, both \mathcal{L} and \mathcal{L}_1^* have the same cardinality. Nevertheless, \mathcal{L}^* can be larger than \mathcal{L}_1^* . We also remark that the elements of \mathcal{L}_1^* are subsets of \mathcal{L} .

It is worth noting that for totally ordered sets, we always have $\mathcal{L}^* = \mathcal{L}_1^*$, which has the same cardinality as \mathcal{L} . Thus, the difference between lower ideals and lower level sets is only observable for posets.

Examples 1) For the totally ordered set $(\mathcal{L}, \leq) = (\{0, 1\}, \leq)$ we obtain the lower level sets as follows:

$$[0] = \{0\}, \quad [1] = \{0, 1\}, \quad \mathcal{L}_1^* = \{[0], [1]\} = \mathcal{L}^*.$$

2) For the poset $(\mathcal{L}, \leq) = (\{0, 1\} \times \{0, 1\}, \leq)$ we obtain

$$\begin{aligned} [(0, 0)] &= \{(0, 0)\}, & [(1, 0)] &= \{(0, 0), (1, 0)\}, \\ [(0, 1)] &= \{(0, 0), (0, 1)\}, & [(1, 1)] &= \mathcal{L}, \end{aligned}$$

therefore $\mathcal{L}_1^* = \{(0, 0), [(1, 0)], [(0, 1)], [(1, 1)]\} \cong \{0, 1, 2, 3\}$ and we have

$$\mathcal{L}^* = \mathcal{L}_1^* \cup \{[(1, 0)] \cup [(0, 1)]\} \cong \{0, 1, 2, 3, A_{12}\}. \quad (3)$$

Thus, for posets there is a difference between lower level sets and lower ideals (see Figure 1). We will refer to this difference

$$\mathcal{L}^A := \mathcal{L}^* - \mathcal{L}_1^* \subset 2^{\mathcal{L}} \quad (4)$$

as the *augmented label set*, or equivalently $\mathcal{L}^* = \mathcal{L}_1^* \cup \mathcal{L}^A$. Please note that the cardinality of \mathcal{L}^A may grow exponentially with respect to $|\mathcal{L}|$. In Section 2.2 we will see how these augmented labels appear if we lift our energy (1). In fact, the augmented labels result in an infeasible solution. To obtain a feasible solution without augmented labels, we propose a heuristic projection scheme.

2.2 Energy Lifting

From now on we assume a poset $(\mathcal{L}, \leq) = (\mathcal{L}, \subset)$, where $\mathcal{L} = \mathcal{L}_1 \times \dots \times \mathcal{L}_k$ is the Cartesian product of k totally ordered sets and $\mathcal{H} = (\mathcal{L}, \mathcal{E}_{\mathcal{L}})$ its Hasse diagram. Let E be of the form (1). Furthermore, we assume that the smoothness term E_{ij} is of the form (2) and that $d(f_i, f_j) = g(f_i - f_j)$ can be represented via an even, separable convex function g . We want to construct a graph \mathcal{G} such that each labeling $f: \mathcal{V} \rightarrow \mathcal{L}$ corresponds to an s - t cut of \mathcal{G} with $E(f)$ as its cut value [11].

Totally Ordered Label Set In the simple case $k = 1$, thus $\mathcal{L} = \mathcal{L}_1$ is a totally ordered set and we can follow the construction of Ishikawa [8] to design a graph with the desired properties. The used vertices consist of a *source* s , a *sink* t and the *internal nodes* $\mathcal{V} \times \mathcal{L}$. The edges can be divided into three different classes.

The *constraint edges* between (i, ℓ) and $(i, \ell - 1)$ of infinite capacities guarantee that in an optimal cut the binary labeling of the set $\{i\} \times \mathcal{L}$ has the form $(1, \dots, 1, 0, \dots, 0)$, where 1 indicates that a vertex is connected with the source.

The *data edges* can be designed as terminal links between s , respectively t , and (i, ℓ) . This formulation is due to [19] and differs from the original formulation of [8]. The *smoothness edges* of capacity $w_{ij} \cdot c_{\delta}$ between vertices $(i, \ell + \delta)$ and (j, ℓ) for all $(i, j) \in \mathcal{E}$ model the convex function g . This is done by using the non-negative values

$$c_0 = g_1 - g_0 \quad \text{and} \quad c_{\delta} = g_{\delta+1} - 2g_{\delta} + g_{\delta-1} \quad (\forall \delta > 0).$$

For more details we refer to [8].

Partially Ordered Label Set For the general case $k > 1$ we want to design a different graph with the desired properties. Like before, the used vertices are the source s , the sink t and the internal vertices $\mathcal{V} \times \mathcal{L}$. Also, we introduce *constraint edges*, *data edges* and *smoothness edges*. While these edges will be different from Ishikawa's construction [8], they serve nonetheless the same purpose.

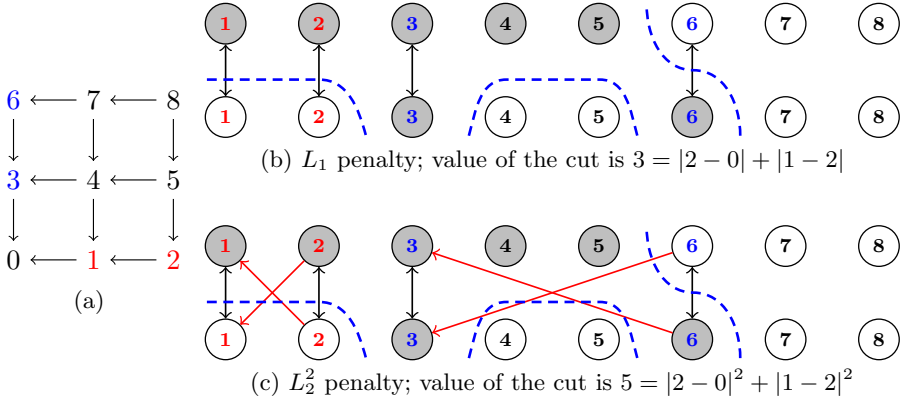


Fig. 2. Graph construction for smoothness terms. (a) Hasse diagram of the poset $\mathcal{L} = \{0, 1, 2\} \times \{0, 1, 2\}$. (b), (c) Graph construction for L_1 and L_2^2 penalties, resp., where the gray and white nodes connected to the source and sink (corresponding to 1 and 0), resp. This example shows the case that $f_i = (2, 1) \in \mathcal{L}$, $f_j = (0, 2) \in \mathcal{L}$. The cut is shown by the blue dashed lines. Note that only 1-0 edges should be cut

The *constraint edges* should also connect a label ℓ with its immediate predecessor ℓ' . Due to the partial ordering, ℓ' is not unique. Thus, we use the Hasse diagram $\mathcal{H} = (\mathcal{L}, \mathcal{E}_{\mathcal{L}})$ of \mathcal{L} and introduce an edge of infinite capacity between (i, ℓ) and (i, ℓ') for each $(\ell, \ell') \in \mathcal{E}_{\mathcal{L}}$. As a consequence, we obtain a labeling $\hat{f}: \mathcal{V} \rightarrow \mathcal{L}^*$ instead of $f: \mathcal{V} \rightarrow \mathcal{L}$. Note that the set \mathcal{L}^* of lower ideals contains the lower level sets \mathcal{L}_1^* and the augmented labels \mathcal{L}^A . Since there is a one-to-one relationship between \mathcal{L}_1^* and \mathcal{L} , *i.e.* they are isomorph⁴, we can understand \hat{f} as a *relaxation* of f and we will denote \hat{f} as $\hat{f}: \mathcal{V} \rightarrow (\mathcal{L} \cup \mathcal{L}^A)$.

The *data edges* should reflect the data terms $E_i(f_i)$. Since a label $f_i \in \mathcal{L}$ is now represented by the lower level set $[f_i] \in \mathcal{L}_1^*$, we have to associate a unary data cost of $D_{i,\ell}$ with the vertex (i, ℓ) such that the following holds

$$\sum_{\ell \in [f_i]} D_{i,\ell} = E_i(f_i) \quad \forall i \in \mathcal{V}, f_i \in \mathcal{L}. \quad (5)$$

Since the Hasse diagram is a DAG, the matrix of this system of linear equations is (after permutation) in upper triangular form. Therefore, the Problem (5) can be readily solved by successive substitution. If the resulting $D_{i,\ell}$ is positive, it results in an edge of capacity $D_{i,\ell}$ from (i, ℓ) to the sink t . Otherwise, it results in an edge of capacity $-D_{i,\ell}$ from the source s to (i, ℓ) [9].

The *smoothness edges* should reflect the pairwise smoothness terms, that is, $E_{ij}(f_i, f_j) = w_{ij} \cdot g(f_i - f_j)$. Here, the special structure of our posets comes into play. $\mathcal{L} = \mathcal{L}_1 \times \dots \times \mathcal{L}_k$ results in k -dimensional labels, therefore, we write $f_i = (f_{i,1}, \dots, f_{i,k})$ and $f_j = (f_{j,1}, \dots, f_{j,k})$. Since we assume that g is an even,

⁴ Two posets are said to be isomorph, when their Hasse diagrams as graphs are isomorph to each other

separable convex function, we have k even, convex functions g_κ for $\kappa = 1, \dots, k$ such that

$$d(f_i, f_j) = \sum_{\kappa=1}^k g_\kappa(f_{i,\kappa} - f_{j,\kappa}). \quad (6)$$

Since a label $f_i \in \mathcal{L}$ is now represented by its lower level set $[f_i]$, this lower level set also contains

$$(0_1, \dots, 0_{\kappa-1}, f_{i,\kappa}, 0_{\kappa+1}, \dots, 0_k), \quad (7)$$

where $0_{\kappa'}$ denotes the minimal element of the totally ordered set $\mathcal{L}_{\kappa'}$. Therefore, it is enough to encode g_κ on

$$\hat{\mathcal{L}}_\kappa := \{0_1\} \times \dots \times \{0_{\kappa-1}\} \times \mathcal{L}_\kappa \times \{0_{\kappa+1}\} \times \dots \times \{0_k\}.$$

Note that $\hat{\mathcal{L}}_\kappa$ is a totally ordered set and we can therefore replicate Ishikawa's idea for all $\kappa = 1, \dots, k$ in order to design the smoothness edges. Note that this is possible since g is separable convex. For more details we refer to Figure 2.

Overall, we have proved the following theorem.

Theorem 1. *Let \mathcal{L} be a poset that can be represented as the Cartesian product of k totally ordered sets \mathcal{L}_κ , $\kappa = 1, \dots, k$. Further consider the multi-labeling problem of minimizing the energy (1) for $f: \mathcal{V} \rightarrow \mathcal{L}$*

$$E(f) = \sum_{i \in \mathcal{V}} E_i(f_i) + \sum_{(i,j) \in \mathcal{E}} E_{i,j}(f_i, f_j),$$

where the smoothness term is given as

$$E_{i,j}(f_i, f_j) = w_{ij} \cdot d(f_i, f_j) = w_{ij} \sum_{\kappa=1}^k g_\kappa(f_{i,\kappa} - f_{j,\kappa}) \quad w_{ij} \geq 0$$

for even, convex functions g_κ for all $\kappa = 1, \dots, k$. Then we can define a lifted, sub-modular, graph-representable functional $D: [\mathcal{V} \rightarrow (\mathcal{L} \cup \mathcal{L}^A)] \rightarrow \mathbb{R}$ such that

$$D(f) = E(f) \quad \text{if } f: \mathcal{V} \rightarrow \mathcal{L}. \quad (8)$$

So far, we found an optimal labeling $f: \mathcal{V} \rightarrow (\mathcal{L} \cup \mathcal{L}^A)$. If this labeling is in fact a labeling $f: \mathcal{V} \rightarrow \mathcal{L}$ that excludes augmented labels, we globally solved the original multi-labeling problem. This can happen if the considered data terms are very pronounced. Nonetheless, we should assume that in practice augmented labels will occur. While $D(f) = E(f)$ is satisfied for the lower level sets, we like to emphasize that our energy (1) is in general not sub-modular⁵. We consider an energy with sub-modular pairwise terms, however, the arbitrary unary terms make the energy non-submodular. The proposed relaxation is graph-representable, thus it is sub-modular⁶. Thus, we can compute the global optimum of the relaxed energy at the cost of having augmented labels. In the next section we provide a heuristics in order to remove these augmented labels.

⁵ The proof is contained in the supplementary material

⁶ Graph-representability implies sub-modularity [9]

2.3 Resolving Augmented Labels

Assume that $\mathcal{L}^A \ni f_i = \bigcup_{\mu=1}^m [\alpha_\mu]$. One way of resolving the ambiguity would be to apply move-making methods like $\alpha - \beta$ swaps [1] over the labels $[\alpha_1], \dots, [\alpha_m]$. Nonetheless, we like to point to a different heuristic that takes the structure of the poset better into account. The idea is to also consider those labels that can be constructed by the join operation \vee

$$\alpha \vee \beta = \min\{\gamma \mid \alpha \leq \gamma \text{ and } \beta \leq \gamma\}.$$

Let us consider, for example, the label space $\mathcal{L} = \{0, 1\} \times \{0, 1\}$ and let $f_i = [(1, 0)] \cup [(0, 1)]$. In this case we consider all $\alpha - \beta$ swaps with respect to $\{(1, 0), (0, 1), (1, 1)\}$. The rationale is that the energy with respect to f_i accumulated the data terms of $[(1, 0)]$ and $[(0, 1)]$. Since the energy with respect to $[(1, 1)]$ also accumulates these data terms (and the data term of $(1, 1)$), it makes sense to broaden the label space for the move-making methods.

Discussion In many applications the label set is defined as a *lattice*⁷, (*i.e.* regular grid). Topkis [18] presented a theory of sub-modular energy minimization on a lattice. Although our label set also forms a lattice, our energy (1) is not sub-modular. In [20] a general hierarchical model is introduced, where the label space forms an arbitrary tree specifying a partial ordering over the labels. The authors proposed effective multi-labeling moves, called Path-Moves [20]. The Path-Moves algorithm can be seen as a combination of well-known α -expansion [1] and Ishikawa's construction [8]. Nonetheless, the label set that we consider in this paper is a lattice, rather than a tree, therefore Path-Moves algorithm cannot be directly applied.

3 Coarse-to-fine Strategy

In practice, the minimization of the lifted energy (8) becomes quickly intractable as the number of labels grows. Therefore, it is beneficial to have the number of possible labels as small as possible. In addition, we deal with the relaxation to our original energy. There is no guarantee that we obtain a feasible solution. Accordingly, for some pixels we may obtain augmented labels (*i.e.* combination of labels), that we need to resolve so as to get a feasible solution. Note that the number of the augmented labels grows exponentially by increasing the size of the label sets, which makes the augmented label removal very challenging. To overcome these issues, we consider the following *coarse-to-fine* approach.

To simplify the notation we assume that $k = 2$ and $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2$. In the first iteration we consider only $m \times n$ labels for each pixel, where m and n are divisors of the size of \mathcal{L}_1 and \mathcal{L}_2 , respectively. Each of the coarse labels correspond

⁷ If two elements α and β of a poset have a least upper bound (greatest lower bound), denoted by $\alpha \vee \beta$ ($\alpha \wedge \beta$), it is their *join* (*meet*). A poset that contains the join and the meet for each pair of its element is a *lattice* [18]

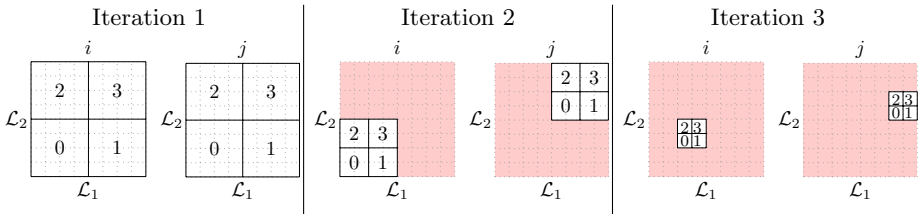


Fig. 3. Illustration of the proposed coarse-to-fine strategy over the label space $\mathcal{L} = \{0, \dots, 7\} \times \{0, \dots, 7\}$, where $m = n = 2$. In each iteration the search space for each pixel is partitioned into $mn = 4$ equal regions, indexed by, resp., 0,1,2 and 3, and the optimal region is sought. Only this optimal region of the labels space will be considered in the next iteration. The rest of the labels, shown in red, will be ignored

to a region of labels. After a decision on the coarsest level, the next iteration only considers the region, that has been selected in the previous iteration. This common approach is illustrated in Figure 3. After some iterations either \mathcal{L}_1 or \mathcal{L}_2 cannot be divided anymore. This means that the remaining part of the optimization boils down the minimization over a totally ordered set, which can be globally solved via Ishikawa’s construction [8].

For the data term on the coarse level we apply *min pooling* over the labels belonging to the same region. Thus, we have a strong guidance for the optimization at the current level. For the smoothness terms we are using the distance between the centers of the selected patches.

It is important to note that, in contrast to the previous works [21], we apply a coarse-to-fine approach in the label space instead of the image domain. Moreover, the goal of our method is to compute labelings that provide useful results in practice, even if not all labels can be chosen optimally. Like α -expansion, our method tries to find a local optimum as quickly as possible. For that reason we can only provide a weak-persistency guarantee, namely that the global optimum is found if no augmented label is inferred.

4 Numerical Experiments

In this section we discuss the implementation details of the proposed minimization scheme and illustrate it through optical flow estimation.

4.1 Implementation Details

We ran our experiments on a machine with Intel Xeon E5-2697 CPU@2.3GHz under Linux in `Matlab` with `C/C++ mex` extensions. For the maximum flow calculation and for move-making algorithms (*i.e.* $\alpha - \beta$ swap and α -expansion) we used the publicly available GCO library [1, 9, 11]. In order to have a fair comparison with other methods we used `float` representation of the energy terms. Our implementation is publicly available at <https://github.com/csabdomokos/MRFOptimizationOnPosets>.

Minimization In order to minimize our relaxed energy, we applied the BK algorithm [11]. During the flow graph construction, for each pixel an augmenting path is sought through the data edges and the constraint edges corresponding to the given pixel. This pre-processing has linear time complexity and ends up a better runtime of the BK algorithm, since the BK algorithm has the worst case complexity $\mathcal{O}(|\mathcal{E}| |\mathcal{V}|^2 C)$, where C is the value of the minimum cut in the flow graph [11].

Augmented labels In order to resolve augmented labels, *i.e.* unfeasible solutions, we applied the heuristics that we explored in Section 2.3. That is, we considered a 2×2 label space in each iteration of the proposed coarse-to-fine approach. Therefore we only have one augmented label, *i.e.* $\alpha = [(0, 1)] \cup [(1, 0)]$, and we select a feasible label among the labels $\{(0, 1), (1, 0), (1, 1)\}$ via standard $\alpha - \beta$ swap moves [1]. More precisely, the augmented labels are replaced with a feasible label corresponding to the lowest data cost for the given pixel. Afterwards the $\alpha - \beta$ swap algorithm [1] is run over all three label pairs. The $\alpha - \beta$ swap algorithm requires the pairwise terms to be semi-metric, which is satisfied in our case, since we assume even functions in our energy.

4.2 Optical Flow Estimation

To substantiate the quality of our optimization we focus on the optical flow application. Assuming an input image pair I_1 and I_2 , the classical optical flow estimation aims to find the displacement between pixels in I_1 and corresponding pixels in I_2 [13]. In a discrete setting one can consider totally ordered (finite) label sets \mathcal{L}_1 and \mathcal{L}_2 to model the horizontal and vertical displacements. The labels for each pixel is taken from the poset $\mathcal{L}_1 \times \mathcal{L}_2$. The goal is to find an optimal labeling $f: \mathcal{V} \rightarrow \mathcal{L}_1 \times \mathcal{L}_2$ such that $I_1(p_i) = I_2(p_i + f_i)$.

Recently, Chen and Koltun [6] have proposed an efficient solution for optical flow estimation. Here, we defined our energy, adopted from [6], as

$$E(f) = \sum_{i \in \mathcal{V}} E_i(f_i) + \lambda \sum_{(i,j) \in \mathcal{E}} w_{ij} |f_i - f_j|, \quad (9)$$

where $\lambda = 0.021$ and w_{ij} represents the contrast-sensitive weighting factors. The data cost has the form of $E_i(f_i) = 1 - \max(0, \text{NCC}(i, f_i))$, where $\text{NCC}(i, f_i)$ is the *normalized cross-correlation* between the patches of size 3×3 centered at pixels i and $i + f_i$, respectively. In order to prevent the penalty of negatively correlated patches, negative values are clamped to zero. The pairwise smoothness terms are defined as the *contrast-sensitive Potts model* [24], that is, the edge based weighting factors w_{ij} are calculated as

$$w_{ij} = \exp\left(-\frac{\|I_1(i) - I_2(j)\|_2^2}{2\sigma^2}\right), \quad \text{where } \sigma = \frac{1}{\sqrt{6}}.$$

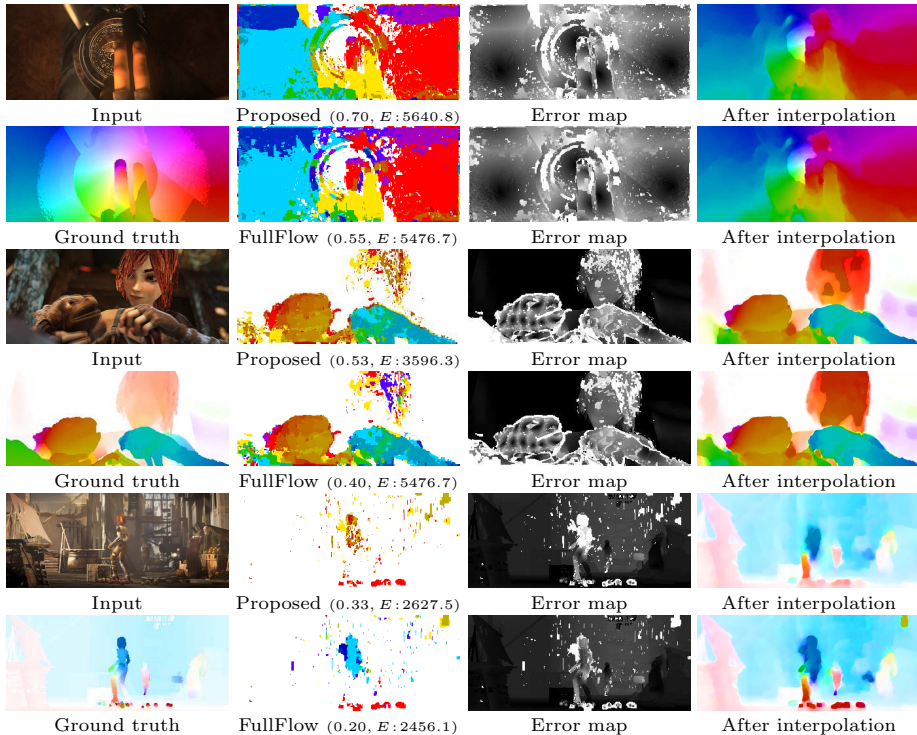


Fig. 4. Qualitative results on the Sintel dataset [22]. The input images along with the ground truth are in the first column. The results obtained by our method and the FullFlow [6] method, resp., are shown in the second column. The average endpoint errors and the energy values E are in parenthesis. The corresponding error maps are in the third column. The results in the last column are obtained after EpicFlow [23] interpolation

Post-processing In several methods, the estimated optical flow is interpolated further to obtain sub-pixel accuracy [25, 6]. Recently, it has been a common technique to apply EpicFlow interpolation [23] as post-processing. EpicFlow requires point matches as an input and the final result is achieved through variational optimization. We adopted the interpolation from the paper [6]. Accordingly, we also used EpicFlow interpolation [23] (see Figure 4).

4.3 Evaluation

For evaluation we used the MPI Sintel dataset [22], which is a naturalistic optical flow dataset derived from a 3D animated film Sintel. Each image has a resolution of 438×1024 pixels. The data set includes a variety of challenging features like long sequences, large motions, specular reflections, motion blur, defocus blur and atmospheric effects. We ran our experiments on the *training* set with the *final* sequences, including motion blur. By following the settings in [6], we first rescaled

Sequence	$ \mathcal{L} $	Baseline		$\alpha - \beta$ swap		α -expansion		FullFlow [6]		Proposed		[12]
		EPE	rt.	EPE	rt.	EPE	rt.	EPE	rt.	EPE	rt.	EPE
sleeping_1	8×8	1.12	2.23	0.56	3.06	0.55	3.16	0.52	0.86	0.60	0.35	1.77
sleeping_2	8×8	0.71	1.50	0.53	2.82	0.53	2.52	0.47	0.86	0.53	0.24	1.46
shaman_3	16×16	1.10	6.44	0.76	27.48	0.73	14.19	0.62	2.74	0.85	0.44	1.45
alley_1	32×32	1.45	20.79	0.85	377.34	0.77	61.85	0.58	10.93	0.92	0.56	1.44
alley_2	32×32	3.69	29.45	1.24	445.39	1.16	74.93	0.74	11.23	1.32	0.70	3.48
bandage_2	32×32	0.93	19.44	0.54	314.45	0.53	59.49	0.40	10.94	0.57	0.45	0.93
shaman_2	32×32	1.29	16.03	0.65	377.79	0.58	67.21	0.36	10.97	0.95	0.62	0.92
ambush_7	64×64	1.97	57.80	1.58	9278.76	1.36	286.73	0.65	47.33	1.40	0.89	3.30
market_2	64×64	1.67	79.07	1.02	5851.91	1.19	306.62	0.58	47.96	1.61	0.84	1.27

Table 1. Quantitative comparison to other combinatorial optimization approaches on the Sintel dataset [22]. EPE and rt., resp., stand for the mean value of the average endpoint error and the runtime (sec.). All experiments were ran on a single CPU core

the input images by a factor of $1/3$. We considered sequences, having 50 images, with various maximum displacements of 10, 22, 46 and 94, which correspond to the label set of size 8×8 , 16×16 , 32×32 and 64×64 , respectively, after rescaling. As evaluation measure the *average endpoint error* was used. Some qualitative results can be seen in Figure 4.

Comparison Our experiments were targeted at providing a comprehensive comparison to state of the art combinatorial optimization approaches. As a baseline we ran alternating optimization, initialized from the zero flow, where the global optimization method [8] was used for each direction. We considered classical move-making algorithms, that is, $\alpha - \beta$ swap and α -expansion [1]. In case of the TRW-S method, we used the implementation of the FullFlow method [6]. In contrast to [6], we ran the code on a single CPU core in order to have a fair runtime comparison. Only three iterations of the TRW-S method were computed. For the sake of completeness, we also ran the method of Shekhovtsov *et al.* [12]. We used the authors implementation with similar settings as in the case of other methods. We remark that the implementation of [12] applies the TRW-S method as inference, however, the considered energy is not the same as the energy (9), therefore, this comparison is not completely fair.

The quantitative results are shown in Table 1. We can observe that the classical move-making algorithms become quickly prohibitive as the size of the label set grows. Our proposed method provides comparable accuracy to those methods. The FullFlow method always provided the least average endpoint error, but its runtime grows linearly with respect to $|\mathcal{L}|$. Our method provided moderately worse results comparing to the FullFlow method, however, the runtime of our method increases very slowly and always stayed below a second. The method [12] provided larger errors than the other methods.

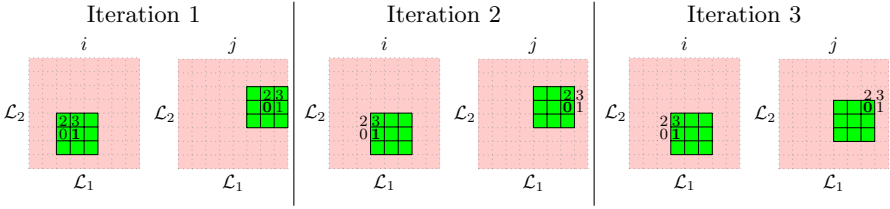


Fig. 5. Illustration of three iterations of label refinement. At the given level of the coarse-to-fine approach, we have the (coarse) labels $f_i = 1$ and $f_j = 0$, and consider their 3×3 neighborhoods in the (coarse) label space $\{0, \dots, 7\} \times \{0, \dots, 7\}$, shown by green, for refinement. In the next iteration the 3×3 neighborhood of the refined label is considered

Label Refinement One can observe from Table 1 that the error obtained by our method grows with the size of the label set. In fact, there is an inherent limitation of our coarse-to-fine strategy. When it makes a decision for a pixel at a current level, then only the corresponding region of labels will be taken into account in the later iterations. Although, the min-pooling operation provides a strong guidance, the labeling at the current level is not necessarily optimal. To overcome this limitation, we investigated a label refinement technique.

In each iteration, we get a feasible solution, which is then refined by applying local move-making cuts. More precisely, for the current labeling we consider only the labels at the given level of the coarse-to-fine approach, and explore 3×3 neighborhoods in the label space (see Figure 5). The classical $\alpha - \beta$ swap algorithm is used over the 3×3 regions in order to refine the current labeling. We reconsider the resulting labels again and use the same process until no more improvement is possible. As the $\alpha - \beta$ swap always decreases the energy, convergence is guaranteed. We observed slightly improvement of the results, however, at the price of higher runtime (see the supplementary material).

5 Conclusions

In this work we have presented a new approach to compute a (locally) optimal labeling for a specific class of partially ordered label sets. We assume that the label set \mathcal{L} can be represented as the Cartesian product of k different totally ordered label sets \mathcal{L}_k . Under the assumption that the convex prior on \mathcal{L} is separable with respect to the k totally ordered label sets, we were able to design a graph-representable sub-modular energy. While this energy leads to a relaxed solution, we could show that the relaxation helps us to guide local move-making methods. In combination with variational post-processing, we were able to provide optical flow results that are comparable with state-of-the-art methods, based on combinatorial approaches, at reduced time complexity.

Acknowledgment This work was partially supported by the Alexander von Humboldt Foundation.

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(11) (November 2001) 1222–1239
2. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(10) (October 2006) 1568–1583
3. Komodakis, N., Tziritas, G.: Approximate labeling via graph cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(8) (August 2007) 1436–1453
4. Ladický, L., Russell, C., Kohli, P., Torr, P.H.S.: Associative hierarchical random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(6) (June 2013) 1056–1077
5. Menze, M., Heipke, C., Geiger, A.: Discrete optimization for optical flow. In Gall, J., Gehler, P., Leibe, B., eds.: *Proceedings of German Conference on Pattern Recognition*. Volume 9358 of LNCS., Aachen, Germany, Springer (2015) 16–28
6. Chen, Q., Koltun, V.: Full flow: Optical flow estimation by global optimization over regular grids. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, IEEE* (June 2015)
7. Li, M., Shekhovtsov, A., Huber, D.: Complexity of discrete energy minimization problems. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: *Proceedings of European Conference on Computer Vision*. Volume 9906 of *Lecture Notes in Computer Science*., Amsterdam, The Netherlands, Springer (October 2016) 834–852
8. Ishikawa, H.: Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10) (October 2003) 1333–1336
9. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(2) (February 2004) 147–159
10. Ramalingam, S., Kohli, P., Alahari, K., Torr, P.H.S.: Exact inference in multi-label CRFs with higher order cliques. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, IEEE* (June 2008)
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9) (September 2004) 1124–1137
12. Shekhovtsov, A., Kovtun, I., Hlaváč, V.: Efficient MRF deformation model for non-rigid image matching. *Computer Vision and Image Understanding* **112**(1) (October 2008) 91–99
13. Horn, B.K., Schunck, B.G.: Determining optical flow. *Artificial Intelligence* **17**(1–3) (August 1981) 185–203
14. Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., Torr, P.H.S.: On partial optimality in multi-label MRFs. In: *Proceedings of International Conference on Machine Learning, Helsinki, Finland, ACM* (July 2008) 480–487
15. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts—A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(7) (July 2007) 1274–1279
16. Goldstein, T., Bresson, X., Osher, S.: Global minimization of Markov random fields with applications to optical flow. *Inverse Problems & Imaging* **6**(4) (November 2012) 623–644

17. Goldluecke, B., Strelakovski, E., Cremers, D.: Tight convex relaxations for vector-valued labeling. *SIAM Journal on Imaging Sciences* **6**(3) (March 2013) 1626–1664
18. Topkis, D.M.: Minimizing a submodular function on a lattice. *Operations Research* **26**(2) (March–April 1978) 305–321
19. Li, K., Wu, X., Chen, D., Sonka, M.: Optimal surface segmentation in volumetric images - A graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(1) (November 2006) 119–134
20. Isack, H., Veksler, O., Oguz, I., Sonka, M., Boykov, Y.: Efficient optimization for hierarchically-structured interacting segments (HINTS). In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, IEEE* (July 2017)
21. Carr, P., Hartley, R.: Solving multilabel graph cut problems with multilabel swap. In: *Proceedings of International Conference on Digital Image Computing, Melbourne, Australia, IEEE* (December 2009)
22. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: *Proceedings of European Conference on Computer Vision*. Volume 7577 of *Lecture Notes in Computer Science.*, Springer (October 2012) 611–625
23. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, IEEE* (June 2015)
24. Kohli, P., Ladický, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, IEEE* (2008)
25. Bailer, C., Taetz, B., Stricker, D.: Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In: *Proceedings of IEEE International Conference on Computer Vision, Santiago, Chile, IEEE* (December 2015) 4015–4023