Full Flow: Optical Flow Estimation By Global Optimization over Regular Grids

Qifeng Chen
Stanford University

Vladlen Koltun
Intel Labs

Abstract

We present a global optimization approach to optical flow estimation. The approach optimizes a classical optical flow objective over the full space of mappings between discrete grids. No descriptor matching is used. The highly regular structure of the space of mappings enables optimizations that reduce the computational complexity of the algorithm’s inner loop from quadratic to linear and support efficient matching of tens of thousands of nodes to tens of thousands of displacements. We show that one-shot global optimization of a classical Horn-Schunck-type objective over regular grids at a single resolution is sufficient to initialize continuous interpolation and achieve state-of-the-art performance on challenging modern benchmarks.

1. Introduction

Optical flow is a vital source of information for visual perception. Animals use optical flow to track and control self-motion, to estimate the spatial layout of the environment, and to perceive the shape and motion of objects [36, 43]. In computer vision, optical flow is used for visual odometry, three-dimensional reconstruction, object segmentation and tracking, and recognition.

The classical approach to dense optical flow estimation is to optimize an objective of the form

$$E(f) = E_{\text{data}}(f) + \lambda E_{\text{reg}}(f),$$

where $f$ is the estimated flow field, $E_{\text{data}}$ is a data term that penalizes association of visually dissimilar areas, and $E_{\text{reg}}$ is a regularization term that penalizes incoherent motion [22]. Traditionally, this objective is optimized by iterative local refinement that maintains and updates a single candidate flow [39]. This local refinement does not optimize the objective globally over the full space of flows and is prone to local minima.

Global optimization of the flow objective has generally been considered intractable unless significant restrictions are imposed [38]. Menze et al. [30] achieved impressive results with a discrete optimization approach, but had to heuristically prune the space of flows using descriptor matching.

In this work, we develop a global optimization approach that optimizes the classical flow objective (1) over the full space of mappings between discrete grids. Our work demonstrates that a direct application of global optimization over full regular grids has significant benefits. Since the highly regular structure of the space of mappings is preserved, we can employ optimizations that take advantage of this structure to reduce the computational complexity of the algorithm’s inner loop. The overall approach is simple and does not involve separately-defined descriptor matching modules: simply optimizing the classical flow objective over full grids is sufficient. We show that this minimalistic approach yields state-of-the-art accuracy on both the Sintel [8] and the KITTI 2015 [29] optical flow benchmarks.

2. Background

The variational approach to optical flow originates with Horn and Schunck [22]. This elegant approach posits a clear global objective (1) and produces a dense flow field connecting the two images. Since the space of flows is so large, the variational objective has traditionally been optimized locally. Starting with a simple initialization, the flow is iteratively updated by gradient-based steps [4, 6, 39]. Through these iterations, a single candidate flow is maintained. While this local refinement approach can be accurate when displacements are small [3, 39], it does not optimize the objective globally over the full space of flows and is prone to local minima.

Recent methods have used descriptor matching and nearest neighbor search to initialize the continuous refinement [7, 45, 11, 44, 32, 2]. This more sophisticated initialization is known to significantly improve results in the presence of large displacements. However, the descriptor matching module is trained separately, does not optimize a coherent objective over the provided correspondence sets, and can yield globally suboptimal initializations. We show that state-of-the-art accuracy can be achieved by globally optimizing the classical objective (1), with no separately trained or designed descriptors.

A number of approaches to global optimization for op-
tical flow estimation have been proposed. Steinbrücker et al. [37] use an alternating scheme to optimize a quadratic relaxation of the global objective. This formulation relies on the assumption that the regularizer is convex. A number of subsequent approaches use functional lifting to map the problem into a higher-dimensional space, where the optimization reduces to estimating a collection of hypersurfaces [20, 19, 38]. These schemes likewise impose certain assumptions on the model, such as requiring the data term or the regularizer to be convex. In general, these approaches have not been shown to produce state-of-the-art results on modern benchmarks.

Our approach treats objective (1) as a Markov random field and uses discrete optimization techniques. The Markov random field perspective on optical flow estimation dates back to the 80s and discrete optimization techniques have been applied to the problem in different forms since that time [26, 21]. Glockner et al. [18, 17] applied MRF optimization to sets of control points in coarse-to-fine schemes. In contrast, we operate on dense grids with large two-dimensional label spaces. A number of works considered a simplified MRF formulation that decomposes the horizontal and vertical components of the flow [34, 27, 47]. In contrast, we demonstrate the feasibility of operating on much larger models with two-dimensional label spaces. Lempitsky et al. [28] iteratively improved the estimated flow field by generating proposals and integrating them using the QPBO algorithm. In contrast, we optimize over the full space of mappings between discrete grids. Komodakis et al. [25] evaluated MRF optimization on optical flow estimation with small displacements. In contrast, we show that global optimization over full two-dimensional label spaces is tractable and yields state-of-the-art performance on challenging large-displacement problems.

Menze et al. [30] pruned the space of flows using feature descriptors and optimized an MRF on the pruned label space. In contrast, we argue that operating on the full space is both feasible and desirable. First, we avoid heuristic pruning and the reliance on separately-defined feature descriptors that are not motivated by the flow objective itself. Second, pruning destroys the highly regular structure of the space of mappings. We show that optimization over the full space can be significantly accelerated due to the regularity of the space. In particular, the full regular structure enables the use of highly optimized min-convolution algorithms that reduce the complexity of message passing from quadratic to linear [15, 9].

3. Model

Let $I_1, I_2 : \Omega \to \mathbb{R}^3$ be two color images, where $\Omega \subset \mathbb{Z}^2$ is the image domain. Let $f = (f^1, f^2) : \Omega \to [-\varsigma, \varsigma]^2$ be a flow field that maps each pixel $p$ in $I_1$ to $(p + f_p)$ in an augmented domain $\bar{\Omega} \supset \Omega$, which contains $\Omega$ and a large surrounding buffer zone. The buffer zone absorbs pixels that flow out of the visual field. The augmented domain $\bar{\Omega} \subset \mathbb{Z}^2$ is the Minkowski sum of $\Omega$ and $[-\varsigma, \varsigma]^2 \cap \mathbb{Z}^2$, where $\varsigma$ is the maximal empirical displacement magnitude. The maximal empirical displacement magnitude is measured by taking the maximal displacement observed in a training set.

For example, the maximal displacement magnitude on the KITTI training set [16, 29] is 242 pixels. We perform the optimization on 1/3-resolution images, so $\varsigma = 81$ for the KITTI dataset.

Our objective function is

$$E(f) = \sum_{p \in I_1} \rho_D(p, f_p, I_1, I_2) + \lambda \sum_{\{p,q\} \in \mathcal{N}} w_{p,q} \rho_S(f_p - f_q),$$

(2)

where $\mathcal{N} \subset \Omega^2$ is the 4-connected pixel grid. See Figure 1 for illustration. The data term $\rho_D(p, f_p, I_1, I_2)$ penalizes flow fields that connect dissimilar pixels $p$ and $(p + f_p)$. We use truncated normalized cross-correlation [41]:

$$\rho_D(p, f_p, I_1, I_2) = 1 - \max(NCC, 0),$$

(3)

where $NCC$ is the normalized cross-correlation between two patches, one centered at $p$ in $I_1$ and one centered at $(p + f_p)$ in $I_2$, computed in each color channel and averaged. The truncation at zero prevents penalization of negatively correlated patches. If $(p + f_p)$ is in the buffer zone $\bar{\Omega} \setminus \Omega$, the data term is set to a constant penalty $\varsigma$.

Our optimization approach assumes that the regularization term has the following form:

$$\rho_S(f) = \min \left( \rho(f^1) + \rho(f^2), \tau \right),$$

(4)
where $f^1, f^2$ are the two components of vector $f$ and $\rho(\cdot)$ is a penalty function, such as the $L^1$ norm or the Charbonnier penalty. Our formulation and the general solution strategy can accommodate non-convex functions $\rho$, such as the Lorentzian and the generalized Charbonnier penalties. The reduction of message passing complexity from quadratic to linear, described in Section 4.2, applies to such functions as well [9]. The highly efficient min-convolution algorithm described in Section 4.3 will assume that the function $\rho$ is convex. Other linear-time algorithms can be used if this assumption doesn’t hold [9].

Note that the regularization term (4) couples the horizontal and vertical components of the flow. We apply a Laplace weight to attenuate the regularization along color discontinuities:

$$w_{p,q} = \exp\left(-\frac{\|I_1(p) - I_2(q)\|}{\beta}\right).$$

### 4. Optimization

Objective (2) is a discrete Markov random field with a two-dimensional label space [5]. At first glance, global optimization of this model may appear intractable. The number of nodes and the number of labels are both in the tens of thousands. The most sophisticated prior application of discrete optimization to this problem resorted to pruning of the label space to bring the size of the problem under control [30]. We show that the full problem is tractable.

#### 4.1. Message passing algorithm

The label space of the model is $[-\zeta, \zeta]^2 \cap \mathbb{Z}^2$. Let $N = |\Omega|$ be the number of nodes and let $M = (2\zeta + 1)^2$ be the size of the label space. To optimize the model, we use TRW-S, which optimizes the dual of a natural linear programming relaxation of the problem [42, 24]. We choose TRW-S due to its effectiveness in optimizing models with large label spaces [40, 10]. Note that TRW-S optimizes the dual objective and will generally not yield the optimal solution to the primal problem.

For notational simplicity, we omit the weights $w_{p,q}$, although incorporating the weights into the presented method is straightforward. We first write down objective (2) as an equivalent discrete labeling problem. Let $L = [-\zeta, \zeta]^2 \cap \mathbb{Z}^2$ be the candidate flow vectors and $|L| = M = (2\zeta + 1)^2$. We optimize the following objective with respect to the labeling $\theta: \Omega \rightarrow L$:

$$\theta(1) = \sum_{p \in \Omega} \theta_p(l_p) + \sum_{(p,q) \in \mathcal{N}} \theta_{pq}(l_p, l_q),$$

$$\theta_p(s) = \rho_{D}(p, s, I_1, I_2),$$

$$\theta_{pq}(s,t) = \rho_{S}(s-t).$$

$\theta$ are the potentials for the data and pairwise terms. Instead of directly minimizing objective (5), TRW-S maximizes a lower bound that arises from a reparametrization of $\theta$. $\tilde{\theta}$ is said to be a reparametrization of $\theta$ iff $\tilde{\theta}(1) \equiv \theta(1)$. TRW-S constructs reparametrizations of the following form:

$$\tilde{\theta}_p(s) = \theta_p(s) + \sum_{(p,q) \in \mathcal{N}} m_{q\rightarrow p}(s),$$

$$\tilde{\theta}_{pq}(s,t) = \theta_{pq}(s,t) - m_{q\rightarrow p}(s) - m_{p\rightarrow q}(t),$$

where $m_{p\rightarrow q}$ is a message (a vector of size $M$) that pixel $p$ sends to its neighbor $q$ [24]. Given a reparametrization $\tilde{\theta}$, a lower bound for $\theta(1)$ can be obtained by summing the minima of all the potentials:

$$\Phi(\tilde{\theta}) = \sum_{p \in \Omega} \min_s \tilde{\theta}_p(s) + \sum_{(p,q) \in \mathcal{N}} \min_{s,t} \tilde{\theta}_{pq}(s,t).$$

TRW-S maximizes $\Phi(\tilde{\theta})$ by first iterating over all pixels in a given order (e.g., in scanline order or by sweeping a diagonal wavefront from one corner of the image to its antipode). When pixel $p$ is reached, the following message update rule is applied to each $m_{p\rightarrow q}$ for which $q$ has not been visited yet:

$$m_{p\rightarrow q}(t) = \min_s \left( \frac{1}{2} \left( \theta_p(s) + \sum_r m_{r\rightarrow p}(s) \right) - m_{q\rightarrow p}(s) + \theta_{pq}(s,t) \right).$$

The pixels are then visited in reverse order with analogous message update rules. This completes one forward-backward pass, considered to be one iteration. A number of iterations are performed.

Given updated messages $m$, we can compute a solution $\tilde{\theta}$ greedily [24]. We determine the labels sequentially in a given order. Upon reaching pixel $p$, we choose a label assignment $l_p$ that minimizes $\theta_p(l_p) + \sum_{q < p} \theta_{pq}(l_p, l_q) + \sum_{p < q} m_{q\rightarrow p}(l_p)$, where $p < q$ means that $p$ precedes $q$ in the order.

#### 4.2. Complexity reduction

A brute-force implementation of a message update requires $O(M^2)$ operations as there are $M$ elements in each message and computing each element directly requires $O(M)$ operations according to update rule (6). We now show that a message update can be performed using $O(M)$ operations in our model. This builds on the min-convolution acceleration scheme developed by Felzenszwalb and Huttenlocher [14, 15]. A general treatment of the one-dimensional case was presented by Chen and Koltun [9].

We begin by rewriting the message update rule (6) as

$$m_{p\rightarrow q}(t) = \min_s \left( \phi_{pq}(s) + \theta_{pq}(s,t) \right),$$

$$\phi_{pq}(s) = \frac{1}{2} \left( \theta_p(s) + \sum_r m_{r\rightarrow p}(s) \right) - m_{q\rightarrow p}(s).$$
Each $\phi_{pq}(s)$ can be computed using $O(1)$ operations, thus $\phi_{pq}$ can be computed using $O(M)$ operations in total. We now show that, given $\phi_{pq}$, all elements of the message $m_{p,q}$ can be computed in $O(M)$ operations as well. Recall that $\theta_{pq}(s, t) = \rho_S(t - s)$ and that $\rho_S(\cdot)$ has the form given in equation (4). Rearranging terms, we obtain

$$m_{p,q}(t) = \min\left(D_{pq}(t), T_{pq}\right),$$

where

$$D_{pq}(t) = \min_s \left(\phi_{pq}(s) + \rho(t^1 - s^1) + \rho(t^2 - s^2)\right),$$

$$T_{pq} = \min_s \left(\phi_{pq}(s) + \tau\right).$$

$T_{pq}$ can be computed using $O(M)$ operations given $\phi_{pq}$. We now show that $D_{pq}$ can also be computed using $O(M)$ operations in total. Note that $s$ and $t$ are 2D vectors. Abusing notation somewhat, we can rewrite $D_{pq}(t)$ as a two-dimensional min-convolution:

$$D_{pq}(t^1, t^2) = \min_{s^1, s^2} \left(\phi_{pq}(s^1, s^2) + \rho(t^1 - s^1) + \rho(t^2 - s^2)\right)$$

$$= \min_{s^2} \left(\min_{s^1} \left(\phi_{pq}(s^1, s^2) + \rho(t^1 - s^1)\right) + \rho(t^2 - s^2)\right).$$

This can be decomposed into two sets of $O(\sqrt{M})$ one-dimensional min-convolutions:

$$D_{pq}(t^1, t^2) = \min_{s^2} D_{pq|s^2}(t^1) + \rho(t^2 - s^2), \quad (8)$$

$$D_{pq|s^2}(t^1) = \min_{s^1} \phi_{pq}(s^1, s^2) + \rho(t^1 - s^1). \quad (9)$$

For each $s^2$, $D_{pq|s^2}(t^1)$ can be computed for all $t_1$ by a 1D min-convolution. Then, for each $t^1$, $D_{pq}(t^1, t^2)$ can be computed for all $t^2$ by a 1D min-convolution. Each min-convolution can be evaluated in $O(\sqrt{M})$ operations, for a total complexity of $O(M)$.

### 4.3. Further acceleration

A min-convolution has the following general form:

$$h(i) = \min_j g(j) + \rho(i - j). \quad (10)$$

It is well-known that the min-convolution can be computed using $O(n)$ operations, where $n = \sqrt{M} = 2^k + 1$ [9]. However, commonly used algorithms require computing intersections of shifted copies of the function $\rho$. While each intersection can be computed in time $O(1)$, this computation can be numerically intensive for some penalty functions. Since this computation is in the inner loop, it can slow the optimization down. We now review an alternative algorithm that can be used to compute the min-convolution without computing intersections. This algorithm relies on the assumption that $\rho$ is convex, which is otherwise not necessary. Related algorithms are reviewed by Eppstein [13].

The algorithm is based on totally monotone matrix searching [1]. Let $A$ be an $n \times n$ matrix, such that $A(i, j) = g(j) + \rho(i - j)$. Let ind$_A(i)$ be the column index of the minimal element in the $i$th row of $A$. The min-convolution $h$ can be defined as $h(i) = A(i, \text{ind}_A(i))$. The challenge is to evaluate ind$_A$ in time $O(n)$ without explicitly constructing the matrix $A$.

The convexity of $\rho$ implies that $A$ is totally monotone. The totally monotone matrix search algorithm computes ind$_A$ in $O(n)$ operations by divide-and-conquer. The algorithm first constructs an $\frac{n}{2} \times n$ submatrix $B$ by removing every other row of $A$. Then $B$ is reduced to an $\frac{n}{2} \times \frac{n}{2}$ submatrix $C$ by removing columns that do not contain minima of the rows of $B$. The minima ind$_C$ are computed recursively, after which the missing elements of ind$_A$ are filled in. As shown by Aggarwal et al. [1], all steps can be performed in time $O(n)$. Crucially, all steps can be performed without explicitly constructing $A$.

### 5. Implementation

**Parallelization.** To reduce wall-clock time, we implemented a parallelized TRW-S solver. This general-purpose solver along with the rest of our implementation will be made freely available. At each step of TRW-S, a pixel is ready to be processed if all of its predecessors have already been updated during the current iteration. Thus at any time there is a wavefront of pixels that can be processed in parallel. A grid can be swept diagonally. In the first step only one node can be processed, but the size of the wavefront grows rapidly and all nodes on the wavefront can be processed in parallel. This parallelization scheme has previously been explored on special-purpose hardware for stereo matching [12]. We have implemented the scheme on general-purpose processors. Our implementation is evaluated on a workstation with a 6-core Intel i7-4960X CPU. Parallelization with hyper-threading reduces the running time of each iteration of TRW-S from 256 to 39 seconds, a factor of 6.6. Since the size of the wavefront is $\Theta(\varsigma)$ for most of the iteration, increased hardware parallelism is expected to directly translate to reduction in wall-clock time. We also refer the reader to the concurrent work of Shekhovtsov et al. [35], who developed a parallelized energy minimization scheme that may be applicable to our setting.

**Occlusion handling.** Some pixels in $I_1$ may not have corresponding points in $I_2$. The computed forward flow field on these occlusion pixels is likely incorrect. We adopt the common tactic of forward-backward consistency checking: compute the forward flow from $I_1$ to $I_2$ and the backward flow from $I_2$ to $I_1$, and discard inconsistent matches [31]. Given the forward flow field $f$ and the backward flow field
The computation of the 1.3 billion pixel optical flow datasets, MPI Sintel [8] and KITTI 2015 [29].

The presented approach is implemented in Matlab, with a C++ wrapper for the parallelized TRW-S solver. Our Matlab code is less than 50 lines long, not including the general-purpose solver.

The experiments are performed on two challenging optical flow datasets, MPI Sintel [8] and KITTI 2015 [29]. We use a workstation with a 6-core Intel i7-4960X 3.6GHz CPU and 64GB of RAM. The computation of the 1.3 billion values in the unary cost volume takes 51 seconds. We use 3x3 patches for NCC in 1/3-resolution images. Performing 3 iterations of TRW-S using the general optimization framework described in Section 4 takes about 2 minutes on either Sintel or KITTI images, downsampled by a factor of 3, with any penalty function $\rho$. When the penalty function is the $L^1$ norm, we can accelerate the optimization further with the $L^1$ distance transform [15], which reduces the running time to about 30 seconds for 3 iterations of TRW-S. EpicFlow interpolation takes 3 seconds. For each dataset, we train the parameters on 5% of the training set by grid search. We use the same parameters for the ‘final’ and ‘clean’ sequences in the Sintel dataset.

### 6.1. Comparison to prior work

In experiments reported in this section, we use the $L^1$ norm for regularization ($\rho(x) = |x|$) and no truncation ($\tau = \infty$). This decision is motivated by the controlled experiments reported in Section 6.2.

**MPI Sintel.** MPI Sintel is a dataset for large-displacement optical flow [8]. There are two types of sequences in the dataset, clean and final. The clean sequences exhibit a variety of illumination and shading effects including specular reflectance and soft shadows. The final sequences additionally have motion blur, depth of field, and atmospheric effects.

The experimental results are provided in Table 1. We use the 10 metrics reported by Bailer et al. [2], including all, noc, occ, d0-10, and s40+ for both clean and final test sequences. All the errors are measured as endpoint error (EPE), which is the Euclidean distance between the estimated flow vector and the ground truth. Since some error metrics are extremely close for different methods, and because the average EPE is sensitive to outliers (the top methods generally have errors of 20 to 40 pixels on a number of pixels).
of challenging sequences), we highlight every method that achieves within 1% of the lowest reported error as one of the top methods according to that error metric.

Our approach outperforms EpicFlow [32], TF+OFM [23], NNF-Local [11], PH-Flow [46], and Classic+NL [39] on almost all metrics. Our approach ranks 2nd on the key EPE-all metric for both final and clean sequences. Compared to EpicFlow, our approach reduces EPE-all by 6.2% on the final sequences and by 12.5% on the clean sequences.

KITTI 2015. KITTI Optical Flow 2015 is an optical flow dataset that comprises outdoor images of dynamic scenes captured from a car [16, 29]. The car is equipped with a LiDAR sensor and color cameras. Ground-truth flow is obtained by rigid alignment of the static environment and by fitting CAD models to moving objects. Ground-truth correspondences are sparse. The dataset contains 200 training sequences and 200 test sequences. A flow vector is considered an outlier if its endpoint error is 3 pixels or higher. Table 2 lists the most accurate methods on this dataset, along with the classical Horn-Schunck algorithm for reference. Note that SOF [33] was developed concurrently with our work and uses substantially more information at training time, at the cost of generality.

| Method          | all    | noc    | occ    | d0-10  | s40+   | all    | noc    | occ    | d0-10  | s40+   |
|-----------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| FlowFields [2]   | 5.810  | 2.621  | 31.799 | 4.851  | 33.890 | 3.748  | 1.056  | 25.700 | 2.784  | 23.602 |
| FullFlow         | 5.895  | 2.838  | 30.793 | 4.905  | 35.592 | 3.601  | 1.296  | 22.424 | 2.944  | 20.612 |
| DiscreteFlow [30]| 6.077  | 2.937  | 31.685 | 5.106  | 36.339 | 3.567  | 1.108  | 23.626 | 3.398  | 20.906 |
| EpicFlow [32]    | 6.285  | 3.060  | 32.564 | 5.205  | 38.021 | 4.115  | 1.360  | 26.595 | 3.660  | 25.859 |
| TF+OFM [23]      | 6.727  | 3.388  | 33.929 | 5.544  | 39.761 | 4.917  | 1.874  | 29.735 | 3.676  | 31.391 |
| NNF-Local [11]   | 7.249  | 2.973  | 42.088 | 4.896  | 44.866 | 5.386  | 1.397  | 37.896 | 3.770  | 36.342 |
| PH-Flow [46]     | 7.423  | 3.795  | 36.960 | 5.550  | 44.926 | 4.388  | 1.714  | 26.202 | 3.612  | 27.997 |
| Classic+NL [39]  | 9.153  | 4.814  | 44.509 | 7.215  | 60.291 | 7.961  | 3.770  | 42.079 | 6.191  | 57.374 |

Table 2. Accuracy of different methods on the KITTI 2015 test set. This table lists the percentage of outliers on all pixels and on non-occluded pixels.

Qualitative results. In Figure 4, we compare our visual results to EpicFlow and DiscreteFlow on three scenes from MPI Sintel and three scenes from KITTI 2015. On MPI Sintel, our approach performs well on regions with large displacements (we rank first on s40+ in Table 1). This is also reflected in the visual results. See the flapping wings in scene 1 and the flying butterfly in scene 2. In scene 3, all three methods fail but our approach and DiscreteFlow recover more of the flow field than EpicFlow. On KITTI 2015, our approach is visually similar to DiscreteFlow in most street scenes (for example, scene 1). In some cases, our approach is visually more accurate (scene 2), but not on others (outliers on the white line in scene 3). Both our approach and DiscreteFlow are visually more accurate than EpicFlow.

6.2. Controlled experiments

The generality of the presented optimization framework enables a comprehensive evaluation of different data terms and regularization terms. We perform such an evaluation using 5% of the MPI Sintel training set (final pass). In all conditions, we optimize variants of the model presented in Section 3 using the method presented in Sections 4 and 5. We evaluate two data terms: the patch-based truncated NCC term given in equation (3) and the classical pixelwise Horn-Schunck data term given by the squared Euclidean distance in RGB color space. We also evaluate three penalty functions for the regularization term (equation 4): $L^1(p(x) = |x|)$, squared $L^2(p(x) = x^2)$, and Charbonnier $(\rho(x) = \sqrt{x^2 + \varepsilon^2},$ where $\varepsilon = 5$). For each penalty function, we evaluate a truncated regularizer ($\tau$ is determined by grid search) and a non-truncated convex form ($\tau = \infty$). All free parameters are determined by grid search.

The results are shown in Table 3, which provides the average EPE over the images used for the evaluation for each combination of the three factors (data term, penalty function, truncation). The results suggest that the data term is of primary importance: the patch-based truncated NCC term is much more effective than the pixelwise Horn-Schunck data term, irrespective of the regularizer. Note that the non-truncated HS+$L^2$ condition corresponds to global optimiza-
Table 3. Controlled evaluation of the data term, penalty function, and truncation. Lower is better. The patch-based NCC data term is much more effective than the pixelwise HS data term.

| Data Term            | Truncated | Non-truncated |
|----------------------|-----------|---------------|
| NCC+L₁               | 2.710     | 2.710         |
| NCC+Charbonnier      | 2.883     | 2.888         |
| NCC+L²               | 2.896     | 2.976         |
| HS+L₁                | 5.972     | 6.523         |
| HS+Charbonnier       | 5.847     | 6.181         |
| HS+L²                | 6.337     | 6.529         |

Figure 3. Average endpoint error for each tested image, sorted by magnitude. The tested images are sorted independently for each condition: for example, image #40 is not the same for different conditions.

To summarize, the presented optimization approach was designed to support global optimization with very general data and regularization terms. The generality of the presented framework enabled a controlled evaluation of global optimization with different data terms and regularizers. The results indicate that within a global optimization framework the detailed form of the regularizer is less important than the data term, the classical quadratic regularizer yields competitive performance, and the highest accuracy is achieved using the L¹ penalty.

7. Conclusion

We have shown that optimizing a classical Horn-Schunck-type objective globally over full regular grids is sufficient to initialize continuous interpolation and obtain state-of-the-art accuracy on challenging modern optical flow benchmarks. In particular, this demonstrates that state-of-the-art accuracy on large-displacement optical flow estimation can be achieved without externally defined descriptors. The flow objective itself is sufficiently powerful to produce accurate mappings even in the presence of large displacements. We have shown that optimizing the objective globally over the full space of mappings between regular grids is feasible and that the regular structure of the space enables significant optimizations.

Our Matlab implementation is less than 50 lines long, excluding the general-purpose TRW-S solver. We hope that the simplicity of our approach will support further advances. More advanced data terms can easily be integrated into our global optimization framework and are likely to yield even more accurate results. In addition, we believe that there is scope for further improvement in continuous interpolation accuracy, building on the impressive performance of the interpolation scheme of Revaud et al. [32]. The output of the presented global optimization approach can serve as a canonical initialization for benchmarking such continuous interpolation schemes.

References

[1] A. Aggarwal, M. M. Klawe, S. Moran, P. W. Shor, and R. E. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2, 1987. 4
[2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015. 1, 5, 6
[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1), 2011. 1
[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1), 1996. 1
[5] A. Blake, P. Kohli, and C. Rother. *Markov Random Fields for Vision and Image Processing*, MIT Press, 2011. 3
[6] T. Brox, A. Bruhn, N. Papenber, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 1
[7] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, 33(3), 2011. 1
Figure 4. Qualitative comparison on three scenes from the MPI Sintel training set (top) and three scenes from the KITTI 2015 training set (bottom). Two rows of images per scene. (a) shows the average of two input images. (b-d) show the flow fields, and (f-h) show the corresponding EPE maps, truncated at 10 pixels. The arrangement of images is the same for each scene.
