Motion Detail Preserving Optical Flow Estimation

Li Xu, *Member, IEEE,* Jiaya Jia, *Senior Member, IEEE,* Yasuyuki Matsushita, *Senior Member, IEEE*

Abstract—A common problem of optical flow estimation in the multi-scale variational framework is that fine motion structures cannot always be correctly estimated, especially for regions with significant and abrupt displacement variation. A novel *extended coarse-to-fine* (EC2F) refinement framework is introduced in this paper to address this issue, which reduces the reliance of flow estimates on their initial values propagated from the coarse level and enables recovering many motion details in each scale. The contribution of this paper also includes adaptation of the objective function to handle outliers and development of a new optimization procedure. The effectiveness of our algorithm is demonstrated using the Middlebury optical flow benchmark and by experiments on challenging examples that involve large-displacement motion.

1 Introduction

The variational framework [18], together with coarse-to-fine refinement [2], [23], is widely used in optical flow estimation [10], [12]. In the Middlebury optical flow evaluation website [3], [4], almost all top-ranked methods adopt this scheme.

However, the conventional coarse-to-fine warping framework has a fundamental limitation in handling motion details. Brox *et al.* [9], in computing large-displacement optical flow, pointed out that if flow structure is smaller than its displacement, the latter may not be well estimated. In this paper, we show that this issue can be even more serious, as it also applies to small-displacement motion. Taking Fig. 1 as an example, due to the camera motion, the foreground toy deer has motion significantly different from that of the background (average displacements $d = -2$ and $d = 21$ respectively). This example is very challenging for the coarse-to-fine variational optical flow estimation.

As shown in Fig. 1(e), in a coarse level, the narrow neck does not exist and only the significant background motion is estimated. This makes the actual motion of the foreground pixels in the finer scale (Fig. 1(f)) drastically different from the initial estimate from the background, violating the linearization assumption and accordingly leading to highly unstable motion estimation. The final flow result shown in (c) includes considerable errors. This example discloses one problem of the general coarse-to-fine variational model – that is, the inclination to diminish small motion structures when *spatially significant and abrupt change* of the displacement exists.

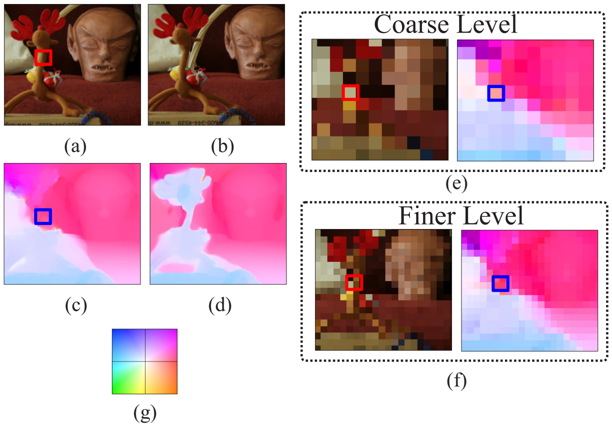


Fig. 1. Motion detail preserving problem. (a)-(b) Two input patches. (c) Flow estimate using the coarse-to-fine variational setting. (d) Our flow estimate. (e)-(f) Two consecutive levels in the pyramid. Flow fields are visualized using the color code in (g). The input data is from [29].

We address this problem in this paper and propose a unified framework for high-quality flow estimation in *both large and small displacement settings*. Central to our method is a novel selection scheme to compute extended initial flow vectors in each image level. This makes the following optimization not completely rely on the result at the previous scale, and is thus capable of refining the estimation correctly in a top-down fashion. Our flow result shown in Fig. 1(d) contains small structures. More examples are included in Section 5.

This paper also contributes in the following ways. First, we use robust sparse features, together with patch matching, to produce extended flow initialization, which helps enforce the linearization condition in the variational setting. Second, in the flow estimation model, we propose the selective combination of color and gradient constraints in defining the data term, robust to outliers. Third, we propose a fast variable-splitting-based optimization method to refine flow maps. It is highly parallel.

Finally, we employ the Mean Field approximation to enable solving the objective function, which involves both

- *L. Xu and J. Jia are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. E-mail: xuli@cse.cuhk.edu.hk, leojia@cse.cuhk.edu.hk*
- *Y. Matsushita is with Microsoft Research Asia, Beijing, China. Email: yasumat@microsoft.com*

discrete and continuous variables, commonly regarded as challenging to handle. Extensive experiments visually and quantitatively validate the performance of our approach for both large- and small-displacement motion.

This manuscript extends its conference version [42] with the following major differences. (1) We provide more discussion and derivation details of the data term and its Mean-Field approximation. (2) The extended coarse-to-fine scheme is further generalized in order to handle non-rigid motion with dense patch matching. (3) The occlusion is progressively handled in each scale. (4) We have experimented with more challenging examples in this paper.

The rest of the paper is organized as follows. Section 2 reviews related work. In Section 3, we introduce the flow energy. An extended coarse-to-fine framework together with the efficient solver is presented in Section 4. Section 5 shows results for both large- and small-displacement optical flow. We conclude this paper in Section 6.

2 Related Work

Following the variational model of Horn and Schunck [18], modern optical flow estimation is usually posed as an energy minimization problem, with the energy function containing a data term and a smoothness term.

One important improvement over the original variational model is the introduction of robust statistics for both the energy terms. Black and Anandan [7] replaced the quadratic penalty functions in [18] by non-convex robust ones to reject outliers. ℓ_1-norm, or its variation (*e.g.*, the Charbonnier function), is also commonly used [10], [12], [37], [43]. Learning-based methods construct the distribution from empirical data. In particular, Roth and Black [26] learned spatial smoothness using Field-of-Experts, and combined it with a Charbonnier data term; Sun *et al.* [34] proposed a learning framework to fit distributions using Gaussian Scale Mixture (GSM). In [33], Sun *et al.* empirically demonstrated that the simple Charbonnier function (ℓ_1-norm) actually outperforms other highly non-convex robust functions due to its convex property.

Efforts also have been put into improving the optical flow constraints. Haussecker and Fleet [17] proposed a physical constraint to model brightness change. Wedel *et al.* [37] proposed a structure-texture decomposition method to reduce the discrepancy between two frames caused by illumination change. Lempitsky *et al.* [20] computed the matching cost only using high frequency components. Pre-filtering on the input images was suggested in [34] and [25] to handle illumination variation. These models are flexible, but at the same time require pre-processing or advanced optimization to solve complex objective functions.

In [10], Brox *et al.* introduced the gradient constancy constraint to complement the brightness constraint. In [12], separate penalties are imposed on the brightness and gradient constraints. Zimmer *et al.* [45] further employed the normalized brightness and gradient constraints. We will show later that the way to combine the brightness and gradient terms can be improved by a selection model.

To preserve motion discontinuities, anisotropic- [39], steerable- [34], [44], [45], and adaptive-smoothness [1], [36] terms were studied. Segmentation information was incorporated to regularize flow estimates in [19], [24], [41]. Recently, the non-local smoothing strategy [33], [38] demonstrated the potential to handle displacement discontinuities and occlusion, which is tightly linked to explicit refinement of the flow field using image filtering [28], [40].

Almost all the above methods rely on coarse-to-fine warping to deal with motion larger than one pixel [2], [6]. As discussed in Section 1, this strategy could fail to recover small-scale structures. Handling incorrect initialization by adapting windows for stereo matching [31] is a solution. It however assumes at least that the nearby disparities are correctly initialized, which may not be true for small-scale structures that are totally eliminated in the coarse level.

Using discrete optimization, Lempitsky *et al.* [20] proposed fusing flow proposals obtained from different flow estimation methods with various parameter settings. It is effective at finding the optimal values among the given proposals. But the sufficiency and optimality of the proposals cannot be controlled. Because the proposals are still generated by the conventional coarse-to-fine warping, it is possible that none of the proposals preserve small-scale motion structure. In comparison, our method computes high confidence flow candidates in each level, and thus is not entirely dependent on flow obtained from the coarse scale.

Related work also includes recent large-displacement optical flow estimation [9], [11], where region-based descriptor matching was introduced. It is an effective method except for occasional vulnerability to matching outliers, due to the data term. As discussed in [11], descriptor matching could decrease the performance in small-motion regions.

By extending the numerical scheme of [43] and by searching possible values to minimize the data energy [32], large displacement optical flow estimation can be achieved. As the smoothness prior is not enforced, the results can possibly be noisy and lack sub-pixel accuracy. In this paper, an extended coarse-to-fine method is proposed, which can significantly improve both the large- and small-displacement optical flow estimation in a unified framework.

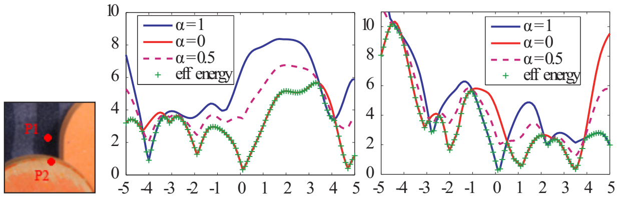
3 Optical Flow Model

We introduce in this section our objective function. We base our data penalty function on the ℓ_1 norm to reject outliers and use the Total Variation (TV) for regularization.

3.1 Robust Data Function

As the color constancy constraint is often violated when illumination or exposure changes, combining the gradient constraint was adopted [10], [12]. Denoting by $\mathbf{u} = (u, v)^T$ the flow field that represents the displacement between frames $\mathbf{I_1}$ and $\mathbf{I_2}$, one choice of the data term for flow estimation is

$$\begin{aligned} E_D(\mathbf{u}) = \sum_{\mathbf{x}} \frac{1}{2}\|\mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \mathbf{I_1}(\mathbf{x})\| + \\ \frac{1}{2}\tau\|\nabla\mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \nabla\mathbf{I_1}(\mathbf{x})\|, \end{aligned} \tag{1}$$



(a) (b) Data costs for P1 (c) Data costs for P2

Fig. 2. Data cost distributions for two points. (a) A patch in the "RubberWhale" example, where two points P1 and P2 are highlighted. (b)-(c) Plots of different data costs (heights of the points) for P1 and P2. The ground truth displacement is moved to 0 in the horizontal axis for ease of illustration.

where $\mathbf{x} \in \mathbb{Z}^2$ indexes the 2D coordinates, τ is a weight balancing the two matching costs. ∇ is the discrete approximation of the gradient operator. This function, due to the addition of two terms, is less accurate in modeling pixel correspondence than only using one out of the two terms.

Fig. 2 shows an example where the patch in (a) contains two points P1 and P2. Their data cost distributions with respect to different displacement values are plotted in (b) and (c) respectively (ground truth displacements are shifted to 0). It is noticeable that the color constraint (blue curve in (b)) does not produce the minimum energy near the ground truth value because the color constancy is violated given point P1 moving out of the shadow. Adding the color and gradient terms using Eq. (1) also results in an undesirable distribution (dashed magenta curve) as the cost at the ground truth point is not even a local minimum. Similarly, in Fig. 2(c), only the color constancy holds because point P2 undergoes rotational motion, which alters image gradients. It is not ideal as well to add the two constraints in the data function definition.

The above analysis indicates that a good model should *only* use the more fitting constraint, but not both of them. We accordingly define a binary weight map $\alpha(\mathbf{x}) : \mathbb{Z}^2 \mapsto \{0, 1\}$ to switch between the two terms. The new data function is expressed as

$$E_D(\mathbf{u}, \alpha) = \sum_{\mathbf{x}} \alpha(\mathbf{x}) \|\mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \mathbf{I_1}(\mathbf{x})\| + (1-\alpha(\mathbf{x}))\tau \|\nabla \mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \nabla \mathbf{I_1}(\mathbf{x})\|. \quad (2)$$

When $\alpha(\mathbf{x}) = 1$, the gradient constraint is favored. Otherwise, we select color constancy. Our empirical investigation provided in Section 5 shows that this model can produce higher quality results than various alternatives.

3.2 Edge-Preserving Regularization

The regularization term for optical flow estimation is generally designed to be edge preserving [34], [36], [45]. We define our smoothness term as

$$E_S(\mathbf{u}) = \sum_{\mathbf{x}} \omega(\mathbf{x}) \|\nabla \mathbf{u}(\mathbf{x})\|, \quad (3)$$

where $\|\nabla \mathbf{u}(\mathbf{x})\|$ is the common TV regularizer. $\omega(\mathbf{x})$ is the simple structure adaptive map that maintains motion discontinuity [1], [36]:

$$\omega(\mathbf{x}) = \exp(-\|\nabla \mathbf{I_1}\|^{\kappa}), \quad (4)$$

where we set $\kappa = 0.8$ in our experiments. For simplicity, we use the brightness derivatives (2 channels) to compute $\|\nabla \mathbf{I_1}\|^{\kappa}$. The final objective function is defined as

$$E(\mathbf{u}, \alpha) = E_D(\mathbf{u}, \alpha) + \lambda E_S(\mathbf{u}), \quad (5)$$

where λ is the regularization weight.

3.3 Mean Field Approximation

Minimizing Eq. (5) involves simultaneously computing two fields: continuous $\mathbf{u}$ and binary α, which is computationally challenging. We employ the Mean Field (MF) approximation [14] to simplify the problem by first canceling out the binary process by integration over α. The probability of a particular state of the system is given by

$$P(\mathbf{u}, \alpha) = \frac{1}{Z} e^{-\beta E(\mathbf{u}, \alpha)}, \quad (6)$$

where β is the inverse temperature and Z is the partition function, defined as

$$Z = \sum_{\{\mathbf{u}\}} \sum_{\{\alpha=0,1\}} e^{-\beta E(\mathbf{u}, \alpha)}. \quad (7)$$

We then compute the sum over all possible αs with the saddle point approximation (see Appendix in the supplementary file for the derivation), yielding:

$$E^{eff}(\mathbf{u}) = \lambda E_S(\mathbf{u}) - \sum_{\mathbf{x}} \frac{1}{\beta} \ln(e^{-\beta \mathcal{D}_{\mathbf{I}}(\mathbf{u}, \mathbf{x})} + e^{-\beta \mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u}, \mathbf{x})}), \quad (8)$$

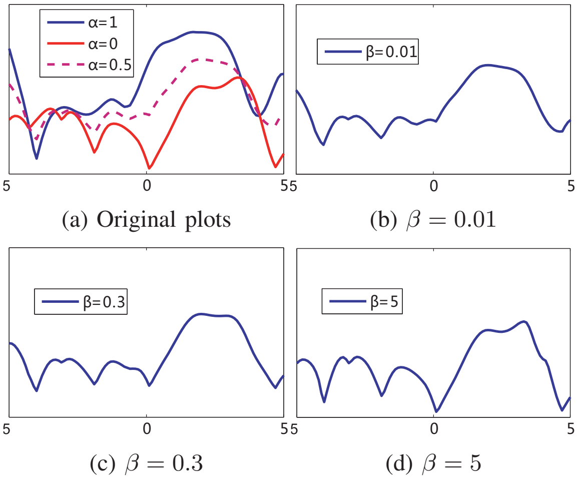
where $\mathcal{D}_{\mathbf{I}}(\mathbf{u}, \mathbf{x}) = \|\mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \mathbf{I_1}(\mathbf{x})\|$ and $\mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u}, \mathbf{x}) = \tau \|\nabla \mathbf{I_2}(\mathbf{x}+\mathbf{u}) - \nabla \mathbf{I_1}(\mathbf{x})\|$. It indicates that the flow estimate by minimizing Eq. (8) is actually the Mean Field (MF) approximation of minimizing Eq. (5). The effective energy is therefore written as

$$E^{eff}(\mathbf{u}) = E_D^{eff}(\mathbf{u}) + \lambda E_S(\mathbf{u}), \quad (9)$$

where the effective data function is

$$E_D^{eff}(\mathbf{u}) = \sum_{\mathbf{x}} -\frac{1}{\beta} \ln(e^{-\beta \mathcal{D}_{\mathbf{I}}(\mathbf{u}, \mathbf{x})} + e^{-\beta \mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u}, \mathbf{x})}). \quad (10)$$

The optimality of Eq. (9) does not depend on the estimate of α. Eq. (10) defines a robust function and β plays a key role in shaping it. When $\beta \longrightarrow 0$, Eq. (10) acts as the average of the two data costs in Eq. (1), while $\beta \longrightarrow \infty$ leads to the lower envelope of the two costs in Eq. (10). We show in Fig. 3 several examples on how the effective function is affected by varying β. Fig. 3(a) contains plots with different α values, same as the ones shown in (b). In (b)-(d), we show distributions of the effective data costs by varying β. Note that a small β makes the distribution (plotted in (b)) close to the original one with $\alpha = 0.5$ (shown in (a)) while a relatively large β (shown in (d)) yields the distribution approaching the lower envelope of the costs with $\alpha = 0$ and $\alpha = 1$, which is what we need for accurate flow estimation. The effective data costs (with $\beta = 5$) are also plotted in



(a) Original plots

(b) $\beta = 0.01$

(c) $\beta = 0.3$

(d) $\beta = 5$

Fig. 3. Effective data cost distributions with different β values. (a) shows data costs with different α values. (b)-(d) show effective data costs by varying β.

```
Input: a pair of images for optical flow estimation
1. Construct pyramids for both of the images and set the initial
   level $l = 0$ and $\mathbf{u}_l = 0$ for all pixels.
2. Propagate $\mathbf{u}_l$ to level $l + 1$.
3. Extended Flow Initialization (Section 4.1)
   3.1. Detect and match SIFT features in level $l + 1$.
   3.2. Perform patch matching in level $l + 1$.
   3.3. Generate multiple flow vectors as candidates.
   3.4. Optimize flow using QPBO (Eq. (9)).
4. Continuous Flow Optimization (Section 4.2)
   4.1. Compute the $\bar{\alpha}$ map (Eq. (12)).
   4.2. Solve the TV/$\ell_1$ energy function in Eq. (15).
5. Occlusion-aware Refinement (Section 4.3)
6. If $l \neq n - 1$ where $n$ is the total number of levels, $l = l + 1$
   and go to Step 2.
Output: the optical flow field
```

TABLE 1
Method Overview

Fig. 2(b) and (c) using the green crossed curves. They are coincident with the smallest-value curves. Our method always keeps $\beta \geq 1$ empirically.

We optimize Eq. (9) using an iteratively reweighted optimization strategy. The difficulty of minimizing Eq. (9) stems from the non-convex data function. Taking the partial derivative with respect to the variable u yields

$$\partial_u E_D^{eff}(\mathbf{u}) = \sum_{\mathbf{x}} \bar{\alpha}(\mathbf{x}) \partial_u \mathcal{D}_{\mathbf{I}} + (1 - \bar{\alpha}(\mathbf{x})) \partial_u \mathcal{D}_{\nabla \mathbf{I}}, \quad (11)$$

where $\bar{\alpha}(\mathbf{x})$ is the flow-dependent weight, written as

$$\bar{\alpha}(\mathbf{x}) = \frac{1}{1 + e^{\beta(\mathcal{D}_{\mathbf{I}}(\mathbf{u},\mathbf{x}) - \mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u},\mathbf{x}))}}. \quad (12)$$

It indicates that the energy can be minimized by iteratively updating $\bar{\alpha}$ in the outer loop and by solving for $\mathbf{u}$ with the computed weights afterwards. In addition, although Eq. (9) is non-convex and difficult to solve using continuous optimization, there is no obstacle to apply discrete optimization if candidate labels can be obtained. We propose a robust algorithm, described in the next section, to estimate $\mathbf{u}$.

The solver can also be interpreted from another perspective. Note that $\bar{\alpha}(\mathbf{x})$ is actually the MF-approximation of $\alpha(\mathbf{x})$ (see Appendix in the supplementary file), and thus can be updated once $\mathbf{u}$ is obtained. It has an effect similar to $\alpha(\mathbf{x})$ (given in Eq. (2)) in constraint selection.

4 Optimization Framework

Traditional optical flow estimation, ascribed to the use of the variational setting, relies excessively on the coarse-to-fine refinement. As discussed in Section 1, this process could fail to recover ubiquitous fine motion details given the possible large discrepancy between the initial flow and the ground truth displacements in each image level.

In this section, based on E^{eff} and $\bar{\alpha}$, we propose an iterative method to optimize Eq. (5). Specifically, because $E_D^{eff}(\mathbf{u})$ is independent of $\bar{\alpha}$, we first infer multiple high-confidence flow candidates and apply discrete optimization to select the optimal ones. With this result, $\bar{\alpha}$ in Eq. (12) can then be quickly estimated. We finally improve the subpixel accuracy of flow with the estimated $\bar{\alpha}$ using continuous optimization. This procedure is found to be surprisingly effective in dampening estimation errors.

Our overall algorithm is sketched in Table 1 based on iteratively processing images in a top-down fashion. The steps are detailed further below.

4.1 Extended Flow Initialization

We address the general flow initialization problem in each image scale by finding *multiple* extended displacements (denoted as $\{\mathbf{u}_0^v, ..., \mathbf{u}_n^v\}$) through sparse feature matching and dense patch matching to improve estimation in $\mathbf{u}^c$, which is the flow field computed in the immediately coarser level. The following steps are adopted to obtain the extended displacements.

4.1.1 SIFT feature detection

SIFT feature detection and matching [22] can efficiently capture large motion for objects undergoing translational and rotational motion. Instead of computing a dense descriptor field as in scene matching [21], we only employ sparse matching of discriminative points, which avoids introducing many ambiguous correspondences and outliers. One example is shown in Fig. 4(b). Note that some matches could still be wrong. But this is not a problem as we will eventually employ discrete optimization to only select the most credible candidates.

4.1.2 Selection

The computed displacement vectors by feature matching are denoted as $\{\mathbf{s}_0, ..., \mathbf{s}_n\}$, as shown in Fig. 4(b). They are new potential flow candidates except those that already exist in the initial flow field $\mathbf{u}^c$ (shown in Fig. 4(c)). To robustly screen out the duplicated vectors, we compute the Euclidean distance between each $\mathbf{s}_i$ and all $\mathbf{u}_j^c$s where pixel j is within a 5×5 window centered at the reference feature $\mathbf{s}_i$. If all results are greater than 1 (pixel), we regard $\mathbf{s}_i$ as a new flow candidate. We repeat this process for all is, and denote the m remaining candidate vectors as $\{\mathbf{s}_{k_0}, ..., \mathbf{s}_{k_{m-1}}\}$, as shown in Fig. 4(d).

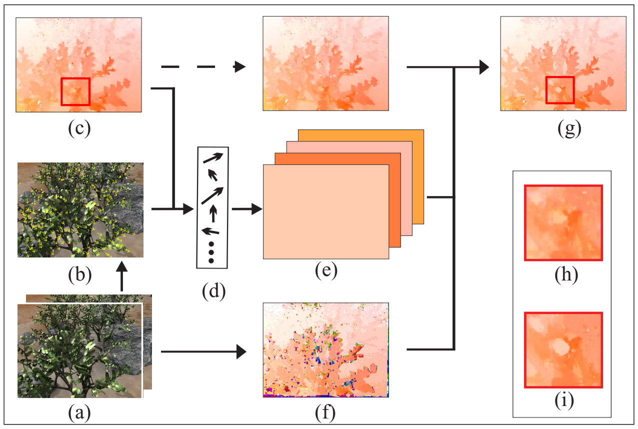


Fig. 4. Extended flow initialization. (a) Two input frames. (b) One of the images overlaid with the computed feature motion vectors $\mathbf{s}_i$. (c) Flow field $\mathbf{u}^c$ propagated from the coarse level. (d) New displacements $\{\mathbf{s}_{k_0}, ..., \mathbf{s}_{k_{m-1}}\}$ computed using (b) and (c). (e) New displacement maps. Each $\mathbf{u}_i^s$ is expanded from $\mathbf{s}_{k_i}$ and therefore is a constant-value map. (f) Dense nearest-neighbor patch matching field $\mathbf{u}^n$. (g) Optimized flow map $\mathbf{u}^0$ with respect to all candidates in the current image scale. (h)-(i) show close-ups of (c) and (g).

This strategy significantly reduces the system dependence on the coarse-scale flow estimation. It is notable as well that feature matching initially produces many vectors distributed in the whole image, as shown in Fig. 4(a); but they reduce to less than 15 candidates after local comparison with $\mathbf{u}^c$ in the given example. Only the most distinct flow vectors are retained.

4.1.3 Expansion

The m remaining vectors $\{\mathbf{s}_{k_0}, ..., \mathbf{s}_{k_{m-1}}\}$ represent possible missing motion in the present flow field $\mathbf{u}^c$. To determine whether or not they are better estimates to replace the original ones in $\mathbf{u}^c$, we expand each displacement vector $\mathbf{s}_{k_i}$ to a constant-value flow field $\mathbf{u}_i^s$ for further fusion. The fields are shown in Fig. 4(e).

4.1.4 Patch matching

SIFT Feature matching, albeit very effective, sometimes still misses motion vectors. It is because small textureless objects may not have distinct features, making their detection problematic. Another main reason is that to let SIFT descriptors gather enough information for 128-dimension feature vector formation, the patches on which they operate should at least contain 16 × 16 samples as suggested. The size could be too large for non-rigid motion as edge statistics may change a lot for successive two frames.

We resort to dense nearest-neighbor patch matching for amelioration. The patches we use can be as small as 5 × 5. They are more flexible to describe motion of small textureless regions, as shown in Fig. 4(f). Specifically, we compute the matching field $\mathbf{u}^n$ by minimizing energy

$$E(\mathbf{u}^n, \mathbf{x}) = \sum_{\mathbf{y}\in N(\mathbf{x})} \sum_k \|\mathrm{I}_2^k(\mathbf{y}+\mathbf{u}^n(\mathbf{x})) - \mathrm{I}_1^k(\mathbf{y})\|^2, \quad (13)$$

where $\mathrm{I}^k \in \{\mathrm{I}^r, \mathrm{I}^g, \mathrm{I}^b, \partial_x \mathrm{I}, \partial_y \mathrm{I}\}$, denoting a total of 5 color and gradient channels. $N(\mathbf{x})$ is a 5×5 window centered at $\mathbf{x}$. Although noise is generated by this method, it can be quickly rejected in the following optimization step with the collection of a set of flow candidates for each pixel.

The energy (13) was employed in [13] as well. But linearization was performed eventually in [13], confining only local refinement. In comparison, we do not impose any smoothness constraint at this stage. So estimates for very large displacement can be obtained.

4.1.5 Matching field fusion

The $m+1$ new motion fields $\{\mathbf{u}_0^s, ..., \mathbf{u}_{m-1}^s, \mathbf{u}^n\}$, together with the original $\mathbf{u}^c$, comprise several motion candidates for each pixel in the present image scale. Selection of the optimal flow among the $m+2$ candidates for each pixel is a labeling problem, with the objective function in Eq. (9). It can be solved by discrete optimization efficiently because on the one hand the number of candidates is small, thanks to the carefully designed selection process; on the other hand, Eq. (9) does not involve α, simplifying computation.

We adopt the Quadratic Pseudo-Boolean Optimization (QPBO) [27] to solve this problem. The fusion move step [20] is used to repeatedly fuse the candidates until each gets visited twice. Also, to suppress the checker-board-like artifacts commonly produced near motion boundaries in discrete optimization, we employ the anisotropic representation of the TV regularizer $\|\nabla \mathbf{u}\| = \|\nabla u\|_1 + \|\nabla v\|_1$ with 8-neighbor discretization [15]. This method turns the checker-board-like boundaries to octagons, a better approximation of the original smooth boundaries. The output is the flow map denoted as $\mathbf{u}^0$. One result is shown in Fig. 4(g), which contains better recovered motion structure compared to the field $\mathbf{u}^c$ in (c). Close-ups are shown in (h) and (i).

Note that an alternative is to directly discretize the original 2D solution space and fuse all candidate flows. It however may suffer from expensive and possibly unstable computation because hundreds of labels can be produced simultaneously in the original resolution.

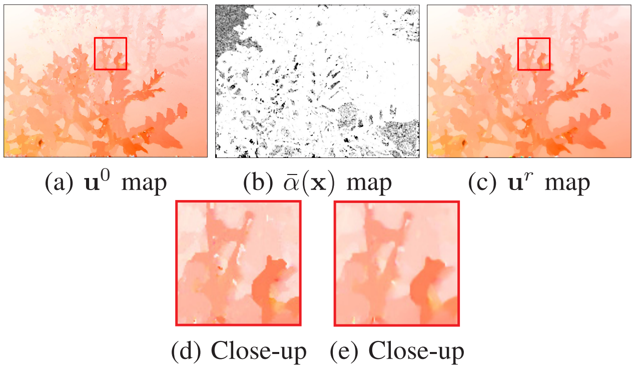


Fig. 5. Continuous optimization. Errors are further reduced in this step. (d) and (e) show close-ups of (a) and (c).

4.2 Continuous Flow Optimization

We now refine flow $\mathbf{u}^0$ through continuous optimization, by iteratively updating $\bar{\alpha}$ in Eq. (12) and $\mathbf{u}$. The initial flow field is taken into Eq. (12) to estimate $\bar{\alpha}$, as shown in Fig. 5(b). Considering that Eq. (10) is highly non-convex, we then take $\bar{\alpha}$ back to Eq. (5) for optimization in the variational model.

As color images are used, we still denote by $\mathrm{I}^k \in \{\mathrm{I}^r, \mathrm{I}^g, \mathrm{I}^b, \partial_x \mathrm{I}, \partial_y \mathrm{I}\}$ the set of channels included in the data term and use $\alpha^k \in \{\bar{\alpha}, \bar{\alpha}, \bar{\alpha}, (1-\bar{\alpha})\tau, (1-\bar{\alpha})\tau\}$ to represent the corresponding weights. The energy in Eq. (5) is thus written as

$$E(\mathbf{u}) = \sum_{\mathbf{x}} \sum_{k} \alpha^k(\mathbf{x}) \|\mathrm{I}_2^{\mathrm{k}}(\mathbf{x}+\mathbf{u}) - \mathrm{I}_1^{\mathrm{k}}(\mathbf{x})\| + \lambda(\mathbf{x}) \|\nabla \mathbf{u}(\mathbf{x})\|, \tag{14}$$

where $\lambda(\mathbf{x}) := \lambda \omega(\mathbf{x})$. With the initial flow $\mathbf{u}^0$ estimated in the previous step, we solve for the increments $\mathbf{du} = (du, dv)^{\mathrm{T}}$ by minimizing Eq. (15). The final flow vector is $\mathbf{u} = \mathbf{u}^0 + \mathbf{du}$. By convention, the Taylor expansion of Eq. (15) at point $\mathbf{x} + \mathbf{u}^0$ yields

$$\begin{aligned} E(\mathbf{u}) \;\; = \;\; & \sum_{\mathbf{x}} \sum_{k} \alpha^k(\mathbf{x}) \|\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}}\| + \\ & \lambda(\mathbf{x}) \|\nabla(\mathbf{u}^0 + \mathbf{du})(\mathbf{x})\|, \end{aligned} \tag{15}$$

given small $\mathbf{du}$. In Eq. (15),

$$\begin{aligned} \mathrm{I}_{\mathrm{x}} &= \partial_x \mathrm{I}_2(\mathbf{x} + \mathbf{u}^0), \\ \mathrm{I}_{\mathrm{y}} &= \partial_y \mathrm{I}_2(\mathbf{x} + \mathbf{u}^0), \\ \mathrm{I}_{\mathrm{t}} &= \mathrm{I}_2(\mathbf{x} + \mathbf{u}^0) - \mathrm{I}_1(\mathbf{x}). \end{aligned}$$

To preserve motion discontinuity, we employ the rotational invariant isotropic form of the TV regularizer, written as

$$\|\nabla \mathbf{u}\| = \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_x v)^2 + (\partial_y v)^2}. \tag{16}$$

Our Solver We propose decomposing the optimization into three simpler problems, each of which can have the globally optimal solution. The key technique is a variable-splitting method [35] with auxiliary variables p and $\mathbf{w}$, representing the substituted data cost and flow derivatives respectively, to move a few terms out of the non-differentiable ℓ_1-norm expression. This scheme is found to be efficient and is crucial to produce high quality results.

The derivatives of each flow vector comprise four elements, *i.e.*,

$$\nabla \mathbf{du} = (\partial_x du, \partial_y du, \partial_x dv, \partial_y dv)^{\mathrm{T}}.$$

For each element, we introduce a corresponding auxiliary variable. The set of the variables is denoted as

$$\mathbf{w} = (w^{du_x}, w^{du_y}, w^{dv_x}, w^{dv_y})^{\mathrm{T}}.$$

Then Eq. (15) is transformed to

$$\begin{aligned} & \sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta} \|\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}} - \mathrm{p}^k\|^2 + \alpha^k \|\mathrm{p}^k\| + \\ & \qquad \frac{1}{2\theta} \|\nabla \mathbf{du} - \mathbf{w}\|^2 + \lambda \|\nabla \mathbf{u}^0 + \mathbf{w}\|. \end{aligned} \tag{17}$$

In this function, $\frac{1}{2\eta} \|\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}} - \mathrm{p}^k\|^2 + \alpha^k \|\mathrm{p}^k\|$ encourages p^k to approach $\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}}$, and $\frac{1}{2\theta} \|\nabla \mathbf{du} - \mathbf{w}\|^2 + \lambda \|\nabla \mathbf{u}^0 + \mathbf{w}\|_2$ makes $\mathbf{w}$ similar to $\nabla \mathbf{du}$. It can be observed as well that Eq. (17) approaches Eq. (15) when $\theta \rightarrow 0$ and $\eta \rightarrow 0$. Our algorithm proceeds with the following iterations with initial $\mathbf{u} := \mathbf{u}^0$.

```
Input: images $\mathrm{I}^k$, initial flow field $\mathbf{u}^0$, weights $\alpha^k$.
Perform linearization at $\mathbf{u}^0$
$\eta \leftarrow \eta_0$
repeat
    Compute $\mathrm{p}^k$ using Eq. (19)
    $\theta \leftarrow \theta_0$
    repeat
        Compute $\mathbf{w}$ using Eq. (21).
        Compute $\mathbf{du}$ by solving Eq. (22).
        $\theta \leftarrow \theta/3$
    until $\theta < \theta_{\min}$
    $\eta \leftarrow \eta/3$
until $\eta < \eta_{\min}$
$\mathbf{u^r} = \mathbf{u}^0 + \mathbf{du}$
Output: refined flow field $\mathbf{u}^r$.
```

TABLE 2
Algorithm for continuous flow optimization

1. Fix $\mathbf{u}$ to estimate p. The simplified objective function is

$$\min \sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta} \|\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}} - \mathrm{p}^k\|^2 + \alpha^k \|\mathrm{p}^k\|. \tag{18}$$

Single variable optimization can be used in this step. The optimal solution is given by the shrinkage formula [16]

$$\mathrm{p}^k = \mathrm{sign}(\mathrm{o}^k) \max(|\mathrm{o}^k| - \eta \alpha^k, 0), \tag{19}$$

where $\mathrm{o}^k := \mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}}$ is the flow constraint.

2. Fix $\mathbf{u}$ to estimate $\mathbf{w}$. The function reduces to

$$\min \sum_{\mathbf{x}} \frac{1}{2\theta} \|\nabla \mathbf{du} - \mathbf{w}\|^2 + \lambda(\mathbf{x}) \|\nabla \mathbf{u}^0 + \mathbf{w}\|_2. \tag{20}$$

Similarly, the following solution can be obtained by the shrinkage formula

$$w^{du_x} = \max(\|\nabla \mathbf{u}\|_2 - \theta\lambda, 0) \frac{\partial_x u}{\|\nabla \mathbf{u}\|_2} - \partial_x u^0, \tag{21}$$

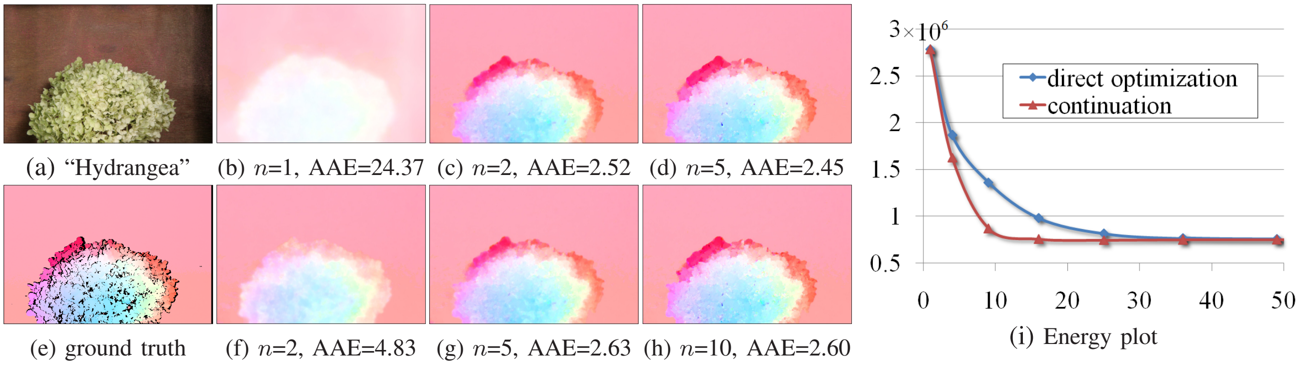
where $\mathbf{u} = \mathbf{u}^0 + \mathbf{du}$. Solutions for w^{du_y}, w^{dv_x}, and w^{dv_y} can similarly be derived. The computation in this step is also quick and is highly parallel by nature.

3. Fix $\mathbf{w}, \mathrm{p}$ and solve for $\mathbf{u}$. The objective function is

$$\min \sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta} \|\mathrm{I}_{\mathrm{x}}^{\mathrm{k}} du + \mathrm{I}_{\mathrm{y}}^{\mathrm{k}} dv + \mathrm{I}_{\mathrm{t}}^{\mathrm{k}} - \mathrm{p}^k\|^2 + \frac{1}{2\theta} \|\nabla \mathbf{du} - \mathbf{w}\|^2. \tag{22}$$

It is quadratic and the corresponding Euler-Lagrange equations of Eq. (22) are linear w.r.t. du and dv. A globally optimal solution can be obtained by solving the linear system in this step.

Our method iterates among optimizing (19), (21), and (22) until convergence. Note that cost function decomposition with auxiliary variables was used in [38], [39], [43] for flow estimation. Their steps use the primal-dual solvers. In comparison, our scheme consists of a set of simpler subproblems, each with guaranteed global optimality. It thus


(a) “Hydrangea” (b) n=1, AAE=24.37 (c) n=2, AAE=2.52 (d) n=5, AAE=2.45

(e) ground truth (f) n=2, AAE=4.83 (g) n=5, AAE=2.63 (h) n=10, AAE=2.60

(i) Energy plot

Fig. 6. Continuation scheme. (b)-(d) show our results obtained using the algorithm in Table 2 with the continuation scheme. n is set to 1, 2, and, 5 respectively. The error is already very small when $n = 2$. (f)-(h) show results with fixed $\eta = 0.1$ and $\theta = 0.01$ in all iterations. AAE stands for “average angular error” [3]. (i) Energy decreasing w.r.t. the number of iterations with and without continuation.

differs from previous methods in the way of formulating the problem and of proposing the solver to each sub-problem.

In practice, θ and η are critical parameters that should be small. It was found that fixing them to constants typically results in slow convergence. We thus adopt the continuation scheme [16] for speedup, which initially sets θ and η to large values to allow *warm-starting*, and then decreases them in iterations toward the desired convergence. Our algorithm is sketched in Table 2, where $\eta_{\min}$ and $\theta_{\min}$ are set to 0.1 and 0.01 respectively. η_0 and θ_0 are the respective initial values, configured as $\eta_0 = 3^n \times \eta_{\min}$ and $\theta_0 = 3^n \times \theta_{\min}$, where n controls the number of iterations. Fig. 5(d) and (e) show flow fields before and after the continuous refinement in an image scale. We denote by $\mathbf{u}^r$ the refined flow field.

Fig. 6 demonstrates the effectiveness of this continuation scheme (that is, by altering η and θ in iterations) and compares results obtained with and without using it. We set different iteration numbers in experiments. The top row shows results with $n = 1$, $n = 2$, and $n = 5$ using the continuation scheme. The bottom row contains estimates using the algorithm shown in Table 2, by fixing $\eta = 0.1$ and $\theta = 0.01$ in all iterations. Energy decreasing w.r.t. the number of iteration is plotted in (i). It is clear from the comparison that our algorithm with the continuation scheme converges more efficiently.

4.3 Occlusion-Aware Refinement

Motion vectors for occluded pixels generally cannot be determined due to the lack of correspondences. In this step, we handle occlusion in the computed flow field. Although cross-checking is effective in occlusion detection, it needs to compute optical flow bidirectionally. Our strategy is based on an observation that multiple pixels mapping to the same point in the target image using forward warping are possibly occluded by each other.

Thus, we detect occlusion using the mapping uniqueness criterion [8], expressed as

$$o(\mathbf{x}) = T_{0,1}(\mathrm{f}(\mathbf{x} + \mathbf{u}(\mathbf{x})) - 1), \tag{23}$$

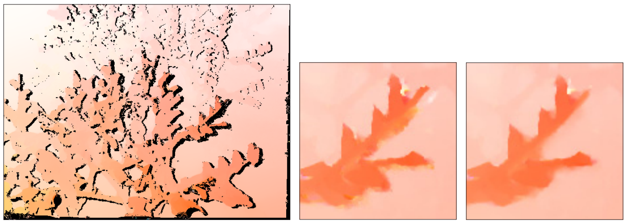

(a) Occlusion and flow (b) $\mathbf{u}^r$ (c) Refined

Fig. 7. Occlusion-aware refinement. (a) Flow estimate overlaid with the occlusion map ($o(\mathbf{x}) > 0.5$). (b) and (c) show results before and after the final refinement in an image scale.

where $\mathrm{f}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ is the count of reference pixels mapped to position $\mathbf{x} + \mathbf{u}(\mathbf{x})$ in the target view using forward warping. $T_{l,h}(a)$ is a function that truncates the value of a if it is out of the range $[l, h]$. Eq. (23) indicates if there exist more than one reference pixel mapping to $\mathbf{x} + \mathbf{u}(\mathbf{x})$, the occlusion label for the reference $\mathbf{x}$ is set. Although this simple method sometimes fattens the occlusion region, it seldom leaves out true occluded pixels, and thus is useful in the final flow estimation. In practice, we apply a small Gaussian filter on the computed $o(\mathbf{x})$ to reduce noise.

Our measure of the data confidence based on the occlusion detection is expressed as

$$\mathrm{c}(\mathbf{x}) = \max(1 - o(\mathbf{x}), 0.01). \tag{24}$$

The value 0.01 is to make $\mathrm{c}(\mathbf{x})$ always larger than 0. The metric is used in the following two ways to improve flow estimation in the occluded regions. First, we explicitly perform cross bilateral filtering for the detected occluded pixels where $o(\mathbf{x}) > 0.5$. Each pixel is further weighted by the measure $\mathrm{c}(\mathbf{x})$ so that occluded pixels have weaker influence in filtering. This scheme was shown to be effective in occlusion handling [28], [40] and was used in defining the flow function [33], [38].

Second, based on the fact that we should not trust the data term with large $o(\mathbf{x})$, the energy function is updated with respect to the occlusion confidence, which makes flow computation for the occluded pixels depend more on the

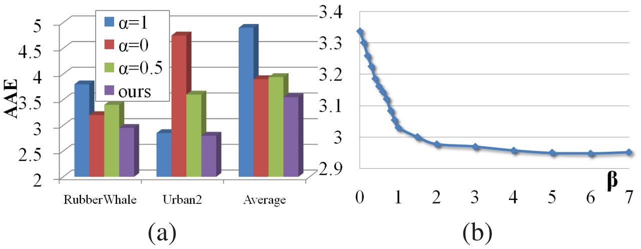


Fig. 8. Flow estimation error comparison. (a) Estimation errors w.r.t. α. $\alpha = 1$ and $\alpha = 0$ indicate respectively that only the color or gradient constancy constraint is used. $\alpha = 0.5$ refers to weighted addition of the two constraints. (b) Estimation errors w.r.t. β on the "RubberWhale" example.

local smoothness constraint:

$$E'(\mathbf{u}) = \mathrm{c}(\mathbf{x})E_D(\mathbf{u}) + \lambda E_S(\mathbf{u}). \tag{25}$$

It can be efficiently optimized also with our solver.

The occlusion-aware flow refinement is applied at each scale with the computed vectors from the continuous estimation step. The final result of the "Grove" example in one image scale is shown in Fig. 7 where the detected occlusion map is overlaid on the flow estimate. We compare the $\mathbf{u}^r$ maps obtained before and after our occlusion-aware refinement in (b) and (c).

5 Evaluation and Experiments

In this section, we present our results in both small- and large-displacement settings. τ in Eq. (2) is set to $1/1.4$ to normalize the color and gradient constraints, which is learned from the Middlebury training image set by setting the color and gradient costs to be equal. In order to reduce the sampling artifacts in Eq. (12), we filter $\mathcal{D}_\mathbf{I}$ and $\mathcal{D}_{\nabla\mathbf{I}}$ with a small Gaussian kernel with the standard deviation 1.0. β, λ, η, and θ are empirically set to 5, 12, 0.1, and 0.01 respectively. For feature detection, we use the implementation of Lowe [22]. Matches are retained only if ratios between the best and the second best matching scores are smaller than 0.6. For patch matching, we adopt the randomized nearest-neighbor method approximation [5] with patch size 5×5.

5.1 Evaluation of the Data Term

We evaluate the selective combination strategy in defining the data cost function. We compare our method with those using fixed weights $\alpha = 0.5$, $\alpha = 1$, and $\alpha = 0$ on the Middlebury training set [4], where the ground truth data are available. To demonstrate the alpha influence not involving our other steps, we employ the classic coarse-to-fine warping framework. The errors are listed in Fig. 8(a), calculated on the two representative examples "RubberWhale" and "Urban2". It can be noticed that the average angular error (AAE) for "Urban2" is small when using the color constraint alone while the gradient constraint is more favored in "RubberWhale" due primarily to illumination variation. Simply adding these two constraints ($\alpha = 0.5$) produces AAE in between. Our method locally selects the more optimal term and thus performs better.

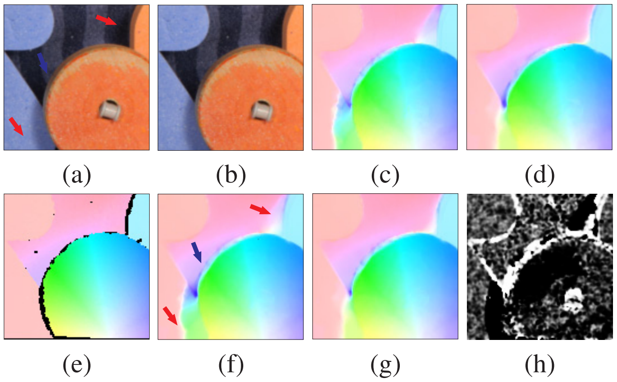


Fig. 9. Visual comparison with different α settings. (a) and (b) show two image patches. (c) and (d) show flow results computed using the color and gradient constraints respectively. (e) is the ground truth flow field. (f) shows the result with $\alpha = 0.5$. (g) is the flow map obtained using our selective combination model. (h) shows the $\bar{\alpha}$ map.

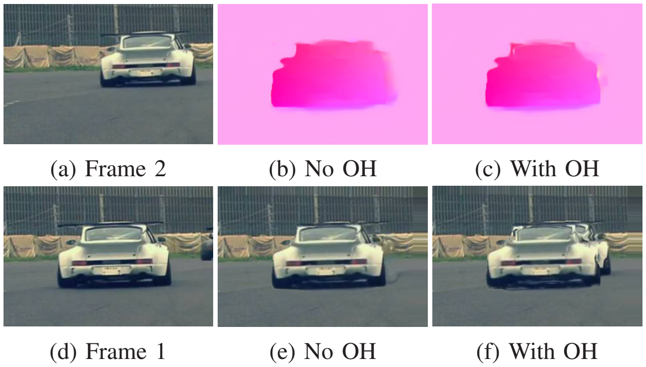


Fig. 10. Flow estimation of a car moving away from the camera. (a) and (d) are two input images. (b) and (e) show respectively the flow field produced without occlusion handling and the backward warping result. (c) and (f) are the flow field and the warping result with our occlusion handling. "OH" stands for occlusion handling.

Fig. 8(b) shows how the estimate changes with respect to different β for the "RubberWhale" example. $\beta = 0$ corresponds to weighted addition of the two normalized constraints. Note that the average error of the flow field decreases quickly with the increase of β, in line with our understanding (explained in Section 3.3).

In Fig. 9, we show a visual comparison. Red arrows in (a) and (f) indicate pixels violating the color constancy assumption. The blue arrows highlight the edge of the wheel, of which the gradient varies. (c) and (d) show results by respectively setting $\alpha = 1$ and $\alpha = 0$. (f) shows the result with $\alpha = 0.5$, where problems caused by using either of the constraints is still present. Our selective combination model helps robustly reject outliers, as shown in (g).

For quantitative comparison, a series of experiments with different optimization strategies are conducted, varying from traditional coarse-to-fine to our full optimization with EC2F and occlusion refinement. The error statistics are listed in Table 3, where "F" represents setting $\alpha = 0.5$ and

Method	RubberWhale	Hydrangea	Dimetrodon	Grove2	Grove3	Urban2	Urban3	Venus
F+C2F	3.29	2.25	2.58	2.26	5.88	3.61	5.23	6.45
A+C2F	2.95	2.02	2.50	2.09	5.51	2.81	4.28	6.07
F+EC2F	2.95	2.13	2.54	2.07	5.22	2.66	4.09	4.27
A+EC2F	2.61	2.06	2.51	2.00	5.17	2.40	3.32	3.86
F+EC2F+o	2.93	2.08	2.55	2.04	5.13	2.46	4.02	4.07
A+EC2F+o	2.59	2.02	2.52	1.92	4.87	2.22	3.21	3.56
F+EC2F+O	2.92	2.08	2.54	1.97	4.88	2.40	4.03	3.70
A+EC2F+O	2.59	1.97	2.51	1.88	4.77	2.15	3.15	3.55

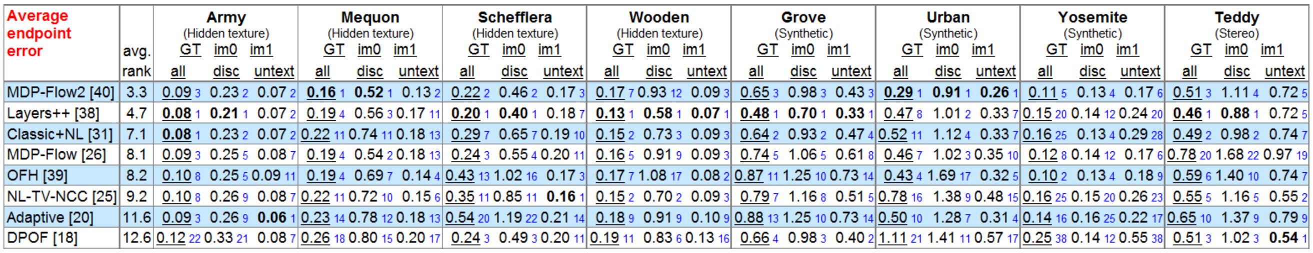
TABLE 3

AAEs yielded by different strategies on the Middlebury optical flow training data

Initial Flow	RubberWhale	Hydrangea	Dimetrodon	Grove2	Grove3	Urban2	Urban3	Venus
C2F	2.71	2.00	2.51	2.08	5.19	2.77	4.14	5.87
C+SIFT	2.61	1.97	2.50	1.94	4.79	2.17	3.20	3.65
C+PM	2.66	2.00	2.51	1.95	4.83	2.27	3.87	4.13
All	2.59	1.97	2.51	1.88	4.77	2.15	3.15	3.55

TABLE 4

AAEs on the Middlebury training data under different flow initialization



Average endpoint error	avg.	Army (Hidden texture)			Mequon (Hidden texture)			Schefflera (Hidden texture)			Wooden (Hidden texture)			Grove (Synthetic)			Urban (Synthetic)			Yosemite (Synthetic)			Teddy (Stereo)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [40]	3.3	0.09 3	0.23 2	0.07 2	**0.16** 1	**0.52** 1	0.13 2	0.22 2	0.46 2	0.17 3	0.17 7	0.93 12	0.09 3	0.65 3	0.98 3	0.43 3	**0.29** 1	**0.91** 1	**0.26** 1	0.11 5	0.13 4	0.17 6	0.51 3	1.11 4	0.72 5
Layers++ [38]	4.7	**0.08** 1	**0.21** 1	0.07 2	0.19 4	0.56 3	0.17 11	**0.20** 1	**0.40** 1	0.18 7	**0.13** 1	**0.58** 1	**0.07** 1	**0.48** 1	**0.70** 1	**0.33** 1	0.47 8	1.01 2	0.33 7	0.15 20	0.14 12	0.24 20	**0.46** 1	**0.88** 1	0.72 5
Classic+NL [31]	7.1	**0.08** 1	0.23 2	0.07 2	0.22 11	0.74 11	0.18 13	0.29 7	0.65 7	0.19 10	0.15 2	0.73 3	0.09 3	0.64 2	0.93 2	0.47 4	0.52 11	1.12 4	0.33 7	0.16 25	0.13 4	0.29 28	0.49 2	0.98 2	0.74 7
MDP-Flow [26]	8.1	0.09 3	0.25 5	0.08 7	0.19 4	0.54 2	0.18 13	0.24 3	0.55 4	0.20 11	0.16 5	0.91 9	0.09 3	0.74 5	1.06 5	0.61 8	0.46 7	1.02 3	0.35 10	0.12 8	0.14 12	0.17 6	0.78 20	1.68 22	0.97 19
OFH [39]	8.2	0.10 8	0.25 5	0.09 11	0.19 4	0.69 7	0.14 4	0.43 13	1.02 16	0.17 3	0.17 7	1.08 17	0.08 2	0.87 11	1.25 10	0.73 14	0.43 4	1.69 17	0.32 5	0.10 2	0.13 4	0.18 9	0.59 6	1.40 10	0.74 7
NL-TV-NCC [25]	9.2	0.10 8	0.26 9	0.08 7	0.22 11	0.72 10	0.15 6	0.35 11	0.85 11	**0.16** 1	0.15 2	0.70 2	0.09 3	0.79 7	1.16 8	0.51 5	0.78 16	1.38 9	0.48 15	0.16 25	0.15 20	0.26 23	0.55 5	1.16 5	0.55 2
Adaptive [20]	11.6	0.09 3	0.26 9	**0.06** 1	0.23 14	0.78 12	0.18 13	0.54 20	1.19 22	0.21 14	0.18 9	0.91 9	0.10 9	0.88 13	1.25 10	0.73 14	0.50 10	1.28 7	0.31 4	0.14 16	0.16 25	0.22 17	0.65 10	1.37 9	0.79 9
DPOF [18]	12.6	0.12 22	0.33 21	0.08 7	0.26 18	0.80 15	0.20 17	0.24 3	0.49 3	0.20 11	0.19 11	0.83 6	0.13 16	0.66 4	0.98 3	0.40 2	1.11 21	1.41 11	0.57 17	0.25 38	0.14 12	0.55 38	0.51 3	1.02 3	**0.54** 1

Fig. 11. The average end-point errors (EPEs) on the benchmark data as of Oct, 2010, copied from the Middlebury website [4]. Our method is denoted as "MDP-Flow2".

"A" stands for our adaptive α scheme. "C2F" and "EC2F" represent the classic and extended coarse-to-fine schemes respectively. As described in Section 4.1, "EC2F" uses extended flow initialization at each scale. Our method yields consistent quality improvement over other alternatives.

5.2 Evaluation of Occlusion Handling

We also evaluate the occlusion-aware refinement step. The bottom several rows of Table 3 list the statistics produced without ("*+EC2F") and with occlusion-aware refinement ("*+EC2F+O"). "*+EC2F+o" stands for occlusion handling used in the early version of the system [42], where cross bilateral filtering is not employed. Both occlusion handling methods yield reasonable results.

For the special case that an object moves away from the camera, the occlusion regions could be largely fattened because multiple pixels on the object when it is near could be mapped to one pixel when it is far. Even in this case, the flow estimates can still be refined because our occlusion handling in essence seeks flow discontinuity alignment with image edges.

We show in Fig. 10 an example. (b) and (e) show our flow and backward warping results without occlusion handling. The moving car is correctly reconstructed but the flow is not accurate at the occluded region (rightmost part of the car). With occlusion handling, the results are those shown in (c) and (f). The seemingly incorrect warping result in (f) in fact indicates correct handling of occlusion. The flow near the boundary is a bit noisy in (c), owing to the fattened occlusion region jeopardizing proper flow regularization. It is one of the limitations.

5.3 Evaluation of Extended Coarse-to-Fine

We also evaluate our coarse-to-fine framework with extended flow initialization. Specifically, we have tested 1) classical flow initialization in the coarse-to-fine framework; 2) flow initialization extended by SIFT feature matching only; 3) extended flow initialization with patch matching; 4) our flow initialization with both SIFT and patch matching. Their abbreviations are "C2F", "C+SIFT", "C+PM" and "All". The statistics are listed in Table 4. The results indicate that extended flow initialization ("C+SIFT", "C+PM", "All") can greatly improve estimation.

5.4 Middlebury Optical Flow Benchmark

We now evaluate our method on the Middlebury optical flow benchmark data. The table in Fig. 11 is copied in part

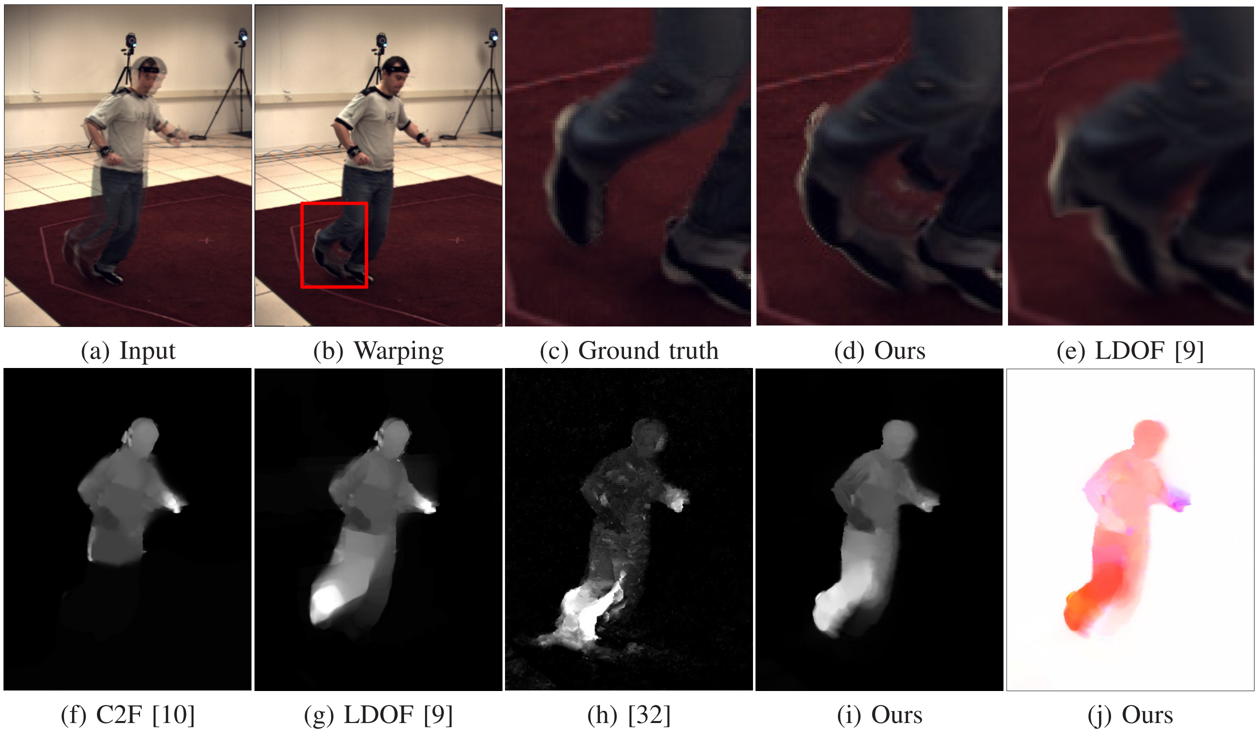


Fig. 12. Visual comparison on a large-displacement optical flow example from the HumanEva-II data set [30].

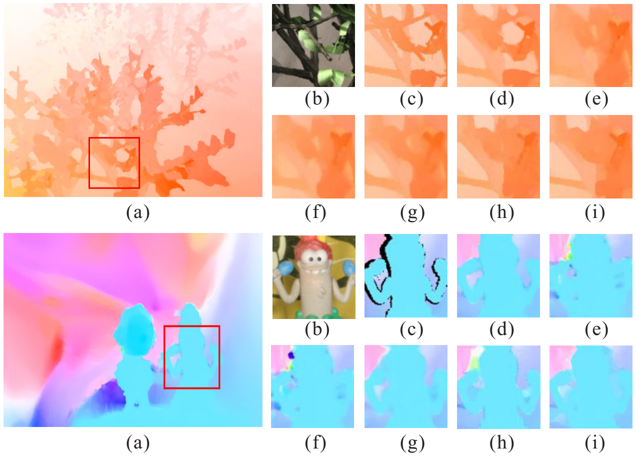


Fig. 13. Visual comparison of the small-displacement optical flow results on two examples. (a) Our flow results. Close-ups of (b) the input image, (c) ground truth flow, (d) our estimate, and of results of (e) Brox *et al.* [10], (f) LDOF [11], (g) Zimmer *et al.* [45], (h) Werlberger *et al.* [38], and (i) of Sun *et al.* [33], are shown.

from the evaluation website [4]. Our method, denoted as "MDP-Flow2", ranked 1st at the time of submission (as of Oct. 29, 2010).

Regarding the running time, in our current CPU implementation, the whole program takes 420s to compute a high quality flow field for an image pair with resolution 640×480 in, for instance, the *Urban* sequence. The running time is reported on a laptop computer containing an Intel Core i7 CPU @2.13GHz and 2GB Memory.

We show our flow results for two examples in Fig. 13(a)-(b). Methods of Brox *et al.* [10] that uses TV/ℓ_1 model for flow estimation, the large-displacement optical flow estimator [11] that incorporates descriptor matching in the data term, and of three top-performing methods that provide motion-discontinuity-preserving regularization terms, produce results shown in (e)-(i).

5.5 Large-Displacement Optical Flow Estimation

Our method by nature can deal with large-displacement flow, without any modification of the framework. One example from the HumanEva-II benchmark dataset [30] is shown in Fig. 12. It contains significant articulated motion of a running person. The fast foot movement cannot be estimated correctly in the conventional coarse-to-fine scheme [10], as shown in (f). (b) shows the backward warping result based on our dense flow estimate. The close-ups are shown in (d). Our method successfully recovers the shape of the left foot. The pixels in the occluded region are simply unknown for all optical flow estimation methods. The flow magnitude maps are shown in the second row. The maps in (g) and (h) are produced by two representative large-displacement optical flow methods.

The flexibility of our method is boosted by patch matching especially for non-rigid large-displacement motion estimation. Fig. 14(a) and (b) show two frames. The duck head undergoes very large motion. So it is not surprising that other optical flow methods based on the traditional coarse-to-fine scheme [10], [33] cannot cope with it well. The large-displacement methods using descriptor matching [11], [21], [42] produce results shown in Fig. 14(g)-(l). There are also errors. The flow and warping results produced by extended coarse-to-fine with only patch matching are

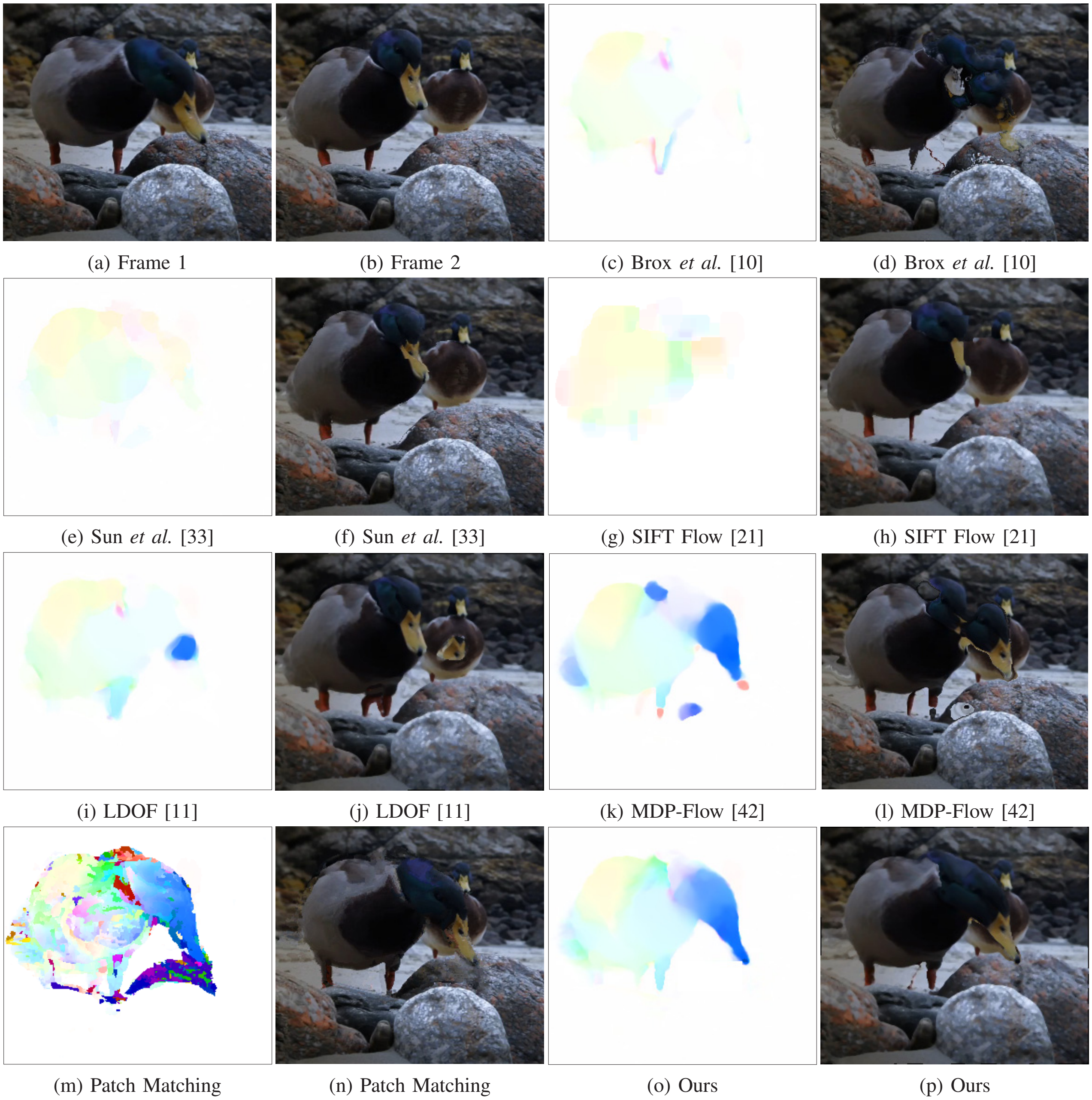

(a) Frame 1 (b) Frame 2 (c) Brox *et al.* [10] (d) Brox *et al.* [10]
(e) Sun *et al.* [33] (f) Sun *et al.* [33] (g) SIFT Flow [21] (h) SIFT Flow [21]
(i) LDOF [11] (j) LDOF [11] (k) MDP-Flow [42] (l) MDP-Flow [42]
(m) Patch Matching (n) Patch Matching (o) Ours (p) Ours

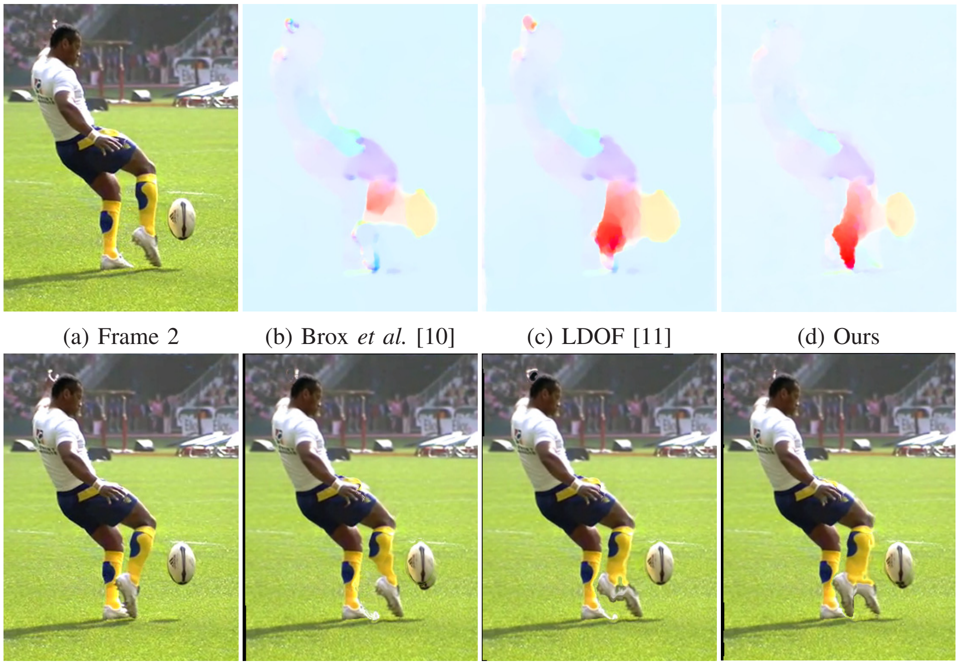
Fig. 14. A challenging example for large-displacement optical flow estimation. (a) and (b) show two input images. Seven flow estimation and the corresponding backward warping results are shown in (c)-(p).

shown in (m) and (n). Although the field is noisy, it roughly captures the head motion. Our final flow estimate, yielded with the complete EC2F scheme that involves both patch matching and feature matching, is shown in (o). Its quality is much higher.

Another example is shown in Figs. 15 and 16, which is a low-frame-rate sequence containing a football player. Fig. 15 contains the results of the conventional coarse-to-fine warping method [10], the large-displacement estimator [11], and of our method. Fig. 16 shows a few results in the sequence. All examples demonstrate that in terms of handling large motion of small-size regions, our method reduces the dependence on the linearization condition in the variational model and thus can generate good results.

6 Discussion and Conclusion

We have presented a new optical flow estimation framework to reduce the reliance on the coarse level estimation in the variational setting for small-size salient motion estimation. Differing from previous efforts mainly to improve the model, we instead revise flow initialization in the coarse-to-fine setting, which yields a unified framework to preserve motion details in both small- and large- displacement scenarios. The proposed method also takes advantage of the accurate variational coarse-to-fine framework and of non-local search/matching. Other main contributions include the selective combination of the color and gradient constraints, sparse feature matching and dense patch matching to collect



(a) Frame 2 (b) Brox *et al.* [10] (c) LDOF [11] (d) Ours

(e) Frame 1 (f) Brox *et al.* [10] (g)LDOF [11] (h) Ours

Fig. 15. Large-displacement optical flow results. (f)-(h) show the backward warping results based on the flow estimates in (b)-(d) respectively.

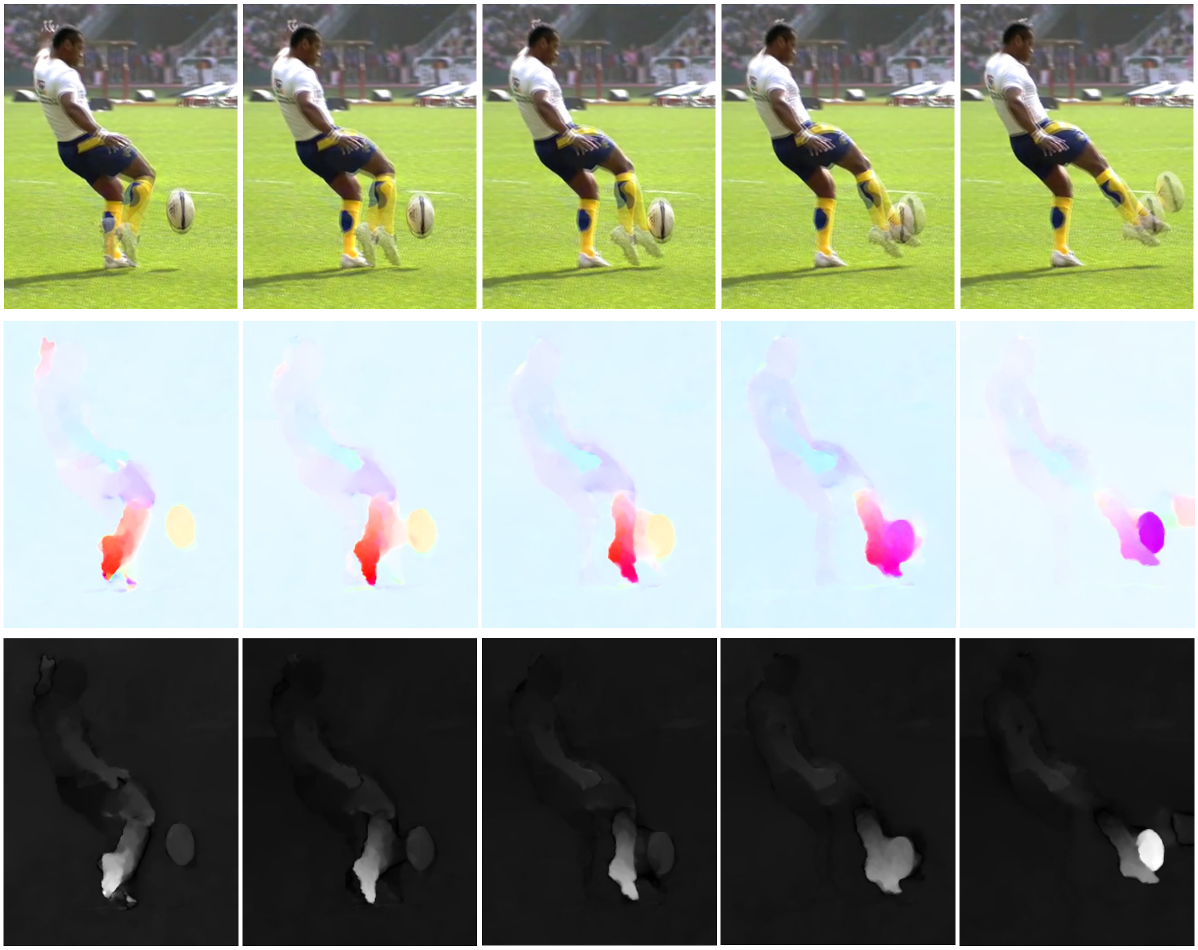
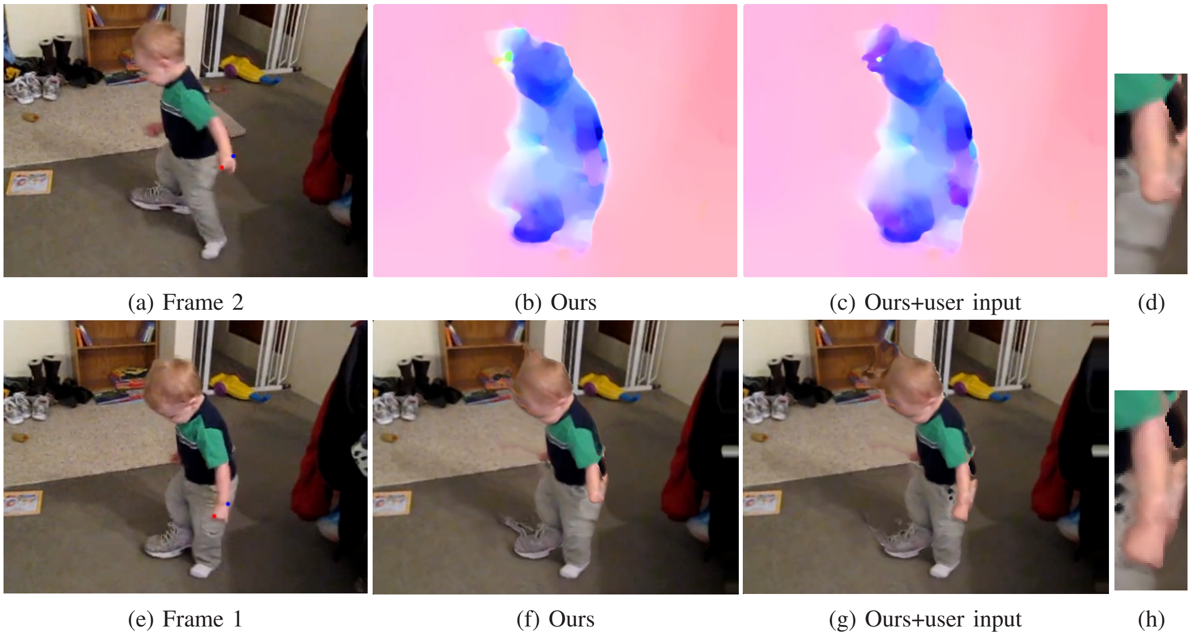


Fig. 16. Optical flow estimation in consecutive frames in a low-frame-rate sequence. First row: two-body-overlaid images to visualize the large displacement. Second row: our flow estimates. Third row: magnitude maps.


(a) Frame 2 (b) Ours (c) Ours+user input (d)

(e) Frame 1 (f) Ours (g) Ours+user input (h)

Fig. 17. A challenging example. (a) and (e) are two input frames. (b) and (f) show our automatically computed flow field and the backward warping result. (c) and (g) are the flow field and the backward warping result with simple user indication of two additional pairs of correspondence, shown as the red and blue dots in (a) and (e). (d) and (h) are close-ups of (f) and (g).

appropriate motion candidates, the mean field approximation to simplify optimization, and a variable splitting technique to enable fast and reliable flow estimation. Our future work will be system acceleration using GPU.

Limitations There are several limitations. First, although sparse feature matching and dense patch matching complement each other in proposing new flow candidates, they could still be insufficient especially for motion in textureless or regularly-patterned regions, where large matching ambiguity could occur. Other information such as simple user input may help.

We show one example in Fig. 17. (a) and (e) are input frames where the boy's right leg and arms undergo large motion. (b) and (f) show our estimated flow field as well as the backward warping result. Primary large-displacement motion (of the shoe, for example) is correctly computed except for the left arm. Note that textureless regions not only fail feature detection, but also create ambiguities for nearest-neighbor matching. In this example, we manually specify two corresponding points in the input frames (the red and blue dots in (a) and (e)), and then take the displacement vectors as new constant flow candidates to improve estimation. Final results are shown in (c) and (g) with close-ups in (d) and (h). They indicate that simple user interaction can decisively improve flow estimation in challenging regions.

Second, motion inference for large occluded regions is still an open problem due to lack of correspondence. Our current occlusion handling relies on a heuristic smoothness assumption, which could fail in texture- or color-rich regions when occlusion is significant. Incorporating other clues, such as color segmentation, may remedy the problem.

Acknowledgments

The authors would like to thank the associate editor and all the anonymous reviewers for their time and effort. This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (Project No. 412911) and by NSF of China (key project No. 61133009).

References

[1] L. Alvarez, J. Esclarin, M. Lefebure, and J. Sanchez. A pde model for computing the optical flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, 1999.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal on Computer Vision (IJCV)*, 2:283–310, 1989.

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. Technical report MSR-TR-2009-179, Microsoft Research, 2009. http://vision.middlebury.edu/flow/.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3), 2009.

[6] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, 1992.

[7] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding (CVIU)*, 63(1):75–104, 1996.

[8] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):993–1008, 2003.

[9] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *CVPR*, 2009.

[10] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 4, pages 25–36, 2004.

[11] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(3):500–513, 2011.

[12] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV*, pages 749–755, 2005.

[13] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal on Computer Vision (IJCV)*, 61(3):211–231, 2005.

[14] D. Geiger and F. Girosi. Parallel and deterministic algorithms for mrfs surface reconstruction and integration. A.I. Memo 1114, MIT, 1989.

[15] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. Technical Report 07-09, Rice University, 2007.

[16] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[17] H. W. Haussecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):661–673, 2001.

[18] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981.

[19] C. Lei and Y.-H. Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *ICCV*, 2009.

[20] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008.

[21] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, volume 3, pages 28–42, 2008.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision (IJCV)*, 60(2):91–110, 2004.

[23] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[24] É. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal on Computer Vision (IJCV)*, 46(2):129–155, 2002.

[25] S. M.Seitz and S. Baker. Filter flow. In *ICCV*, 2009.

[26] S. Roth and M. J. Black. On the spatial statistics of optical flow. In *ICCV*, pages 42–49, 2005.

[27] C. Rother, V. Kolmogorov, V. S. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, 2007.

[28] P. Sand and S. J. Teller. Particle video: Long-range motion estimation using point trajectories. In *CVPR*, volume 2, pages 2195–2202, 2006.

[29] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[30] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal on Computer Vision (IJCV)*, 87(1-2):4–27, 2010.

[31] M. Sizintsev and R. P. Wildes. Efficient stereo with accurate 3-d boundaries. In *BMVC*, pages 237–246, 1996.

[32] F. Steinbrücker and T. Pock. Large displacement optical flow computation without warping. In *ICCV*, 2009.

[33] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[34] D. Sun, S. Roth, J. P. Lewis, and M. J. Black. Learning optical flow. In *ECCV*, volume 3, pages 83–97, 2008.

[35] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[36] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *ICCV*, 2009.

[37] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l1 optical flow computation. In *Dagstuhl Visual Motion Analysis Workshop*, 2008.

[38] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *CVPR*, 2010.

[39] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, 2009.

[40] J. Xiao, H. Cheng, H. S. Sawhney, C. Rao, and M. A. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, volume 1, pages 211–224, 2006.

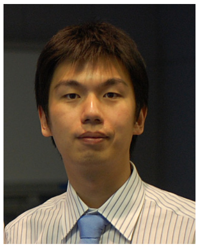
[41] L. Xu, J. Chen, and J. Jia. A segmentation based variational model for accurate optical flow estimation. In *ECCV*, volume 1, pages 671–684, 2008.

[42] L. Xu, J. Jia, and Y. Matsushita. Motion Detail Preserving Optical Flow Estimation. In *CVPR*, 2010.

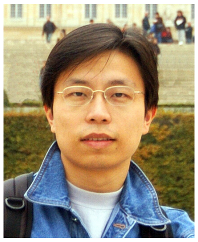
[43] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. *Pattern Recognition (Proc. DAGM)*, pages 214–223, 2007.

[44] H. Zimmer, A. Bruhn, and J. Weickert. Optic flow in harmony. *International Journal on Computer Vision (IJCV)*, 93(3):368–388, 2011.

[45] H. Zimmer, A. Bruhn, J. Weickert, B. R. Levi Valgaerts and, Agustín Salgado, and H.-P. Seidel. Complementary optic flow. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2009.



Li Xu received the BS and MS degrees in computer science and engineering from Shanghai JiaoTong University in 2004 and 2007 respectively, and the PhD degree in 2010 in computer science and engineering from the Chinese University of Hong Kong, where he is currently a postdoctoral fellow. He received the Microsoft Research Asia Fellowship Award in 2008 and served as reviewers for several major computer vision and graphics conferences and journals. His research interests include motion estimation, motion deblurring, image/video analysis and enhancement. He is a member of the IEEE.



Jiaya Jia received the PhD degree in Computer Science from Hong Kong University of Science and Technology in 2004 and is currently an associate professor in Department of Computer Science and Engineering at the Chinese University of Hong Kong (CUHK). He was a visiting scholar at Microsoft Research Asia from March 2004 to August 2005 and conducted collaborative research at Adobe Systems in 2007. He leads the research group in CUHK, focusing specifically on computational photography, 3D reconstruction, practical optimization, and motion estimation. He serves as an associate editor for TPAMI and as an area chair for ICCV 2011. He was on the program committees of several major conferences, including ICCV, ECCV, and CVPR, and co-chaired the Workshop on Interactive Computer Vision, in conjunction with ICCV 2007. He received the Young Researcher Award 2008 and Research Excellence Award 2009 from CUHK. He is a senior member of the IEEE.

Yasuyuki Matsushita received his B.S., M.S. and Ph.D. degrees in EECS from the University of Tokyo in 1998, 2000, and 2003, respectively. He joined Microsoft Research Asia in April 2003. He is a Lead Researcher in Visual Computing Group of MSRA. His major areas of research are computer vision (photometric techniques, such as radiometric calibration, photometric stereo, shape-from-shading), computer graphics (image relighting, video analysis and synthesis). Dr. Matsushita served as an Area Chair for IEEE Computer Vision and Pattern Recognition (CVPR) 2009 and International Conference on Computer Vision (ICCV) 2009, and he is on the editorial board member of International Journal of Computer Vision (IJCV), IPSJ Journal of Computer Vision and Applications (CVA), The Visual Computer Journal, and Encyclopedia of Computer Vision. He serves as a Program Co-Chair of PSIVT 2010, Demo Co-Chair for ICCV 2011, and Program Co-Chair for 3DIMPVT 2011. He is appointed as a Guest Associate Professor at Osaka University (April 2010-) and National Institute of Informatics, Japan (April 2011-). He is a senior member of IEEE.