Motion Detail Preserving Optical Flow Estimation[*]

Li Xu   Jiaya Jia
The Chinese University of Hong Kong
{xuli,leojia}@cse.cuhk.edu.hk

Yasuyuki Matsushita
Microsoft Research Asia
yasumat@microsoft.com

Abstract

We discuss the cause of a severe optical flow estimation problem that fine motion structures cannot always be correctly reconstructed in the commonly employed multi-scale variational framework. Our major finding is that significant and abrupt displacement transition wrecks small-scale motion structures in the coarse-to-fine refinement. A novel optical flow estimation method is proposed in this paper to address this issue, which reduces the reliance of the flow estimates on their initial values propagated from the coarser level and enables recovering many motion details in each scale. The contribution of this paper also includes adaption of the objective function and development of a new optimization procedure. The effectiveness of our method is borne out by experiments for both large- and small-displacement optical flow estimation.
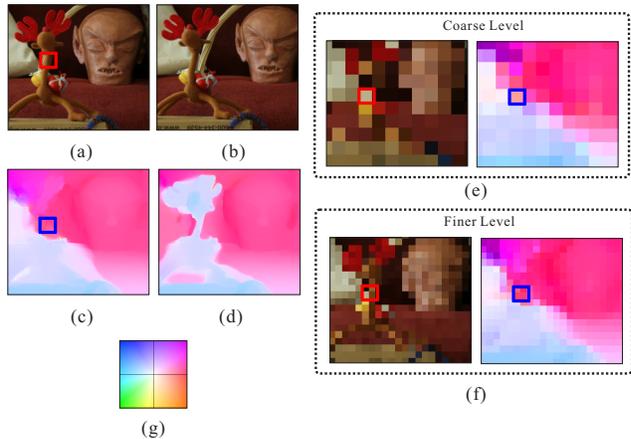
Figure 1. Motion detail preserving problem. (a)-(b) Two input patches. (c) Flow estimate using the coarse-to-fine variational setting. (d) Our flow estimate. (e)-(f) Two consecutive levels in the pyramid. Flow maps are visualized using the color code in (g).

1. Introduction

The variational framework [13], together with the coarse-to-fine refinement [1], is widely adopted in optical flow estimation [7, 8]. In the Middlebury optical flow evaluation website [2], almost all top-ranked methods use this strategy.

Brox *et al.* [6], in computing large-displacement optical flow, pointed out that if the flow structures are smaller than their displacements, the latter may not be well estimated. In this paper, we show that this issue also applies to *small-displacement motion*. Taking Figure 1 as an example, due to the camera motion, the foreground toy deer has its motion significantly differing from that of the background (average displacements $d = -2$ and $d = 21$ respectively). This example is in fact very challenging for the coarse-to-fine variational optical flow estimation.

As shown in Figure 1(e), in a coarse level, the narrow neck entirely disappears and only the significant background motion is estimated. This makes the emerging foreground pixels in the finer scale (Figure 1(f)) are with their actual motion significantly different from the initial esti-

mate from the background, violating the linearization assumption and accordingly leading to a highly unstable motion estimation process. The final flow result shown in Figure 1(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 displacements exists.

We address the motion detail preserving 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 extensive initial flow vectors in each image level. This makes the following optimization not completely rely on the result from the previous scale and thus capable to correct estimation error in the top-down refinement. Our flow result shown in Figure 1(d) contains fine structures. More examples are included in Section 5 and in the technical report [24].

This paper also contributes in the following ways. First, we use robust sparse feature 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 the

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, compatible with modern GPU computation architecture.

Finally, we employ the Mean Field approximation to enable solving the objective function, which involves both discrete and continuous variables and is commonly regarded challenging to solve. Extensive experiments visually and quantitatively validate the performance of our approach in maintaining details for both large- and small-displacement motion.

2. Related Work

Modern optical flow estimation is usually posed as an energy minimization problem. Black and Anandan [4] replaced the quadratic penalty functions in [13] with non-convex robust functions to reject outliers. Sun *et al.* [21] used a learning-based framework for both the matching cost (data term) and flow derivatives (smoothness term).

Efforts also have been put into improving the optical flow constraints. Haussecker and Fleet [12] proposed a physical constraint to model brightness change. Lempitsky *et al.* [14] computed the matching cost only using high frequency components. Pre-filtering on the input images was suggested in [21] and [17] to handle illumination variation. These models are flexible, but at the same time require to solve highly non-convex objective functions.

In [7], Brox *et al.* introduced the gradient constancy constraint to complement the brightness one. L^1 norm is used as the penalty function for both the data and smoothness terms so that the energy is convex after linearization. Similar compromise between robustness and complexity was also made in [8, 26]. However, we will show later that direct addition of the brightness and gradient terms is not optimal and propose a selection model to improve it.

Almost all the above methods rely on the coarse-to-fine warping to deal with motion larger than one pixel [1, 3]. As discussed in Section 1, this strategy has inherent problem to recover small-scale structures in many situations. Adaptive window is used in stereo matching [19] to handle incorrect initialization near depth boundary. It assumes at least the nearby disparities are correctly initialized, which might not be true for small-scale structures that are totally eliminated in the coarse level.

Using discrete optimization, Lempitsky *et al.* [14] proposed fusing flow proposals obtained from different flow estimation methods with various parameter settings. This is proven effective to find the optimal values among the given proposals. But the sufficiency and optimality of the proposals cannot be controlled. Also, the methods [16, 13] that generate the proposals still employ the conventional coarse-to-fine warping. So it is possible that none of the proposals
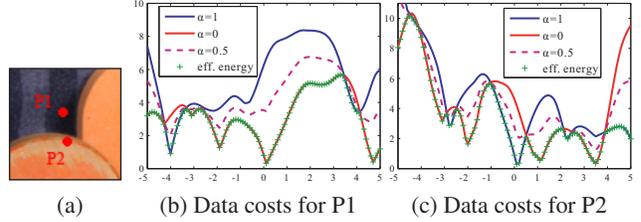

(a)   (b) Data costs for P1   (c) Data costs for P2

Figure 2. Data cost distributions for two points. (a) shows a patch of the "RubberWhale" example, where two points P1 (138,278) and P2 (141,299) are highlighted. (b) and (c) plot different data costs (vertical axis) for P1 and P2. The ground truth displacement is moved to 0 (horizontal axis) for illustration.

preserve small-scale motion structures. In comparison, our method computes a few high confidence flow candidates in each level, and thus is not entirely dependent of the flow initialization from the previous scale.

In recent large displacement optical flow estimation, Brox *et al.* [6] performed region-based descriptor matching. This method can effectively recover large displacement flow by adjusting the objective to favor matching results, albeit sometimes vulnerable to matching outliers. Steinbrücker and Pock [20] extended the numerical scheme of [25] and searched over all possible values for the large displacement flow. As discrete labels are used in the search step, results can be lack of sub-pixel accuracy.

3. Optical Flow Model

The Total Variation/L^1 model [7, 8, 25] was proven very effective in flow estimation. 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, adding gradient constancy constraints was proposed [7, 8]. Denoting by $\mathbf{u} = (u, v)^T$ the flow vector representing the displacement between frames $\mathbf{I_1}$ and $\mathbf{I_2}$, the data term for flow estimation can generally be written as

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

where τ is a weight. This function, due to the addition of two terms, is less accurate in terms of modeling the confidence of pixel correspondence than only using one out of the two terms that is more appropriate.

Figure 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 Figure 2(c), only the color constancy holds as point P2 undergoes rotational motion which alters image gradients. So 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* incorporate the more informative 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

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

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

3.2. Edge-Preserving Regularization

The regularization term for optical flow estimation is generally edge preserving [21, 22]. We define our smoothness term as

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

where $\mathbf{x} \in \mathbb{Z}^2$, indexing the 2D coordinates and $\|\nabla \mathbf{u}(\mathbf{x})\|$ is the common TV regularizer. $\omega(\mathbf{x})$ is the simple structure adaptive map that maintains motion discontinuity [22]:

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

where $\kappa = 0.8$ in our experiments. The final objective function is thus 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 commonly regarded as computationally intractable. We employ the Mean Field (MF) Approximation [9] to simplify the problem by first canceling out the binary process by integration over α [24]. 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 (as described in [24]) with the saddle point approximation, yielding

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

and the *effective* potential

$$
E^{eff}(\mathbf{u}) = \lambda E_S(\mathbf{u}) - \sum_{\mathbf{x}} \frac{1}{\beta} \ln(e^{-\beta \mathcal{D}_{\mathbf{I}}(\mathbf{u}, \mathbf{x})} + e^{-\beta \mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u}, \mathbf{x})}). \quad (9)
$$

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. (9) is actually the Mean Field (MF) approximation of minimizing Eq. (5). The effective energy is therefore written as

$$
E^{eff}(\mathbf{u}) = E_D^{eff}(\mathbf{u}) + \lambda E_S(\mathbf{u}), \quad (10)
$$

where the effective data function is

$$
E_D^{eff}(\mathbf{u}) = \sum_{\mathbf{x}} -\frac{1}{\beta} \ln(e^{-\beta \mathcal{D}_{\mathbf{I}}(\mathbf{u}, \mathbf{x})} + e^{-\beta \mathcal{D}_{\nabla \mathbf{I}}(\mathbf{u}, \mathbf{x})}). \quad (11)
$$

The optimality of Eq. (10) does not depend on the estimate of α. Moreover, although Eq. (10) is non-convex and is not easy 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 $\mathbf{u}$.

Note that the effective data term can also be deemed as a robust function which *selectively* combines the color and gradient constancy constraints. This can be clarified by taking the partial derivative with respect to the variable u on the data term, which yields

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

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

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

$\bar{\alpha}(\mathbf{x})$ is the MF-approximation of $\alpha(\mathbf{x})$. So its effect equates that of $\alpha(\mathbf{x})$ (Eq. (2)) in constraint selection. The cost distributions of the new effective data function are plotted in Figures 2(b) and (c) using green crossed curves. They indicate that the effective energy approximates the lower envelope of the two data costs ($\alpha = 0$ and $\alpha = 1$), which is exactly what we need for accurate flow estimation.

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. Generate multiple flow vectors as candidates.
3.3. Optimize flow using QPBO (Eq. (10)).
4. Continuous Flow Optimization (Section 4.2)
4.1. Compute the α map (Eq.(12)).
4.2. Solve the TV/L^1 energy function in Eq.(14).
5. If $l \neq n-1$ where n is the total number of levels, $l = l+1$ and go to Step 2.
6. Occlusion-aware Refinement (Section 4.3)
Output: optical flow map

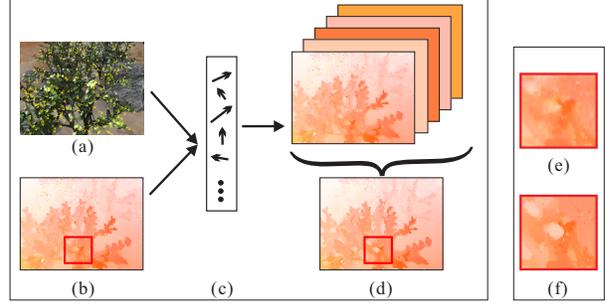Table 1. Overview of our method.

Figure 3. Extended flow initialization. (a) One of the images overlaid with the computed feature motion vectors. (b) Flow field $\mathbf{u}^c$ propagated from the coarser level. (c) New displacements computed using (a) and (b). They are candidate flow vectors for all pixels. (d) Optimized flow map $\mathbf{u}^0$ with respect to the candidates in the current image level. (e)-(f) show close-ups of (b) and (d).

4. Optimization Framework

Traditional optical flow estimation, due 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 due to the possible large discrepancy between the initial flow estimates and the ground truth displacements in each level.

In this section, based on E^{eff} and $\bar{\alpha}$, we propose an unconventional method to optimize Eq. (5). Specifically, because $E_D^{eff}(\mathbf{u})$ is not dependant 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) is then quickly estimated. We finally improve the subpixel accuracy of the flow estimates with the estimated $\bar{\alpha}$ using continuous optimization. This procedure is found surprisingly effective to dampen estimation errors caused by the occasionally biased flow results from the coarse level computation.

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 level by estimating *multiple* displacements from the reference to target images using the SIFT features detection and matching [15]. The displacement vectors are denoted as $\{\mathbf{u}_0^v, ..., \mathbf{u}_n^v\}$, as shown in Figure 3(a). They are new potential flow candidates except those that already exist in the flow map $\mathbf{u}^c$ propagated from the immediate coarser scale (Figure 3(b)). To robustly screen out the duplicated vectors, we compute the Euclidean distance between each $\mathbf{u}_i^v$ and all $\mathbf{u}_j^c$s where pixel j is within a 5×5 window centered at the reference feature of $\mathbf{u}_i^v$. If all results are greater than 1 (pixel), we regard $\mathbf{u}_i^v$ as a genuine flow candidate. We repeat this process for all is, and denote the m new can-

didates as $\mathbf{u}_{k_0}^v, ..., \mathbf{u}_{k_{m-1}}^v$. Figure 3(c) shows an example.

This strategy significantly reduces the system dependence on the coarse-scale flow estimation. It is notable as well that feature matching initially produces considerable vectors distributed in the whole image, as shown in Figure 3(a); but they reduce to less than 15 candidates after local comparison with $\mathbf{u}^c$ in the given example. Only the most distinctive flow vectors are retained.

The m new vectors $\mathbf{u}_{k_0}^v, ..., \mathbf{u}_{k_{m-1}}^v$, together with the original $\mathbf{u}^c$, represent possible motion in the present image scale. We model the selection of the optimal flow among the $m + 1$ candidates for each pixel as a labeling problem, where the objective function is given in Eq. (10). Upper of Figure 3(d) demonstrates the color coded labels. This problem can be solved by discrete optimization efficiently because on the one hand the number of candidates is small, thanks to the screening; on the other hand, Eq. (10) does not involve α, simplifying the computation.

We adopt the Quadratic Pseudo-Boolean Optimization (QPBO) [18] to solve this MRF problem. The fusion move step [14] is used to repeatedly fuse the candidates until each gets visited twice. Also, to avoid 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 [10]. The output is the flow map denoted as $\mathbf{u}^0$. One result is shown in Figure 3(d), which contains better recovered motion structure compared to the map $\mathbf{u}^c$ in Figure 3(b). Close-ups are shown in Figures 3(e) and (f).

Our method can work directly on the input images without employing the multscale framework. But it will suffer from expensive and possibly unstable computation because hundreds of or more labels might be produced simultaneously in the original resolution.
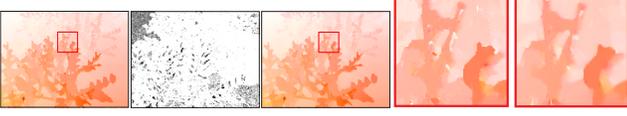
(a) $\mathbf{u}^0$ map (b) $\bar{\alpha}(\mathbf{x})$ map (c) $\mathbf{u}^r$ map (d) Close-up (e) Close-up
Figure 4. Continuous optimization. Errors are further reduced in this step. (d) and (e) show close-ups of (a) and (c).

4.2. Continuous Flow Optimization

The flow estimates from the previous step are taken into Eq. (12) to compute $\bar{\alpha}$. One result is shown in Figure 4(b). Considering that Eq. (11) is highly non-convex, we take $\bar{\alpha}$ back to Eq. (5) for optimization in the variational model.

As color images are used, we denote by $I^k \in \{I^r, I^g, I^b, \partial_x I, \partial_y I\}$ the set of channels to be 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. Then the energy in Eq. (5) is written as

$$E(\mathbf{u}) = \sum_{\mathbf{x}} \sum_{k} \alpha^k(\mathbf{x})|I_2^k(\mathbf{x}+\mathbf{u}) - I_1^k(\mathbf{x})| + \lambda(\mathbf{x})\|\nabla\mathbf{u}(\mathbf{x})\|, \quad (13)$$

where $\lambda(\mathbf{x}) := \lambda\omega(\mathbf{x})$. With the initially computed flow $\mathbf{u}^0$ from the previous step, we solve for the flow increments $\mathbf{du} = (du, dv)^{\mathrm{T}}$ by minimizing Eq. (13). The final flow vector is $\mathbf{u} = \mathbf{u}^0 + \mathbf{du}$. By convention, the Taylor expansion of Eq. (13) at point $\mathbf{x} + \mathbf{u}^0$ yields the linearized function

$$E(\mathbf{u}) = \sum_{\mathbf{x}} \sum_{k} \alpha^k(\mathbf{x})|I_x^k du + I_y^k dv + I_t^k|$$
$$+ \lambda(\mathbf{x})\|\nabla(\mathbf{u}^0 + \mathbf{du})(\mathbf{x})\| \quad (14)$$

given small $\mathbf{du}$. In Eq. (14), $I_x = \partial_x I_2(\mathbf{x}+\mathbf{u}^0)$, $I_y = \partial_y I_2(\mathbf{x}+\mathbf{u}^0)$, and $I_t = I_2(\mathbf{x}+\mathbf{u}^0) - I_1(\mathbf{x})$. To preserve motion discontinuity, we employ the rotational invariant isotropic form of the TV regularizer, written as

$$\|\nabla\mathbf{u}\| = \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_x v)^2 + (\partial_y v)^2}. \quad (15)$$

Our Solver We propose decomposing the optimization problem into three simpler ones, each of which can have the globally optimal solution. The key technique is a variable-splitting method with auxiliary variables $\mathbf{p}$ 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 very efficient and essential 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. (14) can be transformed into

$$\sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta}\|I_x^k du + I_y^k dv + I_t^k - p^k\|^2 + \alpha^k\|p^k\|$$
$$+ \frac{1}{2\theta}\|\nabla\mathbf{du} - \mathbf{w}\|^2 + \lambda\|\nabla\mathbf{u}^0 + \mathbf{w}\|. \quad (16)$$

In this function, $\frac{1}{2\eta}\|I_x^k du + I_y^k dv + I_t^k - p^k\|^2 + \alpha^k\|p^k\|$ encourages that p^k approaches $I_x^k du + I_y^k dv + I_t^k$, 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 also can be observed that Eq. (16) is equivalent to Eq. (14) upon convergence when $\theta \to 0$ and $\eta \to 0$.

Besides efficiency and reliability, Eq. (16) makes optimization highly parallel, fully compatible with GPU acceleration. The result optimality is guaranteed in each step. Our algorithm proceeds with the following iterations where the initial $\mathbf{u} = \mathbf{u}^0$.

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

$$\min \sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta}\|I_x^k du + I_y^k dv + I_t^k - p^k\|^2 + \alpha^k\|p^k\|. \quad (17)$$

Single variable optimization can be achieved in this step. The optimal solution is given by the shrinkage formula [11]

$$p^k = \text{sign}(o^k)\max(|o^k| - \eta\alpha^k, 0), \quad (18)$$

where $o^k := I_x^k du + I_y^k dv + I_t^k$ is the optical flow constraint.

2. Fix $\mathbf{u}$ and solve for $\mathbf{w}$. The function reduces to

$$\min \sum_{\mathbf{x}} \frac{1}{2\theta}\|\nabla\mathbf{du} - \mathbf{w}\|^2 + \lambda\|\nabla\mathbf{u}^0 + \mathbf{w}\|_2. \quad (19)$$

Similarly, unique solution is guaranteed by the shrinkage formula

$$w^{du_x} = \max(\|\nabla\mathbf{u}\|_2 - \theta\lambda, 0)\frac{\partial_x u}{\|\nabla\mathbf{u}\|_2} - \partial_x u^0, \quad (20)$$

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 in nature.

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

$$\min \sum_{\mathbf{x}} \sum_{k} \frac{1}{2\eta}\|I_x^k du + I_y^k dv + I_t^k - p^k\|^2 + \frac{1}{2\theta}\|\nabla\mathbf{du} - \mathbf{w}\|^2. \quad (21)$$

It is quadratic and thus the corresponding Euler-Lagrange equations of Eq. (21) are linear w.r.t. du and dv. Globally optimal solution can be directly obtained by solving the linear system [13] in this step.

Our method iterates among optimizing (18), (20) and (21) until convergence. Note that θ and η are critical parameters that should have very small values. It was found however fixing them constants typically results in slow convergence. We thus adopt the continuation scheme [11], which initially sets θ and η to relatively 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 denotes the number of iterations. More explanations are given in [24].

Continuous Flow Optimization
$\eta \leftarrow \eta_0$
repeat
Compute p^k using Eq.(18)
$\theta \leftarrow \theta_0$
repeat
Compute $\mathbf{w}$ using Eq. (20).
Compute $\mathbf{du}$ by solving Eq. (21).
$\theta \leftarrow \theta/3$
until $\theta < \theta_{\min}$
$\eta \leftarrow \eta/3$
until $\eta < \eta_{\min}$

Table 2. Algorithm for continuous flow optimization.
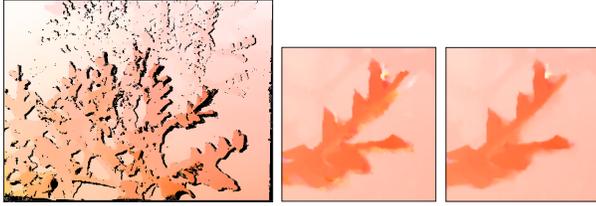
(a) Occlusion and flow   (b) $\mathbf{u}^r$ (c) Final result

Figure 5. 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.

Figures 4(d) and (e) show flow maps before and after the continuous refinement in an image scale. We denote by $\mathbf{u}^r$ the refined flow map.

4.3. Occlusion-Aware Refinement

Our final step is for handling large occlusion in the computed flow map. Although cross-checking is effective in occlusion detection, it needs to compute flow fields 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 [5], expressed as

$$o(\mathbf{x}) = T(m(\mathbf{x} + \mathbf{u}(\mathbf{x})) - 1, 0, 1), \qquad (22)$$

where $m(\mathbf{x})$ is the count of reference pixels mapped to position $\mathbf{x} + \mathbf{u}(\mathbf{x})$ in the target view using forward warping. $T(a, l, h)$ is a function that truncates the value of a if it is out of $[l, h]$. Eq. (22) indicates if there exist more than one reference pixels 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 does not harm the final flow estimation.

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

$$c(\mathbf{x}) = \max(1 - o(\mathbf{x}), 0.01). \qquad (23)$$

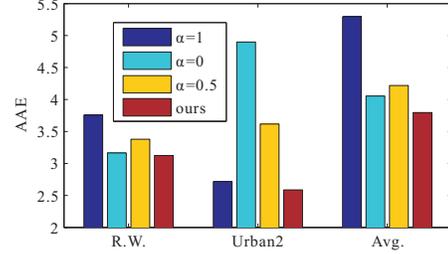Figure 6. Flow estimation errors w.r.t. different αs.

The larger $o(\mathbf{x})$ is, the less we trust the data term. 0.01 is to make $c(\mathbf{x})$ always larger than 0. The final energy used to refine the disparity map is

$$E'(\mathbf{u}) = c(\mathbf{x})E_D(\mathbf{u}) + \lambda E_S(\mathbf{u}), \qquad (24)$$

which can be efficiently optimized with our continuous solver. The final result of the "Grove" example in one image scale is shown in Figure 5 where the detected occlusion map overlays the flow estimate. (b) and (c) compare the $\mathbf{u}^r$ map computed from the continuous flow optimization step with the final occlusion-aware refinement result.

5. Evaluation and Experimental Results

In this section, we present our results and comparison in both small- and large-displacement settings. τ in Eq. (2) is set to $1/1.4$, which is learned from the Middlebury training image set by equaling the color and gradient costs. 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 matching, we use the implementation of Lowe [15] with default parameter values.

5.1. Evaluation of the Data Term

We first evaluate the effectiveness of our selective combination strategy in defining the data function. We compare our method with the those that set $\alpha = 0.5$, $\alpha = 1$, and $\alpha = 0$ respectively. For fairness, we do not use the complete framework to generate our results, which would otherwise produce flow estimates with even smaller error. Instead, we optimize Eq. (11) simply by fusing the two flow maps computed with $\alpha = 1$ and $\alpha = 0$ respectively using graph cuts.

Figure 6 shows the error comparison on the Middlebury training data [2] where the ground truth flow map is available. The two representative examples are "RubberWhale" ("R.W." for short) and "Urban2". It can be noticed that the average angular error (AAE) for "Urban2" is small when using the color constraint while the gradient constraint is favored in "RubberWhale" due to illumination change. Simply adding these two constraints produces AAE always in between [8]. In comparison, our method locally selects the optimal term and is more effective for energy minimization.
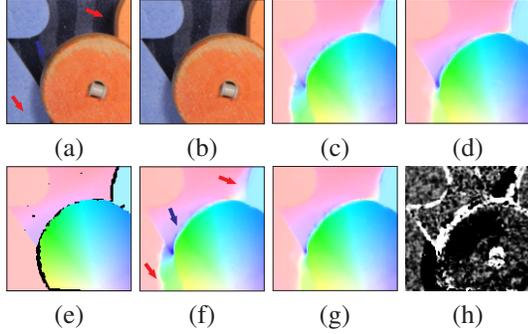
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. 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 corresponding $\bar{\alpha}$ map.
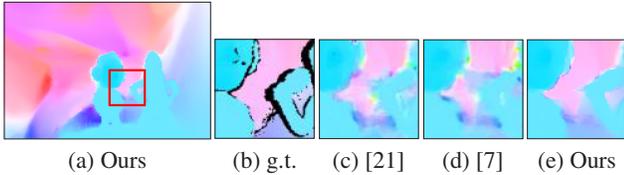

(a) Ours (b) g.t. (c) [21] (d) [7] (e) Ours

Figure 8. Visual comparison of the small-displacement optical flow results. (b) shows the ground truth flow map.

(a) (b) g.t. (c) [4] (d) [21] (e) [7]

(f) [14] (g) [26] (h) [22] (i) [23] (j) Ours

Figure 9. Extensive visual comparison.

(a) Inputs (b) Warping (c) g.t. (d) Ours (e) [6]

(f) C2F [7] (g) [6] (h) [20] (i) Ours (j) Flow

Figure 10. Visual comparison on a large-displacement optical flow example [6].

Figure 7 shows the 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 still present. Our selective combination model helps more robustly reject outliers, as shown in 7(g).

5.2. Middlebury Optical Flow Benchmarking

In this subsection, we evaluate our optical flow estimation method using the traditional small displacement data. Table 3 lists the average ranks of the top-performing flow estimation methods on Middlebury evaluation website. Many of the small-scale motion structures can be recovered by our method.

Regarding the running time, the extended flow initialization uses about 3 minutes in the fine image level (resolution 640×480), taking the *Urban* sequence as an example. The continuous flow refinement takes about one minute using the single thread CPU implementation. With GPU acceleration, this process can be further speeded up.

We visually compare flow results for one example in Figure 8, which shows that our flow estimate contains more motion details. In Figure 9, we show an extensive comparison of results produced by a number of state-of-the-art optical flow me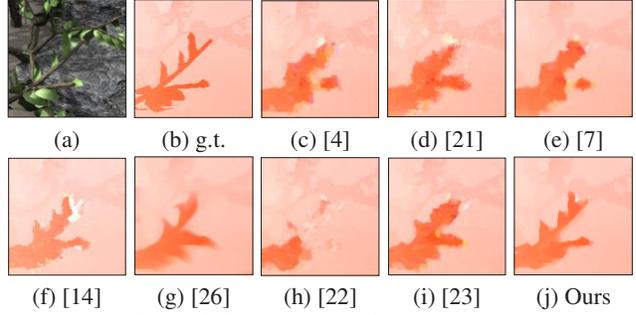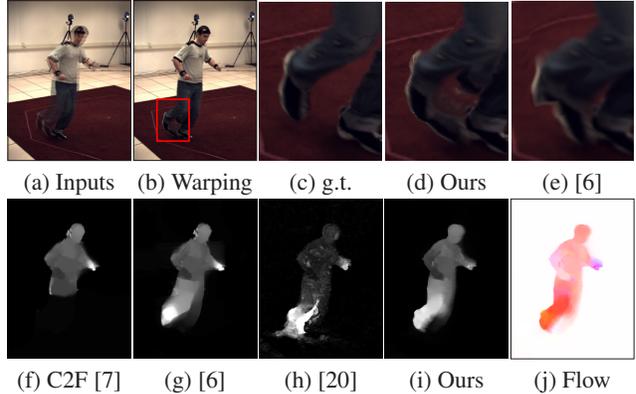thods, including those employing non-convex penalty functions [4, 21], using TV/L^1 model to reject outliers [7], minimizing energy in a continuous-discrete fashion [14], and applying advanced smoothness terms to handle motion discontinuity [26, 22, 23]. The inadequate ability to handle large motion discrepancy on narrow objects in the traditional multiscale variational framework makes many results still lack of a few details.

5.3. Large-Displacement Optical Flow Estimation

Our method can naturally deal with large-displacement flow estimation without any modification on the framework. Figure 10 shows an example containing significant articulated motion of a running person (published in [6]). (a) shows a two-object-overlaid image from the HumanEva-II benchmark dataset. Note that the fast foot movement cannot be estimated correctly in the conventional coarse-to-fine scheme [7], as shown in Figure 10(f). (b) shows the backward warping result using out dense flow estimate. The close-ups are shown in (d) and (e) for comparison. Our method successfully recovers the shape of the left foot. Note that the pixels in the occluded region are unrecoverable 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 the methods of [6] and [20], both of which dedicate to large-displacement flow estimation and

Method	Our Method	Adaptive [22]	Comp. OF [26]	Aniso. Hurber-L1 [23]	Brox *et al*. [7]
Avg. AAE Rank	4.5	6.0	5.9	7.6	11.5
Avg. EPE Rank	4.0	5.9	7.3	7.8	11.1

Table 3. The average ranking of the methods with top performance on the Middlebury optical flow evaluation website (at the time of submission). The two types of ranking are based on average angular errors (AAEs) and average end-point errors (EPEs) respectively.

do not perform best in handling small displacement motion. Several other examples are included in [24].

6. Concluding Remarks

In this paper, we have presented a novel optical flow estimation method to reduce the reliance on the coarse level flow estimation in the variational setting for small-size salient motion structure estimation. Other main contributions include the selective combination of color and gradient constraints, feature matching to find appropriate motion candidates, the mean field approximation to simplify optimization, and a variable splitting technique to enable fast and reliable flow estimation.

It is notable that although the sparse feature matching is a useful strategy to find novel flow candidates, occasionally it might not perform well enough especially when a very small region is entirely textureless. Exhaustive search can be used to solve this problem with higher computational cost.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual. *IJCV*, 2:283–310, 1989.

[2] 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.

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

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

[5] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):993–1008, 2003.

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

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

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

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

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

[11] 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.

[12] H. W. Haussecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):661–673, 2001.

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

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

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

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

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

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

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

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

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

[23] M. Werlberger, W. Trobin1, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, 2009.

[24] L. Xu, J. Jia, and Y. Matsushita. A unified framework for large- and small-displacement optical flow estimation. Technical report, The Chinese University of Hong Kong, 2010. www.cse.cuhk.edu.hk/%7eleojia/projects/flow/index.html.

[25] 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.

[26] H. Zimmer, A. Bruhn, J. Weickert, B. R. Levi Valgaerts and, Agustín Salgado, and H.-P. Seidel. Complementary optic flow. In *EMMCVPR*, 2009.