Large Displacement Optical Flow from Nearest Neighbor Fields

Zhuoyuan Chen[1] Hailin Jin[2] Zhe Lin[2] Scott Cohen[2] Ying Wu[1]

[1]Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{zch318, yingwu}@eecs.northwestern.edu

[2]Adobe Research
345 Park Ave, San Jose, CA 95110
{hljin,zlin,scohen}@adobe.com

Abstract

We present an optical flow algorithm for large displacement motions. Most existing optical flow methods use the standard coarse-to-fine framework to deal with large displacement motions which has intrinsic limitations. Instead, we formulate the motion estimation problem as a motion segmentation problem. We use approximate nearest neighbor fields to compute an initial motion field and use a robust algorithm to compute a set of similarity transformations as the motion candidates for segmentation. To account for deviations from similarity transformations, we add local deformations in the segmentation process. We also observe that small objects can be better recovered using translations as the motion candidates. We fuse the motion results obtained under similarity transformations and under translations together before a final refinement. Experimental validation shows that our method can successfully handle large displacement motions. Although we particularly focus on large displacement motions in this work, we make no sacrifice in terms of overall performance. In particular, our method ranks at the top of the Middlebury benchmark.

1. Introduction

Inferring a dense correspondence field between two images is one of the most fundamental problems in Computer Vision. It started with Horn and Schunck's original optical flow work [15] in the early eighties. There have since been many great advances. However, a good solution still remains elusive in challenging situations such as occlusions, motion boundaries, texture-less regions, and/or large displacement motions. This paper addresses particularly the issue of large displacement motions in optical flow.

Most existing optical flow formulations are based on linearizing the optical flow constraint which requires an initial motion field between the two images. In the absence of any prior knowledge, they use zero as the initial motion field which is then refined by a gradient-based optimization technique. Gradient-based optimization methods can only recover small deviations around the initial value. To handle large deviations, i.e. large displacement motions, most optical flow methods adopt a multi-scale coarse-to-fine framework which sub-samples the images when going from a fine scale to a coarse scale. Sub-sampling reduces the size of the images and the motion within, but at the same time, the reduction in image size leads to a loss of motion details that any algorithm can recover. Because of sub-sampling, most methods that rely on the coarse-to-fine framework perform poorly on image structures with motions larger than their size. This is an intrinsic limitation of the coarse-to-fine framework.

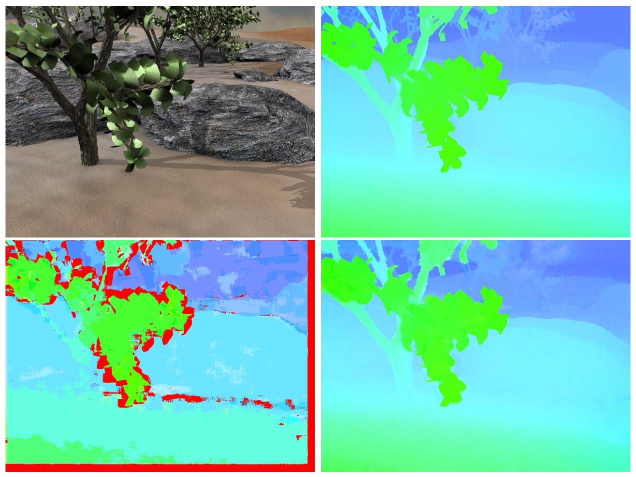
Figure 1. **Top left**: one of two "Grove2" images from the Middlebury dataset. **Top right**: color-coded ground-truth motion. **Bottom left**: color-coded motion computed from an approximate nearest neighbor field computed using [17]. We color the locations with incorrect motions as red. One can see that the motion of many parts of the image is correct. **Bottom Right**: the final motion result from our algorithm.

The intrinsic limitation of the coarse-to-fine framework comes from the zero motion assumption which makes sense when there is no prior information on the motion. However, it turns out that it is possible to obtain reliable motion information for a sparse set of distinct image locations using robust keypoint detection and matching such as [19]. One can

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.316

incorporate the sparse matches into a dense field through either motion segmentation [16], constraints [9] or fusion [26]. Since keypoints are detected and matched in the entire image, in theory these methods have no restrictions on the amount of motion they can handle. However, in practice they are subjective to the performance of the keypoint detection and matching algorithms. In particular, regions with weak texture often do not yield reliable keypoints and their correspondence problems remain ambiguous.

In this paper, we propose to incorporate a different type of correspondence information between two images, namely *nearest neighbor fields* [5]. A nearest neighbor field between two images is defined as, for each patch in one image, the most similar patch in the other image. Computing exact nearest neighbor fields can be computationally expensive depending on the size of images but there exist efficient approximate algorithms such as [5, 13, 17, 21]. Approximate nearest neighbor field algorithms are shown to be effective in terms of finding visually similar patches between two images. The first key observation in this paper is that although not designed for the motion estimation problem, approximate nearest neighbor fields contain a sufficiently high number of patches with approximately correct motions (see Figure 1). However, one cannot directly use nearest neighbor fields as the input for a nonlinear refinement because they often contain a huge amount of noise. Our second key observation is that most images are composed of a small number of spatially contiguous regions that have similar motion patterns. Based on this observation, we can view the problem as a motion segmentation problem. In particular, we segment the images into a set of regions that have similar motion patterns using a multi-label graph-cut algorithm [8]. We compute the motion patterns from a noisy nearest neighbor field using an algorithm that is robust to noise.

There are two issues in the motion segmentation formulation which are the type of motion patterns and the number of them. These two issues are related in that more complex motion patterns can describe the image with fewer patterns. However, the more complex are the patterns, the more difficult they are in terms of inferring them from a noisy nearest neighbor field. In this paper, we choose to use similarity transformations as the motion pattern. Most images contain small deformations that cannot be modeled by similarity transformations. This leads to errors in motion segmentation such as bleeding observed in the results of [16]. Our third key observation is that it is not necessary for the motion segmentation to obtain perfect results in terms of motion estimation, as long as the error is within the limit of a typical optical flow refinement. Based on this observation, we propose to allow for small deformations on top of the similarity transformations in motion segmentation.

Finally, we observe that although motion segmentation with similarity transformations and local deformations is very effective in terms of capturing the overall motion between two images, it may sometimes miss objects of small scale. We experimentally observed that small objects can be reliably recovered under a translational motion pattern. Therefore, we perform a fusion between the motion segmentation result under translations and that under the similarity transformations before a final refinement.

1.1. Related Work

There is a huge body of literature on optical flow following the original work of Horn and Schunck [15]. It is beyond the scope of this paper to review the entire literature. We only discuss the papers that address the large displacement motion problem as that is the main focus of this work.

The coarse-to-fine framework was first proposed in [2, 12]. It has since been adopted by most optical flow algorithms to handle large displacement motions. [1] was probably the first to note that the standard coarse-to-fine framework may not be sufficient. They proposed to modify the standard framework by using a linear scale-space focusing strategy to avoid convergence to incorrect local minima. Instead, Steinbruecker et al. [22] proposed a new framework which avoids warping and linearization. However, the algorithm performs an exhaustive search for candidate correspondences which can be computationally expensive. Brox and Malik [9] proposed to add robust keypoint detection and matching such as the SIFT features [19] into the classical optical flow framework which can handle arbitrarily large motion without any performance sacrifice. Xu et al. [26] also proposed to use similar robust keypoints. Instead of adding keypoint matches as constraints, they expand the matches into candidate motion fields and fuse them with the standard optical flow result. Both [9] and [26] are based on keypoint detection and matching algorithms and may suffer in regions with weak texture due to lack of reliable keypoints.

Our work is related to [16] where motion segmentation is also used to deal with large displacement motions. The differences are that we use nearest neighbor fields as the input and we allow local deformations in motion segmentation, both of which are shown to improve the overall performance. Motion segmentation is related to the so-called layer representation in optical flow [24, 25]. The advantages having an explicit segmentation are that we can incorporate simple models such as translation or similarity transformations to describe the motion between two images and integrate the correspondence information within individual segments. Our work is related to [7, 24] in terms of using local deformations and to [18, 26] in terms of fusing flow proposals obtained with different algorithms or with different parameter settings. Finally, our work is related to [6] in terms of using nearest neighbor fields to compute correspondence. The differences are that [6] uses a variant of Belief Propagation to regularize the noisy nearest neighbor fields and we use motion segmentation.

1.2. Contributions

The main contribution of this work is an optical flow that can handle large displacement motions. In particular we improve upon existing methods in the following ways:

- We use approximate nearest neighbor field algorithms to compute an initial dense correspondence field. It turns out the approximate nearest neighbor field contains a high percentage of approximately accurate motions which can be used by robust algorithms to recover the dominant motion patterns.
- We formulate the motion estimation problem as a motion segmentation problem and allow local deformations in the segmentation process. Having local deformations significantly reduces the number of motion patterns needed to describe the motion and therefore improves the robustness of the algorithm.
- We find experimentally small objects with large motions are easier to discover under translations. We use a novel fusion algorithm to merge the motion result under translations with that under similarity transformations.

Admittedly, our method focuses on the large displacement motion issue in optical flow and does not explicitly address other outstanding issues, such as occlusions, motion boundaries, etc. However, our algorithm achieved a top ranking on the Middlebury benchmark.

2. Our Approach

Our optical flow estimation algorithm consists of the following four steps: (1) Computing an approximate nearest neighbor field between the two input images, (2) Identifying dominant motion patterns from the nearest neighbor field, (3) Performing motion segmentation with dominant motion patterns, (4) Local flow refinement by traditional optical flow formulation.

2.1. Nearest Neighbor Fields (NNF)

Given a pair of input images, we first compute an approximate nearest neighbor field between them using Coherency Sensitive Hashing (CSH) [17]. As noted in introduction, the nearest neighbor field is approximately consistent to the ground truth flow field in majority of pixel locations. Our empirical study shows that this is a valid assumption for most cases in optical flow estimation.

Under this assumption, there are two advantages in leveraging the nearest neighbor field for optical flow problems. Firstly, since nearest neighbor field algorithms are not restricted by the magnitude of motions, they can provide valuable information for handling large motions in optical flow estimation, which has been a main challenge for traditional optical flow algorithms. Secondly, although the nearest neighbor fields are generally noisy, they retain motion details for small image structures, which would most likely be ignored by traditional optical flow algorithms.

Directly applying the nearest nearest neighbor field as an initialization to traditional optical flow algorithms cannot recover from large errors in the nearest neighbor field since these algorithms only refine flows locally, which makes noise handling crucial in our formulation.

The patch size w used in the NNF computation is an important parameter in our algorithm. On the one hand, a larger kernel eliminate matching ambiguity. Especially in repetitive patterns and textureless regions, a large range of context is required for accurate matching. On the other hand, a large kernel has more risk to contain multiple motion modes. Typically, we choose $w = 16$ for sequences of size 640×480 and $w = 32$ for high-resolution movies.

2.2. Dominant Motion Patterns

To suppress noise on the nearest neighbor field, here we propose a motion segmentation-based method by restricting the initial nearest neighbor field to a sparse set of dominant motion patterns. To achieve this, we fist identify those patterns robustly based on simple geometric transformation models between the two images based on an iterative RANSAC algorithm and then use them to compute motion segmentation from the nearest neighbor field.

We can simply adopt histogram statistics as in [14] to extract K most frequent motion modes and use them as the dominant motion patterns. This would work well if the underlying motions only consist of translations. However, when the scene contains more complex rigid motions such as rotation and scaling, the number of modes required for accurately representing the underlying motion field can be very large.

To address this problem, we can extract the dominant motion patterns under more sophisticated models such as similarity/affine transformations. In other words, the problem is to estimate dominant projection matrices $P = \{P_1, ..., P_J\}$ from the nearest neighbor field. There have been several works (e.g. [16]) adaptively estimating multiple homographies from sparse SIFT correspondences by RANSAC [20], but they generally do not eliminate correspondences deemed as "inliers" and add in perturbed interest points to avoid elimination of too many correspondences and true independent motions, which is also named as "Phantom motion fields" in [16].

As in these methods, if we do not eliminate "inliers", the procedure will be biased towards large moving objects,

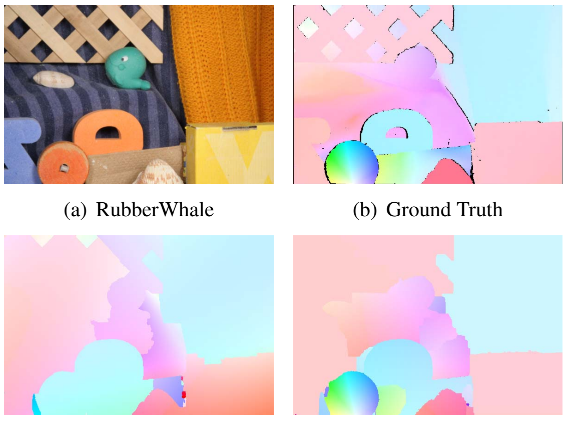
(a) RubberWhale (b) Ground Truth (c) SIFT [16] (d) PatchMatch

Figure 2. A comparison of dominant motion patterns extracted from sparse SIFT correspondences and a dense nearest neighbor field. (a) The RubberWhale example [4], (b) The ground truth, (c)(d) Th similarity transformations inferred with dominant motion patterns extracted from sparse feature correspondences and the nearest neighbor field, respectively.

which may cause a problem for relatively small objects such as a flying tennis. It is possible that the true motion pattern corresponding to a small object cannot be identified even with a large number of RANSAC trials.

Here, we adopt a more robust approach by removing only those "inliers" with high confidence values (samples which are sufficiently close to the current motion pattern) during the iterative RANSAC-based motion estimation process. Also, the large number of potential correspondences offered by the nearest neighbor field allows us to estimate the motion patterns robustly even for small objects or non-texture scenes.

Figure 2 shows an example from the Middlebury benchmark [4], which demonstrates the advantage of our method over a sparse feature correspondence-based method (e.g. [16, 26]). As we can see from the figure, due to lack of textures on the cylinder, the SIFT-based estimation [16] cannot reconstruct the motion of the rotating cylinder accurately. In contrast, our dense correspondence-based method closely reconstructs the ground truth with a similarity transformation-based motion pattern.

In our implementation, we use the dominant motion patterns extracted from both translation and similarity transformation. The reason is that motion modes from offset histograms is complementary to motion patterns from similarity transformations, i.e. translation models can more robustly identify motions on small independently moving objects and covers motions unexplainable with the set of estimated similarity transformations. We also tried to complement our motion models with affine transformation, but we found that it is quite sensitive to errors in the original nearest neighbor field due to its increased DOF.

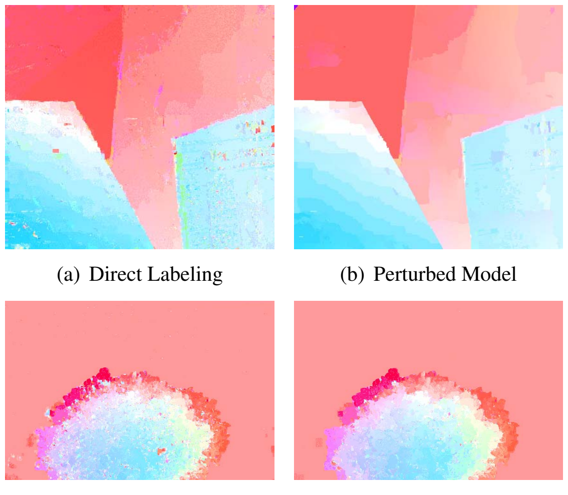
(a) Direct Labeling (b) Perturbed Model (c) Direct Labeling (d) Perturbed Model

Figure 3. The Effect of Motion Pattern Perturbation. Motion segmentation result on the Venus and Hydrangea examples [4]. (a)(c) motion estimation *without* motion pattern perturbation and (b),(d) *with* motion pattern perturbation.

2.3. Local Deformations

The set of dominant motion patterns can reconstruct the ground truth motion field quite well in some cases, but when there exists non-rigid transformation or local deformation, it is obvious that the dominant motion patterns alone are not sufficient to well reconstruct the underlying field. To deal with this problem, we allow a small perturbation around each motion pattern.

We define the set $\Omega(\mathbf{u}) := \{\mathbf{u}' | \ ||\mathbf{u}' - \mathbf{u}||_{L^2} \leq \epsilon\}$ as a small perturbation around motion pattern $\mathbf{u}$. Then, for each dominant offset $\mathbf{u} \in \{\mathbf{u}_1, ..., \mathbf{u}_K\}$ or motion pattern $\mathbf{u} \in \{P_1 \circ \mathbf{x}, ..., P_J \circ \mathbf{x}\}$, we choose $\mathbf{u}' \in \Omega(\mathbf{u}_i)$ that achieves the smallest matching error $|I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}')|$, where I_1 and I_2 are the input images, and $\mathbf{x}$ denotes a location in I_1.

This perturbation step is essential in improving the quality of motion segmentation using dominant motion patterns. It can be regarded as a kind of relaxation where we allow each motion pattern to vary locally. An example is shown in Figure 3 to demonstrate the advantages of local perturbation on regularizing motion fields and obtaining more accurate motion segmentation.

Another benefit of the perturbation model is that we can use a compact set of motion patterns to represent a much wider range of motions: since all $\mathbf{u} \in \{\mathbf{u}_i\}$ is perturbed from a single motion pattern, we greatly reduce the number of dominant patterns required to describe the underlying motion. The time complexity of the motion segmentation step (described in the following) is super-linear to the number of motion patterns, so we typically achieve at least 4X speed up by allowing perturbation (e.g. choosing $\epsilon = 1$).

2.4. Motion Segmentation with Dominant Motion Patterns

Given the set of K candidate motion patterns and their perturbation models, we formulate the dense motion estimation procedure as a labeling problem:

$$E(\mathbf{u}) = \sum_{\mathbf{x}} \Phi_D(I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}(\mathbf{x}))) + \sum_{(\mathbf{x},\mathbf{x}')|\mathbf{x}\in N(\mathbf{x}')} \Phi_S(\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}')) \quad (1)$$
$$s.t. \quad \mathbf{u}(\mathbf{x}) \in \{\Omega(\mathbf{u}_1), ..., \Omega(P_1 \circ \mathbf{x}), ...\},$$

where $N(\mathbf{x})$ refers to 4-connected neighbors of $\mathbf{x}$; Φ_D and Φ_S are robust functions of data consistency and motion smoothness. The penalty of assigning $\mathbf{u}(\mathbf{x})$ to the concept $\mathbf{u}_i$ is defined as the minimum matching error in $\Omega(\mathbf{u}_i)$:

$$\Phi_D(\mathbf{u}_i) := \min_{\mathbf{u}'\in\Omega(\mathbf{u}_i)} \Phi(I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}'(\mathbf{x}))) \quad (2)$$

and the edge preserving motion smoothness is defined as:

$$\Phi_S(\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}')) := w(\mathbf{x})\Phi(\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}')) \quad (3)$$

where we use the edge-preserving term $w(\mathbf{x}) = \exp(-||\nabla I_1||^{\kappa})$ as in [3, 26]. In both Eqn (2) and (3), we choose the slightly non-convex robust influence function $\Phi(s) = (s^2 + \epsilon^2)^{\alpha}$ (also named as generalized Charbonnier penalty) with $\alpha = 0.45$, which has been proved to work well in motion analysis [23].

Directly optimizing Eqn (1) is a NP-hard problem, and the sub-modularity makes it even more challenging. Accordingly, we optimize it by a two-stage fusion process which approximates the global minimum. First, we estimate the motion configuration that best explains the data by choosing from motion patterns obtained from translation and similarity transformation separately as:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \hat{E}(\mathbf{u}) \quad s.t.\ \mathbf{u}(\mathbf{x}) \in \{\Omega(\mathbf{u}_1), ..., \Omega(\mathbf{u}_\mathbf{K})\} \quad (4)$$

and

$$\mathbf{u}^{**} = \arg\min_{\mathbf{u}} \hat{E}(\mathbf{u}) \quad s.t. \quad \mathbf{u}(\mathbf{x}) \in \{\Omega(P_1 \circ \mathbf{x}), ..., \Omega(P_J \circ \mathbf{x}))\}, \quad (5)$$

where the energy function $\hat{E}$ takes the same form as E in Eqn (3) but with a smoothness term on the motion pattern type rather than actual flow vectors. We use multi-label graph-cut [8] to optimize the above two equations. This step is equivalent to solving for motion segmentation with the two motion models separately, which is reasonable given that motion patterns from these two models are estimated independently so they have overlaps.

Then, we apply a fusion algorithm to adaptively choose between $\mathbf{u}^*$ and $\mathbf{u}^{**}$ to obtain the final result $\hat{\mathbf{u}}$ as:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} E(\mathbf{u}) \qquad s.t. \qquad \mathbf{u} \in \{\mathbf{u}^*, \mathbf{u}^{**}\} \quad (6)$$

Due to its sub-modular condition, we apply a QPBO fusion [10] similar to [26].

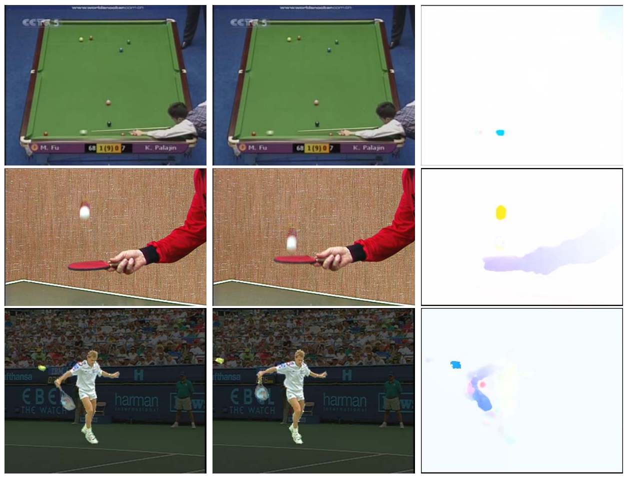

Figure 4. Large Motion of Small Objects. The first and second columns show the input frames. The third column shows that our flow result captures the motion of the fast moving, motion-blurred, textureless balls in each of these three examples.

2.5. Continuous Flow Refinement

For generating final optical flow with sub-pixel accuracy, we need final continuous refinement. We achieve this by simply initializing the motion field with $\hat{\mathbf{u}}$ and estimate the sub-pixel motion field by a continuous optical flow framework [23].

3. Experimental Results

Regarding the running time, NNF takes about 2s; RANSAC similarity transformation takes 20s; the multi-label graph-cut takes about 60s; the final continuous refinement takes 240s. The whole program takes 362s to compute a high quality flow field for an image pair with resolution 640×480 in, for instance, the Urban sequence. Unless otherwise noted, our flow results in this section include the final continuous flow refinement step.

Validation of Large Motion Handling

Our first experiment is to validate our NNF-based flow approach for handling large motions. Figure 4 shows that we can capture large motions of small textureless objects, namely the pool ball, the ping pong ball, and the tennis ball. Although all three balls are also heavily motion-blurred and there is no overlap of the balls between the frames, our NNF-based method captures their large, fast motions accurately. Feature-based methods will have problems here because there are no reliable features to track on the balls. Pyramid-based methods that rely on a small motion assumption will also have problems because the balls will be heavily blurred at the pyramid level for which the small motion assumption holds.

Figure 5 also shows large motion tests on the Middlebury benchmark. We can see that LDOF [9] fails on some regions

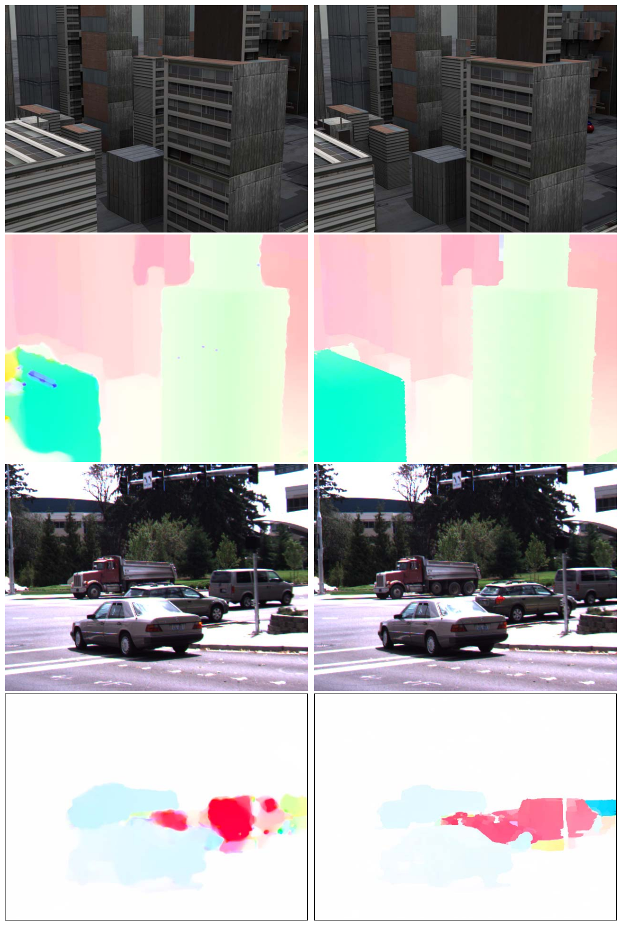

Figure 5. Large Motion on Middlebury [4]. (row 1) Urban input frames. (row 2) LDOF flow [9] (left), Our flow (right). (row 3) Truck input frames. (row 4) LDOF flow (left), Our flow (right).

due to ambiguity of repetitive patterns and occlusions, and direct initialization by wrong sparse matching leads to noisy results. In contrast, our algorithm achieves accurate motion estimation by fusion of different motion patterns.

Figure 6 also shows large motion tests. But these examples have large motions of large objects, namely the umbrellas and the tree. Figure 6 (c),(h) show the noisy NNFs that nonetheless capture the dominant large motions in the scenes. Our flow results in Figure 6 (d),(i) show that we can compute these large motions of large objects quite effectively from the NNF and dominant translations (not shown). As shown in Figure 6(e),(j), the state-of-the-art MDPOF method [26] fails to capture many of the motions of the fast moving, low texture umbrellas and has difficulties at the motion discontinuity on the left side of the tree.

Further Qualitative Evaluation

Recall that our method computes the flow directly at the full input resolution. Although we focus on computing large motions, this property of our algorithm also aids in capturing fine motion details. We show the advantage of *direct finest-scale operation* in Figure 8. In (a) and (b) we show the Schefflera sequence at different scales. A direct coarse-to-fine framework is at risk of missing motion details, due to being trapped in a local minimum during previous iteration as shown in Figure 8(c); in comparison, our method is able to preserve motion details, benefiting from use of the NNF and handling motion estimation in the finest resolution, as shown in Figure 8(c,d,e).

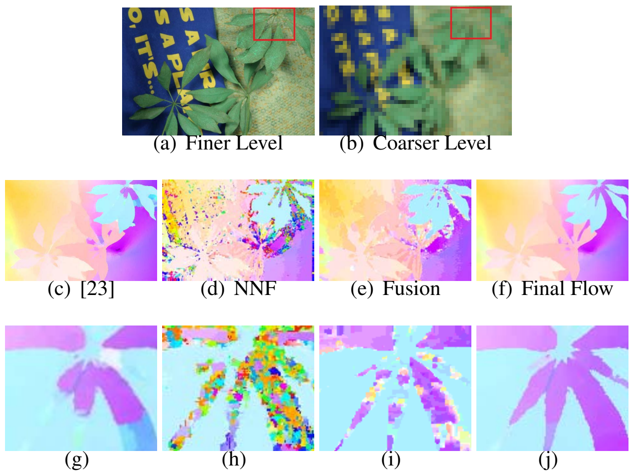

Figure 8. An example of motion detail preserving problem. (a,b) Schefflera sequences at different scales; (c) classical coarse-to-fine method[23] tends to miss some structures; (d) NNF (e) discrete result motion $\hat{\mathbf{u}}$; (f) refined flow. (g)(h)(i)(j) are close-up views of the red box region in (c)(d)(e)(f) respectively.

To further evaluate our approach, we apply our algorithm on the Middlebury benchmark [4]. Some results are shown in Figure 7. As we can see, for sequences in the first column, we estimate motions with the translation modes to get $\mathbf{u}^*$ and the similarity modes to get $\mathbf{u}^{**}$ in the third and fourth column, and fuse them adaptively to obtain $\hat{\mathbf{u}}$ as in fifth column. The fused result is quite consistent with ground truth (second column), and is an improvement over using only translations or only similarity transformations.

Close scrutiny reveals that for deforming objects (e.g., the cloth in the sequence "Demetrodon") the dominant offset model $\mathbf{u}^*$ performs much better, and for rigid-body motion (e.g. the cylinder in the sequence "RubberWhale") $\mathbf{u}^{**}$ generally outperforms $\mathbf{u}^*$. The final fusion step adaptively selects the "best" model between $\mathbf{u}^*$ and $\mathbf{u}^{**}$, and obtains quite satisfactory results. For example, the fused Rubber Whale flow in the fifth column uses the small hole flows in the letter D and the fence from the translation result and the rotating cylinder flow from the similarity result. The translation flow roughly approximates the cylinder's large rotation with several translations.

Moreover, we find that $\hat{\mathbf{u}}$ can also be served as a good "initialization" for further continuous refinement. We show some results in the last column of Figure 7.

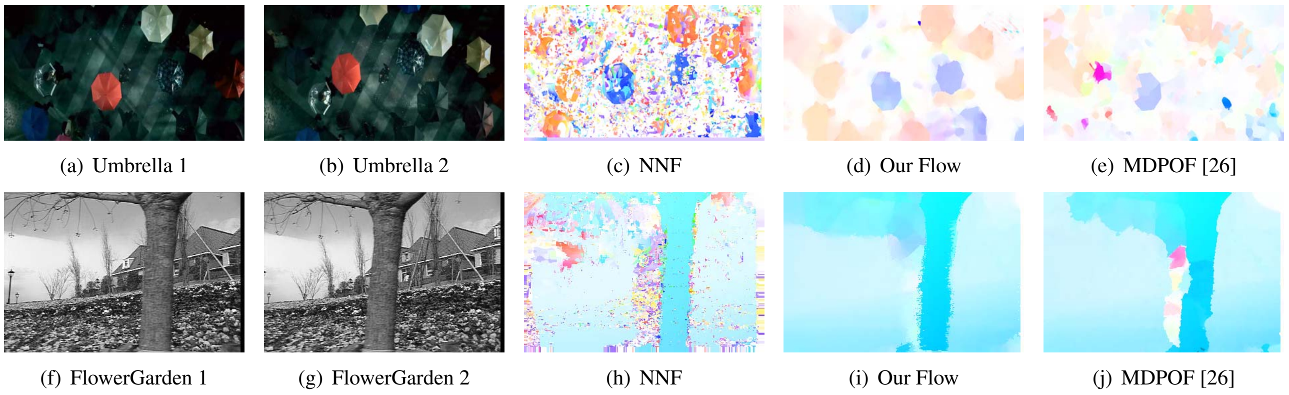

Figure 6. Large Motion of Large Objects. An example result of our method compared to the state-of-the-art optical flow estimation algorithms. (a,b) two frames in the Movie "Resident Evil: Afterlife". (c) The flow implied by the Nearest Neighbor Field. (d) Our final flow after continuous refinement. (e) the state-of-the-art algorithm [26] fails on the fast moving umbrellas. The second row shows the same quantities for the FlowerGarden frames (f),(g). Note that, although noisy as a flow field, the NNF in (h) captures the dominant motions in this scene. (i) Our method produces a good flow result. (j) MDPOF [26] struggles at the large motion discontinuity to the left of the tree.

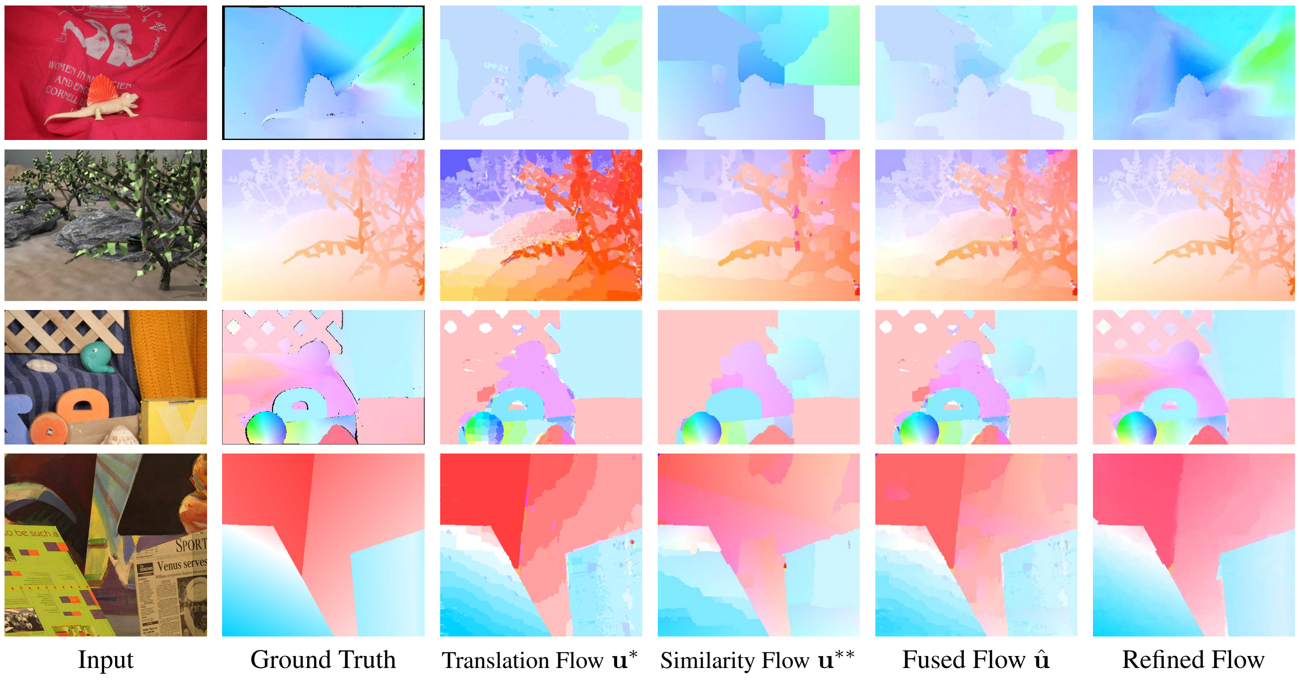
Input
Ground Truth
Translation Flow $\mathbf{u}^*$
Similarity Flow $\mathbf{u}^{**}$
Fused Flow $\hat{\mathbf{u}}$
Refined Flow

Figure 7. Our flow results on the Middlebury benchmark [4]. (1st row) Dimetrodon, (2nd row) Grove3, (3rd row) RubberWhale, (4th row) Venus2. First column: image sequences; second column: ground truth; third column: flow $\mathbf{u}^*$ by translation in Eqn (4); fourth column: similarity flow $\mathbf{u}^{**}$ in Eqn (5); fifth column: fusion result $\hat{\mathbf{u}}$ by Eqn (6); last column: further refined optical flow initialized with $\hat{\mathbf{u}}$.

Quantitative Evaluation

Finally, we quantitatively evaluate our algorithm on the Middlebury benchmark database. In Table 1, we listed the Average End-point Error and Average Angle Error (AAE) of our algorithm. At the time of publishing, our method achieves state-of-the-art quantitative results on the benchmark which contains small motions as well.

4. Conclusion

In this work, we introduce a novel PatchMatch flow method for motion estimation. By exploiting the statistics of ground truth and PatchMatch correspondences, we find that they are *approximately consistent*. Based on this observation, we extract dominant offsets and rigid-body motion modes. Then, we propose to solve the discrete optimization problem by multi-label graph-cut.

The obtained motion can be served as a good start-

	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
AEPE	0.07	0.15	0.18	0.10	0.41	0.23	0.10	0.34
AAE	2.89	2.10	2.27	1.58	2.35	1.89	2.43	1.01

Table 1. Experimental results of our algorithm on Middlebury test set [4].

ing point of further continuous flow refinement. Our experiments on the Middlebury benchmark database clearly show that our approach can achieve very satisfactory performance.

There are still some limitations of our work. Although direct dense matching provides pixel-wise motion candidates, raw patch feature is not robust to large appearance variations such as changes of radiation condition, as well as scaling and rotation. Sometimes repetitive patterns also cause problems in PatchMatch process since ambiguity could occur. Our next step will focus on a generalized PatchMatch [11] to handle these problems.

5. Acknowledgement

This work is done partially when the first author was an intern at Adobe. This work was supported in part by National Science Foundation grant IIS-0916607, IIS-1217302, US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504, and DARPA Award FA 8650-11-1-7149.

References

[1] L. Alvarez, J. Weickert, and J. Sanchez. Reliable estimation of dense optical flow fields with large displacements. *IJCV*, 39(1):41–56, 2000.

[2] Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2(3):283–310, 1989.

[3] W. Andreas, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. *ICCV*, 2009.

[4] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM SIGGRAPH*, 24, 2009.

[6] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: Patchmatch belief propagation for correspondence field estimation. In *BMVC*, 2012.

[7] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, 1996.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[9] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–513, 2011.

[10] R. Carsten, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. *CVPR*, 2007.

[11] B. Connelly, E. Shechtman, D. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. *ECCV*, 2010.

[12] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. In *Computer Vision, Graphics, and Image Processing*, 1988.

[13] K. He and J. Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. *CVPR*, 2012.

[14] K. He and J. Sun. Statistics of patch offsets for image completion. *ECCV*, 2012.

[15] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 16:185–203, 1981.

[16] W. Josh, S. Agarwal, and S. Belongie. What went where. *CVPR*, 2003.

[17] S. Korman and S. Avidan. Coherency sensitive hashing. *ICCV*, 2011.

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

[19] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.

[20] F. Martin and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[21] I. Olonetsky and S. Avidan. Treecann - k-d tree coherence approximate nearest neighbor algorithm. In *ECCV*, 2012.

[22] F. Steinbruecker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *ICCV*, 2009.

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

[24] D. Sun, E. B. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *NIPS*, 2010.

[25] M. Unger, M. Werlberger, T. Pock, and H. Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *CVPR*, 2012.

[26] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *PAMI*, 16(9):1744–1757, 2012.