Unsupervised Flow Refinement near Motion Boundaries

Shuzhi Yu
shuzhiyu@cs.duke.edu
Hannah Halin Kim
hannah@cs.duke.edu
Shuai Yuan
shuai@cs.duke.edu
Carlo Tomasi
tomasi@cs.duke.edu

Duke University
Durham, NC, USA

Abstract

Unsupervised optical flow estimators based on deep learning have attracted increasing attention due to the cost and difficulty of annotating for ground truth. Although performance measured by average End-Point Error (EPE) has improved over the years, flow estimates are still poorer along motion boundaries (MBs), where the flow is not smooth, as is typically assumed, and where features computed by neural networks are contaminated by multiple motions. To improve flow in the unsupervised settings, we design a framework that detects MBs by analyzing visual changes along boundary candidates and replaces motions close to detections with motions farther away. Our proposed algorithm detects boundaries more accurately than a baseline method with the same inputs and is shown to improve estimates from different flow predictors without additional training.

1 Introduction

Optical flow estimation is an important problem in computer vision as it enables high-level tasks such as motion segmentation [22, 34], action recognition [38], and object tracking [45]. Unsupervised prediction has been attracting increasing attention [19, 24, 25, 30, 41, 46] since annotating real video is both expensive and difficult [50], and it is still not clear how well synthetic video can simulate real data. The state-of-the-art *average* End-Point Error (EPE) of both unsupervised [24, 41] and supervised [43, 44] estimators on benchmark datasets has been decreasing ever since the introduction of Convolutional Neural Networks (CNNs) for this problem. The *maximum* EPE of the estimated flow, on the other hand, is typically much larger and occurs mainly along Motion Boundaries (MBs).

To illustrate the performance degradation near MBs, Figure 1 (left) shows the performance of top flow estimators, both unsupervised and supervised, stratified by pixel distance to the closest MB on MPI-Sintel. The EPE increases with decreasing distance from a boundary, regardless of whether supervision is available. Fundamentally, estimating motion near

MBs is harder than elsewhere. First, flow is discontinuous across MBs, while typical estimators assume smoothness. Second, the features matched to find point correspondences between frames have wide receptive fields [9, 16, 43, 44] and straddle MBs when they are near them. Appearance on the two sides of a boundary typically changes in different ways from frame to frame, because of the different motions. As a consequence, feature correspondences across frames are often poor along boundaries, and this often results in poor flow estimates.

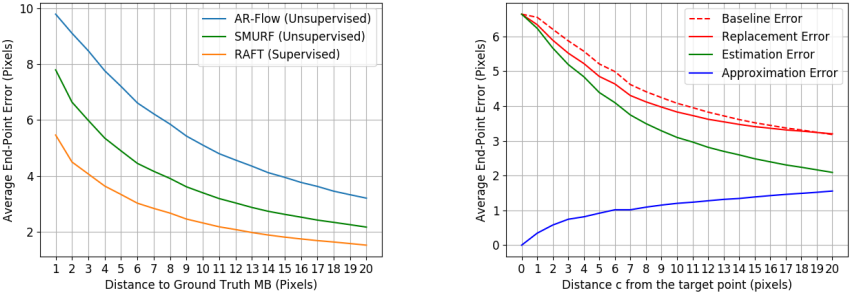

Figure 1: Left: EPE, flow estimation error $e = \|\hat{F} - F\|$ ($\hat{F}$ is the estimate, F is true flow), for a top supervised flow estimator, RAFT [44] (*orange*), and two top unsupervised flow estimators, AR-Flow [24] (*blue*) and SMURF [41] (*green*) versus distance to the closest true MB, averaged on all of MPI Sintel (clean). Estimates worsen near MBs. Right: Plots of the two error sources for replacement flow. Every point $\mathbf{p}$ that is 2 pixels away from a true MB is a replacement target point. If $\mathbf{b}$ is the closest MB point to $\mathbf{p}$, let a unit vector $\mathbf{u}$ point from $\mathbf{b}$ to $\mathbf{p}$. A point $\mathbf{q} = \mathbf{p} + c\mathbf{u}$, for $c \in [0, 20]$, is used for this plot as long as every point in the line segment from $\mathbf{p}$ to $\mathbf{q}$ is within the frame and closer to $\mathbf{b}$ than to any other MB point. For these $\mathbf{p}, \mathbf{q}$ pairs, the plot shows the estimation error $e = \|\hat{F}(\mathbf{q}) - F(\mathbf{q})\|$ (*green*), the approximation error $a = \|F(\mathbf{q}) - F(\mathbf{p})\|$ (*blue*), and the replacement error $r = \|F(\mathbf{p}) - \hat{F}(\mathbf{q})\|$ (*solid red*) averaged on all of MPI-Sintel (clean) and with $\hat{F}$ values from SMURF [41]. Flow replacement is favorable (the solid red line is under the EPE at $\mathbf{p}$, *dashed red*) over a wide range of values for c.

In this paper, we propose a method to detect MBs and refine optical flow near them *without supervision*. Our method first detects MBs from the input flow estimated by an existing unsupervised estimator. Near these boundaries, it improves each flow estimate by replacing it with the flow a bit farther away from the MB. The possible improvement on the *average* EPE over the whole image is bounded by the fact that MBs are a small fraction of any image: Only around 1% of all the pixels in the dataset used in Figure 1 are on true MBs. However, improvements in the *maximum* EPE are important for applications that require clean MBs and accurate flow estimates near them. For example, accurate MBs would help sharpen the segmentation boundaries of moving objects in video object segmentation [5, 28] and help prevent color bleeding in the color propagation of moving objects in video editing [33]. In addition, recent video interpolation methods require flow estimates as input and interpolation results degrade near MBs because of poor flow estimates near them [2, 35].

There has been little work explicitly detecting MBs [1] and improving the flow near them without supervision [18, 21, 48]. Typically, a *baseline method* detects MBs by thresholding

the magnitude of the flow gradient. However, results are often poor because flow estimates are both inaccurate and smooth near MBs. We show that accurate prediction of MBs from imperfect flow estimates also helps improve flow estimates near MBs. Our proposed MB detection method uses hysteresis thresholding [6] on maps of flow gradient magnitude, image edge maps, and on novel maps we propose in this paper. These new maps identify locations in the input flow map where MBs are likely to exist, based on the observation that *changes of appearance in the foreground and background on the two sides of a MB are more consistent with their own motion than with the motion on the other side.* Thus, given a point on a MB candidate, we consider two points nearby, one on each side of the boundary, and we measure how the appearance of each point would change when subjected in turn to the motions measured at either of them. If the candidate is away from a boundary, all four combinations of point and motion typically yield good matches between frames. If the candidate is on a boundary, at least two combinations often yield poorer matches.

When replacing flow values near boundaries with values farther away, we face two contrasting sources of error: The *approximation error* comes from the fact that motion measured at one pixel replaces motion measured elsewhere. This error decreases as the replacement motion is taken closer to the replacement candidate. The *estimation error* stems from the fact that even the replacement estimate is not exact. This error increases closer to MBs, where flow estimates degrade. Figure 1 (right) shows plots of these two errors. This trade-off explains why the replacement method cannot make improvement away from MBs, where the estimation error is flat. Even near MBs, not every point can improve through replacement. We observe that the difference in the flow values on the two sides of the MBs is a useful indicator of which points can benefit from replacement.

Empirically, our MB detection method improves over the baseline methods both quantitatively and qualitatively. Our replacement algorithm improves flow at promising candidate points near MBs when compared to the state-of-the-art unsupervised flow estimators on both synthetic and real video benchmarks. We also analyze various properties of flow estimates near MBs. To the best of our knowledge, this is the first work that specifically improves and analyzes MB detection and nearby flow estimates in the unsupervised setting.

2 Related Work

Flow Estimation and Refinement Flow estimation has been studied for a long time [3, 7, 11, 12, 26, 47], culminating with recent work with CNNs [9, 13, 14, 15, 16, 17, 25, 36, 43, 44]. Recently, vision transformers and attention mechanisms have also led to better results [42, 49]. Supervised CNN methods are trained on datasets like MPI Sintel [9] or KITTI [10]. The aperture problem requires regularization, and all systems assume a smooth flow, either explicitly or implicitly and either during inference (classical methods) or during training (deep learning). State-of-the-art methods [44] achieve good average sub-pixel accuracy, but predictions are typically at a quarter or even an eighth of the original resolution because of computation cost. Final predictions are then up-sampled, again assuming smoothness. Also, features in these systems have wide receptive fields, with the negative implications discussed earlier. As a result, the maximum EPE tends to be quite large (Figure 1) even when the average EPE is small, a problem that has so far attracted little attention.

Since annotating realistic flow datasets is hard and expensive, many unsupervised CNN methods [19, 24, 25, 30, 41, 46] have been proposed since the pioneer work by Yu *et al*. [20] and Ren *et al*. [37]. The top unsupervised networks often evolve from the best supervised

ones. The two top unsupervised estimators used in our experiments are AR-Flow [24] and SMURF [41], and are based on the top supervised networks PWC-Net [43] and RAFT [44] respectively. While achieving good average EPEs on benchmarks, the unsupervised flow estimators also degrade in performance near MBs.

Our method explicitly detects MBs and refines the flow near them without requiring motion labels. A related unsupervised method [27] additionally predicts a so-called interpolation flow and weight for each pixel. The interpolation flow guides where to find the replacement flow and the weight indicates the contribution of the replacement to the final prediction. Another supervised method [14] also replaces low-confidence flow values by nearby higher-confidence flow values under certain conditions. However, these methods end up replacing flow values far from MBs as well, which can be problematic.

It would seem at first that methods that sharpen depth maps near edges [39] can be adapted to correcting flow estimates near MBs. However, improving flow near MBs is different from simply sharpening the MBs, as it requires identifying both the poor flow estimates and good flow substitution candidates. So it is not clear how this adaptation would work.

Some flow estimators use additional frames as input and achieve better flow estimates in occluded areas (which are different from, but closely related to MBs) [19, 24, 25]. However, estimation errors still increase with decreasing distance from MBs. Our method uses 3 frames to detect MBs, which none of these estimators detect, and specifically refine flow estimates near them. Thus, our motion refiner can also benefit these models near MBs.

Motion Boundary Detection Thanks to new datasets with MB labels, recent supervised approaches use machine learning algorithms to detect MBs. LMDB [48] uses structured random forests [8] that take as inputs two consecutive images, forward and backward optical flow estimates, and image warping errors. More recent approaches use multi-task learning. Lei *et al*. propose a fully convolutional Siamese network that jointly estimates both object boundaries and the motion of the pixels on them [23]. Ilg *et al*. [18] simultaneously estimate occlusions, depth boundaries, MBs, optical flow, disparities, motion segmentation, and scene flow (FlowNet-CSS). The state-of-the-art approach, MONets [21], jointly predicts MBs and occlusions by explicitly exploiting the relationship between the two tasks.

These supervised methods require ground-truth flow. In contrast, our MB detection method is unsupervised. Alhersh and Stuckenschmidt [1] work in a similar direction. They propose to use unsupervised loss to fine-tune the pre-trained flow estimators and thereby improve the detection of MBs based on the gradient of the estimated optical flow field. In comparison, we do not use any supervised pre-trained model and require no additional training. In addition, our approach detects MBs first and then improves flow estimates near them.

3 Method

Our method takes as input three consecutive video frames I_1, I_2, and I_3 and two optical flow maps $\hat{F}_{21}$ (from I_2 to I_1) and $\hat{F}_{23} \in \mathbb{R}^{h\times w\times 2}$ (from I_2 to I_3) estimated using any unsupervised flow predictor. We first detect MBs $\hat{B}_{23} \in \{0,1\}^{h\times w}$ from frame I_2 to I_3. We then identify points $\mathbf{p}$ near boundaries whose flow estimates can be potentially improved by replacing them with those at nearby points $\mathbf{q}$. These replacements yield a refined map $\hat{F}^r_{23} \in \mathbb{R}^{h\times w\times 2}$. We now explain the components of boundary detection and flow refinement [1].

[1] Code is available at https://github.com/pszyu/unsupervised-mb-flow-refinement.

3.1 Motion Boundary Detection

MBs are detected by hysteresis thresholding [6] from three feature maps, all in $\{0,1\}^{h\times w}$: The image edge map M_e, the motion discrepancy map M_{md}, and our proposed map of invalid smooth motion M_{ism}. Specifically, M_e is computed by the DexiNed [40] edge detector in our experiments. M_{md} is obtained by thresholding the magnitude of the flow gradient. M_{ism} complements M_{md} by also flagging pixels where the smooth motion in the estimated flow map is unlikely to be correct, as explained below. The detector based on these maps is described in Section 3.1.2.

3.1.1 Invalid smooth motion map

Flow estimators tend to over-smooth their predictions across MBs, and the M_{ism} map flags areas where the spatial smoothness of flow estimates may be unwarranted. The map is computed by analyzing two patches, one on each side of MB candidates. Let $\mathbf{b}$ be a point in frame I_2 where the gradient $\mathbf{g}$ of image brightness is nonzero and let $\mathbf{u}$ be either of the two unit vectors parallel to $\mathbf{g}$. MB typically align with image edges, so if $\mathbf{b}$ is on a MB β, then the following two points are likely on opposite sides of β:

$$\mathbf{a} = \mathbf{b} + \sigma\mathbf{u} \quad \text{and} \quad \mathbf{c} = \mathbf{b} - \sigma\mathbf{u}\,. \tag{1}$$

We use $\sigma = 5$ in our experiments and order $\mathbf{a},\mathbf{c}$ so that $\mathbf{a}$ has the smaller EPE.

Our proposed method for detecting the invalid smooth motion is based on the observation that the flow estimates on one side of MB are often much better than those of the other side. To illustrate, Figure 2 shows scatter plots of the EPEs of all the $\mathbf{a},\mathbf{c}$ pairs for the whole Sintel sequence "alley_2". The point clouds have long vertical tails in both the clean (left) and final (right) passes. For instance, in this specific sequence (clean), the EPE of flow estimates for 54% of the true MB points is sub-pixel at $\mathbf{a}$ but not at $\mathbf{c}$. Moreover, the asymmetry is 5 pixels per frame or greater in about 36% of these asymmetric cases. The statistics are similar for the final pass. One reason for this asymmetry is that flow estimates tend to be poor on the background side of occluding MBs, where points in one frame have no match in the other.

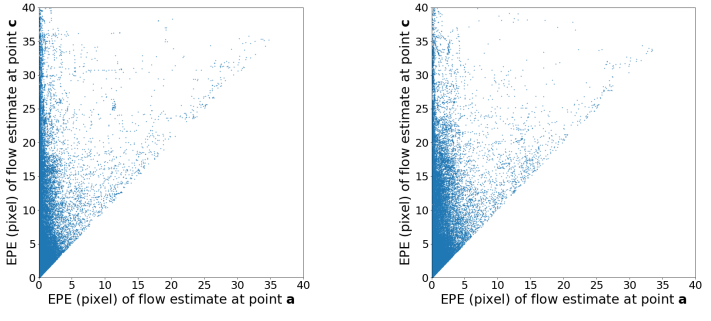

Figure 2: The scatter plots of the EPEs of point $\mathbf{a}$ (x-axis) and $\mathbf{c}$ (y-axis), as defined in Equation 1, for all the true MB points in the Sintel sequence "alley_2" clean (left) and final (right) pass. Without loss of generality, $\mathbf{a}$ has smaller EPE.

Let $\mathbf{p}$ be either $\mathbf{a}$ or $\mathbf{c}$. The appearance change resulting from matching point $\mathbf{p}$ in frame I_i to point $\mathbf{p}+\mathbf{v}$ by motion $\mathbf{v}$ in frame I_j is measured by the matching cost $c_{ij}(\mathbf{p},\mathbf{v}) =$

$-s_{ij}(\mathbf{p}, \mathbf{p}+\mathbf{v})$ where $s_{ij}(\cdot)$ is the Pearson correlation between the features of its arguments, *i.e.*, $s(\mathbf{p},\mathbf{q}) = \mathbf{f}_\mathbf{p}^T \mathbf{f}_\mathbf{q} / (||\mathbf{f}_\mathbf{p}|| \cdot ||\mathbf{f}_\mathbf{q}||) \in [-1,1]$. In our experiments $\mathbf{f}_\mathbf{p}$ is the vector of the RGB values of a 3×3 patch around $\mathbf{p}$, centered by subtracting the mean patch color.

Matching costs under forward motion For simplicity, let $\hat{F} = \hat{F}_{23}$. Without knowing which side the EPE is smaller, if we use $\hat{F}(\mathbf{a})$ or $\hat{F}(\mathbf{c})$ to match either $\mathbf{a}$ or $\mathbf{c}$ in frame I_2 to a point in frame I_3, we get four cost measurements $m_{\mathbf{aa}}, m_{\mathbf{ac}}, m_{\mathbf{ca}}, m_{\mathbf{cc}}$, where $m_{\mathbf{ac}} = c(\mathbf{a}, \hat{F}(\mathbf{c}))$ and so forth (Figure 3 (a)).

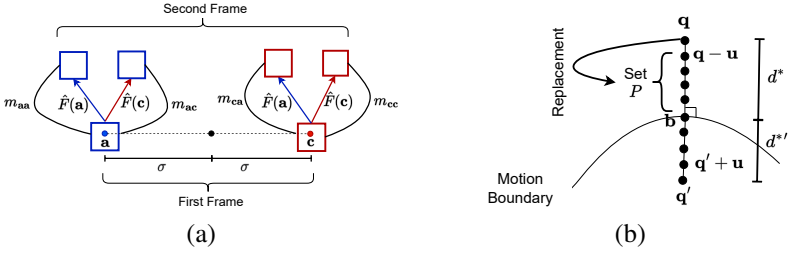

Figure 3: (a) The four matching costs that are used in checking whether the smooth motion across a point is valid. The blue and red arrows represent the estimated motion of the blue and red patches respectively. (b) Notation for the replacement algorithm. The line through $\mathbf{q}$, $\mathbf{b}$, $\mathbf{q}'$ is perpendicular to the boundary at $\mathbf{b}$, and parallel to unit vector $\mathbf{u}$. Only the estimated flow of one side of the boundary, in set P, may be replaced by $\mathbf{q}$.

Away from MBs and with no occlusion, the estimated flow is often accurate. Since flow is smooth there, we typically have $m_{\mathbf{aa}} - m_{\mathbf{ca}} \leq \theta_{ism}$ for some pre-defined small threshold θ_{ism}. Similarly, $m_{\mathbf{ac}} - m_{\mathbf{cc}} \leq \theta_{ism}$. However, occlusions may also occur away from MBs in the presence of large motions [21]. In that case, the inequalities above are likely to hold with *backward motion*, as explained later.

Near MBs, and where the EPE is asymmetric as explained earlier, let $\mathbf{a}$ be the point with the smaller EPE, without loss of generality. We assume that the matching cost of $\mathbf{c}$ is large following the motion of $\mathbf{a}$, *i.e.* $m_{\mathbf{ca}}$ is large. On the other hand, $m_{\mathbf{aa}}$ is likely small. Similar considerations follow when $\mathbf{a}$ and $\mathbf{c}$ are switched, and we accordingly define the map

$$M_{ism}(\mathbf{b}) = \max\{m_{\mathbf{ac}} - m_{\mathbf{cc}},\ m_{\mathbf{ca}} - m_{\mathbf{aa}}\} > \theta_{ism}\ . \tag{2}$$

There will be false negatives if the flow estimates are inaccurate on both sides of $\mathbf{b}$, and false positives if there is a large matching cost away from MBs. Using backward motion and hysteresis thresholding, as explained next, will mitigate these false predictions.

Adding backward motion Suppose the motion is smooth between frames 1 and 3. An additional frame I_1 is helpful to the detection of the MB because an occluding MB forward in time is a dis-occluding MB in reverse time, and flow near dis-occlusions are often more accurate than near occlusions. We therefore redefine the cost as follows:

$$\tilde{m}_{\mathbf{xy}} = \min\{c_{23}(\mathbf{x}, \hat{F}_{23}(\mathbf{y})), c_{21}(\mathbf{x}, \hat{F}_{21}(\mathbf{y}))\}\ . \tag{3}$$

3.1.2 Detection algorithm

We use a notion similar to hysteresis thresholding [6] to combine the edge map M_e, motion discrepancy map M_{md}, and invalid smooth motion map M_{ism}. Specifically, points for which

M_{md} is true are classified as *strong MBs*. A point is classified as *weak* MB where M_{md} is false and both M_e and M_{ism} are true. Other points are classified as non-motion-boundary points. A weak MB point becomes a strong one if it is spatially connected with some strong MB points. The final strong MB points are the final predictions from the detector.

3.2 Flow Replacement

From the analysis above, replacing flow at $\mathbf{p}$ with flow at $\mathbf{q}$ near a MB is most effective where the improvement from a smaller estimation error at $\mathbf{q}$ trumps the increasing approximation error as $\mathbf{p}$ and $\mathbf{q}$ are taken farther apart. This turns out to occur most often for points $\mathbf{p}$ where true flow is small and true flow opposite the MB is large. At these points $\mathbf{p}$, the approximation error is smaller because the flow is smaller and therefore typically changes less rapidly. The estimation error is smaller as well, because flow predictors do better on smaller motions.

To identify these points, a method is needed to detect points on the two sides of a MB candidate where flow can be estimated well. Since flow is smooth on either side of a MB, most of the variation in flow with the distance $d = \|\mathbf{q}-\mathbf{b}\|$ of $\mathbf{q}$ from the MB is caused by interference from the flow across the MB. Figure 1 (left) shows that this interference tends to drop and saturate as the distance d from the MB increases. Thus, flow estimates tend to change more rapidly for smaller values of d than for larger ones. Formally, let $\hat{f}(d) = \hat{F}(\mathbf{b}+d\mathbf{u})$ for brevity. Then, the change $\delta(d,d+1) = \|\hat{f}(d)-\hat{f}(d+1)\|$ resulting from a one-pixel increment from d to $d+1$ drops and saturates as well. We measure this drop relative to the overall change in flow from distance 1 to distance d, that is, relative to $\delta(1,d) = \|\hat{f}(1)-\hat{f}(d)\|$ and define the *smallest safe distance* d^* to be the smallest distance for which the ratio between $\delta(d,d+1)$ and $\delta(1,d)$ drops below some threshold $\tau \in (0,1)$:

$$d^* = \min\left\{ d \;\middle|\; \frac{\|\hat{f}(d)-\hat{f}(d+1)\|}{\|\hat{f}(1)-\hat{f}(d)\|} < \tau \right\}. \tag{4}$$

We use $\tau = 0.2$ in all our experiments.

Let now $\mathbf{q} = \mathbf{b}+d^*\mathbf{u}$ be the first safe point on one side of the MB, and similarly define a first safe point $\mathbf{q}'$ on the other side. Then, we define the set P containing all the replacement candidates near $\mathbf{b}$ as follows:

$$P = \left\{ \mathbf{p} = \mathbf{b}+d\mathbf{u} \;\middle|\; \underbrace{0<d<d^*}_{\text{close to MB}} \;\&\; \underbrace{\|\hat{F}(\mathbf{q})\| < \|\hat{F}(\mathbf{q}')\|}_{\text{side with smaller flow}} \;\&\; \underbrace{\|\hat{F}(\mathbf{q})-\hat{F}(\mathbf{q}')\| \geq \alpha\|\hat{F}(\mathbf{q})\|}_{\text{large difference across MB}} \right\}, \tag{5}$$

We use $\alpha = 0.2$ in our experiments. The refined flow is (Figure 3 (b))

$$\hat{F}^r(\mathbf{p}) = \hat{F}(\mathbf{q}) \quad \text{if } \mathbf{p} \in P \quad \text{and} \quad \hat{F}^r(\mathbf{p}) = \hat{F}(\mathbf{p}) \quad \text{otherwise.} \tag{6}$$

4 Empirical Results

4.1 Experiment Settings

Datasets: MPI-Sintel [4] dataset provides the optical flow labels for each frame of 23 high resolution synthetic sequences of 20 to 50 frames each in its training set. Fast motion and large occluded areas make this dataset challenging. We follow LDMB [48] to compute the

ground-truth MB labels. **KITTI-2015** [31, 32] is a realistic dataset that is commonly used as a benchmark in flow estimation. However, its training set, consisting of 200 image pairs, only provides sparse ground truth optical flow, and thus accurate true MBs cannot be inferred. We use these two training sets to demonstrate the improvement of our algorithm on the detection of MBs and the flow estimates near them.

Performance evaluation: We evaluate estimated MBs by F_1-score. We follow the literature [18, 48] and use the BSDS benchmark [29] to compute MB detection performance. A prediction is a true positive as long as it is within 0.75% of the image diagonal away from a true MB [18]. Optical flow is evaluated by End-Point Error (EPE).

Flow estimators: All the flow estimators are trained without ground-truth labels, and are provided by their authors [24, 41] (More details in the supplementary material).

Hyper-parameters: The threshold for the motion discrepancy map is set to be 1 for Sintel and 3 for KITTI. For MB detection, we set the threshold θ_{ism} for the maps of invalid smooth motion as 0.2 for Sintel dataset and 0.6 for KITTI. The impact of the hyper-parameters is analyzed in the supplementary material.

4.2 Motion Boundary Detection

In Table 1, we compare our method with the baseline method, *i.e.* map M_{md}, over different input flows to show our method's robustness. The performance is evaluated on both the clean and final passes of the MPI Sintel training set, following the literature [21].

The table shows that the performance of both the baseline method and ours improves with better input flow, except that the baseline method does better with LDOF than with AR-Flow on Sintel clean. Our method consistently outperforms the baseline method across all three flow estimators and on both passes. The improvement ranges from 5.97% (70.3 to 74.5) on Sintel clean with SMURF to 21.09% (53.1 to 64.3) on Sintel clean with AR-Flow. The largest improvements on both passes are with AR-Flow. On one hand, the good performance by the baseline method with SMURF limits the margin for improvement. On the other hand, the estimated flow needs to be accurate at least on one side of a MB for good detection, and thus the inaccurate input flow from LDOF limit the improvement margin as well.

Figure 4 shows two MB detection examples on Sintel (clean). Our method detects some MBs missed by the baseline method (red ovals in columns 2, 5) thanks to the invalid-smooth-motion maps, even if these are noisy (fourth column).

		SMURF [41]			AR-Flow [24]			LDOF [3]		
		Flow (EPE)	BL (F1)	Ours (F1)	Flow (EPE)	BL (F1)	Ours (F1)	Flow (EPE)	BL (F1)	Ours (F1)
Sintel	Clean	2.01	70.3	**74.5**	2.79	53.1	**64.3**	4.18	54.8	**59.2**
	Final	2.87	63.5	**67.4**	3.73	48.5	**57.1**	6.25	46.7	**51.2**

Table 1: F_1-score for **MB estimation** with different input flow estimates, compared with the baseline method (BL). SMURF [41] and AR-Flow [24] are two top unsupervised flow estimators, and LDOF [3] is a top classical flow estimator.

Map ablation study: Maps M_e and M_{ism} (columns 3, 4 in Figure 4) complement each other: The edge map localizes geometry well but does not convey motion information. The map M_{ism} contains motion information but is noisy. Table 2 shows that using either map alone on top of M_{md} (*i.e.* the baseline method) actually worsens performance. Using both of

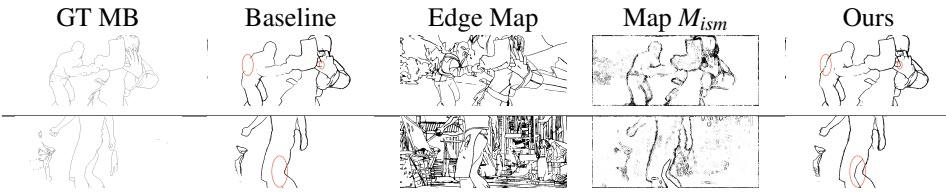

Figure 4: MB detection samples with our method and the baseline on Sintel (clean) with SMURF input. Main differences are highlighted by red ovals.

them with hysteresis improves over M_{md} in both passes of Sintel. The ablation study on the impact of using backward flow on MB detection is shown in the supplementary material.

	Baseline (Map M_{md})	+Map M_e	+Map M_{ism}	Ours (+M_e+M_{ism})
Clean	70.3	39.7	49.7	**74.5**
Final	63.5	42.5	54.2	**67.4**

Table 2: Performance of MB detection (F_1) of our proposed hysteresis scheme with different map combinations. Input flow is estimated by SMURF [41].

4.3 Flow Replacement

Table 3 shows the effects of our flow refinement over the flow estimates on points in the replacement set P. We consistently improve estimates over all datasets.

Figure 5 shows before/after flow quiver plots in two examples. Red vectors are in P. In the Sintel example, the set P is in the background and close to the true MB. In the KITTI example, the set P appears to be moved from the true MB (KITTI has no ground-truth MB labels). However, flow estimates in P are still perturbed by the running car with larger motion. Replacement improves flow estimates in both cases.

Input Flow	Dataset	Input Flow AEPE	Replaced Points			
			% of MB points	Init AEPE	Our AEPE	↓
LDOF [3]	Clean	4.18	51.02	12.81	**10.84**	15.38%
	Final	6.25	33.24	13.68	**11.28**	17.54%
	KITTI	19.63	-	44.95	**43.76**	2.65%
ARFlow [24]	Clean	2.79	48.13	9.52	**7.90**	17.02%
	Final	3.70	34.16	8.96	**7.42**	17.19%
	KITTI	3.46	-	19.29	**18.62**	3.47%
SMURF [41]	Clean	2.03	61.28	5.47	**5.17**	5.48%
	Final	2.90	39.98	5.71	**4.72**	17.34%
	KITTI	1.94	-	15.35	**14.69**	4.30%

Table 3: The average EPE and average EPE improvement with our replacement method near our estimated MBs on the flow estimates by different flow estimators. Note the ARFlow uses 3 frames to estimate the flow. About 1% of all MPI Sintel pixels are true MB points. This information is unknown for KITTI, which has sparse ground truth flow.

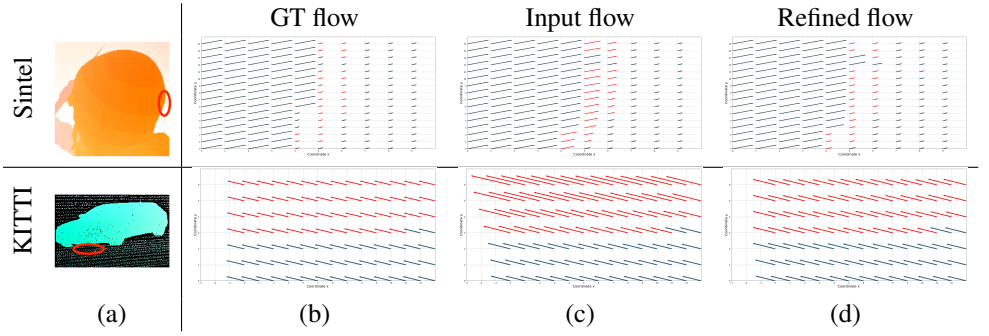

Figure 5: Two examples of flow (quiver plots with flow down-scaled by 70) before and after replacement on set P (red vectors). Input flow is from SMURF [41]. On these two patches, replacement decreases the AEPE from 24.62 to 3.40 pixels per frame for the Sintel example and from 33.23 to 16.87 pixels per frame for the KITTI example.

Impact of MB quality on replaced flow: Table 4 (top half) shows that better MBs lead to better flow replacement on Sintel (clean). This pattern is less clear on the final pass (bottom half), where improvements are comparably large regardless of the quality of the MBs. This is likely because the estimated flow is generally worse on Sintel final, especially near MBs, so there is more room for improvement by replacement.

Dataset		Ours (LDOF)	Ours (AR-Flow)	Ours (SMURF)	GT
Clean	MB (F1)	59.2	64.3	74.5	100.0
	EPE ↓ (%)	0.02	0.75	5.48	7.72
Final	MB (F1)	51.2	57.1	67.4	100.0
	EPE ↓ (%)	17.51	22.50	17.34	20.94

Table 4: Impact of MB quality (by F1) on the performance of flow replacement algorithm on Sintel clean and Sintel final. The flow performance is evaluated on the replacement set P and the input flow is from SMURF [41]. The last column "GT" uses ground-truth MBs.

5 Conclusion

We propose a method that both detects MBs and improves flow estimates near them. The method is plug-and-play and requires no supervision. Fundamentally, it exploits the fact that it may be fruitful to replace a flow vector near a MB with one taken from a pixel that is farther away. This is useful because the error in taking the flow vector from the wrong point is on average smaller than the error caused by proximity to a MB. Figure 1 (right) shows that this balance is favorable on average, and our method exploits that margin fully. Of course, the Figure also shows that the benefit is bounded, and our analysis elucidates the trade-offs. Future work will address methods to tackle the flow estimation error near MBs directly.

Acknowledgements This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract #2021-21040700001; by the National Science Foundation

under Grant No. 1909821; and by an Amazon AWS cloud computing award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, NSF, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] Taha Alhersh and Heiner Stuckenschmidt. Unsupervised fine-tuning of optical flow for better motion boundary estimation. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics (VISIGRAPP)*, 2019.

[2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. doi: 10.1109/TPAMI.2019.2941941.

[3] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011. URL http://lmb.informatik.uni-freiburg.de/Publications/2011/Bro11a.

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conference on Computer Vision*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, 1986.

[7] Haw-Shiuan Chang and Yu-Chiang Frank Wang. Superpixel-based large displacement optical flow. In *2013 IEEE International Conference on Image Processing*, pages 3835–3839. IEEE, 2013.

[8] Piotr Dollár and Lawrence Zitnick. Structured forests for fast edge detection. In *IEEE International Conference on Computer Vision*, 2013. doi: 10.1109/ICCV.2013.231.

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[11] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[12] Yinlin Hu, Yunsong Li, and Rui Song. Robust interpolation of correspondences for large displacement optical flow. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] T. Hui, X. Tang, and C. Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8981–8989, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00936. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00936.

[14] Tak-Wai Hui and Chen Change Loy. Liteflownet3: Resolving correspondence ambiguity for more accurate optical flow estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–184, 2020.

[15] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. URL http://mmlab.ie.cuhk.edu.hk/projects/LiteFlowNet/.

[16] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5747–5756, Long Beach, CA, USA, 2019.

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[18] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *The European Conference on Computer Vision*, September 2018.

[19] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision*, pages 690–706, 2018.

[20] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *Proceedings of the European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[21] Hannah Halin Kim, Shuzhi Yu, and Carlo Tomasi. Joint detection of motion boundaries and occlusions. In *British Machine Vision Conference (BMVC)*, November 2021.

[22] Jens Klappstein, Tobi Vaudrey, Clemens Rabe, Andreas Wedel, and Reinhard Klette. Moving object segmentation using optical flow and depth information. In Toshikazu Wada, Fay Huang, and Stephen Lin, editors, *Advances in Image and Video Technology*, pages 611–623, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-540-92957-4.

[23] Peng Lei, Fuxin Li, and Sinisa Todorovic. Boundary flow: A siamese network that predicts boundary motion without training on motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3290, 2018.

[24] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6489–6498, 2020.

[25] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019.

[26] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981.

[27] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1045–1054, 2021.

[28] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[29] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[30] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[31] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[32] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[33] Simone Meyer, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus H. Gross. Deep video color propagation. In *British Machine Vision Conference (BMVC)*, 2018.

[34] M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. *2013 IEEE International Conference on Computer Vision*, pages 1577–1584, 2013.

[35] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *International Conference on Computer Vision*, 2021.

[36] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2729, 2017.

[37] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[38] Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J. Black. On the integration of optical flow and action recognition. In Thomas Brox, Andrés Bruhn, and Mario Fritz, editors, *Pattern Recognition*, pages 281–297, Cham, 2019. Springer International Publishing. ISBN 978-3-030-12939-2.

[39] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] X. Soria, E. Riba, and A. Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1912–1921, Los Alamitos, CA, USA, mar 2020. IEEE Computer Society. doi: 10.1109/WACV45572.2020.9093290. URL https://doi.ieeecomputersociety.org/10.1109/WACV45572.2020.9093290.

[41] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2021.

[42] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17602–17611, 2022.

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[44] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[45] Mikko Vihlman and Arto Visala. Optical flow in deep visual tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12112–12119, Apr. 2020. doi: 10.1609/aaai.v34i07.6890. URL https://ojs.aaai.org/index.php/AAAI/article/view/6890.

[46] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018.

[47] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision*, Sydney, Australia, December 2013. URL http://hal.inria.fr/hal-00873592.

[48] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Learning to Detect Motion Boundaries. In *Conference on Computer Vision and Pattern Recognition*, Boston, United States, 2015. URL https://hal.inria.fr/hal-01142653.

[49] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022.

[50] Shuai Yuan, Xian Sun, Hannah Kim, Shuzhi Yu, and Carlo Tomasi. Optical flow training under limited label budget via active learning. In *European Conference on Computer Vision (ECCV)*, 2022.