Joint Detection of Motion Boundaries and Occlusions

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

Duke University
Durham, NC, USA

Abstract

We propose MONet, a convolutional neural network that jointly detects motion boundaries (MBs) and occlusion regions (Occs) in video both forward and backward in time. Detection is difficult because optical flow is discontinuous along MBs and undefined in Occs, while many flow estimators assume smoothness and a flow defined everywhere. To reason in the two time directions simultaneously, we direct-warp the estimated maps between the two frames. Since appearance mismatches between frames often signal vicinity to MBs or Occs, we construct a cost block that for each feature in one frame records the lowest discrepancy with matching features in a search range. This cost block is two-dimensional, and much less expensive than the four-dimensional cost volumes used in flow analysis. Cost-block features are computed by an encoder, and MB and Occ estimates are computed by a decoder. We found that arranging decoder layers fine-to-coarse, rather than coarse-to-fine, improves performance. MONet outperforms the prior state of the art for both tasks on the Sintel and FlyingChairsOcc benchmarks without any fine-tuning on them.

1 Introduction

Thanks to large-scale video datasets [2, 5, 20] and advances in deep learning, recent work [9, 30, 31] has rapidly improved dense optical flow estimation via learning with Convolutional Neural Network (CNNs). However, flow predictors still suffer near *motion boundaries* (MBs), the curves across which the optical flow field is discontinuous [32], and in *occlusion regions* (Occs), sets of pixels in one frame that do not have correspondences in the other. First, flow estimates are typically regularized by imposing spatial smoothness, which harms predictions near MBs. Second, flow cannot be measured from the input images in Occs and can only be plausibly guessed from the statistics of the ground truth provided in synthetic datasets. For instance, the top flow estimator RAFT [31] achieves an End-Point Error of 1.4 pixels on Sintel [2], but of 6.5 for pixels that are within 5 pixels from a MB and 4.7 in Occs. The accurate detection of MBs and Occs helps understand where flow estimates can be trusted and provides important visual cues for tracking [27] and video segmentation [12].

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

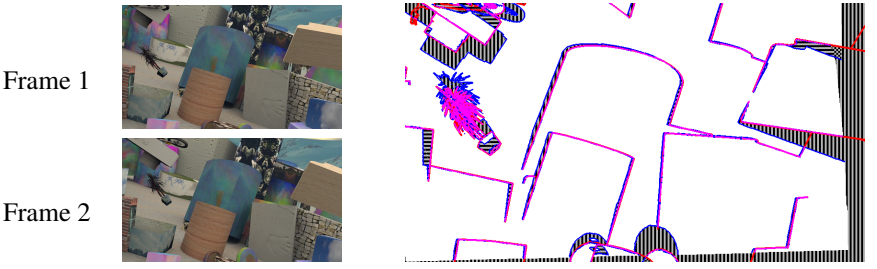
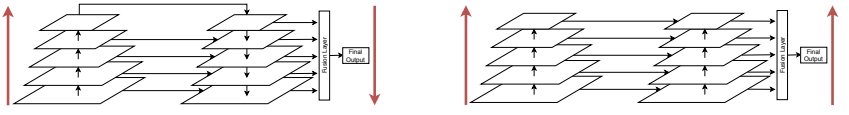


Figure 1: Occs align with MBs in this FlyingThings3D [20] frame pair. Vertical stripes denote O_1, *i.e.*, Occs in frame 1. Horizontal stripes are direct-warps to frame 1 of Occs in frame 2 ($D(O_2)$), and checkerboard patterns are $O_1 \cap D(O_2)$. Red curves are M_1 (similar notation for MBs M as for Occs O), blue ones are $D(M_2)$, and purple ones are $M_1 \cap D(M_2)$.



(a) (b)

Figure 2: (a) An encoder-decoder network [17]. (b) Our architecture with fine-to-coarse predictor (b). The encoder is the same, but information flow in the decoder is reversed.

Occs occur near MBs, where motion is discontinuous (Figure 1). In addition, disocclusions in one time direction are occlusions in the other. This suggests estimating MBs and Occs jointly, and to reason in both time directions simultaneously. Accordingly, we propose a CNN named *MONet* to jointly estimate MBs and Occs given two consecutive images and their estimated flow [30]. The network uses Siamese networks to leverage time symmetry.

Simultaneous reasoning in the two time directions requires mapping all quantities between frames, and we do this in a novel way. All previous flow [5, 30, 31], Occ [9], and MB [11] estimators use the flow from frame b to a to *reverse warp* features from frame a to frame b. We instead use flow from frame a to b to *direct warp* features from frame a to frame b. We show that direct warping both preserves features in Occs and provides additional Occ information through the regions it leaves undefined (Figure 4).

MBs often contour Occs. Based on this observation, we propose to make the MB predictor of MONet focus on Occ boundaries. Specifically, we use an attention mechanism [35] to place MB predictions where the gradient magnitude of the Occ map is large, that is, along predicted Occ boundaries. We use the MB labels to further supervise the attention map.

Since both MBs and Occs disrupt correspondences, MONet computes *cost blocks* that measure the lowest discrepancy between each feature of the first frame and its matching features in a search window in the second frame. A cost block is two-dimensional, and is the minimum over the search window of the four-dimensional cost volumes used in previous estimators of flow [5, 30, 31], MBs [11], and Occs [9]. The light-weight cost blocks are sufficient for our purposes and much less expensive to work with than cost volumes.

Cost blocks are defined on features computed by the MONet encoder, and a two-branch decoder then computes MB and Occ predictions. While the decoders in all previous estimators that use cost volumes [5, 9, 11, 21, 30] process information from coarse to fine, we do so Fine-to-Coarse (F2C) and show empirically the benefits of doing so (see Figure 2).

Summary of Contributions: Direct warping to better preserve information between frames; Spatial attention mechanism to align MB and Occ predictions; Two-dimensional cost blocks to measure feature discrepancies between frames; Fine-to-coarse decoder for higher accuracy; State-of-the-art performance for both MB and Occ detection on Sintel and FlyingChairsOcc without any fine-tuning on either dataset, and even after several ablations.

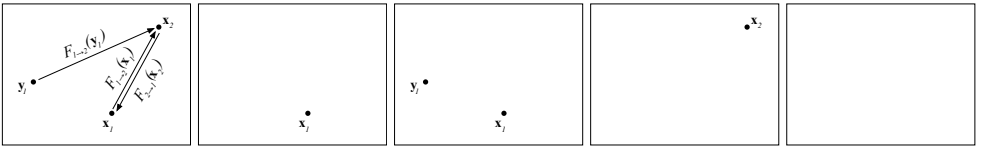
2 Related Work

In spite of continued advances [3, 5, 7, 30, 34], accurate **optical flow** estimation remains an open challenge especially near MBs and Occs. Recent methods [30, 34] starting with Chen and Koltun [3] use a four-dimensional **cost volumes** [24] to compute optical flow. DC Flow [34] and FlowNet [5] build the full cost volume at a single scale, and Sun *et al*. [30] show that building cost volumes at multiple scales leads to better models. Teed and Deng [31] achieve SOTA performance with RAFT, a deep network that builds a complete multi-scale four-dimensional cost volumes for all pairs of pixels on the input images. MONet instead uses a simplified two-dimensional cost block for MB and Occ detection.

The related task of detecting **occlusion regions** has recently received considerable attention [6, 8, 9, 21]. One Occ detector [8] takes as input optical flow estimated with four different algorithms and trains random forests to classify pixels into Occ or non-Occ categories. Fu *et al*. [6] use CNNs to detect Occ boundaries at the patch level and connect these detections to each other with a conditional random field [23]. Hur and Roth [9] achieve SOTA performance using a CNN to infer Occs and flow jointly. Neoral *et al*. [21] consider the two problems sequentially by first detecting Occs and then using Occs to help estimate flow. MONet jointly solves the two closely related problem of MB and Occ detection.

Early work [28] on **motion boundary** estimation exploits the observation that local flow histograms are bimodal near MBs. Liu *et al*. [16] propose to detect MBs by tracking and grouping hypothetical motion edge fragments bottom-up in scale. LDMB [32] uses structured random forests [4] and takes as inputs two consecutive images, optical flow estimates [29], and image warping errors, but produces noisy boundaries and fails on small and thin objects. In addition to the forward flow estimation from frame i to $i+1$, LDMB also takes as input the backward flow estimation from frame i to $i-1$ and requires three frames. MO-Net instead utilizes bi-directional flow between frames i and $i+1$, and only require two frames. Ilg *et al*. [11] uses CNNs incorporating joint training and refinement to simultaneously estimate Occs, MBs, optical flow, disparities, motion segmentation, and scene flow in both temporal directions. However, they simply stack multiple networks for joint-task solving and do not explicitly utilize the relationships between these tasks. MBs have also been used to aid in other low-level vision tasks such as video object segmentation [12].

Kramer's auto-encoder [14] is a precursor to the **encoder-decoder architecture** with three hidden layers with an information bottleneck. Long *et al.* proposes a Fully Convolutional Network, a fully-fledged encoder-decoder. Subsequent work makes upsampling deeper [22] and symmetrizes the architecture into what is called a U-Net [25]. The paper by Schulz and Behnke [26] proposes a shallow encoder followed by an F2C decoder, and uses only the coarsest prediction for inference. Our model is deeper and employs a trainable fusion layer, as we observed that the coarsest prediction is not always the best.



a: flows *b*: direct($\mathbf{x}_2$) *c*: reverse($\mathbf{x}_2$) *d*: direct($\mathbf{y}_1$) *e*: reverse($\mathbf{y}_1$)

Figure 3: Direct and reverse warping in the presence of occlusion. *(a)* Points x and y at $\mathbf{x_1}$ and $\mathbf{y_1}$ in frame 1 move to the same point $\mathbf{x_2}$ in frame 2 and y becomes occluded behind x at $\mathbf{x_2}$. If flow is modeled as a function, there is only one flow $F_{2\to1}(\mathbf{x}_2)$ in the $2\to1$ direction. *(b)* Direct warping of $\mathbf{x_2}$ to frame 1 using $F_{2\to1}$; *(c)* Reverse warping of $\mathbf{x_2}$ to frame 1 using $F_{1\to2}$; *(d)* Direct warping of $\mathbf{y_1}$ to frame 2 using $F_{1\to2}$; *(e)* Reverse warping of $\mathbf{y_1}$ to frame 2 using $F_{2\to1}$. Since the flow from $\mathbf{x}_2$ to $\mathbf{y}_1$ is not defined, nothing gets mapped in this case.

3 Principles

Section 4 describes MONet, a new neural network that jointly predicts MB scores $M \in [0,1]^{w\times h}$ and Occ scores $O \in [0,1]^{w\times h}$ both forward and backward in time (Figure 5). It takes as input two consecutive video frames $I_1, I_2 \in \mathbb{R}^{w\times h\times 3}$ and the corresponding bi-directional flow estimates $F \in \mathbb{R}^{w\times h\times 2}$ from an existing flow estimator. This Section describes the new principles that MONet embodies.

A first idea is that analysis of image motion in one time direction supports analysis in the other. As a result, the same inputs are provided in both temporal orders, $1\to2$ and $2\to1$. The model's parametric complexity is kept constant by sharing weights between temporal directions. Second, Occs align with MBs, and predicting Occs and MBs jointly yields richer insights than separate analyses would. This suggests a network with one Siamese encoder branch and two Siamese decoder branches that exchange information at all levels.

These principles are supported by the following technical ideas, described next: (i) Direct warping is more useful than reverse warping when aligning maps across frames; (ii) Attention can help align MBs and Occs; (iii) Inexpensive cost blocks capture useful information for MB and Occ detection. Architectural considerations are left for Section 4.

3.1 Direct Warping Provides Rich Occlusion Information

Computing MB maps and Occ maps in both frames requires establishing inter-frame correspondences. If a point that is away from both MBs and Occs is at $\mathbf{x}_a$ in frame a and at $\mathbf{x}_b$ in frame b, both flows are defined and unique:

$$\mathbf{x}_b = \mathbf{x}_a + F_{a\to b}(\mathbf{x}_a) \quad \text{and} \quad \mathbf{x}_a = \mathbf{x}_b + F_{b\to a}(\mathbf{x}_b) \quad \text{so that} \quad F_{a\to b}(\mathbf{x}_a) = -F_{b\to a}(\mathbf{x}_b)\,. \tag{1}$$

To map image values I_a from frame a to estimated values $\hat{I}_b$ in frame b, one can then use *direct warping*, which uses the flow defined in the same temporal direction as the map itself:

$$\hat{I}_b(\mathbf{x}_a + F_{a\to b}(\mathbf{x}_a)) = I_a(\mathbf{x}_a) \tag{2}$$

or *reverse warping*, which uses the flow in the direction opposite to the desired map:

$$\hat{I}_b(\mathbf{x}_b) = I_a(\mathbf{x}_b + F_{b\to a}(\mathbf{x}_b))\,. \tag{3}$$

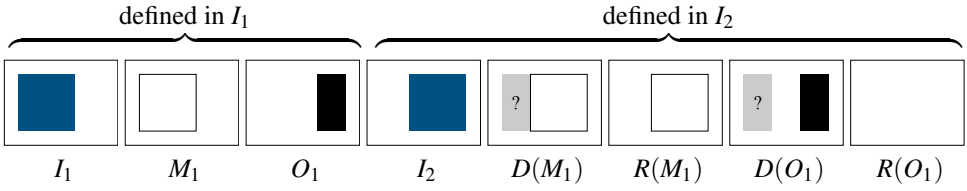


Figure 4: Image frames I_1 and I_2 show a blue square translating to the right on a static white background. The MB map M_1 and Occ map O_1 in frame 1 can be warped with direct ($D(\cdot)$) or reverse ($R(\cdot)$) flows to frame 2. The gray rectangles with question marks in $D(M_1)$ and $D(O_1)$ denote regions for which no map values are available because the flow $F_{1\to 2}$ maps no points into those regions.

Reverse warping is preferred when one loops over a grid of locations $\mathbf{x}_b$ to fill $\hat{I}_b$ completely. Pixel discretization aside, however, the result is the same either way.

The situation is more complex for points in Occs or on MBs. For example, Figure 3 (a) shows two points x and y at locations $\mathbf{x_1}$ and $\mathbf{y_1}$ in the first frame that both move to the same location $\mathbf{x_2}$ in the second frame. Point x remains visible, while point y becomes occluded behind point x. The map in the $2 \to 1$ direction is one-to-many (not a function). However, it is customary (although somewhat arbitrary) to model flow as a function. If we do so, the flow at $\mathbf{x}_2$ in frame 2 maps to $\mathbf{x}_1$ in frame 1, and there is no flow from $\mathbf{x}_2$ to $\mathbf{y}_1$. The effects of direct and reverse mapping in the two directions are illustrated in the remaining panels.

All the previous MB [11], Occ [9, 11], and flow [5, 9, 10, 11, 30, 31] estimators that use cost volumes use reverse warping (equation 3). We are the first to direct warp for motion analysis. As we illustrate in Figure 4 for a $1 \to 2$ mapping, direct warping preserves Occ information *and* provides additional Occ information through the regions that it leaves undefined. Here and elsewhere, we let M_a be the MB maps in frame a, and $D(M_a)$ and $R(M_a)$ be the maps in frame b obtained by direct (equation 2) and reverse (equation 3) warping of M_a. We define O_a, $D(O_a)$, and $R(O_a)$ similarly. First, the Occ in O_1 (black rectangle) is correctly mapped to the second frame in $D(O_1)$ (recall that the background does not move). In contrast, $R(O_1)$ overwrites this information with the no-Occ information from the foreground square, since $F_{2\to 1}$ is defined everywhere on that square. Thus, direct warping preserves Occs but reverse mapping does not. Second, $D(O_1)$ has no values in the gray-shaded rectangle with the question mark, which is the part of the background that has become newly visible in frame 2. None of the points that are visible in frame 1 move to that region, and $D(O_1)$ is therefore undefined there. In contrast, $R(O_1)$ uses $F_{2\to 1}$, which is defined everywhere in frame 2, and $R(O_1)$ is therefore defined everywhere there as well. The presence of undefined values in $D(O_1)$ is a useful source of information for the detection of Occs in the $2 \to 1$ direction, and this information is unavailable when reverse warping is used. Similar considerations hold for the warped MB maps $D(M_1)$ and $R(M_1)$.

3.2 Motion Boundaries Align with Occlusion Regions

In most cases, Occs occur near flow field discontinuities: An object in the foreground moves differently from its background, and the resulting curve of flow discontinuity sweeps over the background to make an Occ. Thus, MBs and Occs align, as Figure 1 illustrates.

Some MBs do cut across Occs. For instance, the right edge of the image adds a third MB to the two MBs between the boxes. Also, Occ patterns for thin regions are complex (small

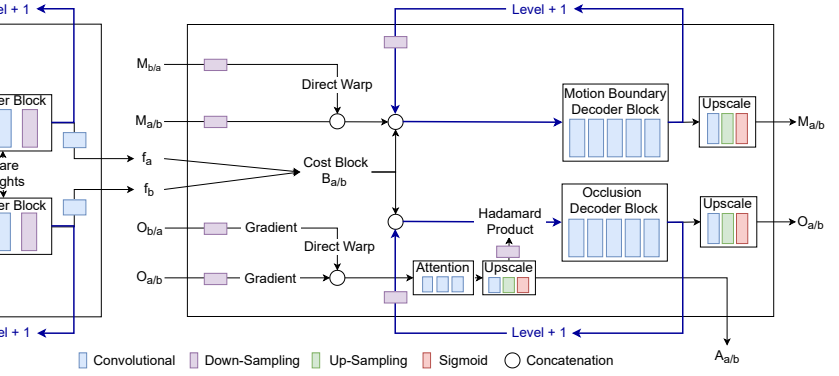


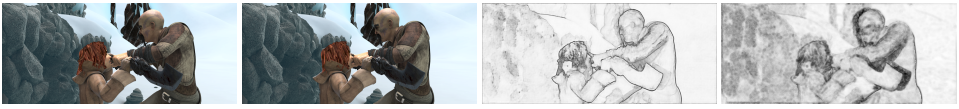
Figure 5: MONet predicts bidirectional $M_{a/b}$ and $O_{a/b}$ maps in each level l using bi-directional predictions (M_a, M_b, O_a, O_b) and cost block ($B_{a/b}$) from the previous level $l-1$. Blue arrows represent the main flow of information across levels.

potted plant on the left). Furthermore, when the motion difference between foreground and background is parallel to the MB, there is no occlusion or dis-occlusion (top of the brown cylinder near the image center). Finally, the "ground truth" used in Figure 1 is not perfect: Some MBs are two pixels thick, when they should ideally have measure zero. Of the two pixels at some point on an MB, one is in the foreground and the other is in the background, so they are warped by different flow values. A $D(M_2)$ (blue) MB pixel that is mistakenly warped by the background motion ends up overlapping an M_1 (red) pixel, giving purple.

Nonetheless, in spite of exceptions and imperfections, the pattern is clear: MBs are adjacent to Occ boundaries. This pattern is also borne by statistics: We found that 80.5% of MB pixels in MPI-Sintel are within one pixel of an Occ boundary, and 89.3% are within three pixels. MONet therefore decodes MB and Occ features jointly in a cascade of *levels* that go fine-to-coarse. (As discussed later on, we found fine-to-coarse to work better than coarse-to-fine.) The magnitudes of the gradients of the Occ maps at one level are sampled and passed to the next-level MB predictor. Conversely, the MBs at one level are sampled and passed to the next level Occ predictor. MONet also incorporates an explicit attention mechanism [35] to focus the MB predictor on boundaries of Occs. Specifically, an attention map $A \in [0,1]^{w \times h}$ is constructed at each level from the gradient of the appropriate Occ map at that level. This map is then is multiplied with the MB feature at that level with the Hadamard product before it is sub-sampled and passed to the subsequent level (see Figure 6 (c)).

3.3 Occlusions or Motion Boundaries Have High Residual Cost

The discrepancies between features computed from the two frames also provide evidence for Occs and MBs. Specifically, suppose that a feature vector $\mathbf{f}_a$ with receptive field R_a centered at pixel $\mathbf{x}_a$ in frame a is tentatively matched to a feature vector $\mathbf{f}_b$ with receptive field R_b centered at $\mathbf{x}_b$ in frame b. Then, if $\mathbf{x}_a$ and $\mathbf{x}_b$ do not correspond to each other because of occlusion, the vectors $\mathbf{f}_a$ and $\mathbf{f}_b$ are likely to be different from each other. In addition, if, say, $\mathbf{x}_a$ is on an MB, then R_a contains both foreground and background pixels. Pixels in these two subsets move differently, so it is unlikely that all of them match up with corresponding pixels in R_b, again leading to differences between $\mathbf{f}_a$ and $\mathbf{f}_b$. To exploit this intuition, we also



(a) I_1 (b) I_2 (c) A_1 (d) B_1

Figure 6: Attention map A_1 (c) and cost block B_1 (d) computed in the finest resolution, using an example from Sintel [2] (a and b). Black is large and white is small for A_1 and B_1.

construct optional *cost blocks* to measure feature discrepancies. Many SOTA Occ and MB methods [9, 11] construct a four-dimensional *cost volume* $V_1(\mathbf{x}_1, \mathbf{d}) \in \mathbb{R}^{w \times h \times (2s+1) \times (2s+1)}$ of the Euclidean distances between a feature at $\mathbf{x_1}$ in the first frame and that of a point at $\mathbf{x}_1 + \mathbf{d}$ for displacement $\mathbf{d}$ within s pixels in each dimension in the second frame. These cost volumes are overkill for us, because we already have flow estimates inputs ($F_{a \to b}$). Instead, we construct two smaller two-dimensional *cost blocks* $B_1, B_2 \in \mathbb{R}^{w \times h}$ where

$$B_a(\mathbf{x}_a) = \min_{\mathbf{d} \in [-s,s]^2} V_a(\mathbf{x}_a, \mathbf{d} + F_{a \to 3-a}(\mathbf{x}_a)) \text{ for } a = 1, 2 \,. \tag{4}$$

High values in either B_1 or B_2 often signal vicinity to MBs or Occs (see Figure 6 (d)).

4 MONet Architecture and Training Details

Encoder We use a fully-convolutional neural network to learn feature maps at multiple scales from fine to coarse, and compute a cost block on these features at each scale. At each scale, four convolutional layers with leaky ReLU activations [18] process down-scaled input images and output 32-channel feature maps. Each feature map is down-sampled with stride-2 convolutions and is passed to the next coarser scale.

Motion boundary and occlusion predictors The two decoders for MBs and Occs are fully-convolutional and Siamese, and process the same set of inputs in opposite temporal directions and in fine-to-coarse manner, where maps predicted in each scale are used to predict maps in the following coarser scale. Each scale has five convolutional layers with 3×3 kernel and leaky ReLUs [18]. Feature maps from each block pass through a 1×1 convolution layer to reduce the channel dimension to 1, a de-convolution layer to up-sample the map to the original resolution, and a sigmoid layer to output a prediction. A fusion layer [33] takes the concatenation of L multi-scale maps and computes a weighted average of the maps using a 1×1 convolution layer with kernels initialized to $1/L$.

Attention modules Each attention module has three convolutional layers with 3×3 kernels followed by a sigmoid. Similar to MB and Occ prediction maps, the output attention feature maps are passed through a 1×1 convolution layer, a de-convolution layer, and a sigmoid layer to output the full-resolution attention maps.

Fine-to-coarse decoders MONet's decoders are Fine-to-Coarse (F2C) (Figure 2 (b)), in contrast with the Coarse-to-Fine (C2F) decoders (Figure 2 (a)) used in current encoder-decoder CNNs that involve cost volume computations [5, 9, 11, 21, 30]. Skip connections feed the encoder feature maps (horizontal arrows) to decoder layers of corresponding resolution (right pyramid). The only differences between the two types of architecture are (i) replacing every up-sampling layer in the decoder with a down-sampling layer, and (ii) connecting every layer to the immediately *finer* layer below it, rather than the coarser one above.

A fusion layer combines the multi-scale predictions into the high-resolution output. Without adding computation, the new F2C architecture improves over C2F (Section 6).

Training We minimize the focal loss [15] on MB, Occ, and attention maps with Adam [13] and with an initial learning rate of 10^{-4}. We use flow estimated from PWC-Net [30] as input for training, and various flow estimators [9, 11, 30, 31] for evaluation. We implement MONet in Tensorflow [1], and is available at https://github.com/hannahhalin/MONet.

5 Datasets and Performance Evaluation

We train on FlyingThings3D [20] dataset and evaluate on FlyingChairsOcc [5, 9] and MPI-Sintel [2] datasets without any fine-tuning. **FlyingThings3D** [20] (FT3D) is created by moving graphics-generated objects along random 3D trajectories and includes 21818 training images and 4248 testing images. **FlyingChairs** [5] is a synthetic dataset generated by applying random affine transformation to Flickr images as backgrounds, and a set of rendered moving 3D chairs as foreground. It consists of 22872 image pairs and corresponding ground truth flow. **FlyingChairsOcc** [9] adds ground-truth forward and backward Occ maps to FlyingChairs. **MPI-Sintel** [2] contains 23 high resolution sequences of 20 to 50 frames each from the open-source computer-animation short "Sintel". Fast motion and large Occs make this dataset challenging. We follow the literature [9, 11, 32] and evaluate Occ predictions by average F_1-score after thresholding the map at 0.5, and evaluate MB predictions by mean Average-Precision (mAP) computed with the BSDS evaluation code [19].

6 Results

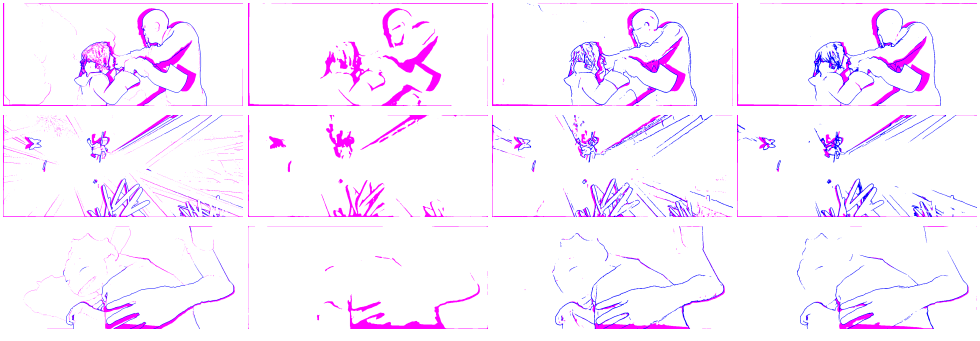
MONet outperforms the state of the art in both MB [11] and Occ [9] detection. Table 1 compares MONet to the existing SOTA MB detectors, LDMB [32] and FlowNet-CSS [11], and to the SOTA Occ detectors, FlowNet-CSSR-ft-sd [11] and IRR-PWC [9]. FlowNet-CSS is trained on FlyingChairs [5] and FT3D [20] to achieve the current SOTA performance on MB detection, and it is further fine-tuned on ChairsSDHom [10] to obtain FlowNet-CSSR-ft-sd for Occ detection. IRR-PWC is trained on FlyingChairsOcc and FT3D, and LDMB [32] is trained on Sintel. Performance of our proposed MONet is obtained by training on FT3D only *without any fine tuning on Sintel or FlyingChairsOcc*. We also include a baseline MB performance by taking the gradient of estimated flow from RAFT. Specifically, we cap the flow gradients at 25%, 50%, 75%, and 100% of the maximum gradient over the entire dataset, and report the highest performance. MONet bests the SOTA methods for both tasks in all datasets (bold). Figure 7 shows precise and clean MB and Occ predictions by MONet.

Figure 8 stratifies occlusion performance in terms of false-positive rate and accuracy by each point's distance to the closest Occ or MB. All three methods, IRR-PWC, FlowNet-CSS, and MONet, degrade closer to MBs or Occs. While FlowNet-CSS and MONet detect MBs and Occs jointly, IRR-PWC does not predict MBs and degrades most sharply. MONet shows the best performance across all distances in all plots: Joint detection helps Occ detection, especially when the relationships between MBs and Occs are leveraged explicitly.

Table 2 shows the effects of removing combinations of the proposed components in MONet. Even without all the components, MONet still improves over the SOTA methods [9, 11] (underlined). Using everything yields the best performance in both tasks (bold).

Dataset	Motion Boundary (*mAP*)				Occlusion (F_1)		
	Baseline	[32]	[11]	MONet	[11]	[9]	MONet
FlyingChairsOcc [9]	-	-	-	-	78.9	75.7	**82.7**
Sintel (Clean) [2]	72.6	76.3	86.3	**93.1**	70.3	71.2	**74.4**
Sintel (Final) [2]	62.9	68.5	79.5	**79.8**	65.4	66.9	**68.7**

Table 1: Average F_1 score for **occlusion detection** (right) and mean average precision (mAP) for **motion boundary detection** (left). Our MONet bests the state of the art for both tasks *without any fine tuning on FlyingChairsOcc or Sintel* (bolded).



Label [9] [11] MONet

Figure 7: MB (blue) and Occ (magenta) predictions of examples from Sintel [2], thresholded at 0.5. MONet yields precise and clean predictions. (Best viewed magnified and in color.)

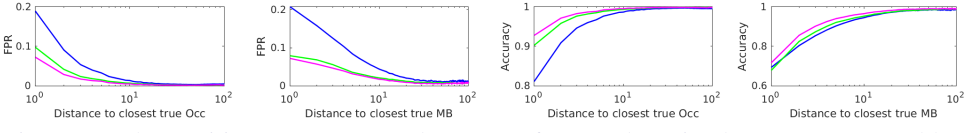


Figure 8: False Positive Rate (FPR) and accuracy for Occ detection by IRR-PWC [9] (blue), FlowNet-CSSR-ft-sd [11] (green), and our MONet (magenta), stratified by the distance to the closest true Occ or MB. Smaller is better for FPR, and larger is better for accuracy.

	$-DAB$	$-AB$	$-B$	$-A$	MONet
MB (*mAP*)	<u>90.8</u>	<u>92.1</u>	<u>91.6</u>	<u>92.2</u>	<u>**93.1**</u>
Occ (F_1)	69.8	<u>71.6</u>	<u>74.0</u>	<u>73.6</u>	<u>**74.4**</u>

Table 2: Effect of proposed components, direct warping (D), attention (A), and Cost Block (B), in Occ and MB detection performance on Sintel. *Even without all the proposed components, MONet still improves over the SOTA methods (underlined).*

Table 3 evaluates MONet with various input flows [9, 11, 29, 30, 31]. Performance in both MB and Occ detection increases with better input flow estimates. Even with Classic+NL [29], also used by LDMB, MO-Net still outperforms the SOTA for MB detection. Similarly, using the flow from the SOTA MB and Occ estimators [9, 11], MONet outperforms SOTA for both MB and Occ detection shown in Table 1. Regardless of flow input quality, MONet improves on the prior SOTA performance for both tasks (underlined).

Table 4 compares the performance of MONet (joint task solving) with that of estimating MBs or Occs separately. To make a single-task version of MONet, we simply remove the de-

Flow (EPE)	6.04 [29]	2.55 [30]	2.08 [11]	1.88 [9]	1.43 [31]
MB (*mAP*)	90.2	91.8	92.1	92.2	**93.1**
Occ (F_1)	69.2	72.3	73.0	73.4	**74.4**

Table 3: Effect of flow estimation input quality (End-Point Error) in Occ and MB detection performance of MONet on Sintel [2]. *Regardless of input flow quality, MONet still improves over the SOTA performers in Table 1 (underlined).*

MB (*mAP*)	Single task	Joint task
	91.6	**93.1**

Occ (F_1)	Single task	Joint task
	72.5	**74.4**

Table 4: Effect of joint task learning in MB and Occ detection performance on Sintel [2].

	Occ ([9])		Occ (ours)		MB (ours)	
Dataset	C2F	F2C	C2F	F2C	C2F	F2C
FlyingChairsOcc	75.7	**78.5**	80.9	**82.7**	-	-
Sintel (Clean)	71.2	**71.9**	73.3	**74.4**	91.2	**93.1**
Sintel (Final)	66.9	**67.1**	65.6	**68.7**	72.4	**79.8**

Table 5: Performance of MB (*mAP*) and Occ (F_1) detection with C2F and F2C decoders. *The F2C version outperforms the C2F version for both IRR-PWC [9] and MONet.*

coder for the other task and all the connections between the two decoders. MONet estimates both MBs and Occs better jointly than separately.

Finally, Table 5 compares C2F and F2C decoders. The current SOTA Occ detector, IRR-PWC [9], utilizes an encoder-decoder architecture, and we simply reverse the information flow of its decoder to make it F2C, using the same training scheme as for its C2F version. The Table shows that the F2C version of each system consistently outperforms its C2F version.

We speculate that the F2C predictor preserves spatial details better when compared to C2F predictor as the finer predictions do not evolve from the bottleneck features in the C2F predictor that suppress spatial details. Specifically, in a F2C decoder, layers at finer resolution process information that is closest to the full resolution of the input, and can focus on getting the initial predictions right. The overall picture is captured well by coarse resolution predictions, which can be upsampled to full resolution and combined with good predictions along boundaries from the finer predictions. This process does not work in a C2F predictor, because the finer-resolution predictions are made many layers away from the input, and the only fine-resolution information they get is from the skip connections. The flow of information in F2C decoder is consistent with what is done in the edge detection literature [33].

7 Conclusion

We propose MONet to jointly detect MBs and Occs in both time directions given two video frames and their estimated bi-directional flow. We direct-warp maps between frames, use an attention mechanism to align MBs and Occs, and provide correspondence information with a cost block within an encoder-decoder architecture with F2C decoders. Fine-to-coarse bests coarse-to-fine both for our architecture and for IRR-PWC. This reversal of information flow can be applied at no cost to any encoder-decoder. MONet improves the SOTA for both MBs and Occs on the Sintel and FlyingChairsOcc without any fine-tuning on either dataset.

Acknowledgments: This material is based upon work supported by the National Science Foundation under Grant No. 1909821 and by an Amazon AWS cloud computing award.

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. URL https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[2] 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.

[3] Qifeng Chen and Vladlen Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4706–4714. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.509. URL https://doi.org/10.1109/CVPR.2016.509.

[4] 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.

[5] 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.

[6] Huan Fu, Chaohui Wang, Dacheng Tao, and Michael J. Black. Occlusion boundary detection via deep exploration of context. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

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

[8] Ahmad Humayun, Oisin Mac Aodha, and Gabriel J. Brostow. Learning to find occlusion regions. *Conference on Computer Vision and Pattern Recognition*, 2011.

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

[10] Eddy Ilg, N. Mayer, Tonmoy Saikia, Margret Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

[11] 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 (ECCV)*, September 2018.

[12] Zahra Kamrani, Ahmad Naghsh-Nilchi, Hamid Sadeghian, Federico Tombari, and Nassir Navab. Joint motion boundary detection and cnn-based feature visualization for video object segmentation. *Neural Computing and Applications*, 2019. doi: 10.1007/s00521-019-04448-7.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

[14] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[15] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 07 2018. doi: 10.1109/TPAMI.2018.2858826.

[16] Ce Liu, William T. Freeman, and Edward H. Adelson. Analysis of contour motions. *Advances in Neural Information Processing Systems*, 2006.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. doi: 10.1109/CVPR.2015.7298965. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298965.

[18] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[19] 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.

[20] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. URL http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16. arXiv:1512.02134.

[21] Michal Neoral, Jan Sochman, and Jiri Matas. Continual occlusion and optical flow estimation. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision - ACCV 2018 - 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part IV*, volume 11364 of *Lecture Notes in Computer Science*, pages 159–174. Springer, 2018.

doi: 10.1007/978-3-030-20870-7_10. URL https://doi.org/10.1007/978-3-030-20870-7_10.

[22] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[23] Xiaofeng Ren, Charless C. Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In *Proc. 10th Int'l. Conf. Computer Vision*, volume 2, pages 1214–1221, 2005.

[24] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, February 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.156. URL http://dx.doi.org/10.1109/TPAMI.2012.156.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[26] Hannes Schulz and Sven Behnke. Learning object-class segmentation with convolutional neural networks. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN), April 2012*, 2012.

[27] J. Shi and C. Tomasi. Good features to track. In *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, USA, 1994. IEEE Comput. Soc. Press.

[28] Anselm Spoerri. *The Early Detection of Motion Boundaries*. PhD thesis, Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences, 1991.

[29] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115–137, 2014.

[30] 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.

[31] 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.

[32] 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, June 2015. URL https://hal.inria.fr/hal-01142653.

[33] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Conference on Computer Vision*, 2015.

[34] Jia Xu, Rene Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5807–5815, Honolulu, HI, 2017.

[35] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.