BrightFlow: Brightness-Change-Aware Unsupervised Learning of Optical Flow

Rémi MARSAL[*†] Florian CHABOT[*] Angelique LOESCH[*] Hichem SAHBI[†]

[*]Université Paris-Saclay, CEA, LIST, F-91120, Palaiseau, France
firstname.lastname@cea.fr
[†]Sorbonne University, CNRS, LIP6 F-75005, Paris, France
firstname.lastname@lip6.fr

Abstract

Unsupervised optical flow estimation relies on the assumption that pixels characterizing the same observed object should exhibit a stable appearance across video frames. With this assumption, the long-standing principle behind flow estimation consists in optimizing a photometric loss that maximizes the similarity between paired pixels in successive frames. However, these frames could be subject to strong brightness changes due to the radiometric properties of scenes as well as their viewing conditions.
In this paper, we present BrightFlow, a new method to train any optical flow estimation network in an unsupervised manner. It consists in training two networks that jointly estimate optical flow and brightness changes. These changes are then compensated in the photometric loss so that reconstruction errors due to shadows or reflections will not affect negatively the training. As this compensation mechanism is only used at training stage, our method does not impact the number of parameters or the complexity at inference. Extensive experiments conducted on standard datasets and optical flow architectures show a consistent gain of our method. Source code is available at https://github.com/CEA-LIST/BrightFlow.

1. Introduction

Optical flow measures the relative motion of each pixel in a given scene acquired at successive instants. This task has many applications including motion segmentation [51, 50, 48], anomaly detection in videos [25, 26, 1] or video representation [11, 32]. While traditional methods [3, 6, 5, 35] are based on optimization problems with handcrafted features, geometric and statistical criteria, more recent ones rely on deep learning approaches [13, 42, 43] that require to train a neural network on a large dataset. Their principle consists in learning a mapping that minimizes a supervised loss between the estimated flow and its underlying ground truth. However, the success of these regression models is highly reliant on the availability of large labeled collections. While densely labeled synthetic collections are abundant [9, 7, 28], they are powerless to capture the inherent variability of real-world scenes, and this may result in deep networks with weak cross-domain generalization. On the contrary, labeled realistic video collections are scarce because their labeling is time/effort-demanding.

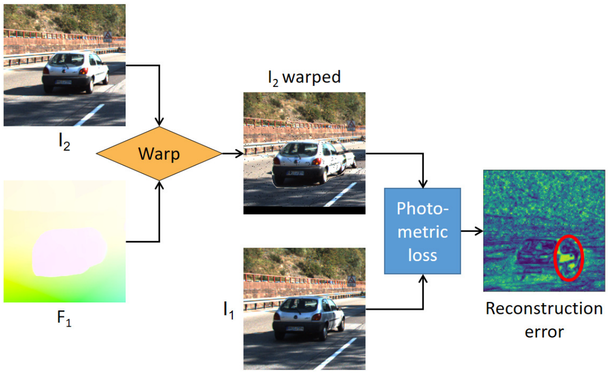
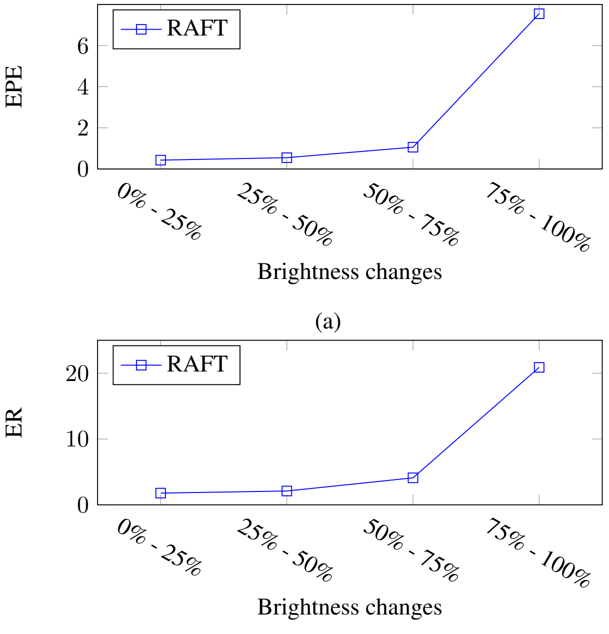


Figure 1: Diagram of the photometric loss to show how it is impacted by brightness changes. Here the rear side of the car has left the shadow in the second image causing a major brightness change. It induces a peak in the photometric loss whereas the optical flow is well estimated.

Unsupervised optical flow estimation is an alternative that circumvents the lack of labeled data. In these methods, training is achieved by minimizing a photometric loss that measures brightness consistency between original and warped frames in videos. Nonetheless, brightness consistency may not hold in practice; on the one hand, the intrinsic content of scenes varies leading to occlusions. On the other hand, acquired scenes are subject to strong changes, even on successive frames due to reflections, shadows,



(a)

(b)

Figure 2: Performances in EPE (a) and ER (b) as a function of brightness changes. All non-occluded pixels in the Sintel final dataset have been divided into four groups depending on the magnitude of brightness changes (from the 25% pixels with the lowest to the 25% pixels with the highest ones). It shows that higher brightness changes lead to worse results.

sensor orientations, etc. In the following, we will group all these appearance variations under the generic term of *brightness changes* that refers to any changes in the appearance of an element of the first image that is still visible in the second one. Hence, occluded pixels are not considered as a part of brightness changes. The issue of occlusion has been widely studied in the literature: there are different ways to estimate them [4, 45]; several methods use knowledge distillation to supervise flow predictions in artificially occluded areas [22, 23] and others have dedicated techniques to estimate optical flow in occluded areas [37, 24]. Comparatively, brightness changes have received much less attention whereas they mislead optical flow estimation because with different appearances, finding corresponding pixels is harder (see figure 2). In this paper, we will focus on preventing brightness changes to affect the photometric loss. Current state-of-the-art unsupervised optical flow estimation methods use the soft census loss [10] in the photometric loss to compare an image and its reconstruction. It is a differentiable version of [53, 36] so it can be used as a loss function to train a neural network. However, whilst being globally robust to many brightness changes (such as multiplicative rescaling and gamma correction), this loss is only invariant to global additive changes; making it suboptimal in some cases of brightness changes (see figure 1). Therefore, brightness changes would still induce errors in the photometric loss, being responsible for unsuitable updates of the optical flow network weights.

Considering the aforementioned issues, we introduce in this paper a novel framework that models brightness changes thanks to a neural network dedicated to this task and whose training is unsupervised. Its output correction map compensates for brightness changes in the photometric loss. The goal is to help the optical flow network to handle brightness changes by reducing the bias induced in the photometric loss. Note that our method does not rely on any rendering model [2, 31] which may require additional information about light source localization or 3D structures of the scenes. Considering the above issues, the proposed work includes the following contributions:

- The brightness correction network, a neural network that takes as input the source image, the warped target image and the underlying occlusion map. It predicts a pixel-wise brightness change map of the source with respect to the target. It is optimized with an unsupervised photometric loss that measures the discrepancy between warped and original frames.
- A novel photometric loss that includes a mechanism of brightness correction gating to make the most of the brightness correction map.
- The proposed method is applicable to any optical flow architecture. Besides, it is considered only during training to enhance the generalization of the optical flow network. This makes run-time and memory footprint of our method similar to the original optical flow network at inference.
- Finally, the consistent gain of our method is shown through extensive experiments involving different datasets and optical flow architectures. We also highlight a better cross-domain generalization.

2. Related work

2.1. Supervised optical estimation

Progress in deep learning has had a major impact in the field of optical flow estimation. Early solutions are based on convolutional networks including the pioneering work of FlowNet [13] as well as its multi-stacked variant [14]. [33] proceeds iteratively by warping images with the optical flow estimated at a lower scale. PWC-Net and LiteFlowNet [42, 12] use cost volumes to measure similarities between feature maps inferred from successive images. Many works have improved PWC-Net [42]: using different correlations in the cost volumes [49, 44] or exploiting occlusion predictions [55]. Whereas the aforementioned methods are coarse-to-fine, a more recent solution RAFT [43] evaluates, at once, a cost volume involving all the paired pixels in successive frames, then refines the flow at a unique resolution using gated recurrent units [8]. Subsequent contributions have either improved RAFT (with attention [17], sparse and more sophisticated cost volumes [18, 38, 54, 56]) or addressed optical flow estimation using transformers [15, 47].

2.2. Unsupervised Optical Flow estimation methods

Due to the prohibitive cost of optical flow ground truth, many unsupervised approaches have been developed. Early works [52, 34] leverage a photometric and a smoothness loss only. Subsequent methods achieve better accuracy by exploiting occlusions [29, 45], forward-backward consistency criteria [29], more than two frames as inputs [16] or knowledge distillation [22, 23]. Handling the optical flow of occluded pixels has also been investigated by OIFlow [24] which proposes a particular architecture to inpaint occlusions and SMURF [37] which inverses the optical flow toward the previous image in order to estimate the motion of occluded objects. SMURF [37] also applies warping on the full image instead of the cropped one in the photometric loss to reduce the amount of boundary occlusions. Architectures are also tuned in unsupervised methods with improved upsampling modules [27] or lighter networks, normalized cost volumes and dropout [19]. Recent approaches rely on complex data augmentation [21] or the generation of a highly varied dataset from the superposition of randomly shaped masks of images [41].

2.3. Brightness changes

Optical flow estimation methods are built upon the assumption that pixels characterizing the same physical objects should exhibit similar appearances across frames in videos. However, this hypothesis becomes wrong in case of photo-realistic datasets since shadows and reflections induce brightness changes (alteration of the appearance of corresponding pixels in different frames). Both supervised and unsupervised optical flow methods are subject to brightness changes through the optical flow network that directly processes images. To get the network agnostic to brightness changes, asymmetric data-augmentation is employed [43, 37]. It consists in exposing the flow estimator to images with artificially generated strong brightness changes, in order to make it resilient to a wide range of brightness changes. More specifically, the two input images receive different photometric data augmentation. In the case of unsupervised learning of optical flow, the photometric loss is also exposed to brightness changes. This may induce false negatives in the photometric loss that could harm the training of the optical flow network. While occlusions have been sufficiently well addressed in the literature, brightness changes have comparatively received much less attention [23, 37]. Our method focuses on the latter issue. Since asymmetric data augmentation emphasizes brightness changes, it could not be a sufficient solution. Commonly used solutions are functions designed to be resilient to some extent to brightness changes like SSIM [46] or the soft census loss [10] that is currently used in state-of-the-art methods. Nevertheless, these functions are handcrafted and can be limited to handle some brightness changes. In this paper, we propose a novel method, BrightFlow, to make the soft census loss in the photometric loss dynamic thanks to a network trained to model brightness changes.

3. Method

3.1. Preliminaries on Unsupervised Optical Flow

Let $\mathcal{T} = \{\mathcal{V}_i\}_i$ be a collection of videos with each one being an ordered sequence of frames denoted as $\mathcal{V}_i = \{I_t^i\}_t$ where $I_t^i \in \mathbb{R}^{H \times W \times C}$ and H, W, C stand for frame height, width and number of channels respectively. Images of $\mathcal{T}$ can be rearranged into the union of consecutive frames $\mathcal{I}$. Let (I_1, I_2) be an element of $\mathcal{I}$, estimating optical flow from I_1 to I_2 consists in inferring the vector field $F_1 \in \mathbb{R}^{H \times W \times 2}$ that explains the 2D relative motion of each pixel in I_1 w.r.t the corresponding pixel in I_2. This field may capture rigid/non-rigid movement of objects and sensors. It is estimated using a mapping function ψ_θ so that $\psi_\theta(I_1, I_2) = F_1$. In practice, ψ_θ corresponds to a deep neural network with learnable parameters θ. Details about the architectures and the training procedure are given subsequently and in the experiments. We define the forward direction when optical flow prediction is made from I_1 to I_2 and the backward direction from I_2 to I_1.

Following an unsupervised setting, and considering a training set $\mathcal{I}$ of consecutive images without optical flow ground truth, one may find the optimal parameters θ^* of ψ_θ as $\theta^* = \arg\min_\theta \mathcal{L}(\theta)$ where $\mathcal{L}$ is a global loss defined as

$$\mathcal{L}(\theta) = \gamma_{ph}\mathcal{L}_{ph1}(\theta) + \gamma_{sm}\mathcal{L}_{sm1}(\theta) + \gamma_{self}\mathcal{L}_{self1}(\theta) \quad (1)$$

where $\mathcal{L}_{ph1}$, $\mathcal{L}_{sm1}$, $\mathcal{L}_{self1}$ stand respectively for photometric, smoothness and self-supervised forward losses; γ_{ph}, γ_{sm}, γ_{self} are their respective weights. Each loss is then computed in the backward direction but (for short) in the rest of the section, we describe only the forward version of each loss. The corresponding backward expression is obtained by swapping indices 1 and 2, then both forward and backward losses are averaged.

Photometric loss. Without ground truth, optical flow can be learned because warping I_2 with the optical flow F_1 provides a reconstruction of I_1: $\hat{I}_1 = w(I_2, F_1) \approx I_1$, being

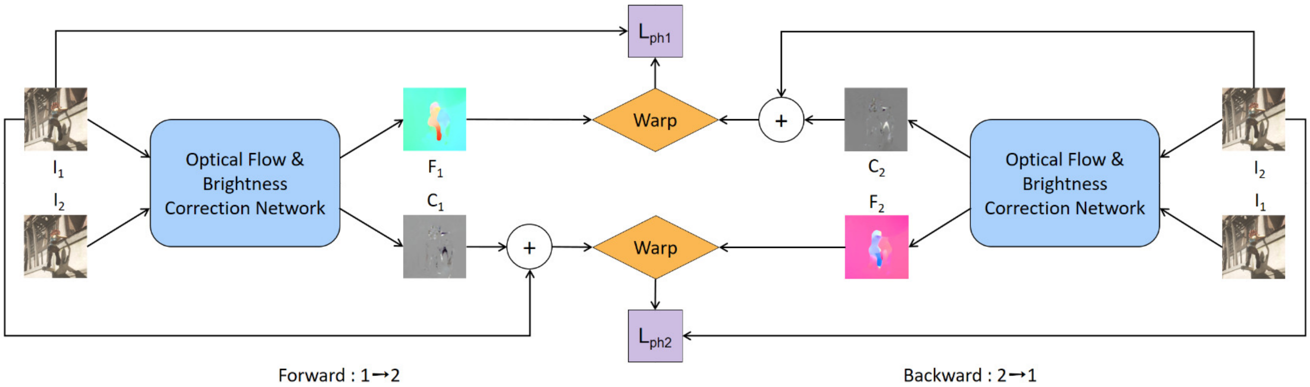

L_ph1
Optical Flow & Brightness Correction Network
I1
I2
F1
C1
Warp
+
Warp
+
C2
F2
Optical Flow & Brightness Correction Network
I2
I1
L_ph2
Forward : 1⟶2
Backward : 2⟶1

Figure 3: **Overall BrightFlow training architecture** Both optical flow and brightness correction network blocks share weights (more details available in figure 4). They predict the optical flow F_1 and the correction map C_1 from image I_1 and I_2 in the forward direction. Symmetrically, they also predict the optical flow F_2 and the correction map C_2 from image I_2 and I_1 in the backward direction. Then, in the photometric loss L_{ph1} the image I_1 is compared to the corrected image $I_2 + C_2$ warped with the optical flow F_1. Likewise, photometric loss L_{ph2} takes as inputs the image I_2 and the corrected image $I_1 + C_1$ warped with the optical flow F_2.

w the warping function of an image with an optical flow. The loss $\mathcal{L}_{ph}$ penalizes photometric error between an image and its reconstruction. However, the consistency of this loss holds only when paired pixels are visible both in I_1 and I_2, so the exact definition of this loss is given as

$$\mathcal{L}_{ph1}(\theta) = \|\mathbf{O}_1 \odot \rho(\hat{I}_1, I_1)\|_1 / \|\mathbf{O}_1\|_1, \tag{2}$$

where $\|.\|_1$ denotes the ℓ_1-norm and $\odot$ the Hadamard product. ρ stands for an entry-wise distance, here the soft census loss [10], which measures the discrepancy between two images (more details about our implementation of the soft census loss are available in the supplementary material). $\mathbf{O}_1$ refers to a binary occlusion mask whose given pixel entry is set to one if and only if the underlying observed point is visible both in I_1 and I_2; otherwise, the entry is set to zero. More information about the computation of the occlusion maps is provided in the implementation detail section.

Smoothness loss. In order to promote object-wise optical flow consistency, the smoothness loss is leveraged

$$\mathcal{L}_{sm1}(\theta) = \frac{1}{HW}\Bigg\| \exp\Bigg\{-\frac{\lambda}{3}\sum_{c\in\{r,g,b\}}\left|\frac{\partial I_{1_c}}{\partial x}\right|\Bigg\} \odot \left|\frac{\partial^k F_1}{\partial x^k}\right| + \exp\Bigg\{-\frac{\lambda}{3}\sum_{c\in\{r,g,b\}}\left|\frac{\partial I_{1_c}}{\partial y}\right|\Bigg\} \odot \left|\frac{\partial^k F_1}{\partial y^k}\right|\Bigg\|_1,$$

where λ is a scalar, k is the smoothness order, I_{1_c} is the c-th channel of I_1 and the exponential is applied entry-wise. With this smoothness term, pixels with low gradient norms make the exponential high and thereby the gradient of the flow is encouraged to take small values. Put differently, pixels belonging to the same object (i.e., with low gradients) should convey similar motion fields; this behavior is disabled on highly textured areas and object boundaries.

Self-supervised loss. A specific solution addresses the issue of pixels that go out of the image frame (boundary occlusions). It relies on the supervision of a student prediction F_1^S by a teacher prediction F_1^T considered as pseudo-ground truth. The teacher prediction F_1^T is obtained by passing through the network ψ_θ images I_1 and I_2 without any data augmentation. The same images are then cropped and augmented with photometric data augmentation only. These images are used as inputs to predict F_1^S. The self-supervision loss involves c the generalized Charbonnier function [39, 40], its expression is:

$$\mathcal{L}_{self1}(\theta) = \frac{1}{HW}||c(F_1^T, F_1^S)||_1,$$

3.2. Brightness Change Correction

The aforementioned losses enable to train an optical flow estimation model in an unsupervised way with specific solutions to deal with occluded pixels. However, the handling of brightness changes in the loss remains overlooked despite their impact on performances as shown in figure 2. Indeed, the soft census loss, while robust, is still sensitive to some brightness changes, misleading the photometric loss that would interpret them as errors in the optical flow estimation. It concerns mainly strong brightness changes, those that induce over/underexposure or very complex ones due to shadows for instance. To address this weakness, we propose BrightFlow, a new optical flow framework that handles brightness changes with no supervision (see figure 3). This method can be built on top of any optical flow network with

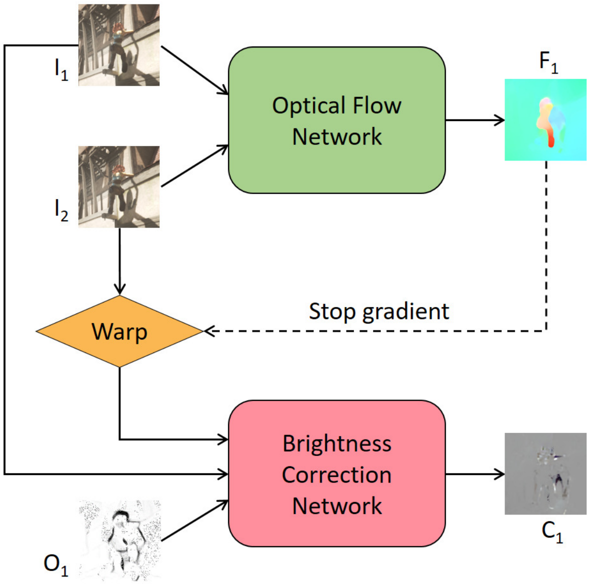


Figure 4: Detailed functioning of the "Optical flow & brightness correction network" block in figure 3. The optical flow network takes as input images I_1 and I_2 to predict optical flow F_1. To return the correction map C_1, the brightness correction network is fed with image I_1, the warped image I_2 with optical flow F_1 and occlusion map O_1.

no impact on its architecture, so at inference, its properties remain the same in terms of computational cost and memory consumption for better performances. The method consists in jointly learning optical flow and a pixel-wise brightness change correction map. The corrections are then used in the photometric loss to compensate for brightness changes. The goal is to prevent the photometric loss to raise reconstruction errors that are due to brightness changes when the flow is well-estimated. To the best of our knowledge, this is the first method with a dynamic photometric loss handling brightness changes.

3.2.1 Brightness Correction Network

The brightness correction network models brightness changes between successive frames. This module, denoted ϕ_{θ_c}, takes as inputs $(I_1, \hat{I}_1, \mathbf{O}_1)$ with $\hat{I}_1 = w(I_2, F_1)$ and predicts a dense brightness correction map $C_1 \in \mathbb{R}^{H\times W\times 3}$ on the three RGB channels as illustrated in figure 4. C_1 compensates for illumination changes from I_1 to I_2 on non-occluded pixels. One may train BrightFlow by plugging $I_1 + C_1$ in $\mathcal{L}_{ph1}$ (equation 2) instead of I_1. Thus, $C_1 = \hat{I}_1 - I_1$ will minimize $\mathcal{L}_{ph1}$. However, since $\hat{I}_1$ and I_1 are in the inputs of the brightness correction network, the latter simply has to infer the difference between $\hat{I}_1$ and I_1. So at the end, the flow estimator will collapse. To address this issue, we propose to re-write the photometric loss as:

$$\mathcal{L}_{ph1}(\theta, \theta_c) = \|\mathbf{O}_1 \odot \rho(\hat{I}_1^c, I_1)\|_1 / \|\mathbf{O}_1\|_1, \tag{3}$$

being $\hat{I}_1^c = w(I_2 + C_2, F_1)$ instead of $\hat{I}_1 = w(I_2, F_1)$ in equation 2. I_2 is replaced by $I_2 + C_2$ so, the photometric loss requires both forward and backward optical flows. The former (F_1) is directly used in the warping operation in $\mathcal{L}_{ph1}$ while the latter (F_2) is used to get the correction C_2. Now, the trivial solution that minimize $\mathcal{L}_{ph1}$ with only C_2 is $C_2 = w^{-1}(I_1, F_1) - I_2$ which is not accessible to ϕ_{θ_c} since F_1 is not in its inputs. However, when the forward and backward flows are consistent then, $w^{-1}(., F_1) = w(., F_2)$ on non-occluded pixels. Hence, one may expect that the model would favor the consistency of F_1 and F_2 at the expense of their ability to model motion, so that minimizing $\mathcal{L}_{ph1}$ could be achieved by predicting $C_2 = \hat{I}_2 - I_2$. To overcome this issue, making forward-backward consistency impossible with a constraint is counter-productive because the true flows are consistent. So our solution is to detach $\hat{I}_2$ from the computational graph of the model in the inputs of ϕ_{θ_c}. Thus the optimization of ϕ_{θ_c} does not impact and potentially tamper with the optical flow network. Photometric errors raised by equation 3 will be back-propagated based on flow F_1 and corrections C_2 but not on flow F_2. Finally, other elements of the model impact the optical flow network in a way that can be incompatible with an excessive forward-backward consistency. It includes the smoothness and self-supervised loss, and the fact that the photometric loss is applied to every prediction of the optical flow network whereas the brightness changes are only estimated from the last prediction of the optical flow network.

3.2.2 Brightness correction gating

In case of errors in the estimation of F_2, warping image I_1 may not reconstruct properly I_2 even in non-occluded areas. Therefore, pixels that do not match would have the same coordinate in I_2 and $\hat{I}_2$. Such inputs could deceive the brightness correction network leading to errors in the correction map C_2. To mitigate their impact in the loss $\mathcal{L}_{ph1}$ (equation 3), only pixels which minimize a reconstruction error ρ' are kept. However, masking poorly estimated corrections prevents their penalization in the loss and so the proper training of the correction estimator. This is why another loss $\mathcal{L}_{ph1}^c$ is added to the previously described photometric loss for the flow estimator (renamed $\mathcal{L}_{ph1}^f$). This new photometric loss is specifically dedicated to the optimization of the corrections estimator. To this end, a gradient stopping is applied to the optical flows F_1 used in the warping of $\hat{I}_1^c$. The final photometric loss with correction of brightness changes is:

$$\mathcal{L}_{ph1}(\theta, \theta_c) = \mathcal{L}_{ph1}^f(\theta) + \gamma_{ph}^c \mathcal{L}_{ph1}^c(\theta_c) \tag{4}$$

Dataset	Sintel clean		Sintel final		KITTI	
Architecture	EPE	ER	EPE	ER	EPE	ER
RAFT	3.93	8.24	3.97	11.22	**2.87**	8.39
RAFT + BrightFlow (Ours)	**3.25**	**7.49**	**3.33**	**10.26**	2.88	**7.98**
GMA	**3.20**	7.42	3.66	10.52	3.47	8.73
GMA + BrightFlow (Ours)	3.24	**7.09**	**3.44**	**10.02**	**3.24**	**8.23**
SCV	3.40	6.77	3.84	10.32	5.00	10.62
SCV + BrightFlow (Ours)	**3.28**	**6.74**	**3.71**	**10.29**	**4.41**	**9.85**

Table 1: Comparison of performances of unsupervised learning without and with our brightness correction network on RAFT [43], GMA [17] and SCV [18] architectures for Sintel and KITTI datasets

$$\mathcal{L}^f_{ph1}(\theta) = \|\mathbf{O}_1 \odot \rho(\mathbf{M}_1 \odot \hat{I}^c_1 + \overline{\mathbf{M}}_1 \odot \hat{I}_1, I_1)\|_1 / \|\mathbf{O}_1\|_1 \quad (5)$$

being $\mathbf{M}_1 = \mathbb{1}_{\{\rho'(\hat{I}^c_1, I_1) \leq \rho'(\hat{I}_1, I_1)\}}$ (with the indicator function $\mathbb{1}_{\{.\leq.\}}$ applied entrywise), $\overline{\mathbf{M}}_1$ its complement, and

$$\mathcal{L}^c_{ph1}(\theta_c) = \|\mathbf{O}_1 \odot \rho'(\hat{I}^c_1, I_1)\|_1 / \|\mathbf{O}_1\|_1. \quad (6)$$

Again ρ is the soft census loss and ρ' corresponds to the ℓ_1-norm in practice. Despite this selection mechanism, some corrections could still be over-estimated leading some pixel intensities to exceed the range values of normal images. To prevent this behavior without impacting the optimization of the brightness correction network, pixel values of the corrected images are clipped in the photometric loss of the optical flow estimator $\mathcal{L}^f_{ph1}$ (equation 5). Also, in practice, $\mathcal{L}^c_{ph1}$ uses augmented images but not $\mathcal{L}^f_{ph1}$ (for more details see BrightFlow pseudo-code in the supplementary material).

4. Experiments

4.1. Datasets

We evaluate the performances of our method on standard datasets, namely Sintel [7], KITTI 2015 [30] and HD1K which exhibit strong brightness changes. These datasets only provide ground truth for the training data. So, we interchanged train and test sets as commonly operated in the related work [19, 37]. Like SMURF [37] dimensions of input images are 296 × 696 for KITTI, and 368 × 496 for Sintel; evaluation is performed at the original image dimensions.

4.2. Implementation Details

We apply BrightFlow on top of the three following architectures of optical flow network: RAFT [43], GMA [17] and SCV [18]. The brightness correction network architecture includes an encoder and an upsampler taken from RAFT to return correction maps at the same resolution as input images. We conducted all experiments from scratch (baselines and training with BrightFlow). We use the same data augmentation as RAFT which includes spatial augmentations (flipping, stretching, rescaling, cropping) and photometric augmentations (random variation of brightness, contrast, saturation and hue) that can be carried independently for each input image. Occlusion masks are estimated with Wang's range-map method [45] for Sintel and Brox's method [4] for KITTI. Similarly to SMURF [37], models are optimized for 75k iterations with a batch size settled to 8 and Adam [20] optimizer. The learning rate initially set to 2×10^{-4} decays exponentially until 2×10^{-7} over the last 20% of the total number of iterations.

The optical flow network is first pretrained for 20k iterations (till reaching decent optical flow performances). Then, for 5k iterations, we include the brightness correction network in the training without applying its corrections in the photometric loss $\mathcal{L}^f_{ph}$ to initialize the brightness correction network. From step 25k we use the corrections in the photometric loss $\mathcal{L}^f_{ph}$. In all the experiments, $\gamma^f_{ph} = 1$ and $\gamma^f_{ph} = 0.1$ unless otherwise stated. The coefficient γ_{self} is set to 0 over the first 40% iterations then is increased linearly to 0.3 during the subsequent 10% iterations and remains constant afterward. Hyperparameters of the smoothness loss depend on the dataset: 1st order smoothness and $\gamma_{sm} = 2.5$ for Sintel; 2nd order smoothness and $\gamma_{sm} = 4$ for KITTI. Only the edge sensibility remains constant: $\lambda = 150$. All losses but $\mathcal{L}^c_{ph}$ are applied on every flow prediction of RAFT, GMA or SCV in the manner of SMURF sequence losses [37]. $\mathcal{L}^c_{ph}$ is only applied with the last optical flow prediction.

4.3. Results

We evaluate the performances of our method using the average End-Point-Error (EPE) in pixel, and Error Rate (ER) in percent. With the latter measure, a flow estimation is considered erroneous at a given pixel if its distance exceeds 3 pixels or 5% w.r.t. its ground truth.

Quantitative results are summarized in table 1, we also provide qualitative results in figure 5 and the supplementary material. It shows that training the RAFT, GMA or SCV optical flow architectures as a part of BrightFlow provide better results than training these networks without brightness correction. This is observed on synthetic datasets like

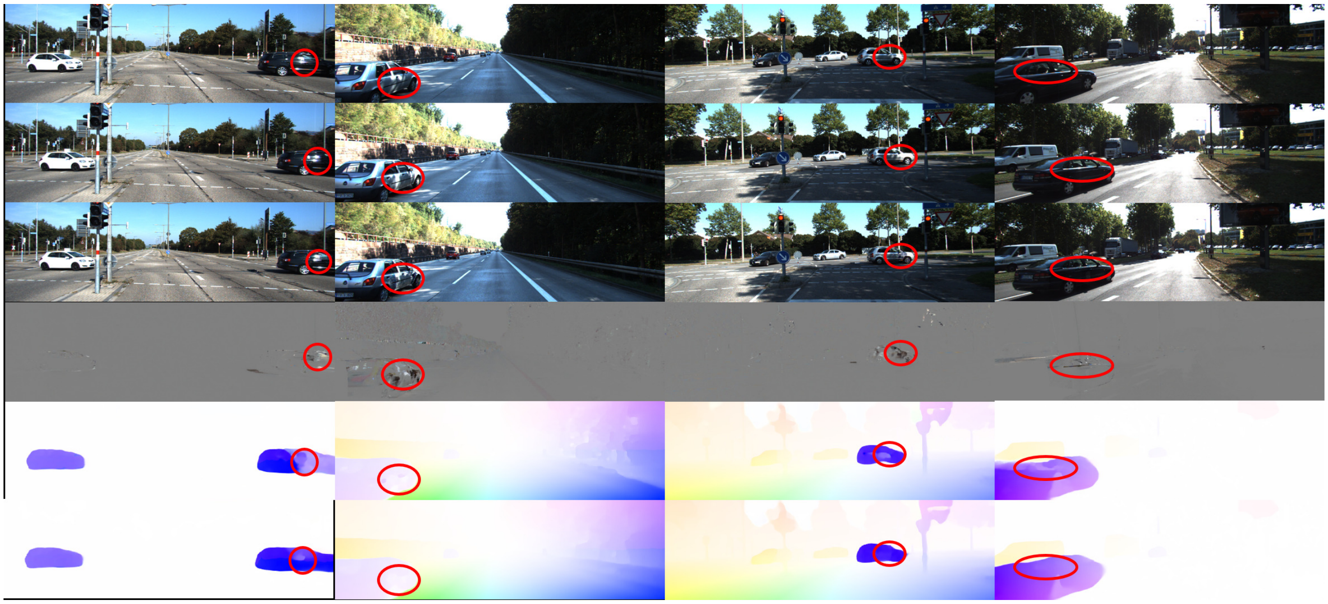


Figure 5: Qualitative results for RAFT trained with or without BrightFlow on KITTI (best viewed in color). From top to bottom, images are I_1, I_2, $I_2 + C_2$, C_2, F_1 predicted by RAFT trained without BrightFlow and then with BrightFlow.

Sintel as well as on photo-realistic data including KITTI. On average BrightFlow enhances unsupervised learning by 7% in EPE and 5% in ER with an improvement peak with RAFT of 11% in EPE and 8% in ER. Two configurations (KITTI+RAFT and Sintel+GAM) perform slightly worse with BrightFlow in terms of EPE metric. It is easily explainable as we conducted our ablation study on Sintel dataset with RAFT architecture only (see 4.4), the hyperparameter γ^c_{ph} may not be the best for other configurations even if at least it always enables to improve ER metrics.

It is worth noticing that the performance ranking of the optical flow architectures with unsupervised training differs from results reported in the related work when their training is supervised (see for instance [18, 17]). Indeed, GMA does not outperform the other architectures on all benchmarks; GMA overtakes the others on synthetic data (Sintel) but undertakes RAFT on the photo-realistic data (KITTI). Likewise, the ranking of SCV against the other architectures is also disparate whereas its performances are expected to be inferior to RAFT and GMA.

Figure 6 illustrates the impact of the brightness correction network on performances depending on different amounts of brightness changes. Whatever their magnitude, training the model with the brightness correction network improves performance. The gain is further amplified on pixels with higher brightness changes. This clearly shows the ability of our method to *bridge the accuracy gap* between pixels that satisfy brightness consistency and the other (more challenging) pixels.

We present the cross-domain generalization results in table 2. It shows that BrightFlow is still beneficial w.r.t a standard unsupervised training of optical flow network. The average improvement of 2% is observed for both EPE and ER. This corroborates the robustness of our method on a more difficult task namely cross dataset evaluation.

4.4. Ablation Study

In this section, we study the impact of each component of BrightFlow on performance, including the brightness correction network, the gradient stopping applied to its inputs, the clipping of corrected images in the photometric loss $\mathcal{L}^f_{ph1}$ and the brightness correction gating mechanism. In order to show their benefit on our method, several trainings have been conducted with each contribution added one at a time. The impact of all these components on the performances is shown in table 3. According to the observed results, leveraging brightness changes in the photometric loss already outperforms the setting without corrections. Besides, when the gradient stopping, the clipping of corrected images and the gating mechanism are enabled, extra gains are also observed. A study for different values of γ^c_{ph} is also provided in table 4.

In order to get a better understanding of what the brightness correction network learns we conducted extra experiments (see supplementary material for more details). As none of these experiments exceed our method, these results confirm the importance of data-augmentation and that the brightness correction network does not simply learn the trivial solution of section 3.2.1 nor a threshold to filter cen-

Optical flow Architecture	BrightFlow	KITTI⟶Sintel clean EPE	 ER	KITTI⟶Sintel final EPE	 ER	KITTI⟶HD1K EPE	 ER	Sintel⟶KITTI EPE	 ER
RAFT		**3.49**	8.38	**3.97**	**11.22**	1.17	5.12	15.12	23.63
RAFT	✓	3.82	**8.35**	4.67	11.76	**1.04**	**4.93**	**13.69**	**23.32**
GMA		4.05	8.61	4.71	12.04	1.25	5.40	15.92	24.56
GMA	✓	**3.82**	**8.38**	**4.53**	**11.6**	**1.18**	**5.25**	**13.88**	**23.57**
SCV		**3.25**	7.53	4.75	12.38	1.19	4.97	**16.81**	**23.56**
SCV	✓	3.31	**7.46**	**4.65**	**12.15**	**1.12**	**4.54**	17.13	23.74

Table 2: Cross-domain generalization. These results compare the ability of optical flow networks trained with or without BrightFlow to make predictions on frames from a different dataset rather than the training one.

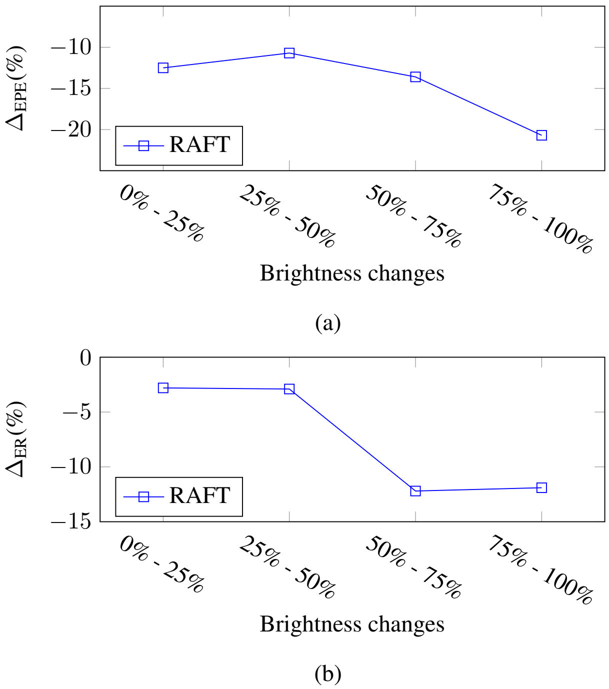


Figure 6: Relative gain in %EPE (a) and ER (b) as a function of brightness change when training RAFT with BrightFlow on Sintel dataset. All non-occluded pixels of final Sintel have been divided into four groups depending on the magnitude of brightness change (from the 25% pixels with the lowest to the 25% pixels with the highest ones).

sus loss outliers. Following these results, we believe that it learns to recognize situations where presumed brightness changes are real (due to shadows or reflections) or errors in the flow estimation and how to handle them.

5. Conclusion

We introduced in this paper BrightFlow, a novel unsupervised method that trains deep neural networks for optical flow estimation. The strength of the proposed method resides in its ability to model brightness changes. An optical flow network is jointly trained with the brightness correction network which model photometric discrepancies due to shadows and reflections to compensate them in the loss. As the latter network is used during training only, it has no impact on time/memory footprint of optical flow estimation during inference. Extensive experiments, conducted on standard datasets, highlight the effectiveness and the consistent gain of our proposed solution w.r.t. the related works.

				Sintel clean		Sintel final	
BCN	GS	CCI	GM	EPE	ER	EPE	ER
				3.93	8.24	3.97	11.22
✓				3.87	7.94	3.67	10.70
✓	✓			3.58	7.81	3.47	10.75
✓	✓	✓		3.78	7.89	3.41	10.39
✓	✓		✓	3.57	7.83	3.36	10.34
✓	✓	✓	✓	**3.25**	**7.49**	**3.33**	**10.26**

Table 3: Impact of each component of our method. It includes the brightness correction network (BCN), the gradient stopping on its inputs (GS), the clipping of corrected images in the photometric loss of the flow (CCI) and the brightness correction gating mechanism (GM). These experiments have been carried out with RAFT as optical flow network on Sintel Dataset.

γ_{ph}^c	Sintel clean EPE	 ER	Sintel final EPE	 ER
0.01	3.66	7.64	3.38	**10.22**
0.1	**3.25**	**7.49**	**3.33**	10.26
1	3.51	7.65	3.43	10.48
10	3.30	7.92	3.43	10.78

Table 4: Effect of varying γ_{ph}^c on performances of RAFT trained with BrightFlow on Sintel dataset.

6. Acknowledgment

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council.

References

[1] Khalil Bergaoui, Yassine Naji, Aleksandr Setkov, Angélique Loesch, Michèle Gouiffès, and Romaric Audigier. Object-centric and memory-guided normality reconstruction for video anomaly detection. *ICIP*, 2022.

[2] Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)*, 39(4):148–1, 2020.

[3] Michael J Black and Padmanabhan Anandan. Robust dynamic motion estimation over time. In *CVPR*, 1991.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[5] 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, 2010.

[6] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005.

[7] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *SSST-8*, 2014.

[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 *ICCV*, 2015.

[10] David Hafner, Oliver Demetz, and Joachim Weickert. Why is the census transform good for robust optic flow computation? In *SSVM*, 2013.

[11] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *NeurIPS*, 2020.

[12] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018.

[13] 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 *CVPR*, 2017.

[14] 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 *CVPR*, 2017.

[15] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022.

[16] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *ECCV*, 2018.

[17] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021.

[18] Shihao Jiang, Yao Lu, Hongdong Li, and Richard Hartley. Learning optical flow from a few matches. In *CVPR*, 2021.

[19] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *ECCV*, 2020.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[21] 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 *CVPR*, 2020.

[22] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *AAAI*, 2019.

[23] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *CVPR*, 2019.

[24] Shuaicheng Liu, Kunming Luo, Nianjin Ye, Chuan Wang, Jue Wang, and Bing Zeng. Oiflow: Occlusion-inpainting optical flow estimation by unsupervised learning. *IEEE Transactions on Image Processing*, 30:6420–6433, 2021.

[25] W. Liu, D. Lian W. Luo, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[27] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *CVPR*, 2021.

[28] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[29] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018.

[30] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[32] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

[33] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017.

[34] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, 2017.

[35] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[36] Fridtjof Stein. Efficient computation of optical flow using the census transform. In *Joint Pattern Recognition Symposium*, 2004.

[37] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *CVPR*, 2021.

[38] 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 *CVPR*, 2022.

[39] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[40] 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*, 106(2):115–137, 2014.

[41] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *CVPR*, 2021.

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[44] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. *NeurIPS*, 2020.

[45] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, June 2018.

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[47] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022.

[48] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[49] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *NeurIPS*, 2019.

[50] Yanchao Yang, Brian Lai, and Stefano Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *CVPR*, 2021.

[51] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *CVPR*, 2019.

[52] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, 2016.

[53] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994.

[54] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021.

[55] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *CVPR*, 2020.

[56] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022.