Joint Unsupervised Learning of Optical Flow and Egomotion with Bi-Level Optimization

Shihao Jiang[1,2,3], Dylan Campbell[1,2], Miaomiao Liu[1,2], Stephen Gould[1,2], Richard Hartley[1,2]
[1]Australian National University, [2]Australian Centre for Robotic Vision, [3]Data61, CSIRO

Abstract

We address the problem of joint optical flow and camera motion estimation in rigid scenes by incorporating geometric constraints into an unsupervised deep learning framework. Unlike existing approaches which rely on brightness constancy and local smoothness for optical flow estimation, we exploit the global relationship between optical flow and camera motion using epipolar geometry. In particular, we formulate the prediction of optical flow and camera motion as a bi-level optimization problem, consisting of an upper-level problem to estimate the flow that conforms to the predicted camera motion, and a lower-level problem to estimate the camera motion given the predicted optical flow. We use implicit differentiation to enable back-propagation through the lower-level geometric optimization layer independent of its implementation, allowing end-to-end training of the network. With globally-enforced geometric constraints, we are able to improve the quality of the estimated optical flow in challenging scenarios, and obtain better camera motion estimates compared to other unsupervised learning methods.

1. Introduction

Dense optical flow estimation is a fundamental problem in computer vision for determining the apparent motion of pixels in an image as the camera and scene moves. It has broad applications in action recognition [38], 3D reconstruction [24], and motion segmentation [31]. The seminal work by Horn and Schunck [21] sets the foundation for solving optical flow estimation problems by enforcing brightness constancy and local smoothness constraints in a variational setting. In the last few decades, the quality of optical flow estimation has improved dramatically with the introduction of ideas such as piece-wise smoothness [5], coarse-to-fine refinement for large displacements [6], and layered formulations for handling occlusions [47].

Like many problems in computer vision, approaches based on the supervised learning of convolutional neural networks (CNNs) now achieve the state-of-the-art results for optical flow estimation [9, 22, 42]. However, the difficulty of obtaining large volumes of ground-truth optical flow limits the applicability of these approaches in many scenarios. Unsupervised learning approaches are a promising alternative, which encode brightness constancy and local smoothness constraints in a loss function for training deep networks [23]. While such constraints perform well in feature-rich regions, they often fail on challenging scenes with featureless or repetitively-textured regions.

To achieve more robust prediction and handle these cases more effectively, we exploit the geometric constraint between the optical flow and camera motion in an unsupervised learning framework. Specifically, we focus on optical flow estimation in mostly rigid scenes, where optical flow is predominantly caused by camera motion [41, 46]. Thus the displacement of corresponding pixels across images, i.e., the optical flow, satisfies the well-known epipolar constraint [20]. We formulate this as an *epipolar geometric loss* defined on an essential matrix determined from the optical flow. Compared to the fundamental matrix used in existing work [53], the essential matrix provides tighter constraints and can be obtained from state-of-the-art geometric algorithms [19] that provide more accurate camera motion estimates than, for example, the 8-point algorithm [20].

To compute the epipolar geometric loss we first require an estimate of the camera motion. As such, we formulate flow estimation as a bi-level optimization problem. The upper-level problem is to estimate the optical flow by minimizing the epipolar geometric loss as well as enforcing the standard brightness constancy constraint. The geometric loss is defined based on an *essential matrix* encoding of camera motion, which is obtained by solving a lower-level optimization problem that estimates the camera motion from the optical flow. To enable end-to-end training, we use implicit differentiation to back-propagate the gradient of the upper-level loss through the essential matrix estimation layer (the lower-level problem). An overview of our training pipeline is shown in Figure 1.

Overall, our key technical contributions include: (1) the introduction of a geometric constraint into an unsupervised deep learning framework for optical flow and camera mo-

tion estimation, and (2) the formulation of an end-to-end trainable model with an embedded optimization layer that estimates the camera motion required for computing the epipolar loss. Importantly, our formulation allows back-propagation through the optimization layer regardless of the algorithmic implementation used for computing the essential matrix. We show that our geometrically-constrained model can accurately estimate optical flow satisfying the epipolar geometry. Our optical flow estimation method outperforms approaches that ignore geometry and produces remarkably good results on cases that previous approaches find challenging, such as featureless regions and regions with repetitive features. Our camera motion estimation also compares favourably against methods that directly use a network to predict camera poses.

2. Related Work

Learning-based optical flow estimation. Optical flow estimation has been studied extensively since the pioneering work of Horn and Schunck [21]. The reader is directed to Fortun et al. [12] and Sun et al. [41] for comprehensive reviews of the literature. Recent approaches formulate optical flow estimation as a supervised deep learning task. Compared to traditional methods, convolutional neural networks (CNNs) have the advantage of fast inference once trained, making real-time prediction possible. For example, PWC-Net [42], building on previous supervised optical flow networks [9, 22], incorporated traditional optical flow techniques into the network such as cost volumes and feature warping, and achieved better performance than traditional methods with a shorter running time.

Despite strong results, supervised deep learning approaches are limited by the need for ground-truth optical flow during training, which is difficult to obtain for real-world scenes. Unsupervised learning instead allows the network to be trained on large volumes of unlabelled data. Several unsupervised loss functions have been proposed, including photometric constancy and local smoothness losses [23], and occlusion-aware bidirectional consistency and robust census losses [29]. Other works explicitly reason about occlusion [44], or synthetically augment data for better occlusion estimation [27, 28]. All of these approaches rely on local matching so cannot handle smooth image regions. Our work is built on some of these previous works, but also enforces global geometric consistency, allowing it to better handle smooth, featureless regions.

Optical Flow and Epipolar Geometry. There has been an extensive study on the relationship between optical flow and epipolar geometry and their applications. Weber and Malik [45] first applied epipolar geometry to estimate and track independently moving objects from optical flow. Early works have also been reviewed in the book by Xu and Zhang [48], which propose to look at the correspondence problem from the standpoint of epipolar geometry. Difficult 2-D search problem can be simplified to a 1-D search problem under the assumption that the epipolar geometry is known *a priori*, which is also demonstrated later by Yamaguchi et al. [49] on the problem of rigid scene optical flow estimation. In contrast, rather than treating the estimated epipolar geometry as known *a priori* and imposing a hard constraint, we propose a soft constraint between optical flow and epipolar geometry, and a joint optimization approach between the two in a deep learning context. Our idea is similar to [43] in that we couple the estimation of epipolar geometry and optical flow to solve a joint optimization problem. However, we propose a method that can be end-to-end trainable in an unsupervised learning framework.

Recently, there have been works that leverage epipolar geometry in unsupervised learning framework [3, 53] with the aim of handling multiple motions. We instead focus on static scenes and demonstrate how to back-propagate the gradients of the loss function through the geometric estimation layer and train the network in an end-to-end manner.

Unsupervised Learning of Camera Motion. Another line of works that has gained popularity in recent years is unsupervised learning of depth and motion from videos since the first work of Zhou et al. [55]. Subsequent works adopted the idea of joint learning of flow, motion and depth and witnessed marginal improvements at the cost of large network parameters [50, 56]. All previous works use a camera motion network to directly regress motion from two images, whereas we propose a hybrid method: asking a network to infer dense correspondences (optical flow), and then use optimization to solve for the camera motion. This hybrid learning idea has also been investigated by Zhou et al. [54] in the context of visual localization.

Differentiable optimization. A few recent works embed optimization problems as layers within a deep learning model [2, 10, 25, 34]. These layers encode complex dependencies and constraints that cannot be easily learned by convolution or fully-connected layers. It is apparently non-trivial to back-propagate gradients through these layers. However, Gould et al. [17, 18] addressed this problem and provided general techniques for differentiating argmin and argmax problems (both constrained and unconstrained). Amos et al. [2] proposed a differentiable optimization layer but was limited to quadratic programs. These techniques relating to differentiable optimization have been used to solve several computer vision problems, such as video classification [10] and meta-learning [25, 34]. In this work, we embed an epipolar geometric constraint into a deep learning framework via bi-level optimization, jointly estimating camera motion and optical flow in an unsupervised fashion.

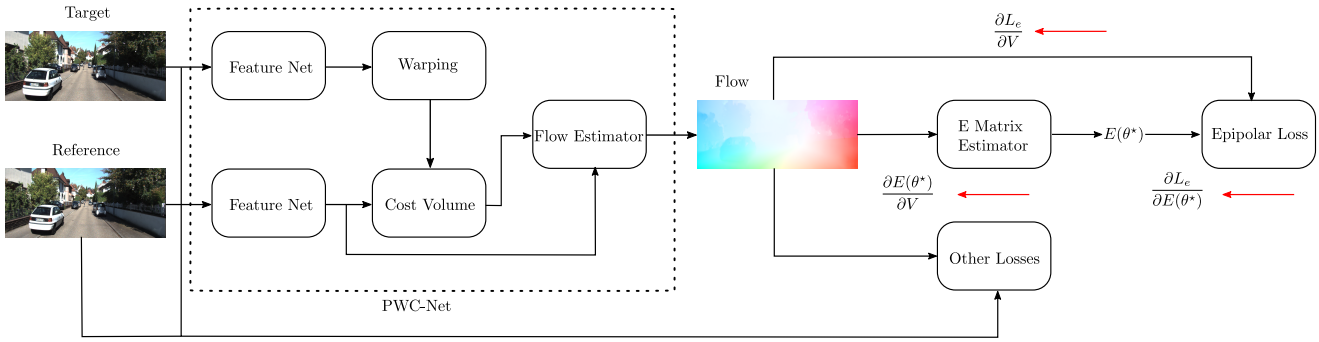


Figure 1. Overall architecture. The dashed block contains the PWC-Net architecture that we use in this work, which can also be replaced with other network architectures. The epipolar loss $L_e(\mathtt{V}, \mathtt{E}(\theta^\star))$ takes input of both the optical flow $\mathtt{V}$ and the estimated essential matrix $\mathtt{E}(\theta^\star)$. The red arrows denote the gradient flow of the proposed loss.

3. Bi-Level Optimization for Optical Flow and Essential Matrix Estimation

We define optical flow as a dense field of displacement vectors, where the displacement vector at each pixel coordinate in one image points to the coordinate of the corresponding pixel in another image. Let a pixel coordinate in image $I \in \mathbb{R}^{W \times H \times 3}$ be denoted by $\mathbf{p} = (u, v)$ and the corresponding pixel in the other image I' by $\mathbf{p}' = (u', v')$. We assume that I and I' are two views of the same scene, typically consecutive frames from a video sequence. The optical flow between I and I' is then the matrix $\mathtt{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N] \in \mathbb{R}^{2 \times N}$ of displacement vectors $\mathbf{v}_i = \mathbf{p}'_i - \mathbf{p}_i$ for every pixel $\mathbf{p}_i$ in image I and corresponding pixel $\mathbf{p}'_i$ in image I'. Here $N = WH$ is the total number of pixels in image I.

Given the camera intrinsic calibration matrices $\mathtt{K}$ and $\mathtt{K}'$ for the image pair I and I', we can obtain the normalized coordinates as $\mathbf{x} = \mathtt{K}^{-1}\tilde{\mathbf{p}}$ and $\mathbf{x}' = \mathtt{K}'^{-1}\tilde{\mathbf{p}}'$, where $\tilde{\mathbf{p}} = (u, v, 1)^\top$ is the pixel $\mathbf{p}$ expressed in homogeneous coordinates. Corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in normalized coordinates satisfy the geometric relationship $\mathbf{x}'^\top \mathtt{E}\mathbf{x} = 0$, known as the *epipolar constraint*. For camera matrices $\mathtt{P} = \mathtt{K}[\mathtt{I}|0]$ and $\mathtt{P}' = \mathtt{K}'[\mathtt{R}|\mathbf{t}]$, the essential matrix can be decomposed into rotation $\mathtt{R}$ and translation $\mathbf{t}$ components as $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ with $\mathbf{t}$ known up to scale [20]. We address the problem of incorporating this epipolar constraint into an optimization procedure for deep optical flow estimation.

3.1. Optimizing an Epipolar Loss Function

We formulate optical flow estimation as a bi-level optimization problem, with an upper-level problem that is solved subject to constraints enforced by a lower-level problem, given by

$$\underset{\mathtt{V}}{\text{minimize}} \;\; L(\mathtt{V}, \boldsymbol{\theta}^\star) \tag{1}$$

$$\text{subject to} \;\; \boldsymbol{\theta}^\star \in \underset{\boldsymbol{\theta} \in \mathbb{R}^5}{\arg\min}\, l(\mathtt{V}, \boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^5$ is the minimal parametrization of $\mathtt{E}$ given in Hartley & Li [19]. The upper-level loss function L comprises several terms, which is fully described in Section 4. To encourage the optical flow to satisfy the epipolar geometry, one of the terms in L is the one-sided epipolar error, given by the global geometric loss function

$$L_{\mathrm{e}}(\mathtt{V}, \boldsymbol{\theta}^\star) = \sum_{i=1}^{N} \frac{(\mathbf{x}_i'^\top \mathtt{E}(\boldsymbol{\theta}^\star)\mathbf{x}_i)^2}{[\mathtt{E}(\boldsymbol{\theta}^\star)\mathbf{x}_i]_1^2 + [\mathtt{E}(\boldsymbol{\theta}^\star)\mathbf{x}_i]_2^2} \tag{3}$$

where the essential matrix $\mathtt{E}$ is a function of its parameters $\boldsymbol{\theta}^\star$ and $[\mathtt{E}\mathbf{x}]_j$ denotes the j^{th} component of the 3-vector $\mathtt{E}\mathbf{x}$. This error measures the sum squared distance of each point $\mathbf{x}'$ to its corresponding epipolar line $\mathtt{E}\mathbf{x}$. Note that $\mathbf{x}_i$ is constant and $\mathbf{x}'_i = \mathtt{K}'^{-1}(\tilde{\mathbf{p}}_i + \tilde{\mathbf{v}}_i)$ is a function of the optical flow $\mathtt{V}$. The optimal essential matrix parameters $\boldsymbol{\theta}^\star$ also depend on the optical flow $\mathtt{V}$. We formulate essential matrix estimation as a lower-level optimization problem (2), with a robust algebraic error objective function given by

$$l(\mathtt{V}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \rho\left(\mathbf{x}_i'^\top \mathtt{E}(\boldsymbol{\theta})\mathbf{x}_i\right), \tag{4}$$

$$\rho(z; \delta) = \begin{cases} \frac{1}{2}z^2, & \text{if } |z| < \delta \\ \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{5}$$

The robust function $\rho(\cdot)$ is a truncated L_2 penalty function with an inlier threshold δ.

To solve the bi-level optimization problem (1) within a deep learning context, we need to back-propagate gradients through the essential matrix optimization layer. During the forward pass, for each image pair we solve problem (2) using iteratively re-weighted least squares (IRLS) to minimize the robust objective function l (4). However, this function is non-convex with many local minima. Hence, we first obtain a robust initial estimate of the essential matrix parameters using RANSAC [11] with the five-point algorithm [26, 32], which we have implemented as an efficient GPU routine.

During the backward pass we need to compute $\mathrm{d}\boldsymbol{\theta}^\star/\mathrm{d}\mathtt{V}$, which amounts to differentiating the $\arg\min$ function. This can be achieved using implicit differentiation described below. As we will see, the gradient computation is agnostic to the method used to solve the lower-level problem, and only requires that a solution be found. Importantly, this means that we do not need to back-propagate through the specific steps of the optimization algorithm.

3.2. Implicit Differentiation

Robustly estimating the essential matrix—via the lower-level optimization problem in (2)—does not have an analytic solution and involves a non-differentiable RANSAC procedure to mitigate the effect of outliers. Obtaining the gradient would therefore not be possible using explicit differentiation or direct automatic differentiation. However, since the objective function of the lower-level optimization problem in (2) is twice-differentiable, we can compute the gradient of the $\arg\min$ function using implicit differentiation knowing only the optimal solution (and not how it was obtained) [17, 37, 7, 33]. The key result, a special case of Dini's implicit function theorem [8, p19] applied to the optimality condition $\mathrm{d}f(x,\mathbf{y})/\mathrm{d}\mathbf{y} = \mathbf{0}$, is given below for completeness.

Lemma 1: (Gould et al. [17]) Let $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ be a continuous function with first and second derivatives. Set $\mathbf{g}(x)$ to be a stationary point of $f(x,\mathbf{y})$ with respect to $\mathbf{y}$, for example $\mathbf{g}(x) \in \arg\min_{\mathbf{y}\in\mathbb{R}^n} f(x,\mathbf{y})$, and let the Hessian $f_{YY}(x,\mathbf{g}(x))$ be nonsingular. Then the vector derivative of $\mathbf{g}$ with respect to x is

$$\frac{\mathrm{d}\mathbf{g}}{\mathrm{d}x} = -f_{YY}(x,\mathbf{g}(x))^{-1} f_{XY}(x,\mathbf{g}(x)) \tag{6}$$

where $f_{YY}(x,\mathbf{y}) \doteq \frac{\partial^2 f(x,\mathbf{y})}{\partial \mathbf{y}^2} \in \mathbb{R}^{n\times n}$ and $f_{XY}(x,\mathbf{y}) \doteq \frac{\partial^2 f(x,\mathbf{y})}{\partial x \partial \mathbf{y}} \in \mathbb{R}^n$.

Proof. The derivative of f with respect to $\mathbf{y}$, evaluated at the stationary point $\mathbf{g}(x)$, is zero by definition. The result follows by differentiating both sides with respect to x using the chain rule, i.e.,

$$f_Y(x,\mathbf{g}(x)) \doteq \frac{\mathrm{d}f(x,\mathbf{y})}{\mathrm{d}\mathbf{y}}\Big|_{\mathbf{y}=\mathbf{g}(x)} = \mathbf{0} \tag{7}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} f_Y(x,\mathbf{g}(x)) = \mathbf{0} \tag{8}$$

$$\therefore\ f_{XY}(x,\mathbf{g}(x)) + f_{YY}(x,\mathbf{g}(x))\frac{\mathrm{d}\mathbf{g}}{\mathrm{d}x} = \mathbf{0} \tag{9}$$

and rearranging the terms. □

For brevity we have shown the derivative with respect to a single parameter. For multiple parameters, the derivative can be computed with respect to each parameter separately. Here, the matrix f_{YY} need only be inverted or decomposed once for the full set of parameters. It is also worth noting that the gradient is valid for any stationary point $\mathbf{g}(x)$ of f, including local minima, maxima and saddle points.

Applied to the problem under consideration, we get

$$\frac{\mathrm{d}\boldsymbol{\theta}^\star}{\mathrm{d}\mathtt{V}} = -\left(\frac{\partial^2 l(\mathtt{V},\boldsymbol{\theta}^\star(\mathtt{V}))}{\partial \Theta^2}\right)^{-1} \frac{\partial^2 l(\mathtt{V},\boldsymbol{\theta}^\star(\mathtt{V}))}{\partial V \partial \Theta}. \tag{10}$$

Automatic differentiation, such as the Autograd package in PyTorch, can be used to compute the necessary Jacobian and Hessian matrices. Observe from (10) that although we require $\boldsymbol{\theta}^\star$ to be a stationary point of l, the computation of the gradient is independent of the algorithmic steps used to determine $\boldsymbol{\theta}^\star$ and hence $\mathtt{E}$. Thus to be clear, automatic differentiation, if used, is applied to the objective function itself and not the algorithmic procedure used to find its minimum, different from standard usage in deep learning models. The total derivative of L in problem (1) is then

$$\frac{\mathrm{d}L(\mathtt{V},\boldsymbol{\theta}^\star(\mathtt{V}))}{\mathrm{d}\mathtt{V}} = \frac{\partial L}{\partial \mathtt{V}} + \frac{\partial L}{\partial \boldsymbol{\theta}^\star}\frac{\mathrm{d}\boldsymbol{\theta}^\star}{\mathrm{d}\mathtt{V}} \tag{11}$$

with all terms evaluated at $(\mathtt{V},\boldsymbol{\theta}^\star)$. This defines the exact gradient of loss function L constrained by (2).

4. Unsupervised Optical Flow Estimation

Now that we have a means of incorporating a global geometric loss function and a geometric estimation layer into an end-to-end learning framework, we can present our full training pipeline as shown in Figure 1. At a high level, our network computes the optical flow from a pair of images and then estimates the camera motion using an embedded robust geometric optimization algorithm. The estimated camera motion is used to self-supervise the optical flow network, alongside standard photometric, smoothness and consistency losses. This approach can be used to enhance any state-of-the-art flow estimation network, which is currently the approach of Liu et al. [28]. As such, we use their unsupervised training strategy, which can be viewed as an intelligent data augmentation approach, in order to improve performance in highly occluded scenes.

Loss functions: Similar to previous works, we use brightness constancy and local smoothness constraints by proposing photometric and smoothness losses. Following Meister et al. [29], we apply a ternary census transform $C(\cdot)$ on the input images, which is robust to real-world violations of the brightness constancy constraint [51, 39]. Since brightness constancy does not hold for occluded pixels, we estimate an occlusion map based on the forward-backward consistency prior [29] and only apply the photometric loss on non-occluded pixels. For these pixels, we also apply a forward-backward consistency loss, to encourage consistent optical flow in both directions.

Let M_i indicate whether the i^{th} pixel is non-occluded and let $Z = \sum_{i=1}^{N} M_i$ be the number of non-occluded pixels. We define the photometric loss L_{p} and the forward-backward consistency loss L_{c} as follows:

$$L_{\text{p}} = \frac{1}{Z} \sum_{i=1}^{N} M_i \, \rho_{\text{c}} \left(C(I, \mathbf{p}_i) - C(I', \mathbf{p}_i + \mathbf{v}_i) \right) \quad (12)$$

$$L_{\text{c}} = \frac{1}{Z} \sum_{i=1}^{N} M_i \, \rho_{\text{c}} \left(\mathtt{V}^{\text{f}}(\mathbf{p}_i) + \mathtt{V}^{\text{b}}(\mathbf{p}_i + \mathbf{v}_i) \right) \quad (13)$$

where $\rho_{\text{c}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} (z_i^2 + \epsilon^2)^{\gamma}$ is the element-wise average of the robust generalized Charbonnier penalty function [41] with $\epsilon = 10^{-3}$ and $\gamma = 0.45$, and $\mathtt{V}^{\text{f}}$ and $\mathtt{V}^{\text{b}}$ are the predicted forward and backward optical flow, indexed by image space coordinates.

To encourage the predicted optical flow to be locally smooth, we apply an edge-aware smoothness loss [15], based on the assumption that motion boundaries often coincide with image edges. The smoothness loss is defined

$$L_{\text{s}} = \frac{1}{2N} \sum_{i=1}^{N} \left(\sum_{a \in \{u,v\}} e^{-\frac{\alpha}{3} \left\| \frac{\partial I(\mathbf{p}_i)}{\partial a} \right\|_1} \left\| \frac{\partial \mathbf{v}_i}{\partial a} \right\|_1 \right). \quad (14)$$

Training strategy: We adopt the training strategy of Liu and co-authors [28, 27] by having a teacher network and a student network in order to artificially generate occlusions and apply supervision on these regions. The teacher network is first trained until convergence with the proposed loss functions. The output of this network is then used to supervise the training of a student network, whose weights are initialized from the teacher network. Following Liu et al. [28], we generate occluded regions by computing SLIC superpixels [1] and replacing randomly-selected superpixels with random noise. Another source of generated occlusions comes from randomly cropping the input images [27]. Pixels warped outside of the cropped image frame are considered to be occluded. These artificial occlusions are only used in the student network, where the output of the teacher network is able to supervise the predicted flow. This makes it possible for the student network to learn to estimate flow more accurately in occluded regions.

Let O_i indicate the i^{th} pixel that is occluded in the synthetically-generated image but non-occluded in the original image, and let $Y = \sum_{i=1}^{N} O_i$ denote the number of such pixels. We define the occlusion loss function for training the student network as

$$L_{\text{o}} = \frac{1}{Y} \sum_{i=1}^{N} O_i \, \rho_{\text{c}} \left(\mathtt{V}(\mathbf{p}_i) - \tilde{\mathtt{V}}(\mathbf{p}_i) \right), \quad (15)$$

where $\mathtt{V}$ denotes the flow predicted by the student network and $\tilde{\mathtt{V}}$ denotes the flow predicted by the teacher network.

We also apply multi-scale training at five different resolutions to handle large motions. The epipolar loss is only applied at the highest resolution. With weights $\lambda_\bullet$ on the loss terms, our total loss for the teacher network is

$$L_{\text{t}} = \sum_{i=1}^{5} \lambda^i \left(\lambda_{\text{p}} L_{\text{p}}^i + \lambda_{\text{c}} L_{\text{c}}^i + \lambda_{\text{s}} L_{\text{s}}^i \right) + \lambda_{\text{e}} L_{\text{e}}, \quad (16)$$

and the total loss for the student network is

$$L_{\text{s}} = \sum_{i=1}^{5} \lambda^i \left(\lambda_{\text{p}} L_{\text{p}}^i + \lambda_{\text{c}} L_{\text{c}}^i + \lambda_{\text{s}} L_{\text{s}}^i + \lambda_{\text{o}} L_{\text{o}}^i \right) + \lambda_{\text{e}} L_{\text{e}}. \quad (17)$$

Note that L_{e} refers to the epipolar loss defined in (3).

5. Experiments

Datasets: We evaluate our method for unsupervised optical flow estimation on two datasets: the standard KITTI 2012 Flow dataset [14] and the more challenging RGB-D SLAM dataset [40]. The KITTI dataset [13] contains outdoor road scenes captured by a car-mounted stereo camera rig. Since we are only estimating rigid flow, we evaluate on the KITTI 2012 Flow subset [14], which has 194 training images with sparse ground-truth optical flow and 195 test images. We do not train on this data, instead we use the KITTI Visual Odometry (VO) dataset which has similar characteristics. This has 22 sequences with 87 060 consecutive image pairs. We leave out sequences 9 and 10 for motion evaluation and use the remainder for training. The RGB-D SLAM dataset [40] is an indoor SLAM dataset with ground-truth pose and depth, from which optical flow can be calculated. The dataset contains varied camera motions, many featureless regions, repetitive patterns, and motion blur, which are well-known to be challenging for optical flow estimation. We select all sequences from the Handheld SLAM, Robot SLAM, and 3D Object Reconstruction categories. We set aside "fr1/360", "fr2/360_hemisphere", "fr2/pioneer_360", and "fr3/teddy" for testing. While most of these sequences are static, we remove those few frames that contain dynamic objects. For the training data, we select image pairs with diverse flow ranges, randomly sampling equally from buckets with a maximum flow of 5–40 pixels, 40–80 pixels and 80–120 pixels. For the test data, we sub-sample the video frames such that the baseline between each image pair is at least 3cm. We obtain training and test sets of 29 106 and 1 667 image pairs.

Metrics: For optical flow, we provide a comparison with a range of recent supervised and unsupervised methods using the standard Average End Point Error (AEPE) metric, given by AEPE $= \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{v}^i - \mathbf{v}_{\text{gt}}^i\|$, where $\mathbf{v}^i$ is the predicted flow at the i^{th} pixel, $\mathbf{v}_{\text{gt}}^i$ is the ground-truth flow, and N is the number of ground-truth pixels. For camera motion evaluation, we use the standard KITTI VO dataset

Table 1. Optical flow performance comparison on the KITTI 2012 and RGBD-SLAM datasets. We report the mean End Point Error (EPE) of the predicted optical flow. Parentheses and the suffix -ft indicate that the models were fine-tuned on the data, missing entries (–) indicate that the results were not reported, asterisks (∗) indicate that the method uses the test set to select the best performing model, and daggers (†) indicate that the results are pre-trained with the Ours-Baseline approach and finetuned with the losses proposed in the papers.

		KITTI 2012 (all)		KITTI 2012 (noc)		RGBD-SLAM	
	Method	train	test	train	test	validation	test
Supervised	SpyNet-ft [35]	(4.13)	4.10	–	2.00	–	–
	FlowNet2-ft [22]	(1.28)	1.80	–	1.00	–	–
	PWC-Net [42]	4.14	–	–	–	4.84	5.41
	PWC-Net-ft [42]	(1.45)	1.70	–	**0.90**	**1.22**	6.71
	PWC-Net-ft* [42]	–	–	–	–	3.64	**5.05***
	SelFlow-ft [28]	(**0.76**)	**1.50**	–	–	–	–
Unsupervised	UnsupFlownet [23]	11.30	9.90	4.30	4.60	–	–
	DSTFlow [36]	10.43	12.40	3.29	4.00	–	–
	DF-Net [56]	3.54	4.40	–	–	–	–
	UnFlow [29]	3.29	–	1.26	–	–	–
	OAFlow [44]	3.55	4.20	–	–	–	–
	EPIFlow [53]	(2.51)	3.40	(0.99)	1.30	5.16†	6.54†
	DDFlow [27]	2.35	3.00	1.02	1.10	5.01†	6.51†
	SelFlow [28]	1.69	2.20	**0.91**	**1.00**	4.95†	6.47†
	Ours-Baseline	3.49	–	1.24	–	5.44	6.89
	Ours-Epipolar	2.61	–	0.99	–	4.82	6.33
	Ours-Occlusion	1.97	–	1.19	–	4.95	6.47
	Ours	**1.56**	**1.90**	0.94	**1.00**	**4.63**	**6.12**

evaluation criterion [14], which evaluates on sub-sequences of length (100, 200, . . . , 800) meters, and report the average relative rotational and translational errors for the test sequences 9 and 10 in Table 2. Let $\Delta\mathtt{T}_{ij} \in \mathrm{SE}(3)$ denote the delta pose difference between the estimated pose and the ground-truth pose given a pair of adjacent frames i and j. The delta pose difference is given by $\Delta\mathtt{T}_{ij} = (\mathtt{T}_{\mathrm{gt},i}^{-1}\mathtt{T}_{\mathrm{gt},j})^{-1}(\mathtt{T}_i^{-1}\mathtt{T}_j)$, where $\mathtt{T}_i$ and $\mathtt{T}_j$ denote the poses at frames (i, j). The relative translation error is given by $t_{\mathrm{err}}^i = \frac{1}{N}\sum_{ij} \|\mathrm{trans}(\Delta\mathtt{T}_{ij})\|$ and relative rotation error is given by $r_{\mathrm{err}} = \frac{1}{N}\sum_{ij} \arccos(0.5(\mathrm{trace}(\mathrm{rot}(\Delta\mathtt{T}_{ij})) - 1))$, where $\mathrm{trans}(\cdot)$ and $\mathrm{rot}(\cdot)$ extract the translation and the rotation parts of $\Delta\mathtt{T}_{ij}$.

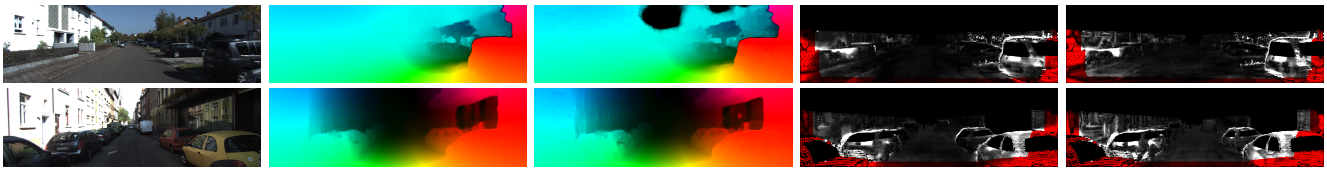
5.1. Implementation Details

We use PWC-Net [42] as our backbone network. Note however that our approach is network-agnostic so other optical flow networks can also be used. Multi-scale supervision is applied to capture large optical flows, especially on the RGB-D SLAM dataset. We generate a five-scale image pyramid starting at the original resolution then halving and warping at each successive level. In the original implementation of PWC-Net [42], a five-level pyramid of optical flow maps is predicted with the highest resolution being a quarter of the original image resolution. Therefore we scale the predicted optical flow by four using bilinear interpolation to match the corresponding image pyramid. For all networks, the weights for the multi-scale losses $(\lambda^1, \lambda^2, \lambda^3, \lambda^4, \lambda^5)$ were set to $(1, 0.34, 0.31, 0.27, 0.08)$ as per Meister et al. [29], modulo a constant factor.

We train directly on the KITTI VO dataset and the RGBD SLAM dataset to obtain our baseline model. When training the baseline model on KITTI VO, we empirically set $(\lambda_\mathrm{p}, \lambda_\mathrm{c}, \lambda_\mathrm{s}, \lambda_\mathrm{e})$ to $(1, 0.1, 0.1, 0)$. We then add the epipolar loss with λ_e set to 1000 to fine-tune the teacher model. When training the student model, we set $(\lambda_\mathrm{p}, \lambda_\mathrm{c}, \lambda_\mathrm{s}, \lambda_\mathrm{e}, \lambda_\mathrm{o})$ to $(1, 0, 0, 1000, 1)$, adding the occlusion loss. We used the same training strategy for the RGBD SLAM dataset, with $(\lambda_\mathrm{p}, \lambda_\mathrm{c}, \lambda_\mathrm{s}, \lambda_\mathrm{e}, \lambda_\mathrm{o})$ set to $(1, 0.1, 1, 100, 1)$. All experiments were run on a PC with a single 11GB RTX 2080 Ti GPU.

5.2. Essential Matrix Estimation

For estimating the essential matrix, we first obtain a robust initial estimate using RANSAC [11] and the five-point algorithm [26, 32]. We randomly sample 10 000 correspondences from the predicted optical flow and each GPU thread randomly selects five points on which to run the five-point algorithm [32]. Hence, each thread provides an essential matrix hypothesis for RANSAC. We select the essential matrix that has the most inliers, with respect to a set of 2 000 test correspondences, to initialize the iteratively re-



(a) Input image (b) Our flow (c) SelFlow [28] flow (d) Our error (e) SelFlow [28] error

Figure 2. Qualitative results on the KITTI 2012 test set. We compare our method with SelFlow [28]. (a) The first image of the input image pair. (b) Optical flow predicted by our model. (c) Optical flow predicted by SelFlow. (d) Optical flow prediction error with respect to the ground-truth using our method. (e) Optical flow prediction error using SelFlow.

weighted least squares (IRLS) algorithm. IRLS iteratively minimizes our robust lower-level objective function l. The stopping criteria is when the objective function is below 10^{-20} or the number of iterations exceeds 200. The inlier threshold δ was empirically set to 0.001.

5.3. Optical Flow Results

Quantitative results for optical flow on the KITTI and RGB-D SLAM datasets are reported in Table 1. For ablation purposes, we also provide results for Ours-baseline, Ours-Epipolar, and Ours-Occlusion. Ours-baseline refers to the model that is only trained with L_{p}, L_{c} and L_{s}. Ours-Epipolar refers to the model that is trained with the same losses but also the epipolar loss L_{e}. This is the result for our teacher network. Ours-Occlusion refers to training the baseline network with the teacher-student strategy described in Section 4, but without our proposed epipolar loss.

For KITTI, our model outperforms previous unsupervised optical flow methods and achieves results very close to supervised methods. The ablation results indicate that the global geometric losses have a significant positive impact on the optical flow quality, decreasing error by approximately 20% on average compared to the baseline. We can also see that the data augmentation technique proposed by Liu et al. [28] only improves on the occluded pixels while our method improves on both the occluded and non-occluded pixels. Note that compared with SelFlow [28], our method only uses two frames as input for training and testing while SelFlow uses five frames for training and three frames for testing. We show improved performance despite using fewer frames as input.

For the RGB-D SLAM dataset, our model outperforms previous state-of-the-art unsupervised optical flow methods and has slightly lower performance compared to a supervised method. Note that the results of other unsupervised methods reported in the table share the same backbone network as ours and are pre-trained using our baseline approach. They are then finetuned with the proposed losses in the respective papers. The supervised method overfits on the training and validation data and therefore uses the test set to select the best performing model. We show that we also make significant improvements over the baseline method on this challenging dataset, indicating the usefulness of the global geometric constraint. This dataset has a much wider variety of camera motions than the KITTI dataset, making optical flow estimation more difficult and camera motion estimation more helpful. Qualitative results for the RGB-D SLAM dataset are shown in Figure 3. They demonstrate that the brightness constancy and smoothness assumptions are insufficient to correctly resolve the flow in challenging scenarios.

Table 2. Odometry comparison for the KITTI VO dataset. We compare with an existing SLAM system and state-of-the-art unsupervised depth and motion learning algorithms. We report the translation error (%) and the rotation error (degrees per 100m).

	Seq. 9		Seq. 10	
Method	$t_{\text{err}}(\%)$	$r_{\text{err}}(^\circ/100\text{m})$	$t_{\text{err}}(\%)$	$r_{\text{err}}(^\circ/100\text{m})$
ORB-SLAM [30]	2.51	0.26	2.10	0.48
Zhou et al. [55]	17.72	6.82	36.57	17.69
Zhan et al. [52]	6.87	3.60	7.87	3.41
Gordon et al. [16]	**3.10**	–	5.40	–
Bian et al. [4]	6.07	2.19	7.56	4.63
Ours	4.36	**0.69**	**4.04**	**1.37**

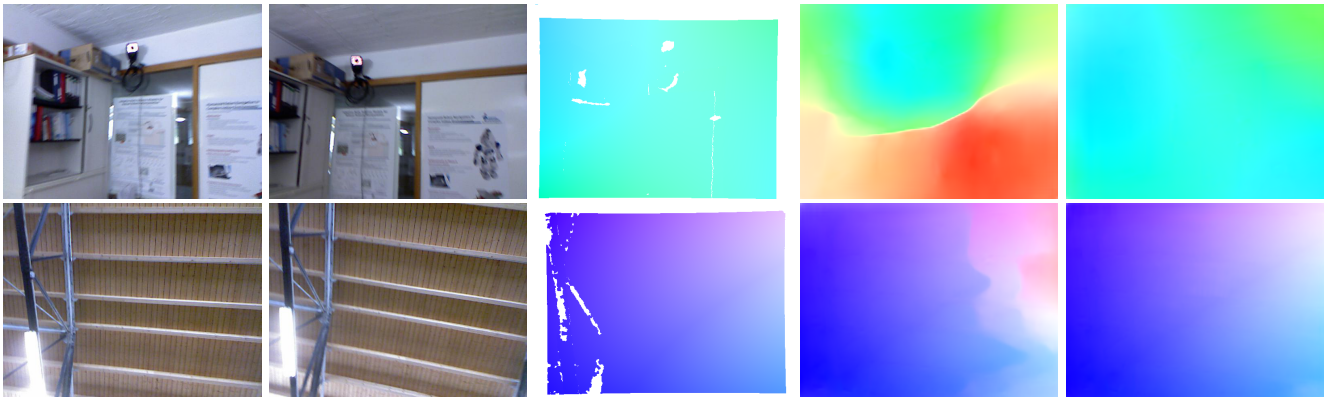
5.4. Ego-motion Results

We estimate the camera pose by decomposing our estimated essential matrix frame-by-frame, without any bundle adjustment, as opposed to ORB-SLAM [30]. Since our estimated essential matrix does not contain any scale information, we have to align our scale with the ground-truth frame-by-frame. For fair comparison, we also align the scales of the compared methods [55, 52, 4].

The quantitative results for ego-motion estimation on KITTI VO are shown in Table 2. Our method of estimating the camera pose significantly outperforms methods that directly regress the pose using a network. Qualitative results for the odometry comparison are shown in Figure 4, from which we can see our algorithm achieves performance close to ORB-SLAM, up to scale.

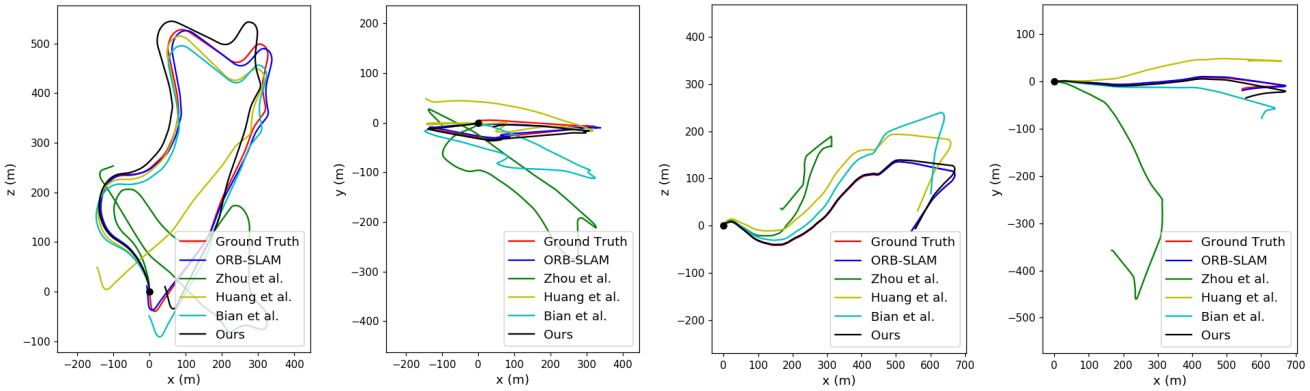
6. Discussion

In this paper, we have proposed a pipeline that is able to learn optical flow and egomotion simultaneously in an un-



(a) Input image 1 (b) Input image 2 (c) GT flow (d) Our flow without L_e (e) Our flow

Figure 3. Qualitative results on the RGBD-SLAM test set. The ground-truth flow is generated from the sparse ground-truth depth and the ground-truth pose provided by the dataset. The examples show that in cases of large motions, featureless regions and repetitive textures, the global geometric loss helps the network to learn to predict correct optical flow.



(a) Seq. 9 (b) Seq. 10

Figure 4. Qualitative results on KITTI VO sequences 9 and 10. For each sequence we provide the xz and xy odometry map. Second to ORB-SLAM, our method is closest to the ground-truth odometry. When visualizing the xy odometry map, it can be seen that unsupervised learning methods [55, 52, 4] have significant error in the y-axis direction, while our method has minimal error.

supervised manner by incorporating global geometric constraints into an optical flow estimation network. In particular, our method uses the implicit differentiation technique to allow back-propagating the gradients through a complicated geometric estimation algorithm without needing to compute the gradient for each algorithmic step. Given that the algorithm is complex, iterative, and involves a non-differentiable RANSAC procedure, it would otherwise be impossible to train end-to-end. Our formulation allows us to estimate the *essential matrix* and back-propagate through this estimation layer. This gives a tighter constraint than the fundamental matrix, having fewer degrees of freedom, and admits the use of state-of-the-art geometric algorithms.

Our model produces state-of-the-art results for unsupervised learning of optical flow, including for challenging data on which existing algorithms are known to perform poorly. We have also demonstrated superior camera motion estimation by optimizing an essential matrix from the predicted optical flow, compared with unsupervised methods that directly regress camera pose. Our approach to including a geometric estimation layer in a deep learning framework can be adapted to many other problems. This work provides a case study that demonstrates the usefulness of implicit differentiation as a tool for improving computer vision models.

7. Acknowledgement

The research was supported in part by the Australian Research Council through the Australian Centre of Excellence for Robotic Vision CE140100016, and the ARC Discovery Project grant DP200102274. We thank all reviewers for their valuable comments.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art

superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 5

[2] B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 136–145. JMLR. org, 2017. 2

[3] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision*, pages 154–170. Springer, 2016. 2

[4] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *arXiv preprint arXiv:1908.10553*, 2019. 7, 8

[5] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996. 1

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36. Springer, 2004. 1

[7] J. Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pages 318–326, 2012. 4

[8] A. L. Dontchev and R. T. Rockafellar. *Implicit Functions and Solution Mappings: A View from Variational Analysis*. Springer-Verlag, 2nd edition, 2014. 4

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 1, 2

[10] B. Fernando and S. Gould. Learning end-to-end video classification with rank-pooling. In *International Conference on Machine Learning*, pages 1187–1196, 2016. 2

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3, 6

[12] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding*, 134:1–21, 2015. 2

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 5, 6

[15] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 5

[16] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. *arXiv preprint arXiv:1904.04998*, 2019. 7

[17] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016. 2, 4

[18] S. Gould, R. Hartley, and D. Campbell. Deep declarative networks: A new hope. *arXiv preprint arXiv:1909.04866*, 2019. 2

[19] R. Hartley and H. Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2303–2314, 2012. 1, 3

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 1, 3

[21] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1, 2

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. 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. 1, 2, 6

[23] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision Workshops*, pages 3–10. Springer, 2016. 1, 2, 6

[24] S. Kumar, Y. Dai, and H. Li. Monocular dense 3D reconstruction of a complex dynamic scene from two perspective frames. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4649–4657, 2017. 1

[25] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. *arXiv preprint arXiv:1904.03758*, 2019. 2

[26] H. Li and R. Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 630–633. IEEE, 2006. 3, 6

[27] P. Liu, I. King, M. R. Lyu, and J. Xu. Ddflow: Learning optical flow with unlabeled data distillation. *arXiv preprint arXiv:1902.09145*, 2019. 2, 5, 6

[28] P. Liu, M. Lyu, I. King, and J. Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 2, 4, 5, 6, 7

[29] S. Meister, J. Hur, and S. Roth. UnFlow: unsupervised learning of optical flow with a bidirectional census loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 4, 6

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 7

[31] M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1577–1584, 2013. 1

[32] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):0756–777, 2004. 3, 6

[33] P. Ochs, R. Ranftl, T. Brox, and T. Pock. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 654–665. Springer, 2015. 4

[34] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine. Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*, 2019. 2

[35] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 6

[36] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 6

[37] K. G. Samuel and M. F. Tappen. Learning optimized MAP estimates in continuously-valued MRF models. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 477–484. IEEE, 2009. 4

[38] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 1

[39] F. Stein. Efficient computation of optical flow using the census transform. In *Joint Pattern Recognition Symposium*, pages 79–86. Springer, 2004. 4

[40] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 5

[41] D. Sun, S. Roth, and M. 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. 1, 2, 5

[42] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1, 2, 6

[43] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. In *Joint Pattern Recognition Symposium*, pages 314–324. Springer, 2008. 2

[44] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018. 2, 6

[45] J. Weber and J. Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):139–143, 1997. 2

[46] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4671–4680, 2017. 1

[47] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *European Conference on Computer Vision*, pages 211–224. Springer, 2006. 1

[48] G. Xu and Z. Zhang. *Epipolar geometry in stereo, motion and object recognition: a unified approach*, volume 6. Springer Science & Business Media, 2013. 2

[49] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1862–1869, 2013. 2

[50] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[51] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, pages 151–158. Springer, 1994. 4

[52] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 7, 8

[53] Y. Zhong, P. Ji, J. Wang, Y. Dai, and H. Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. *arXiv preprint arXiv:1904.03848*, 2019. 1, 2, 6

[54] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. *arXiv preprint arXiv:1908.01293*, 2019. 2

[55] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 2, 7, 8

[56] Y. Zou, Z. Luo, and J.-B. Huang. DF-Net: unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European Conference on Computer Vision*, pages 36–53, 2018. 2, 6