Towards Privacy-Preserving Ego-Motion Estimation using an Extremely Low-Resolution Camera

Armon Shariati[1], Christian Holz[2], and Sudipta Sinha[2]

***Abstract*—Ego-motion estimation is a core task in robotic systems as well as in augmented and virtual reality applications. It is often solved using visual-inertial odometry, which involves using one or more *always-on* cameras on mobile robots and wearable devices. As consumers increasingly use such devices in their homes and workplaces, which are filled with sensitive details, the role of privacy in such camera-based approaches is of ever increasing importance.**

In this paper, we introduce the first solution to perform *privacy-preserving* ego-motion estimation. We recover camera ego-motion from an extremely low-resolution monocular camera by estimating dense optical flow at a higher spatial resolution (i.e., 4x super resolution). We propose *SRFNet* for directly estimating Super-Resolved Flow, a novel convolutional neural network model that is trained in a supervised setting using ground-truth optical flow. We also present a weakly supervised approach for training a variant of SRFNet on real videos where ground truth flow is unavailable. On image pairs with known relative camera orientations, we use SRFNet to predict the auto-epipolar flow that arises from pure camera translation, from which we robustly estimate the camera translation direction. We evaluate our super-resolved optical flow estimates and camera translation direction estimates on the Sintel and KITTI odometry datasets, where our methods outperform several baselines. Our results indicate that robust ego-motion recovery from extremely low-resolution images can be viable when camera orientations and metric scale is recovered from inertial sensors and fused with the estimated translations.

***Index Terms*—Deep Learning in Robotics and Automation, Human-Centered Robotics, SLAM**

I. INTRODUCTION

VISUAL-Inertial Odometry (VIO) is the task of estimating the state (i.e., position, orientation, velocity, etc.) of a device using only one or more cameras and Inertial Measurement Units (IMUs). VIO is used for accurate ego-motion estimation on autonomous mobile robots and movable devices, such as Augmented and Virtual Reality (AR/VR) headsets and modern smartphones. As such devices become ubiquitous, the fact that they rely on one or more *always-on cameras* will potentially be a major privacy concern for consumers, particularly when used in their homes and workplaces.

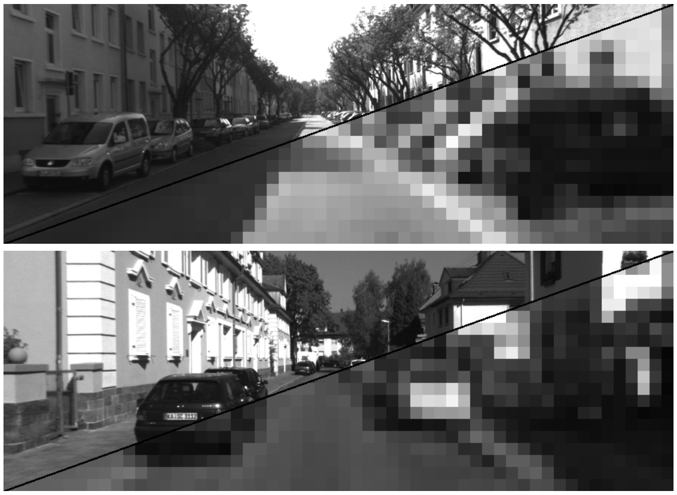


Fig. 1: Split screen visualization of images from the KITTI dataset and low-resolution images (56×20 pixels) generated by resizing the original images (896×320 pixels) to one-sixteenth their size. We focus on developing techniques that can recover ego-motion using cameras that capture such low-resolution images as this will enhance the user's privacy.

While previous work has analyzed privacy risks in robotics, AR and VR [1], [2], [3], less attention has focused on developing *privacy-preserving* approaches suitable for such systems. Only recently, Speciale et al. [4], [5] proposed new camera pose estimation techniques based on privacy-preserving point cloud map and image query representations, envisioning ways to address a key privacy question in AR.

In this paper, we investigate a practical VIO approach for privacy-preserving ego-motion estimation. Existing VIO techniques rely on one or more IMUs and cameras, making the camera the source of privacy concerns. Since visual motion cues are indispensable for accurate state estimation in VIO and are required for reducing drift, our approach is to drastically decreases the camera's pixel resolution to a level that conceals the identity of people, written text, and the presence of other sensitive objects. Maintaining a conventional field of view, our approach operates on images with a resolution as low as $56 \times 20 = 1120$ pixels, i.e., more than two orders of magnitude fewer pixels than are in VGA images. In addition to preserving privacy, low-resolution cameras provide additional benefits such as low-noise sensing, higher frame rates, lower power consumption, lower cost, and greater computational efficiency, all of which are highly appealing for embedded platforms.

Manuscript received: September, 10, 2019; Revised December, 6, 2019; Accepted December, 28, 2019.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Microsoft Corporation

[1] Armon Shariati is with the GRASP Lab, School of Engr. and Applied Science, Univ. of Pennsylvania, USA armon@seas.upenn.edu

[2] Christian Holz and Sudipta Sinha are with Microsoft Research, Redmond, USA {cholz, sudipsin}@microsoft.com

Digital Object Identifier (DOI): see top of this page.

Unfortunately, modern VIO methods designed for higher-resolution images struggle with images composed of only thousands of pixels. At this resolution, feature-based methods can no longer find recurring discriminative keypoints across multiple frames, while direct methods have their underlying assumptions (small patches in the scene with sufficient texture can be registered across images in the sequence) violated. In order to address these shortcomings, we propose a deep convolutional neural network (CNN) based approach that directly computes pixel correspondences at a higher resolution than that of the pair of input images. The design of our network for computing Super-Resolved Optical Flow, which we call SRFNet, involves combining two existing CNN architectures: SRResNet for image super resolution [6] and PWC-Net for coarse to fine optical flow estimation [7]. SRFNet uses SRResNet as a sub-network to super-resolve *feature maps*, rather than images, in order to replace the multi-resolution pyramid features in the original PWC-Net architecture. The super-resolved features then pass directly to the warp and cost volume layers to compute optical flow. We train SRFNet end-to-end in a supervised setting on available optical flow datasets using downsized input images at $^1/_{16}$-th the original resolution and predict optical flow at 4x super resolution, i.e., at $^1/_4$-th the original resolution.

Although SRFNet can predict high-resolution flow from low-resolution videos, it needs more real-world ground-truth optical flow data than what is easily available to achieve good generalization. Therefore, we propose a weakly supervised method that requires only videos with ground truth odometry in order to train SRFNet for ego-motion estimation.

While our approach outlined in this paper ignores the relative orientation between camera frames, we note that in practice, we can include an IMU to obtain a rotation estimate over small time windows in order to warp images with 2D homographies and simulate purely translational camera motion. While this work focuses on the sole task of predicting camera translation directions, in the future, we plan to explore a complete approach that uses IMU measurements for orientation and scale estimation as well.

We test SRFNet on the Sintel dataset [8] and KITTI dataset [9] for optical flow and camera translation estimation respectively. We present our network with downsized image pairs at $1/16^{th}$ of their original resolution (i.e., 48×24 pixels and 56×20 pixels, respectively) as input, which produces flow maps at 4x super-resolution (i.e., 192×96 pixels and 224×80 pixels, respectively). We show that SRFNet is more accurate than several baselines that first super resolve the image and then compute optical flow.

Contributions. (1) We introduce the privacy preserving ego-motion estimation problem and propose key ideas towards solving it. **(2)** We propose a new deep convolutional neural network-based architecture we call *SRFNet* for estimating high-resolution optical flow from low-resolution image pairs. The key novelty to our approach is that instead of super-resolving intensity images and then computing flow, SRFNet bypasses the image super resolution task by super resolving low-resolution feature maps and computing optical flow directly. **(3)** Finally, we propose to fine-tune SRFNet in a weakly supervised setting using a novel auto-epipolar loss function. This enables us to train SRFNet on large amounts of real, unlabeled videos where optical flow ground truth is absent and only camera poses are known.

II. Related Work

In this section, we review existing work on image super-resolution, optical flow and camera ego-motion estimation.

Image Super-Resolution. Classical methods for image super-resolution either learn a better image interpolation function using a database of low-resolution and high-resolution image pairs [10] or solve an image registration problem given multiple low-resolution images at sub-pixel misalignment [11], [12], [13]. The latter framework was extended to jointly solve super-resolution and optical flow on image sequences [14] and later on video using probabilistic formulations [15]. Modern super-resolution methods are based on CNNs [16], [6], [17], [18] with U-Net architectures, i.e., with symmetric encoder and decoder layers connected via skip connections. SRCNN [16] used bicubic upsampling whereas latter methods learn the upsampling filters. SRResNet [6] included a ResNet backbone in their model and later incorporated adversarial training and efficiency [18], [17].

Unlike traditional super-resolution methods, we predict optical flow at a higher spatial resolution instead of image intensities. While it is possible to super-resolve the images and then compute flow from them, we show that it is better to bypass image super-resolution and directly train the model to predict high-resolution flow from low-resolution images.

Optical Flow. Classical methods for optical flow proposed by Horn and Schunk [19], Lukas and Kanade (LK) [20] are optimization-based and rely on brightness constancy constraints alongside suitable regularization terms to compute sparse, semi-dense or dense flow. Recently, revived interest has been in the inverse compositional LK method [14] in the context of CNNs and learning [21]. When optical flow arises from a camera moving within a static scene, flow estimation is often performed in conjunction with camera motion estimation to compute epipolar flow [22] or multi-frame scene flow [23]. Honnegar *et.al* [24] proposed an efficient real-time optical flow method and a FPGA implementation suitable for deployment onboard small UAVs and mobile robots.

Recently, many end-to-end CNN architectures have been proposed for optical flow estimation, including FlowNet [25], SpyNet [26], MirrorFlow [27], PWC-Net [28], and its extension [7] to name a few. U-Nets have also been trained to predict camera motion and depth, such as in DeMoN [29] and SfMNet [30]. Our CNN-based model closely resembles PWC-Net and uses a coarse-to-fine framework as well. The main difference is that in the new architecture, in comparison to PWC-Net, the coarse-to-fine feature pyramid computations are reversed (starting from a low-resolution input to a higher-resolution output, where optical flow is computed).

Ego-motion estimation. Following Nister et al.'s work [31] on autonomous ground vehicles, numerous VO and visual SLAM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.2967307, IEEE Robotics and Automation Letters

algorithms have been recently proposed – SVO [32], [33], DSO [34], PTAM [35], ORB-SLAM [36], LSD-SLAM [37], to name just a few. In parallel, robust filter-based VIO methods performing sensor fusion were used in real-time systems, mobile devices, and other resource constrained settings [38], [39]. Other real-time visual-inertial SLAM, such as OKVIS [40], VINS-Mono [41] and others [38], [42], represent the state-of-the-art in this domain and were recently benchmarked for use on UAVs [43]. These methods are often categorized as either direct [44], [37], [34], indirect [31], [36], [39], or a hybrid [35], [32], [33], [42]. Direct methods minimize an objective based on photometric error, i.e. by directly comparing image intensities. Indirect feature-based methods minimize geometric distance between observed and transformed feature points. Unfortunately, feature-based methods cannot find well localized and repeatable 2D keypoints in images with resolution as low as $56{\times}20$ pixels, thereby adversely affecting the subsequent steps. Direct methods also struggle at this resolution, since single pixels do not correspond to small patches in the scene anymore. In contrast, our approach combines super resolution and optical flow estimation to recover fine pixel correspondence and then uses geometric constraints similar to the feature-based methods to recover camera motion. The dense optical flow estimated by our method are shown to be sufficiently accurate for computing relative camera motion.

III. APPROACH

We describe our approach towards ego-motion estimation using an extremely low-resolution camera in two parts. The first describes our proposed SRFNet architecture while the second then explains our weakly supervised method to fine-tune our model for recovering camera ego-motion.

A. Optical Flow Estimation

The design of our SRFNet architecture builds on both, the PWC-Net [28] and the SRResNet [6] CNN architectures, both of which achieve state-of-the-art performance for their respective tasks. Although PWC-Net is designed for estimating dense optical flow, whereas SRResNet computes single image super resolution (SISR), we demonstrate how these architectures may be integrated into a hybrid network for directly computing super-resolved optical flow. See Figure 2 for an overview. Next, we briefly review PWC-Net [28] and SRResNet [6] before describing our architecture.

1) PWC-Net: The design of PWC-Net, as illustrated in Figure 2a, is inspired by the well-established principle of traditional coarse-to-fine optical flow pipelines [45]; namely pyramidal processing, warping, and the use of a cost volume for computing similarity between image regions. In PWC-Net, these operations take place in feature space as opposed to image space. Once the underlying Feature Pyramid Extractor (FPE), highlighted by the red box in Figure 2a, produces a series of feature maps with decreasing resolution, the warping, cost volume computation, and optical flow estimation also proceeds in a coarse-to-fine fashion.

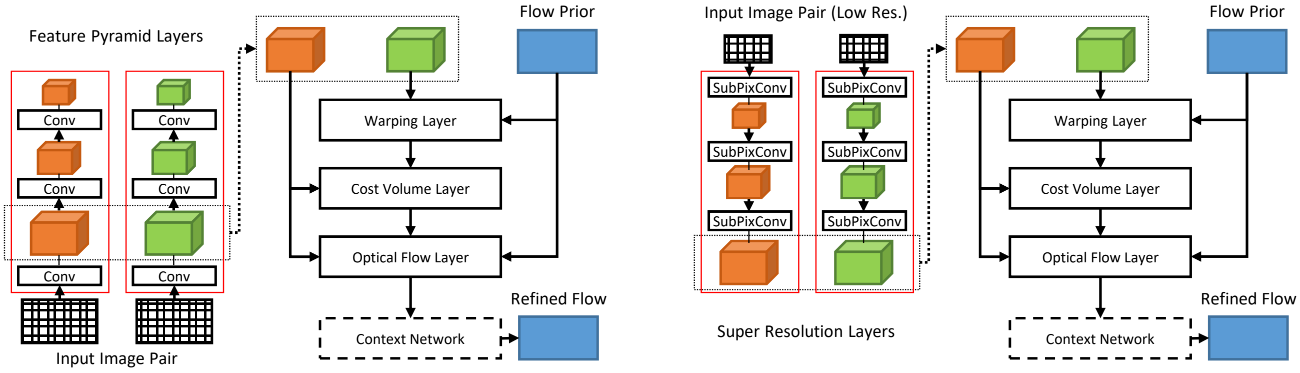
First, features from the second image are warped using the up-sampled flow from the previous iteration at the coarser level of the pyramid. Second, the network computes a cost volume over all corresponding neighborhoods of radius d in the warped and template feature maps. Third, the up-sampled flow, the cost volume, and the feature map from the first image are concatenated into a single feature map, which is then passed through another series of convolutional layers that predicts the flow at the current level of the pyramid. Finally, once the output resolution flow is estimated, it passes through one last set of convolutional layers with a large receptive field, which is designed to refine the flow by incorporating global flow context. This refinement step, however, is optional. For details regarding the architectural specifics of all these layers, please refer to [28].

The purpose of pyramidal processing in optical flow estimation is to effectively mitigate the aliasing problem that occurs due to the fact that image sequences typically have temporal sampling rates lower than that which is required by the sampling theorem to uniquely reconstruct the continuous signal. By filtering high-frequency signals at each level through smoothing and spatial down-sampling, pixel velocities become slower and more stable at coarser scales as spectral replicas disappear. In contrast to this problem, however, pixel velocities at our input resolution are so slow that frame-to-frame displacements are mostly sub-pixel, which leads to a “bleeding” effect that inevitably causes the brightness constancy assumption [20] to be violated. An insufficiently slow spatial sampling rate is the main issue.

2) SRResNet: Although the SRResNet architecture [6] consists of relatively simple components, it demonstrates strong performance in maximizing peak signal-to-noise ratio (PSNR) – a common metric for measuring image reconstruction quality. The network begins by transforming the low-resolution image to feature space using a series of K residual blocks [46]. Following this transformation, the resulting feature map is then interpolated using a series of efficient sub-pixel convolutional layers, as described in [18], whereby each layer increases the spatial resolution of the map by a factor of 2. During training, the network minimizes a mean-squared error loss, which implicitly maximizes PSNR. However, although it is a precise metric for reconstruction quality, a higher PSNR value does not necessarily imply a realistic image. While an adversarial loss can produce more visually appealing results, we are interested in computing accurate optical flow. In Section V, we examine the impact of providing super-resolved images from a network trained only to maximize PSNR. However, our primary interest in the SRResNet architecture is its internal production of a series of feature maps with increasing spatial resolution as it tries to generate a super-resolved image from a low-resolution input.

3) SRFNet: We will now describe the proposed SRFNet model, which is illustrated in Figure 2b. It directly computes super-resolved optical flow and is designed by combining components from both the PWC-Net and the SRResNet architectures. Our main idea is to use a modified SRResNet sub-network to build a feature pyramid. This sub-network replaces the feature pyramid extractor (FPE) in the original PWC-Net network. Unlike the FPE which produces lower-resolution features with each additional convolutional layer,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.2967307, IEEE Robotics and Automation Letters



(a) PWC-Net Architecture (b) Proposed SRFNet Architecture

Fig. 2: Overview of CNN architectures for (a) PWC-Net [28] and (b) our SRFNet model. PWC-Net uses a conventional feature pyramid, while SRFNet's feature pyramid is built in reverse (red) by upsampling lower-resolution feature maps.

our SRResNet sub-network produces a higher-resolution feature map with each additional sub-pixel convolutional layer. In essence, the feature maps in SRFNet are computed in reverse order with respect to those in PWC-Net. We perform an initial transformation of the low-resolution input image using $K = 16$ residual blocks as in the original SRResNet model but this is not shown in Figure 2b for simplicity.

Other than removing the final convolutional layer that computes the final super-resolved image, our sub-network differs from SRResNet in one more way. SRResNet uses a constant number of feature channels following each sub-pixel convolutional layer, but we decrease the number of feature channels by a factor of 2 (as the spatial resolution increases by a factor of 2). This is done so as to match the dimensions of the feature maps generated by PWC-Net's FPE. Our main difference from the original PWC-Net architecture, besides the removal of the FPE itself, is that the feature pyramid has three levels as opposed to six. Given a set of $L = 3$ feature maps, where each contains 64, 32, and 16 channels, respectively, our network proceeds in the same coarse-to-fine manner as the PWC-Net model described previously. Also note that our cost volume computation utilizes a neighborhood size of $d = 4$ for all levels of the feature pyramid, which lets our network handle displacements of up to 64 pixels at the output resolution.

In order to train our SRFNet architecture to predict super-resolved optical flow under general motion, we follow the same supervised training procedure using the same loss functions and hyper-parameters as described in [28]. Furthermore, letting Θ denote the set of weights in our SRFNet model and $\mathbf{w}^l$ denote the flow field at level l in the pyramid, we use the following multi-scale loss for initial training.

$$\mathcal{L}(\Theta) = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} \|\mathbf{w}^l_\Theta(\mathbf{x}) - \mathbf{w}^l_{GT}\|_2 + \gamma\|\Theta\|_2, \quad (1)$$

Subsequently, we fine-tune with the following robust loss.

$$\mathcal{L}(\Theta) = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} \left(|\mathbf{w}^l_\Theta(\mathbf{x}) - \mathbf{w}^l_{GT}| + \epsilon\right)^q + \gamma\|\Theta\|_2. \quad (2)$$

B. Ego-Motion Estimation

Our ultimate goal is to compute ego-motion from real-world videos from a moving camera. While such videos are abundant and easy to obtain, the associated dense ground truth flow for each frame is not. This motivated us to devise a training method that does not rely on dense flow supervision and circumvents the need for ground truth optical flow.

When a camera moves in a static scene, the induced optical flow must satisfy the epipolar constraints over image pairs and is often referred to as epipolar flow [22]. Therefore, one could consider using known epipolar geometry as a source of weak supervision to fine-tune our model to compute epipolar flow. However, at extremely low resolutions, the effect of camera rotation and translation on pixel intensity changes between subsequent frames can be difficult to disambiguate.

On the other hand, if we assume our frame-to-frame rotation can be accurately estimated over a short window – an assumption we would expect to be valid for an IMU – we can compensate for the rotational component of the flow by warping the second image using the homography $H = KR^TK^{-1}$. Given that the fundamental matrix for a pair of cameras can be expressed as

$$F = K^{-T}[\mathbf{t}]_\times RK^{-1} = [\mathbf{e}]_\times(KRK^{-1}), \quad (3)$$

where $\mathbf{e}$ corresponds to the template epipole, and $\mathbf{t}$ and R correspond to the relative translation and rotation between the two, the new fundamental matrix corresponding to the rotation-compensated image and the template image is reduced to $F' = [\mathbf{e}]_\times$. And since $\mathbf{e} = K\mathbf{t}$, estimating the two parameter epipole location amounts to estimating the remaining translational component of the flow up to some scale factor (though scale too, like rotation, may also be provided by an IMU). This is a special case of epipolar flow called *auto-epipolar flow* [47], which is induced by a purely translation relative motion between camera pairs.

This leads us to the following loss function for training,

$$\mathcal{L}(\Theta) = \sum_{l=l_0}^{L} \alpha_l \sum_{\mathbf{x}} \arccos\left(\hat{\mathbf{n}}(\mathbf{x})^T \hat{\mathbf{w}}^l_\Theta(\mathbf{x})\right) + \gamma\|\Theta\|_2, \quad (4)$$

where $\mathbf{n}(\mathbf{x}) = \mathbf{e} - \mathbf{x}$ and $\hat{}$ notation indicates a vector that has been normalized to have unit length. This auto-epipolar loss function minimizes the *angular difference* between the direction of the predicted flow vector at each pixel $\hat{\mathbf{w}}^l_\Theta(\mathbf{x})$ and the direction of the ground truth flow vector which would arise given a relative translation of $\mathbf{t}$ between camera frames $\hat{\mathbf{n}}(\mathbf{x})$. Notice that supervision for this loss, $\hat{\mathbf{n}}(\mathbf{x})$, can be obtained from ground truth odometry alone, which is much easier to obtain than dense flow maps. Also observe that we are only concerned with accurately estimating the flow orientation and not magnitude. Recall that since the flow vectors must satisfy our epipolar constraints, $F = [\mathbf{e}]_\times$ in this case, all flow vectors corresponding to static parts of the scene will end up pointing to the epipole. Therefore, given a dense flow map output by our network trained in this fashion, we can recover the epipole location by finding the approximate point of intersection shared by most pairs of flow vectors using a 2-point RANSAC with an angular difference threshold of 2° to filter outliers. Finally, in order to obtain the translation direction we simply back-project the the epipole position using the camera intrinsics matrix K.

IV. Implementation Details

Since our SRFNet architecture contains two separate sub-networks designed for different tasks, training SRFNet requires multiple stages of pre-training and fine-tuning on various datasets. The order in which different datasets are used during training roughly follows that of [28]. We use an NVIDIA Tesla P100 GPU in our experiments.

We begin by training a 3-level PWC-Net model from scratch on randomly cropped images of size 448×284 from the FlyingChairs dataset [25] for 200K iterations using a learning rate of 10^{-4}, mini-batches of size 8, and the loss function described in Equation 1. We then reduce the learning rate by a factor of 2 and continue training for another 100K iterations. After this intial pre-training is complete, we fine-tune our model on randomly cropped (768×384) images from the FlyingThings3D dataset [48] for 400K iterations using a learning rate of 10^{-5}, mini batches of size 4, and the same loss described in Equation 1. We call this model PWC-Net-T. We use the robust loss described in Equation 2 in the next round of fine-tuning, where we train for 50K iterations on randomly cropped (768×384) images from the Sintel dataset [8] using a learning rate of 10^{-5} and mini batches of size 4. We call the resulting model PWC-Net-S. We further fine-tune it on random crops (896×320) from the KITTI Flow dataset [49] for 20K iterations using a learning rate of 10^{-5}, mini batches of size 4, and the robust loss from Equation 2. We call this model PWC-Net-K. The values of all unspecified hyper-parameters can be found in [28].

In parallel, we train our modified SRResNet architecture from scratch on FlyingChairs, but now using images randomly cropped to (256×256) for 400K iterations with a learning rate of 10^{-5}. We call this model SRResNet-C. We skip training SRResNet on FlyingThings3D and directly fine-tune it on Sintel and KITTI for the same number of iterations and using the same learning rate and mini batch size as those used for training PWC-Net-S and PWC-Net-K, respectively, but using 256×256 crops for SRResNet-S and SRResNet-K. The inputs to all our SRResNet models are produced by down-sizing these (256×256) crops (used as the training targets) by a factor of 4x. Finally, all our SRResNet models are trained using mean-squared loss.

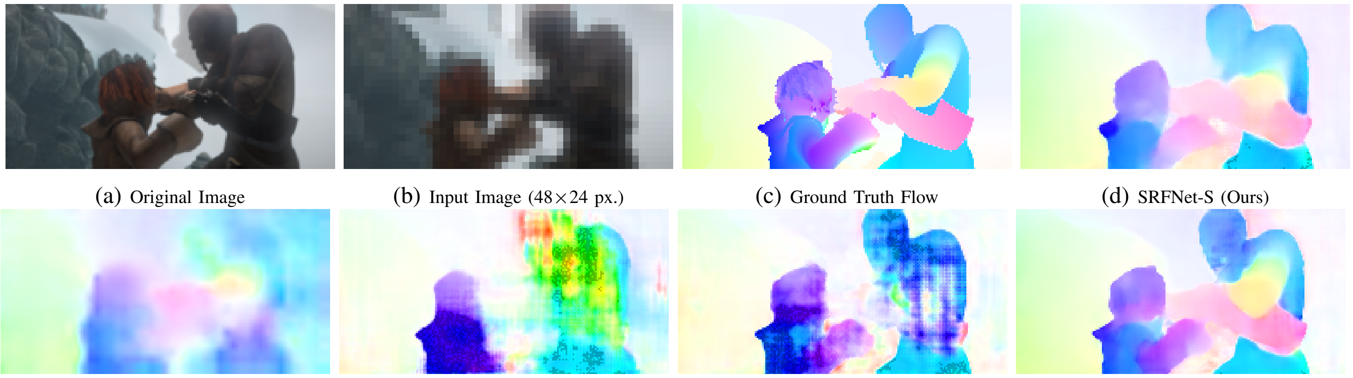
We now describe the training procedure for our SRFNet model. We start by initializing the super resolution layers of our first SRFNet model, in particular the residual blocks and sub-pixel convolutions, using the weights from our SRResNet-C model. Similarly, we initialize the optical flow estimation and context layers of our model with weights from our PWC-Net-T. Note that the warping and cost volume layers have no parameters and thus are identical for all models. Given this initial SRFNet model, we further train the model on 16x downsized random (768×384) crops from the FlyingThings3D dataset for first 60K iterations using a learning rate of 5×10^{-5} and then for an additional 300K iterations using a learning rate of 10^{-5}. We use the same multi-scale loss and batch size as those used for training PWC-Net-T. However, the key distinction is that we do **not** back-propagate through the optical flow and context layers. This is because while these weights have already been exposed to the FlyingThings3D images, the weights within the super resolution layers have not. This model is then further trained in two more rounds of end-to-end fine-tuning; first on Sintel and the second on KITTI Flow, while back-propagating through all layers. We use the same hyper-parameters, iterations, and losses as those used for training PWC-Net-S and PWC-Net-K, while downsizing the random crops by a factor of 16x. Using the same convention, we call these models SRFNet-S and SRFNet-K. Note since Sintel and KITTI Flow do not provide ground truth flow for testing, we randomly select 100 and 30 frames from Sintel and KITTI respectively, to be held out for future evaluation.

Finally, our last model, SRFNet-EK, is trained using our proposed weakly supervised loss function. We take sequences 0–3 from the KITTI VO dataset [9] and partition them into windows of $w = 4$ while skipping every $s = 1$ frames. Using the provided ground truth pose estimates, we compensate for all rotations with respect to the first frame in a given window by applying the homography $H_i = K(R_1^T R_i)^T K^{-1}$ to all the images within the window, where i denotes the index of each frame following the first. We then obtain our ground truth translation vectors by re-expressing the translation component of the given pose vector, which is provided in the global frame, with respect to the first camera frame. Note that the epipoles needed for supervision are obtained by projecting these translations to the image plane, which uses the camera intrinsics. After creating this dataset, we train our model for 90K iterations using a learning rate of 10^{-5} and mini-batches of size 8. We use the same value of α as that used for training all PWC-Net models.

V. Experimental Results

Since our SRFNet model is useful for two different tasks, *i.e.* general optical flow estimation and camera ego-motion estimation, the following experiments are designed to assess

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2020.2967307, IEEE Robotics and Automation Letters



(a) Original Image (b) Input Image (48×24 px.) (c) Ground Truth Flow (d) SRFNet-S (Ours)

(e) PWC-Net-S, No SR (f) PWC-Net-S, Bicubic SR (g) PWC-Net-S, SRResNet-S (h) PWC-Net-S, Oracle SR

Fig. 3: Qualitative results for super-resolved optical flow estimation on an image pair (only one is shown) from Sintel. Our SRFNet model outperforms all the baselines and does almost as well as PWC-Net-S Oracle at 4x super resolution.

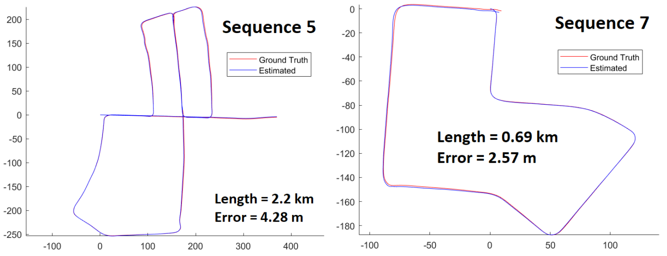
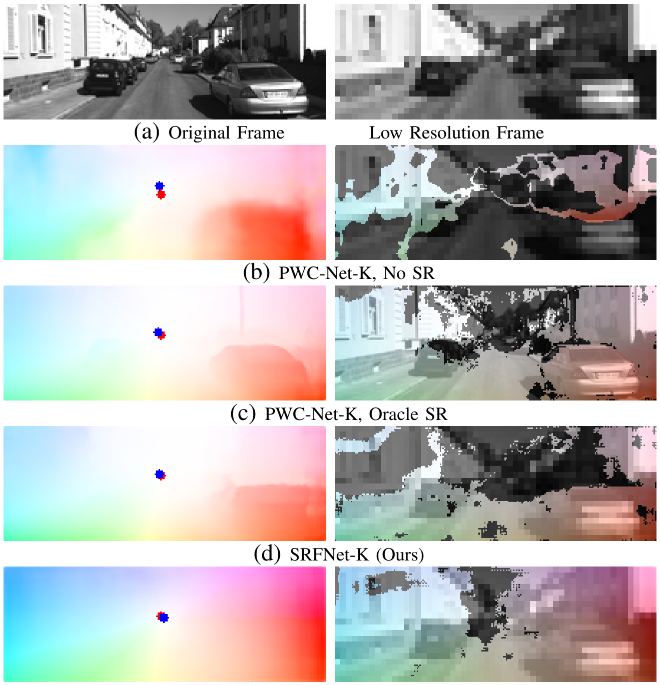


Fig. 4: The ground truth camera trajectories for two KITTI sequences and trajectories by a least squares method that uses the predicted translation directions as well as the ground truth orientation and scale information (see text for details). The low position drift errors in the two sequences confirm that our predicted translation directions are quite accurate.

TABLE I: Ablation experiment comparing different factors of flow super resolution on Sintel. All units in pixels.

	Oracle Baselines			Ours
SR Factor	1x	2x	4x	4x
Flow Method	PWC-Net-S			SRFNet-S
AEPE	1.50	0.99	0.74	0.86

the efficacy of our method on both tasks. On average, a single forward pass through the network takes about 10 ms. All the methods in the experiments perform 4x super resolution (of an image or flow map depending on the method). This resolution was selected based on an ablation study (see Table I for the results). We present our PWC-Net-S model with pairs of images from the Sintel dataset [8] at three different scales – which we refer to as “oracle super resolution” – and measure the average endpoint error of the predicted flow map. The first scale is at the lowest input resolution (48×24), which we refer to as 1x. There, we are measuring the effect of not doing super resolution. The second and third scale is at twice (96×48) and four times (192×96) the original resolution, respectively. We obtain image pairs by downsizing (768×384) center crops from the original images. Our results show that super resolving the input to 4x its original size is more accurate than 1x and 2x. We did not try 8x, since at 4x we already obtain reasonably accurate flow with respect to the (768×384) images.



(a) Original Frame Low Resolution Frame

(b) PWC-Net-K, No SR

(c) PWC-Net-K, Oracle SR

(d) SRFNet-K (Ours)

(e) SRFNet-EK (Ours)

Fig. 5: Qualitative results for epipole prediction on rotation-compensated image pairs from the KITTI dataset. Predicted flow maps (left) and inlier masks (right) are shown for each method. The true and predicted epipoles are shown overlaid on the flow maps using red and blue dots respectively.

TABLE II: Comparison of different baselines for 4x flow super resolution on Sintel. All units in pixels.

	Baselines				Ours
SR Method	None	Bicubic	SRResNet-S	Oracle	
Flow Method	PWC-Net-S				SRFNet-S
AEPE	1.50	2.18	1.71	0.74	0.86

A. *Super-Resolved Optical Flow Estimation*

To study the performance of our integrated SRFNet model for both super resolution and optical flow estimation, we evaluate our system’s performance on its ability to estimate flow from simulated low resolution images (48×24) from the

TABLE III: Comparison of different methods for predicting frame to frame translation

	Baselines				Ours		
SR Method	-	Bicubic	SRResNet-(S / K)	Oracle			
Flow Model	PWC-Net-(S / K)				SRFNet-S	SRFNet-K	SRFNet-EK
$\Delta\theta$ (deg)	6.09 / 6.04	34.91 / 4.73	30.66 / 4.06	2.36 / 2.32	3.5	2.86	2.57
Δd (m)	0.070 / 0.074	0.446 / 0.034	0.384 / 0.033	0.019 / 0.018	0.033	0.023	0.019
Δd (%)	10.44 / 10.39	58.42 / 8.03	51.53 / 6.94	4.01 / 3.94	6.00	4.87	4.38

Sintel dataset [8]. As in the ablation experiment, these images are obtained by downsizing (768×384) center crops of the original images. We compare our model to four different non-integrated baselines, each of which utilizes a separate method for SISR whose output is then provided as input to PWC-Net-S in order to compute optical flow. Our quantitative results measuring average endpoint error are presented in Table II, while qualitative results are shown in Figure 3.

The first method, which we expect to have the worst performance, simply computes optical flow at the input resolution. Since all the methods described below are evaluated at the 4x (192×96) scale, the output flow map must be interpolated to the higher resolution, which we do using a bilinear kernel. The second and third baseline involve super resolving the input images with a bicubic kernel and an SRResNet-S model, respectively, before computing the flow map. We expect these methods to outperform the first model, however, under-perform when compared to our final "oracle" baseline, which directly computes flow at the ground truth super resolved image at (192×96). The oracle baseline serves as an upper bound of performance for our method.

As can be seen in Table II, our integrated model indeed outperforms the first three baselines by a factor of 2, while only suffering a loss of .1 average pixel error compared to the expected upper bound. To our surprise however, the first baseline, which utilizes no super resolution at the input, outperforms the second and third. These results along with those of our initial ablation study indicate that while higher resolution images yield more accurate flow estimates, standard SISR methods are not consistent across consecutive video frames and that hurts the flow accuracy.

B. Camera Translation Estimation

In this set of experiments, we aim to understand how well our proposed SRFNet model performs for our ultimate task of ego-motion estimation. We test three of our models, SRFNet-S, SRFNet-K, and SRFNet-EK, against different SISR methods used in tandem with a PWC-Net model. The baselines selected for comparison parallel those used in the general optical flow experiments, whereby for super resolution we examine the use of no super resolution, bicubic interpolation, and different SRResNet models. We follow the same procedure of downsizing center crops in order to simulate our low resolution inputs. The only difference is that we instead start with a crop size (896×320) and resize them to (56×20). For evaluation, we use rotation-compensated KITTI sequences 4–10 [9], held out during the training. The quantitative results of our experiments are shown in Table III while some qualitative results are shown in Figure 5.

The performance of each super resolution method and flow model combination, as well as each of our integrated models, is measured based on three different metrics for frame-to-frame translation estimation. Recall that since we assume that frame-to-frame rotation is known, the accuracy of complete ego-motion estimation depends entirely on the accuracy of the translation estimates. The first metric is the average angular difference between the direction of the ground truth translation vector and the direction of our predicted translation vector. The second metric is the average endpoint error of the 3D translation vector in meters. Observe that this metric is different from the 2D average endpoint error in pixels used in the previous experiments. Finally, the third metric is the same average endpoint error, however scaled to a percentage in order to account for varying translation magnitudes. Note that we scale our predicted translation directions with the ground truth length for evaluation.

Training the models on real world KITTI Flow [49] yields far better performance as compared to those trained on synthetic Sintel data [8] (see Table III). While this is to be expected, it is interesting to note the degree of improvement exhibited by the PWC-Net models which rely on bicubic interpolation and SRResNet for super resolution when trained on KITTI. In contrast to the results of the previous general optical flow experiment, when trained on the real world data, these two baselines outperform the initial approach of using no super resolution at the input and instead upsampling the output flow map. This suggests that while the general optical flow estimation and ego-motion estimation tasks are closely related, there is some aspect to the latter which makes the idea of super resolving the input images still somewhat viable. Given that, our integrated SRFNet models still outperform the three naive baselines. Notice that even when trained on Sintel, our SRFNet-S model still outperforms the baseline methods that use models fine-tuned on KITTI Flow. Furthermore, the average endpoint error of SRFNet-K is only $< 1\%$ smaller than the oracle.

Ultimately, the result best demonstrating the efficacy of our method are those for our weakly supervised SRFNet-EK model. We see that despite not having ground truth pixel-wise flow for training, our model performs almost as well as the oracle. Thus, the expensive process of collecting accurate dense flow maps can be avoided altogether. It is interesting to note however (see Figure 5), that SRFNet-EK no longer produces flow maps with identifiable scene elements as only flow orientation was used in the training objective.

Finally, we compute full camera trajectories to test the accuracy of our translation estimates. Figure 4 shows the result of a linear pose-graph optimization over the translational measurements. For each frame, we have three linear constraints, one to each of its three consecutive frames. As our network currently predicts only the translation direction, we utilize ground truth scale and orientation information during the trajectory reconstruction step. The estimated trajectories for sequences 5 and 7 from the KITTI dataset have 4.28m and 2.57m position drift over 2.2km and 0.69km respectively.

VI. CONCLUSION

In summary, we present here the first viable approach towards developing a real-time privacy-preserving VIO system by using a CNN model that can accurately estimate the direction of camera translation from extremely low-resolution image pairs – within which faces, text, and other sensitive information is indiscernable. Our results show that our model is more effective compared to methods that first super-resolve the images and then compute optical flow. In the future, we aim to develop a complete VIO system which utilizes an actual low-resolution camera and an IMU for live orientation and scale estimation as well.

REFERENCES

[1] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, and T. Kohno, "A Spotlight on Security and Privacy Risks with Future Household Robots: Attacks and Lessons," in *UbiComp*, 2009, pp. 105–114.

[2] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner, "Towards security and privacy for multi-user augmented reality: Foundations with end users," in *IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 392–408.

[3] F. Pittaluga, S. J. Koppal, S. Bing Kang, and S. N. Sinha, "Revealing Scenes by Inverting Structure From Motion Reconstructions," in *CVPR*, 2019, pp. 145–154.

[4] P. Speciale, J. L. Schonberger, S. B. Kang, S. N. Sinha, and M. Pollefeys, "Privacy Preserving Image-Based Localization," in *CVPR*, 2019.

[5] P. Speciale, J. L. Schonberger, S. N. Sinha, and M. Pollefeys, "Privacy Preserving Image Queries for Camera Localization," in *ICCV*, 2019.

[6] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *CVPR*, 2017, pp. 105–114.

[7] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation," *IEEE TPAMI*, 2019.

[8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *ECCV*, 2012.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *CVPR*, 2012.

[10] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, no. 2, pp. 56–65, 2002.

[11] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.

[12] W. Zhao and H. S. Sawhney, "Is super-resolution with optical flow feasible?" in *ECCV*, 2002, pp. 599–613.

[13] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[14] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *IJCV*, vol. 56, no. 3, pp. 221–255, feb 2004.

[15] C. Liu and D. Sun, "A Bayesian approach to adaptive video super resolution," in *CVPR*, 2011, pp. 209–216.

[16] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE TPAMI*, vol. 38, no. 2, pp. 295–307, 2016.

[17] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in *CVPR Workshop*, 2017, pp. 1132–1140.

[18] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *CVPR*, 2016, pp. 1874–1883.

[19] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, aug 1981.

[20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Intl. Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[21] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *CVPR*, 2017, pp. 2568–2576.

[22] K. Yamaguchi, D. McAllester, and R. Urtasun, "Robust monocular epipolar flow estimation," in *CVPR*, 2013, pp. 1862–1869.

[23] T. Taniai, S. N. Sinha, and Y. Sato, "Fast multi-frame stereo scene flow with motion segmentation," in *CVPR*, 2017, pp. 3939–3948.

[24] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *ICRA*, 2013, pp. 1736–1741.

[25] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks," in *ICCV*, 2015, pp. 2758–2766.

[26] A. Ranjan and M. J. Black, "Optical Flow Estimation Using a Spatial Pyramid Network," in *CVPR*, 2017, pp. 2720–2729.

[27] J. Hur and S. Roth, "MirrorFlow: Exploiting symmetries in joint optical flow and occlusion estimation," in *ICCV*, 2017, pp. 312–321.

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *CVPR*, 2018, pp. 8934–8943.

[29] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *CVPR*, 2017, pp. 5038–5047.

[30] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: Learning of Structure and Motion from Video," *CoRR*, vol. abs/1704.0, apr 2017.

[31] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR*, 2004.

[32] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *ICRA*, 2014, pp. 15–22.

[33] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," *IEEE Trans. on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[34] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE TPAMI*, vol. 40, no. 3, pp. 611–625, mar 2018.

[35] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *ISMAR*. IEEE Computer Society, 2007, pp. 1–10.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[37] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct monocular SLAM," in *ECCV*, 2014.

[38] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *ICRA*, 2007.

[39] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *IJRR*, vol. 32, no. 6, pp. 690–711, 2013.

[40] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *IJRR*, vol. 34, no. 3, pp. 314–334, 2015.

[41] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Trans. on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[42] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Trans. on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[43] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," in *ICRA*, 2018, pp. 2502–2509.

[44] J. Engel, J. Sturm, and D. Cremers, "Semi-dense Visual Odometry for a Monocular Camera," in *ICCV*, 2013, pp. 1449–1456.

[45] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer vision and image understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016, pp. 770–778.

[47] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[48] N. Mayer, E. Ilg, P. Hausser, 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.

[49] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *CVPR*, 2015.