EffiScene: Efficient Per-Pixel Rigidity Inference for Unsupervised Joint Learning of Optical Flow, Depth, Camera Pose and Motion Segmentation

Yang Jiao [*1,2,3], Trac D. Tran [2], Guangming Shi [1,3]
[1] Xidian University, [2] Johns Hopkins University, [3] Xidian Guangzhou Institute of Technology
{yangjiao,yjiao8}@{stu.xidian.edu.cn,jhu.edu}, trac@jhu.edu, gmshi@xidian.edu.cn

Abstract

*This paper addresses the challenging unsupervised scene flow estimation problem by jointly learning four low-level vision sub-tasks: optical flow **F**, stereo-depth **D**, camera pose **P** and motion segmentation **S**. Our key insight is that the rigidity of the scene shares the same inherent geometrical structure with object movements and scene depth. Hence, rigidity from **S** can be inferred by jointly coupling **F**, **D** and **P** to achieve more robust estimation. To this end, we propose a novel scene flow framework named EffiScene with efficient joint rigidity learning, going beyond the existing pipeline with independent auxiliary structures. In EffiScene, we first estimate optical flow and depth at the coarse level and then compute camera pose by Perspective-n-Points method. To jointly learn local rigidity, we design a novel Rigidity From Motion (RfM) layer with three principal components: (i) correlation extraction; (ii) boundary learning; and (iii) outlier exclusion. Final outputs are fused based on the rigid map M_R from RfM at finer levels. To efficiently train EffiScene, two new losses $\mathcal{L}_{bnd}$ and $\mathcal{L}_{unc}$ are designed to prevent trivial solutions and to regularize the flow boundary discontinuity. Extensive experiments on scene flow benchmark KITTI show that our method is effective and significantly improves the state-of-the-art approaches for all sub-tasks, i.e. optical flow ($5.19 \rightarrow 4.20$), depth estimation ($3.78 \rightarrow 3.46$), visual odometry ($0.012 \rightarrow 0.011$) and motion segmentation ($0.57 \rightarrow 0.62$).*

1. Introduction

Scene flow [38, 37] describes the 3D motion of a dynamic scene by 2D optical flow and scene depth, providing essential geometrical clues for numerous practical applications such as self-driving [28] and robotics navigation [1, 31]. However, acquiring dense ground truth for both sub-tasks in real applications are usually expensive or impractical. To overcome this, learning scene flow in an unsupervised way has attracted much attention in recent years, by minimizing the photometric differences between the original-synthesized pixel pairs.

Optimizing pixel-wise photometric error for low-level scene flow task without supervision is not a trivial task. One of the most critical reason is that the pixel correspondence between consecutive frames is highly ambiguous, especially in unstructured or texture-less regions. For example, one pixel from a mountain or a highway surface in frame t can be projected to various surrounding pixels in frame $t+1$ with very low photometric error, often leading to the failure of local scene flow estimation. Unfortunately, this issue always happens in outdoor scenarios due to missing small details due to motion blur. Therefore, additional constraints are strongly needed to eliminate the ambiguities for successful unsupervised scene flow estimation.

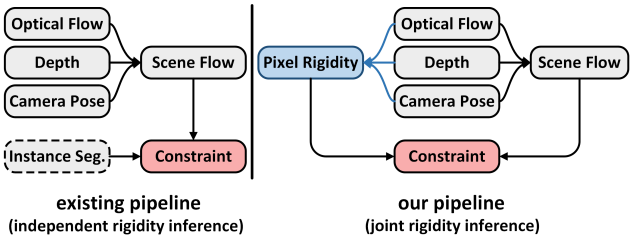

Figure 1. Main idea of our method. Different from independently estimating rigid pixels from the auxiliary instance segmentation in existing pipeline, we jointly learn per-pixel rigidity from optical flow, depth and camera pose for more accurate rigid constraint.

Problems. In recent approaches, rigid constraint is widely employed to separate the scene into static (rigid) and moving (non-rigid) areas. It also restricts the ego-motion of the rigid pixels which obey the rigid scene assumption [25]. To achieve this, current methods [33, 22, 43, 14, 9, 27] follow a popular scene flow pipeline as shown in Fig. 1 (left), where the auxiliary instance segmentation network is designed to predict the rigid pixels that will be constrained by local rigidity. Though impressive scene flow results can be achieved, the performance of the segmentation is often poor, indicating an inaccurate estimation of rigid pixels (static

area), and it could, in turn, harm the rigid constraint. One reason is that the independent rigidity inference in existing pipeline limits the learning of pixel rigidity. More specifically, the segmentation task in existing pipeline is jointly optimized with scene flow sub-tasks in back-propagation, but it is independently launched in forward inference. This independent structure makes inference inefficient, resulting in networks that can only learn pixel-wise rigidity from raw RGB images, but difficult to extract extra geometrical information from flow and depth. Besides, optimizing both deep segmentation network and scene flow multi-networks in current pipeline without ground truth might be very difficult, requiring sophisticated training strategies such as [33].

Motivation & Idea. Inspired from recent works, our key insight is that the rigidity of the scene shares the same inherent geometrical structure with optical flow and depth, hence they are highly correlated and can be mutually beneficial. Based on this observation, instead of designing the auxiliary segmentation structure, we jointly consider optical flow, depth and camera pose for rigidity learning as illustrated in Fig. 1 (right), and propose a novel framework called EffiScene. With the new pipeline, we can go beyond the existing methods by providing: *(i)* more effective rigid constraint via jointly considering scene flow sub-tasks for learning accurate rigid pixels; and *(ii)* more efficient scene flow framework optimization via eliminating the very deep instance segmentation network.

Approach. EffiScene aims to solve the following four unsupervised sub-tasks: *(i)* optical flow **F** estimation; *(ii)* stereo-depth **D** prediction; *(iii)* camera pose **P** for visual odometry; and *(iv)* motion segmentation **S**. We first estimate the optical flow F^o and depth D at the coarse level, then compute the relative camera pose P from time t to $t+1$ by minimizing the reprojection error between the observed coordinates (from F^o) and the projected 3D points (from D) via a Perspective-n-Points (PnP) solver. Next, we propose a novel Rigidity From Motion (RfM) layer to estimate pixel rigidity by explicitly modeling the correlation between optical flow F^o and rigid flow F^r. Our RfM includes three main steps: *(i)* correlation extraction; *(ii)* boundary learning; and *(iii)* outlier exclusion. The rigid map M_R from RfM can be interpreted as motion segmentation. Finally, flows from F^o and F^r are fused to form the final flow, guided by the rigid map M_R at the fine level. In training, two new losses – $\mathcal{L}_{bnd}$ and $\mathcal{L}_{unc}$ – are designed to optimize RfM and regularize the flow boundary discontinuity, respectively. Different from existing methods [41, 16], there are no sensitive thresholds needed to be set manually in EffiScene.

Contributions are summarized as follows.

- We introduce a new structure for unsupervised scene flow estimation, and demonstrate that per-pixel rigidity can be efficiently predicted by jointly learning optical flow, depth and camera pose.
- We design a novel Rigidity from Motion (RfM) layer to recognize rigid regions via explicitly modeling motion correlations. To the best of our knowledge, this is the first deep model for joint rigidity learning.
- We optimize scene flow training by two new losses: $\mathcal{L}_{bnd}$ prevents the trivial solution of RfM whereas $\mathcal{L}_{unc}$ regularizes the optical flow discontinuity in uncovered boundary.

Extensive experiments on KITTI benchmarks [4, 5, 29] show that our method outperforms existing state-of-the-art (SOTA) approaches for all four sub-tasks with highly efficient rigidity inference (RfM with size 0.0032Mb vs. 5.22Mb [33]), i.e. optical flow ($5.19 \rightarrow 4.20$) by a significant 19% improvement, depth estimation ($3.78 \rightarrow 3.46$), visual odometry ($0.012 \rightarrow 0.011$), and motion segmentation ($0.57 \rightarrow 0.62$).

2. Related Work

We first offer a brief review of optical flow and depth estimation, which are jointly learned for efficient rigidity inference in our method. Then, scene flow is discussed.

Optical flow. Deep convolutional neural networks (CNN) are widely used in supervised optical flow methods. FlowNet [2] is the first work using an end-to-end CNN architecture. FlowNet2 [12] improves the results by stacking more layers but can be computationally expensive. Then, simpler deep models such as SpyNet [32], LiteFlowNet [10] and PWC-Net [34] are designed in spatial or feature pyramid fashion. Very recently, recurrent units are designed for decoding all-pair cost volumes in RAFT [36] and it achieves state-of-the-art result. These works [26, 18, 19, 13, 17] provide effective backbones for unsupervised methods in which flow is learned by optimizing photometric loss from synthesis views [45]. To address occlusions and large displacements, novel losses and training strategies are designed, such as bidirectional census loss in UnFlow [26], data distillation in DDFlow [18] and SelFlow [19], and extra forward pass in ARFlow [17].

Different from these works, we construct the final optical flow by fusing the motion from moving and static area guided by the learned rigid map M_R.

Depth estimation. Comparing with monocular estimation, learning depth from stereo images in the absence of the ground truth provides higher quality results. The self-supervision signal comes from the left-right synthesis view. In stereo works, Garg *et al*. [3] firstly adopt an auto-encoder to predict continuous values of disparity. Godard *et al*. [6] introduce a left-right consistency term for geometry constraint, then improves it by associated design choices [7]. Temporal information is also considered in [15, 46]. In monocular based methods, since single-view depth is usu-

ally insufficient for self supervision, extra information is borrowed from consecutive frames [48, 39, 23].

Unsupervised scene flow estimation. Traditional scene flow techniques have achieved impressive results, usually at high computational cost, such as super-pixel scene decomposition [28] and Plane+Parallax framework [42]. Even the fast version [35] still runs in 2-3 seconds per frame. In deep model, GeoNet [44] implicitly represents moving pixels by refining the residual non-rigid flow via ResFlowNet, and DF-Net [50] imposes a cross-task consistency loss for rigid area. Recently, per-pixel rigid constraint is adopted by incorporating deep segmentation network into scene flow. For instance, Ma *et al*. [22] adopt off-the-shelf Mask R-CNN [8] for rigid instance segmentation, and Yang *et al*. [43] predict moving masks via MotionNet followed by a holistic 3D motion parser (HMP). To simplify the multi-task training, Ranjan *et al*. [33] present Collaborative Competitive (CC) to facilitate the network coordination. Wang *et al*. [41] yield static pixels based on the flow residual, and Liu *et al*. [16] extend it via local rigidity. However, both are sensitive to thresholds, which may lead to the inaccurate motion area.

These methods achieve very impressive results, but the ego-motion is constrained in low efficiency due to the independent inference process. In our framework, we efficiently learn per-pixel rigidity by jointly coupling optical flow, depth and camera pose, leading to considerable improvements of each sub-task.

3. EffiScene Method

We first introduce the preliminary geometrical rigid consistency in Sec. 3.1, and then describe the design of RfM in Sec. 3.2. To present the new pipeline, we first construct the overall algorithm in Sec. 3.3, and discuss the design of the new loss as well as regularization functions in Sec. 3.4.

3.1. Geometrical Rigid Consistency

Given two consecutive frames I_t and I_{t+1}, pixel movements can be divided into two categories: *(i)* local motion from moving objects denoted by optical flow $F^o_{t\to t+1}$; and *(ii)* global (or ego) motion from backgrounds described by rigid flow $F^r_{t\to t+1}$. Compared with the optical flow $F^o_{t\to t+1}$, rigid flow $F^r_{t\to t+1}$ strictly follows the rigid geometrical consistency, hence it has lower corresponding ambiguity.

Rigid flow field $F^r_{t\to t+1}$ can be easily determined in 2D cases such as FlyingChairs [2], where the depth D is treated as a constant, by applying a 4-DoF plane affine transformation P. However, in realistic 3D scenes like KITTI suite [4], the geometrical consistency of global motion can be only guaranteed by re-projecting 2D points $\mathbf{x}$ back to 3D world coordinates $\mathbf{X}$ with perspective transformation:

$$[\mathbf{x};1] = \frac{1}{d}\cdot KPM[\mathbf{X};1], \tag{1}$$

where $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic parameter, $P \in \mathbb{R}^{3\times4}$ stands for the relative camera pose and $M \in \mathbb{R}^{4\times4}$ indicates the object movement in world coordinate system. Also, d is the normalization coefficient for $3D \to 2D$ projection, and it indicates the per-pixel depth for $\mathbf{x}$. Based on this, the movement of the static region (with $M = \boldsymbol{I}$) is computed by the differences between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, and can be formulated via the rigid flow $F^r_{t\to t+1}$ as shown below

$$F^r_{t\to t+1} = \frac{1}{d_{t+1}}\cdot KP(d_t\cdot P^{-1}\mathbf{x}_t) - \mathbf{x}_t. \tag{2}$$

Here, the depth d_t of the source image I_t and the camera pose P are the only two unknowns to be estimated in the computation of $F^r_{t\to t+1}$. With this geometrical rigid consistency, global motion from (2) are jointly considered with local motion in RfM for efficient rigid area recognition.

3.2. Rigidity from Motion (RfM)

Instead of independently predicting pixel rigidity from raw images as in [22, 43, 33], we classify the static rigid area based on optical flow $F^o_{t\to t+1}$ and rigid flow $F^r_{t\to t+1}$. This is similar to [41, 16] where a simple threshold is used to binary divide static and moving regions. However, the essential difference in our method comes from the consideration that global motion in $F^r_{t\to t+1}$ is strictly restricted by the geometrical rigid constraint whereas local motion from $F^o_{t\to t+1}$ is free. The static area can be naturally determined by regarding the local motion as the outliers of the global motion. However, finding 3D motion outliers from 2D flow field is non-trivial since *(i)* different 3D motion tensors can be represented by the same 2D flow vector; and *(ii)* the learned flow filed itself might be inaccurate. Hence, we need to design the Rigidity from Motion (RfM) layer to adaptively learn the motion boundary by explicitly modeling the correlation of various flows.

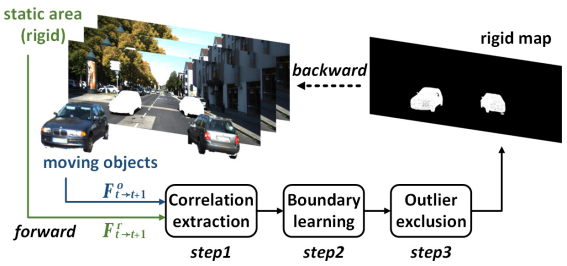

Figure 2. Illustration of RfM. (Flows are replaced by RGBs, and rigid map is shown in inverted colors for better visualization.)

Forward. The RfM involves three steps as illustrated in Fig. 2: *(i)* correlation extraction; *(ii)* boundary learning; and *(iii)* outlier exclusion. In the first place, we construct a pixel wise correlation map C_F as follows:

$$C_F = \mathcal{N}(f_c(F^o_{t\to t+1}, F^r_{t\to t+1})), \tag{3}$$

in which f_c evaluates the per-pixel similarity between $F^o_{t\to t+1}$ and $F^r_{t\to t+1}$, while the operator $\mathcal{N}$ normalizes the correlation value to $[0, 1]$. Usually, inner product can be the first choice to evaluate the correlation between any two vectors. However, it could be insufficient to distinguish the flow in the case that a pixel moves very slowly (near zero) along one direction. For example, in Fig. 3 (a), the inner product between green motion $F^r_{t\to t+1}$ and different blue motions $F^o_{t\to t+1}$ will yield the same results regardless of the v-axis moving of $F^o_{t\to t+1}$. To avoid this issue, we rely on the intuitive but more effective l_2-norm for motion residual (red arrows) to describe the motion similarity by $f_c = ||F^o_{t\to t+1} - F^r_{t\to t+1}||_2$. C_F evaluates the similarity between the two flows $F^o_{t\to t+1}$ and $F^r_{t\to t+1}$, which are more similar (with $C_F = 0$) at rigid region, while dissimilar (with $C_F = 1$) at non-rigid region. According to the Central Limit Theorem, the distribution of C_F in rigid region can be seen as a Gaussian with mean value 0, while non-rigid region fits another Gaussian with mean value 1. Based on this, secondly, we compute the overall histogram h_F of C_F, and naturally separate rigid and non-rigid pixels from the histogram by a Gaussian Mixture Model (GMM). To enforce differentiability, we approximate the GMM by designing a fully connected network $g(h_F|\theta)$ with learnable parameter θ. $g(h_F|\theta)$ automatically regresses the optimal rigid boundary by learning from the input h_F. And lastly, we construct the rigid map M_R in (4) to exclude local motion outliers from the global motion.

$$M_R = 1 - 1/(1 + \alpha \cdot (C_F - g(h_F|\theta))). \tag{4}$$

In this equation, α controls the balance between "hard" mask (large α) and "soft" mask (small α). In M_R, a value close to 1 indicates static rigid region whereas a value near 0 indicates the presence of a moving area.

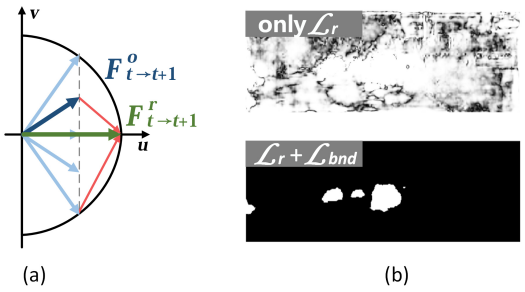

Figure 3. Illustration of (a) flow correlation and (b) effectiveness of boundary loss $\mathcal{L}_{bnd}$ to prevent trivial solutions.

Backward. RfM can be trained in self-supervision without any ground truth. Since the rigid area is given by M_R, the difference between image I_t and its background reconstruction (warped by rigid flow) at rigid area should be zero only when there is no moving object detected in M_R. Hence, we could optimize RfM in an end-to-end fashion via minimizing the rigid photometric loss $\mathcal{L}_r$ between I_t and $w_f(I_{t+1}, -F^r_{t\to t+1})$ on M_R, in which the Warping function $w_f(I, F)$ bilinearly interpolates image I according to the flow F. Formulation details are given in Section 3.4.

However, simply optimizing $\mathcal{L}_r$ is prone to trivial solutions, where $\mathcal{L}_r \to 0$ could be satisfied by generating an all(near)-zero rigid map from RfM as illustrated in Fig. 3 (b). The reason is that RfM tries to minimize $\mathcal{L}_r$ by reducing the number of inliers (rigid pixels). To prevent this, we design a new boundary loss $\mathcal{L}_{bnd}$ to encourage RfM to find out more inliers as much as possible by restricting the area ratio between rigid (M_R) and non-rigid ($1 - M_R$) regions.

$$\mathcal{L}_{bnd} = \frac{||1 - M_R||_1}{||M_R||_1}. \tag{5}$$

In (5), l_1-norm is designed to approximate the area for soft mask and enable end-to-end differentiability.

3.3. Overall Structure

Following the proposed pipeline, we construct our overall EffiScene structure in Fig. 4. In our framework, optical flow and depth are estimated by FlowCNN and DepthCNN, respectively. FlowCNN takes two consecutive frames I_t and I_{t+1} as inputs and computes a double channel optical flow $F^o_{t\to t+1}$ representing the horizontal and vertical pixel movements from time t to $t+1$. DepthCNN uses stereo left-right view image pair I_t and I^R_t at time t, and generates a single channel depth map D. Any existing deep models can be employed here for FlowCNN and DepthCNN. Camera pose $P = [R|\boldsymbol{t}]$ is made up of a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ with respect to the world. Since $F^o_{t\to t+1}$ and D are obtained, P can be computed in (6) by minimizing the reprojection error derived from (1) between the transformed coordinates $\mathbf{x}_{t+1} = \mathbf{x}_t + F^o_{t\to t+1}$ and the projected 3D points $KP\mathbf{X}_t = KP[\mathbf{x}_t + D]$:

$$\arg\min_P \sum ||[\mathbf{x}_{t+1}; 1] - \frac{1}{d} \cdot KP[\mathbf{X}_t; 1]||_2. \tag{6}$$

We follow [16] to solve the arg min problem by adopting the Perspective-n-Points (PnP) method from Simultaneous Localization And Mapping (SLAM) community with a Random Sample Consensus (RANSAC) scheme based on the Levenberg-Marquardt optimization. Once D and P determine the rigid flow $F^r_{t\to t+1}$ via geometrical consistency from (2), RfM will then adaptively recognize the pixel-wise scene rigidity by jointly considering the two flows. Operator E in Fig. 4 stands for the outlier exclusion step in RfM. Finally, we refine the fused flow via $F_{t\to t+1} = M_R F^r_{t\to t+1} + (1 - M_R)F^o_{t\to t+1}$ for more accurate estimation.

In EffiScene, different modules are closely coupled by RfM, and all components can be clearly interpreted from the

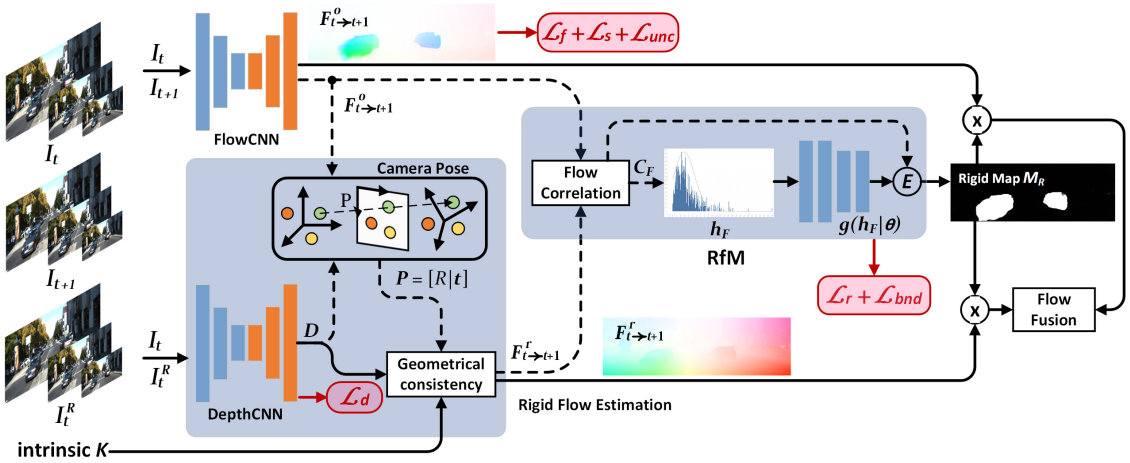

Figure 4. Overall architecture of EffiScene. The solid line and dashed line both indicate the forward propagation, but the gradient can only flow back through the solid line during training process due to the non-differentiable operations. Loss functions are shown in the red boxes.

geometrical view. Therefore, the efficient rigidity inference can be carried out via jointly considering the geometrical information from flow and depth.

3.4. Losses and Regularization

Photometric error evaluates the photometric similarity between two images I and $\hat{I}$ as defined in (7), in which λ_ρ balances the l_1-norm and SSIM term [49]:

$$\rho(I, \hat{I}) = \lambda_\rho l_1(I - \hat{I}) + (1 - \lambda_\rho)\text{SSIM}(I, \hat{I}). \quad (7)$$

We design different loss functions to train EffiScene in an unsupervised manner based on photometric error.

Optical Flow Loss. Optical flow from FlowCNN is optimized by minimizing the photometric error between the original image I_t and its reconstruction $\hat{I}_t^o$ from optical flow $F^o_{t\to t+1}$ on non-occluded region M_{noc}, which is determined by forward-backward flow check [50].

$$\mathcal{L}_f = \frac{1}{\sum M_{noc}} \sum\nolimits_\Omega M_{noc} \cdot \rho(I_t, \hat{I}_t^o). \quad (8)$$

Additionally, edge-aware smooth loss is used to regularize the optical flow on the full image domain Ω.

$$\mathcal{L}_s = \sum\nolimits_\Omega |\nabla^2 F^o_{t\to t+1}| e^{-|\nabla^2 I_t|}. \quad (9)$$

We use 2^{nd} order gradient to eliminate the velocity impact.

Depth Loss. Similar to optical flow, depth map from DepthCNN is trained with photometric loss and smooth loss, but for stereo pairs instead, e.g. left-view image I_t and synthesized image $\tilde{I}_t$ from right-view frame I^R. Left-right consistency from Godard *et al*. [6] is also adopted as penalty to ensure the stereo depth coherence below

$$\mathcal{L}_d = \sum\nolimits_\Omega \rho(I_t, \tilde{I}_t) + |\nabla^2 D| e^{-|\nabla^2 I_t|} + |D - \tilde{D}^L|, \quad (10)$$

where $\tilde{D}^L$ is the projected left-view depth from the right.

RfM Loss. As discussed in Sec. 3.2, the boundary loss $\mathcal{L}_{bnd}$ in (5) and the rigid photometric loss $\mathcal{L}_r$ in (11) are both used to train RfM. In $\mathcal{L}_r$, error between I_t and rigid reconstruction $\hat{I}_t^r$ from rigid flow $F^r_{t\to t+1}$ are evaluated on the rigid map M_R as

$$\mathcal{L}_r = \frac{1}{\sum M_R} \sum\nolimits_\Omega M_R \cdot \rho(I_t, \hat{I}_t^r). \quad (11)$$

Regularization. Minimizing $\mathcal{L}_f$ may cause the discontinuity of optical flow at image boundary due to the uncovered area between two frames. We use rigid flow to rectify the optical flow by enforcing flow consistency based on the learned rigid map M_R from RfM. However, M_R might cover undesired moving objects at the beginning of the training. We improve M_R via generating robust uncover region Ω_{unc} by fusing occlusion mask (M_{occ}), valid optical flow mask (M_{opt}) and valid rigid flow mask (M_{rig}) as shown in Fig. 5. The non-occluded region is defined as

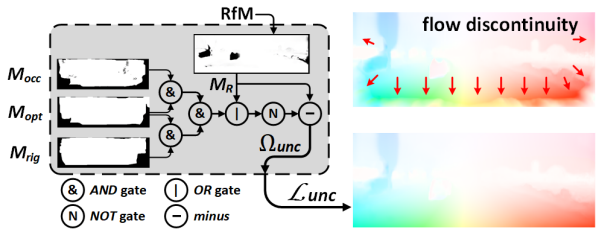

Figure 5. Regularizing optical flow for boundary discontinuity.

M_{occ} while M_{opt} and M_{rig} indicate valid motion from optical flow and rigid flow. Next, the uncover loss for flow regularization is defined as:

$$\mathcal{L}_{unc} = \sum\nolimits_{\Omega_{unc}} ||F^o_{t\to t+1} - F^r_{t\to t+1}||_2^2. \quad (12)$$

All losses are combined to train EffiScene as an energy minimization optimization as follows:

$$E = \lambda_f \mathcal{L}_f + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{L}_r + \lambda_d \mathcal{L}_d + \lambda_{bnd} \mathcal{L}_{bnd} + \lambda_{unc} \mathcal{L}_{unc}, \quad (13)$$

where $\lambda_{f/s/r/d/bnd/unc}$ provide the weighting trade-offs.

4. Experiments

Extensive experiments are conducted to benchmark EffiScene against SOTA scene flow methods in four sub-tasks: *(i)* optical flow estimation; *(ii)* depth prediction; *(iii)* visual odometry; and *(iv)* motion segmentation. Qualitative results are illustrated in Fig. 6. More results are listed in the Supplementary Materials.

4.1. Implementation Details

Dataset. To keep consistency with previous works [44, 50, 33, 41, 16, 6, 43, 21], we use the same dataset and protocol for all experiments. Specifically, 28,968 images out of (42,382) images in KITTI raw set [4] are used to train EffiScene, except for the scenes enrolled in KITTI 2015 [29] training set, which is reserved for optical flow validation as well as depth estimation and motion segmentation with corresponding ground truth. Besides KITTI 2015, KITTI 2012 [5] is also adopted for optical flow evaluation. Different from KITTI 2015, dynamic scenes in KITTI 2012 only contains camera movements but no moving cars. For visual odometry task, we fine-tune our model on sequences 00-08 in KITTI Odometry split [5], then test it on sequences 09 and 10.

Network deployment. For FlowCNN, we employ RAFT [36] as the baseline due to its excellent performance in supervised optical flow estimation and make a few modifications for the unsupervised setting. We also modify PWC-Net [34] for DepthCNN by changing the output of the last convolutional layer from 2 channels to 1 channel to generate a single channel depth map, and replace the *deconv* layer by bilinear upsampling to avoid the checkerboard artifacts. In RfM, $g(f_F|\theta)$ is obtained from two fully connected layers with size 100-32 and 32-1 followed by ReLU and Sigmoid activation, respectively.

Training. We train EffiScene from scratch in three stages without any ground truth. By default, photometric balance λ_ρ in (7) is set to 0.003 for all experiments. Weighting for loss functions denoted by $\{\lambda_f, \lambda_s, \lambda_r, \lambda_d, \lambda_{bnd}, \lambda_{unc}\}$ in (13) are initialized to all zeros, then we adjust them in different stages. In the first stage, we train FlowCNN and fix DepthCNN and RfM to obtain a coarse optical flow $F^o_{t\to t+1}$ by setting $\lambda_f = 1.0$ and $\lambda_s = 0.5$ for 20 epochs. In parallel, we train DepthCNN independently for depth D by setting $\lambda_d = 1.0$ for 50 epochs as suggested in [6]. Once we achieve a reasonable optical flow and depth, we fix FlowCNN and DepthCNN, and train RfM for 10 epochs in the second stage. Here, we set $\lambda_r = 1.0$ and $\lambda_{bnd} = 0.023$, and others to zeros. Finally, in the last stage, we jointly fine-tune all the networks based on the fused flow $F_{t\to t+1}$ by setting $\{\lambda_r, \lambda_d, \lambda_{bnd}, \lambda_{unc}\} = \{1.0, 1.0, 0.023, 1.0\}$ for 10 epochs.

All input images are resized to 256×832, and the AdamW optimizer [20] is utilized for optimization with momentum [0.9, 0.99] and weight decay 1e-5. Batch size is set to 4. The initial learning rate is first set to 1e-4 for the first two training stages, reduced to 1.25e-5 for the last stage, and it is decreased by a factor of 2 for every 50K batch. All models are trained on a single Tesla P40 GPU for about 150 GPU hours. Unlike existing methods [16, 41] whose performances are highly relied on the prefixed thresholds, there is no empirical parameter needed to be set by the user in EffiScene in both training and testing.

4.2. Evaluation

Optical flow estimation. Optical flow comparisons with supervised and unsupervised methods are summarized in Tab. 1. On KITTI 2015, our method achieves the best performances on averaged end-point-error (EPE) across all image regions, e.g. moving area, static area ... Specifically, for the most vital metric EPE-All, EffiScene significantly reduces the existing error by a considerable margin from 5.19 [16] to 4.20 (19.1% relative improvement). We also achieve the best and the second best Fl-all error 14.31% and 13.08% among all approaches on training and testing set, respectively. On KITTI 2012 dataset, EffiScene consistently surpasses UnRigidFlow [16] by 12.5% relative EPE growth (1.68 vs. 1.92), which validates the generalization ability of our method. Unfortunately, since there is no moving object in KITTI 2012, the learned rigidity mask M_R from RfM will cover almost the full image, leading to that the fused flow $F_{t\to t+1}$ will be dominated by the rigid flow $F^r_{t\to t+1}$ (from depth and pose). Hence, it is challenging for EffiScene to benefit from FlowCNN, resulting in a slight drop of EPE to 1.68, comparing to the best achievable EPE=1.64 [41]. However, for occluded region, motion can be better inferred by more accurate depth and pose, and we obtain the best result EPE-Occ of 4.71 (vs. 5.18 [41]). In addition, we also design a variation model EffiScene (-pwc) based on the popular PWC-Net [34] backbone for further evaluation. PWC-based EffiScene also achieves SOTA results for both datasets, demonstrating the consistency as well as robustness of the proposed framework.

Ablation. Optical flow from different training stages are listed in Tab. 2. It is not surprising that optical flow F^o from the 1st training stage yields the worst performance since the geometrical rigidity constraint has not been considered. With the help of RfM, in the 3rd stage, FlowCNN and DepthCNN are jointly optimized based on specific rigid regions M_R, and lower errors can be achieved in moving area (EPE-Move=3.09) as well as static regions (EPE-Static=2.09). By jointly fusing F^o and F^r, the final output F generates a much better result for all regions with EPE=4.20 and Fl-all=14.31%.

Stereo-depth prediction. Depth estimation is evaluated on KITTI train set with standard metrics [23] in Tab. 3. By jointly considering the inherent relation between

Figure 6. Qualitative results of the proposed method for scene flow estimation (denoted by 'est.'.).

Table 1. Quantitative results of optical flow estimation. Averaged end-point-error (EPE) is used for evaluation except for the last two columns which tabulate the percentage of erroneous pixels (Fl-all). 'Noc' and 'Occ' mean non-occluded region and occluded region.

Method	Stereo	Super-vised	KITTI 2012 Train Average EPE			KITTI 2015 Train Average EPE			Train	Test
			Noc	Occ	All	Move	Static	All	Fl-all	Fl-all
FlowNet2 [12]		✓	-	-	4.09	-	-	10.06	30.37%	-
PWC-Net [34]		✓	-	-	4.14	-	-	10.35	33.67%	-
UnFlow-CSS [26]			1.26	-	3.29	-	-	8.10	23.27%	-
DF-Net [50]			-	-	3.54	-	-	8.98	26.01%	25.70%
Self-Mono-SF [11]			-	-	-	-	-	7.51	23.49%	23.54%
CC [33]			-	-	-	5.67	5.04	6.21	26.41%	-
CC-uft [33]			-	-	-	-	-	5.66	20.93%	25.27%
EPC++ [21]	✓		-	-	1.91	-	-	5.43	-	20.52%
UnOS [41]	✓		**1.04**	5.18	**1.64**	5.30	5.39	5.58	-	18.00%
UnRigidFlow [16]	✓		1.09	4.87	1.92	7.92	3.85	5.19	14.68%	**11.66%**
EffiScene (-pwc)	✓		1.19	4.74	1.71	7.63	3.72	4.92	14.55%	-
EffiScene	✓		1.19	**4.71**	1.68	**5.15**	**3.69**	**4.20**	**14.31%**	13.08%

Table 2. Ablation study on optical flow. Subscript $_{t\to t+1}$ of $F^o_{t\to t+1}$, $F^r_{t\to t+1}$ and $F_{t\to t+1}$ has been omitted for clarity.

Flow Type	Train Stage	Train Average EPE			Train Fl-all
		Move	Static	All	
F^o (FlowCNN)	1st	4.38	6.80	6.76	18.89%
F^o (FlowCNN)	3rd	**3.09**	4.81	4.70	15.87%
F^r (DepthCNN)	3rd	38.10	**2.09**	10.42	21.44%
F (EffiScene)	3rd	5.15	3.69	**4.20**	**14.31%**

flow, pose, rigidity and depth, superior depth maps can be predicted comparing with both monocular or stereo based methods. Surprisingly, EffiScene even outperforms SsSMnet [47] and MonoDepth [6] which are specifically designed for the depth estimation task. Since there is no new depth-specific components designed in EffiScene, we hypothesize that the gain in depth estimation may come from the collaborative training process, where optical flow, camera pose and depth are coupled by RfM for mutual reinforcement.

Motion segmentation. Results from RfM is also used to evaluate motion segmentation with advanced scene flow approaches as listed in Tab. 4. We achieve the best pixel accuracy and mean accuracy by surpassing the baseline UnOS [41] 4.5% and 2.8%. However, considering that moving cars just make up only a small portion of the full image in KITTI 2015 (usually less than 5%), high accuracy does not always imply a superior segmentation ability due to severe class imbalance. Therefore, Intersection-Over-Union (IoU) could be a fairer and more compelling benchmark as listed in the last two columns of Tab. 4. Our method improves the mean IoU from SOTA 0.570 [16] to 0.615, and the frequency weighted (f.w.) IoU from 0.900 to 0.926. Note that CC [33] adopts a much more complex and deep auto-encoder for segmentation, but it is still 4.6% lower than the proposed method because of the independent segmentation inference structure.

Visual odometry. Absolute Trajectory Error (ATE) in [30, 48] is utilized for camera pose evaluation in Tab. 5. Two types of technical strategies are summarized: Deep Neural Netwrok (DNN) based and PnP based pose estimation. Usually, DNN based methods run faster than optimization-based PnP, but offer lower accuracy [40, 24, 41]. Since only 2 frames are used in our method, we follow [41, 16] and average the accumulated poses from neighboring 5 frames for fair comparison with multi-frame competitors. For PnP based methods, we outperform all existing methods in both sequences. Note that the performance of PnP heavily depends on the quality of predicted flow and depth, better optical flow and depth estimation from previous steps will definitely contribute to improvement in camera pose accuracy. For DNN approaches, we also give a variation model EffiScene (-dnn) for fair comparison – replacing the PnP method with a 9-layer fully convolutional network with channels [16, 32, 64, 128, 256*4, 6] to regress the 6-DoF pose matrix $P = [R|\boldsymbol{t}]$. It achieves competitive results by surpassing [48] and [50], but it narrowly trails [44] and [33], both having access to more frames than us.

Table 3. Quantitative results of depth estimation conducted on the KITTI 2015 training set. Depth errors (middle columns) and prediction accuracy (right columns) are used for evaluation. All valid depth ranges are capped at 80m.

Method	Stereo	Error (lower is better)				Accuracy, δ (higher is better)		
		AbsRel	SqRel	RMSE	RMSlog	< 1.25	$< 1.25^2$	$< 1.25^3$
CC [33]		0.140	1.070	5.326	0.217	0.826	0.941	0.975
Self-Mono-SF [11]		0.125	0.978	4.877	0.208	0.851	0.950	0.978
EPC [43]	✓	0.109	1.004	6.232	0.203	0.853	0.937	0.975
EPC++ [21]	✓	0.127	0.936	5.008	0.209	0.841	0.946	0.979
SsSMnet [47]	✓	0.075	1.726	4.857	0.165	0.956	0.976	0.985
MonoDepth [6]	✓	0.068	0.835	4.392	0.146	0.942	0.978	0.989
UnRigidFlow [16]	✓	0.051	0.532	3.780	0.126	0.957	0.982	0.991
EffiScene	✓	**0.049**	**0.522**	**3.461**	**0.120**	**0.961**	**0.984**	**0.992**

Table 4. Quantitative results of motion segmentation. IoU based metrics (last two columns) are more meaningful for KITTI 2015.

Method	Pixel Acc.	Mean Acc.	Mean IoU	f.w. IoU
EPC [43]	0.890	0.750	0.520	0.870
EPC++ [21]	0.910	0.760	0.530	0.870
UnOS (full) [41]	0.900	0.820	0.560	0.880
CC [33]	-	-	0.569	-
UnRigidFlow [16]	0.930	0.840	0.570	0.900
EffiScene	**0.945**	**0.848**	**0.615**	**0.926**

Table 5. Comparisons of visual odometry on KITTI Odometry.

Method	Frames	Pose	Sequence 09	Sequence 10
DF-Net [50]	5	DNN	0.017±0.007	0.015±0.009
sfMLearner [48]	5	DNN	0.016±0.009	0.013±0.009
GeoNet [44]	5	DNN	0.012±0.007	0.012±0.009
CC [33]	5	DNN	**0.012±0.007**	**0.012±0.008**
EffiScene (-dnn)	2	DNN	0.013±0.006	0.013±0.008
ORB-SLAM [30]	all	PnP	0.014±0.008	0.012±0.011
Vid2Depth [23]	3	PnP	0.013±0.010	0.012±0.011
UnOS [41]	2	PnP	0.012±0.006	0.013±0.008
UnRigidFlow [16]	2	PnP	0.012±0.007	0.012±0.006
EffiScene	2	PnP	**0.011±0.006**	**0.011±0.008**

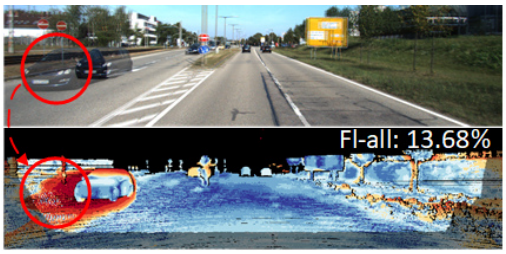

Figure 7. High errors caused by inaccurate R_M and occlusion.

4.3. Analysis

Complexity analysis. Running time and model size are listed in Tab. 6. Our method runs more than 2 times faster than UnOS, but is slower than CC due to the time consuming PnP step as discussed in Sec. 4.2. By replacing PnP with a deep network, EffiScene (-dnn) significantly speeds up the inference with acceptable performance drop (0.002 drop from Tab. 5). EffiScene requires much fewer number of learnable parameters in the full model (#Params) and the segmentation module (#SegParams).

Table 6. Model complexity analysis. Experiments are performed on the same computing platform with a single Tesla P40 GPU.

Method	RunTime (ms)	FPS (f/s)	#Params (Mb)	#SegParams (Mb)
CC [33]	49.55	20.18	74.26	5.22
UnOS [41]	228.16	4.38	17.06	-
UnRigidFlow [16]	87.57	11.42	**10.22**	-
EffiScene	93.11	10.74	10.36	**0.0032**
EffiScene (-dnn)	**47.06**	**21.25**	12.54	**0.0032**

Limitations. Although promising EPE (=4.20) is achieved in our model, test Fl-all error (13.08%) is still 1.42% higher than [16] as depicted in Tab. 1. This indicates that EffiScene learns better optical flow for those 'good' regions (lower EPE), but not for all pixels (higher Fl). One reason is that the learned rigidity map M_R could be wrongly estimated at occluded regions, where local motion is difficult to optimize due to missing pixels, resulting in unreliable motion correlation for RfM. For example, in Fig. 7, backgrounds occluded by the moving cars have much higher errors. Therefore, improving RfM in the large occlusion case seems to be a logical next step.

5. Conclusion

In summary, we propose EffiScene for unsupervised scene flow estimation by coupling several low-level vision sub-tasks. We demonstrate that per-pixel rigidity can be efficiently inferred by jointly exploiting optical flow, depth and camera pose, since they share the same inherent geometrical structure with scene rigidity. By exploring joint rigidity learning, more accurate rigid constraint and efficient network training can be achieved. Extensive experiments on scene flow benchmarks produce SOTA results with simpler model for all four sub-tasks, demonstrating the effectiveness of the proposed method. In our future work, long term dependency will be explored in EffiScene to solve the rigidity inference with large occlusions.

References

[1] Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002. 1

[2] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2, 3

[3] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 740–756, Cham, 2016. Springer International Publishing. 2

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 3, 6

[5] Andreas Geiger, Philip Lenz, and Raquel 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, 2012. 2, 6

[6] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 5, 6, 7, 8

[7] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[9] Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic future prediction for video scene understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 767–785, Cham, 2020. Springer International Publishing. 1

[10] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[11] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7, 8

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 2, 7

[13] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[14] Huaizu Jiang, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz. Sense: A shared encoder network for scene-flow estimation. In *ICCV*, 2019. 1

[15] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291, 2018. 2

[16] Liang Liu, Guangyao Zhai, Wenlong Ye, and Yong Liu. Unsupervised learning of scene flow estimation fusing with local rigidity. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 876–882. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 2, 3, 4, 6, 7, 8

[17] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[18] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *AAAI*, 2019. 2

[19] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[20] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 6

[21] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 6, 7, 8

[22] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 3

[23] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 6, 8

[24] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7

[25] David Marr. *A computational investigation into the human representation and processing of visual information*. the MIT press, Freeman, San Francisco, CA, 1982. 1

[26] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional cen-

sus loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018. 2, 7

[27] Yue Meng, Yongxi Lu, Aman Raj, Samuel Sunarjo, Rui Guo, Tara Javidi, Gaurav Bansal, and Dinesh Bharadia. Signet: Semantic instance aided unsupervised 3d geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 3

[29] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 6

[30] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 7, 8

[31] Naoya Ohnishi and Atsushi Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recognition Letters*, 27(9):1009 – 1021, 2006. 1

[32] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[33] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 6, 7, 8

[34] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan 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 (CVPR)*, June 2018. 2, 6, 7

[35] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[36] Zachary Teed and Jun Deng. Raft: Recurrent all-pairs field transforms for optical flow. *ArXiv*, abs/2003.12039, 2020. 2, 6

[37] Sundar Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480, 2005. 1

[38] Sundar Vedulay, Simon Bakery, Peter Randeryz, Robert Collinsy, , and Takeo Kanadey. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729 vol.2, 1999. 1

[39] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *CoRR*, abs/1704.07804, 2017. 3

[40] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018. 7

[41] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 6, 7, 8

[42] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[43] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018. 1, 3, 6, 8

[44] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 6, 7, 8

[45] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 3–10, Cham, 2016. Springer International Publishing. 2

[46] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian 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 (CVPR)*, June 2018. 2

[47] Yiran Zhong, Yuchao Dai, and Hongdong Li. Self-supervised learning for stereo matching with self-improving ability. *ArXiv*, abs/1709.00930, 2017. 7, 8

[48] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 3, 7, 8

[49] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[50] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3, 5, 6, 7, 8