

Flow-Motion and Depth Network for Monocular Stereo and Beyond

Kaixuan Wang and Shaojie Shen

Abstract—We propose a learning-based method¹ that solves monocular stereo and can be extended to fuse depth information from multiple target frames. Given two unconstrained images from a monocular camera with known intrinsic calibration, our network estimates relative camera poses and the depth map of the source image. The core contribution of the proposed method is threefold. First, a network is tailored for static scenes that jointly estimates the optical flow and camera motion. By the joint estimation, the optical flow search space is gradually reduced resulting in an efficient and accurate flow estimation. Second, a novel triangulation layer is proposed to encode the estimated optical flow and camera motion while avoiding common numerical issues caused by epipolar. Third, beyond two-view depth estimation, we further extend the above networks to fuse depth information from multiple target images and estimate the depth map of the source image. To further benefit the research community, we introduce tools to generate photorealistic structure-from-motion datasets such that deep networks can be well trained and evaluated. The proposed method is compared with previous methods and achieves state-of-the-art results within less time. Images from real-world applications and Google Earth are used to demonstrate the generalization ability of the method.

I. INTRODUCTION

Due to the rich information in images, structure-from-motion (SfM) is of vital importance in computer vision and robotics. Given a set of unconstrained images, SfM aims to estimate the depth maps and the relative camera poses. Traditional systems, for example, COLMAP [1], [2], first estimate the relative poses of cameras by finding correspondences of sparse feature points and then use the estimated camera pose to calculate dense depth maps. The extracted sparse features ignore other information in the images, such as lines, and does not contribute to the following dense depth estimation. Scene priors such as structures and object shapes are also hard to be integrated into the pipeline of traditional methods.

To better utilize image information and exploit context priors, many methods [3]–[5] have been proposed to solve monocular stereo (two-view SfM) problems using convolutional neural networks (CNNs). DeMoN [3] is a pioneering work that first estimates an optical flow and then decomposes it into a depth map and camera pose. The optical flow, depth maps, and camera poses are then iteratively refined by a chain of encoder-decoder networks to handle large viewing angles. LS-Net [4] uses a predicted depth map and camera pose as the initialization to iteratively minimize the photometric reprojection error through a learning-based solver. Different from LS-Net where the update steps are computed by a

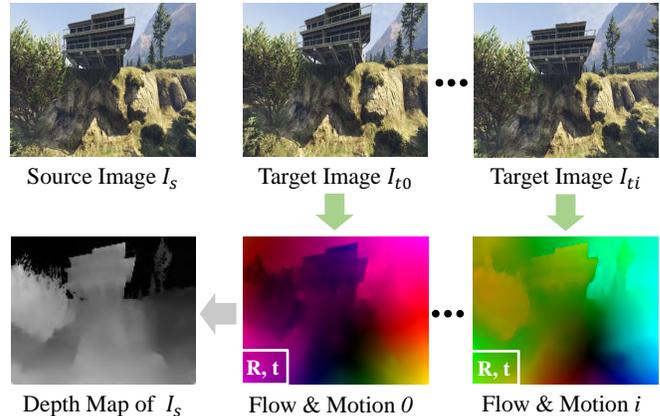


Fig. 1. Illustration of the proposed method. Given multiple images from a moving monocular camera, the flow-motion network (green in the figure) first estimates the optical flow and camera pose between the source image and each target image. The estimated flow and motion are further fused by the depth network (gray in the figure) to compute the depth map of the source image.

network, BA-Net [5] proposes a bundle adjustment layer to predict the damping factor of the Levenberg-Marquardt algorithm [6] and calculates the update. To further reduce the optimization space, BA-Net also parameterizes the depth map as a linear combination of 128 single-view predicted basis maps. Utilizing the information of the whole image, the above methods generate robust camera poses and smooth depth maps. Although these methods achieve impressive results compared with traditional methods, they need multiple iterations (e.g. 15 iterations in LS-Net and BA-Net) to converge, and most methods (e.g. LS-Net and DeMoN) estimate the depth map using only one target frame.

In this letter, we improve both the efficiency and accuracy of the state of the art by incorporating domain knowledge and further extend the method to fuse multiple depth information. The first contribution of our work is a joint estimation of the optical flow and camera poses. We observe that, in monocular stereo problems, the optical flow between multiview images is caused by the ego-motion of a moving camera in static scenes such that the optical flow is constrained along the epipolar lines. By jointly considering the optical flow and camera poses, the pixel search space can be gradually reduced, improving both the efficiency and accuracy. A novel triangulation layer is proposed to encode the estimated optical flow and camera motion without numerical problems caused by unconstrained camera movements. The encoded information from the triangulation layer is used to estimate the depth map of the source image. In many applications, the source image is observed by multiple target images. Beyond the two-view problem, we further extend the networks to fuse

All authors are with ECE, HKUST, Hong Kong, China. {kwangap, eshaojie}@ust.hk.

¹<https://github.com/HKUST-Aerial-Robotics/Flow-Motion-Depth>

depth information from multiple target frames. The depth information from different image pairs is fused by mean-pooling layers and is then used to predict the depth map. Figure 1 shows the workflow of our method estimating the optical flow, camera poses, and depth map given multiple images. By exploiting multiview observations, robust and accurate depth maps can be generated.

Training and evaluating learning-based SfM methods requires lots of images with ground truth camera poses and depth maps. Existing datasets, for example, SUN3D [7] and Scenes11 [3], contain either low-quality images from RGB-D cameras or non-photorealistic synthetic images. To train and evaluate our proposed networks, we develop tools that can generate unlimited high-quality photorealistic images with ground truth depth maps and camera poses from the game Grand Theft Auto V (GTA5). For the benefit of the computer vision community, we release the tools and generated datasets as open source.

To summarize, the contributions of the letter are the following:

- A network that jointly estimates optical flow and camera poses given two-view images. With the estimated camera poses, the optical flow is constrained on epipolar lines such that the flow can be regularized, and the search space is reduced.
- A novel triangulation layer that encodes the estimated optical flow and camera pose so that the depth network can triangulate the depth of each pixel without numerical problems.
- The depth network is further extended to fuse depth information (e.g. flow and motion) from multiple image pairs. By fusing multiple observations, the depth of the source image can be estimated more accurately and robustly.
- Open source tools to customize unlimited photorealistic synthetic images with different daytime, intrinsic parameters, etc. The extracted images serve as a supplementary dataset to train and evaluate learning-based SfM methods.

II. RELATED WORK

In this section, we outline related work using neural networks to estimate the camera poses and depth maps given two or more images.

DeMoN [3] is a pioneering work that jointly estimates depth maps and camera poses given two-view images. To effectively use the two-view observations, DeMoN adapts FlowNetS [8] to first estimate the optical flow between two images, and then decomposes the flow into camera poses and depth maps. To further improve the quality, DeMoN iteratively refines the optical flow, camera pose and depth map using two encoder-decoder networks, and finally upsamples the depth map into a higher resolution.

CodeSLAM [9] and BA-Net [5] parameterize depth maps as compact representations such that both the camera motion and depth map can be solved explicitly by classic optimization methods. CodeSLAM uses an auto-encoder and decoder

to represent the depth map as a function of the corresponding image and unknown code. The unknown code can be solved jointly with the camera pose by minimizing the photometric error and geometric error. Benefiting from the flexibility of the classic optimization, CodeSLAM can simultaneously estimate multiple depth maps and camera poses. To make the depth representation suitable for SfM tasks, BA-Net embeds the bundle adjustment as a differentiable layer into the network and the whole process is end-to-end trainable. Unlike CodeSLAM and BA-Net, LS-Net [4] trains a CNN as a least-square solver to update camera poses and depth values. Starting from initialized depth maps and camera poses, these methods need multiple iterations to converge.

Many approaches have been proposed to solve multiview stereo or camera tracking using neural networks. Given multiple images with known camera poses and intrinsic calibration, DeepMVS [10] generates cost volumes using learned feature maps and then estimates the disparity map by fusing multiple cost volumes. MVDepthNet [11], DPSNet [12] and MVSNet [13], [14] solve the same reconstruction problem but differ in the calculation of cost volumes and the structure of networks. On the other hand, given an RGB-D keyframe, DeepTAM [15] incrementally tracks the pose of a camera using synthetic viewpoints and can further estimate the depth map of the tracked frame.

Here, we propose a method that is different from all the monocular stereo methods mentioned above. The major difference is that our method does not iteratively refine the estimation but rather generates results using only one forward pass in the flow-motion network and depth network. The key to the improved efficiency and quality is the joint estimation of both optical flow and camera motion. The high-quality optical flow directly establishes precise dense pixel correspondences between images, enabling accurate depth triangulation. Also, the proposed method can be extended to estimate the depth map of the source image by fusing the information from multiple target images.

III. NETWORK ARCHITECTURE

As shown in Figure 2, the proposed method consists of two networks: one flow-motion network and one depth network. Given a source image I_s and a target image I_t of a static scene, the flow-motion network estimates the optical flow between two images and the relative camera pose in a coarse-to-fine manner. With camera poses estimation, the search space of the optical flow can be gradually reduced along the epipolar line. Moreover, the aperture problem in optical flow can be reduced by the epipolar line constraint. With the estimated optical flow and camera motion, the depth value of each pixel can be directly triangulated. However, the triangulation step is not numerically stable around the epipolar [16]. Instead, we propose a triangulation layer to encode the information of the estimated optical flow and camera poses. The layer is processed by the depth network to estimate the depth map of I_s . The depth network can also be extended to fuse the information from multiple target images. When the source image is observed by multiple

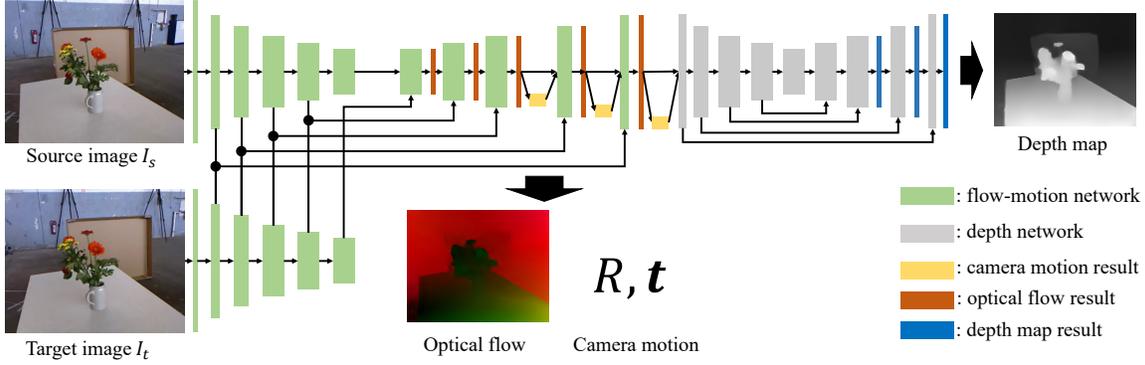


Fig. 2. The architecture of the proposed flow-motion network and depth network. Here, only the two-view architecture is shown for simplicity. The extension to fuse multiple depth information is shown in Figure 5 and is discussed in Section III-C. The flow-motion network jointly estimates the optical flow and camera poses, and the depth network triangulates the depth of each pixel in the source image. Although both networks are based on simple encoder-decoder architectures, the proposed joint estimation (Section III-A) and triangulation layer (Section III-B) enables high-quality and efficient estimation.

target images, the depth map of the source image can be solved by fusing information from *all* source-target pairs.

In the following sections, we first explain the design of the flow-motion network, the depth network that process two-frame SfM problems. In Section III-C, the depth network is further extended to fuse multiple depth information and estimate the depth map of the source image.

A. Flow-Motion Network

A number of works [8], [17]–[19] have shown the success of using CNNs to estimate dense optical flow between two images. The proposed flow-motion network shares similar structures to the state-of-the-art PWC-Net [19] but is tailored for static scenes and jointly estimates camera poses.

To be robust to lighting and viewing angle changes, input images are converted into L-level feature pyramids using a simple CNN. The feature map at the i -th level, \mathbf{f}^i , is processed by three simple convolutional layers to generate the next level feature map \mathbf{f}^{i+1} with the size downsampled by 2. In this work, $L = 6$ pyramid levels are used, with \mathbf{f}^0 being the original 3-channel image. \mathbf{f}_s^i and \mathbf{f}_t^i are used to denote the i -th level feature maps of I_s and I_t , respectively.

The optical flow \mathbf{w} is estimated from coarse to fine to handle large pixel displacement. At the i -th level, the optical flow \mathbf{w}^{i+1} from the $i+1$ -th level is firstly bilinear upsampled into \mathbf{w}_{up}^{i+1} as an initialization of \mathbf{w}^i . A cost volume \mathbf{c}^i is constructed using \mathbf{f}_s^i and \mathbf{f}_t^i . Each element in the cost volume represents the feature similarity between a pixel \mathbf{x}_s in \mathbf{f}_s^i and a pixel \mathbf{x}_t in \mathbf{f}_t^i ,

$$\mathbf{c}^i(\mathbf{x}_s, \mathbf{x}_t) = \frac{1}{N_i} (\mathbf{f}_s^i(\mathbf{x}_s))^T \mathbf{f}_t^i(\mathbf{x}_t), \quad (1)$$

and N_i is the feature dimension of \mathbf{f}_s^i . Due to the coarse-to-fine manner, only a subset of pixels in \mathbf{f}_t^i is needed to calculate the cost volume. The cost volume \mathbf{c}^i , upsampled optical flow \mathbf{w}_{up}^{i+1} , and \mathbf{f}_s^i are used to predict the optical flow \mathbf{w}^i using the DenseNet [20] structure.

The above cost volume construction and optical flow estimation are repeated from coarse to fine until the optical flow of the desired resolution is estimated. In this work, we

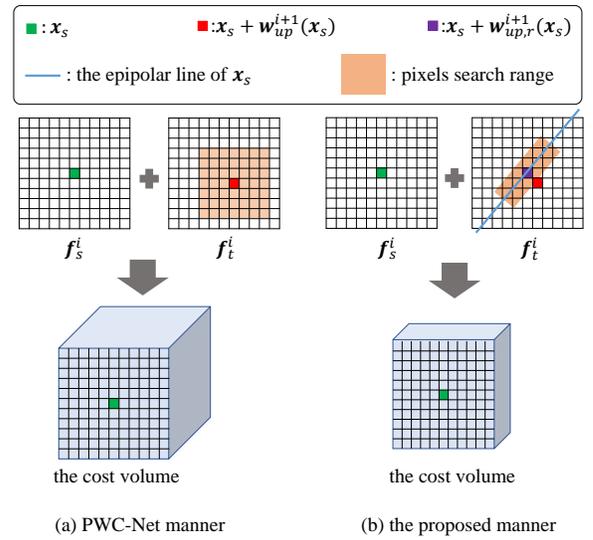


Fig. 3. Differences between the cost volume computation in PWC-Net (left) and the proposed flow-motion network (right). For each pixel \mathbf{x}_s in \mathbf{f}_s^i , PWC-Net matches a fixed set of pixels (colored in orange) around $\mathbf{x}_s + \mathbf{w}_{up}^{i+1}(\mathbf{x}_s)$ to generate the cost volume. On the other hand, the proposed flow-motion network first regularizes the initial flow $\mathbf{w}_{up}^{i+1}(\mathbf{x}_s)$ into $\mathbf{w}_{up,r}^{i+1}(\mathbf{x}_s)$ and matches pixels around the epipolar line.

adapt and improve the above processes by incorporating the static scene prior and jointly estimating the camera pose.

In different pyramid levels, several convolutional layers and linear layers are used to predict the pose of the source frame with respect to the target frame. The pose consists of a rotation matrix R and a translation vector \mathbf{t} . With the estimated camera motion and calibrated intrinsic K , the flow vector of each pixel can be regularized along the corresponding epipolar line and the search space of pixels in the cost volume can be narrowed down.

In static environments, pixel \mathbf{x} in the source image and its optical flow vector $\mathbf{w}(\mathbf{x})$ to the target image have the following relationship,

$$\begin{bmatrix} \mathbf{x} + \mathbf{w}(\mathbf{x}) \\ 1 \end{bmatrix}^T F \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0, \quad (2)$$

where $F = K^{-T} \mathbf{t}_\times R K^{-1}$ is the fundamental matrix. With

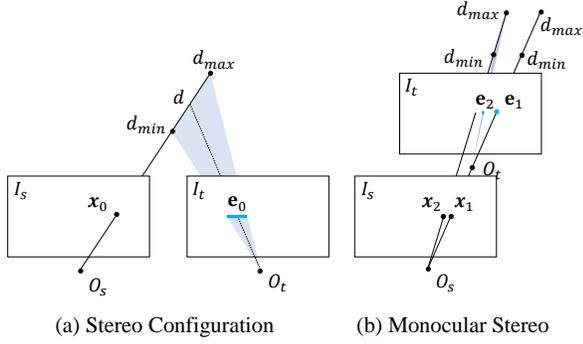


Fig. 4. Example to show the numerical stability in triangulation. O_s and O_t are the optical centers of I_s and I_t , respectively. d_{max} and d_{min} are the maximum and minimum depth of the scene. e_i is the corresponding epipolar line of pixel \mathbf{x}_i . (a) In stereo configurations, the depth can be reliably calculated by finding the corresponding point on e_0 . (b) In unconstrained monocular stereo problems, the epipolar line e_1 (the epipolar point) degenerates into a point, thus the depth is unobservable. For pixels near the epipolar point, such as \mathbf{x}_2 , the epipolar line e_2 is very short, and the result is noise-prone.

the estimated camera pose, the upsampled optical flow of each pixel $\mathbf{w}_{up}^{i+1}(\mathbf{x})$ can be regularized by projecting the corresponding point to the epipolar line,

$$\mathbf{w}_{up,r}^{i+1}(\mathbf{x}) = \frac{1}{e_x^2 + e_y^2} \begin{bmatrix} x'e_y^2 - y'e_x e_y - e_x e_z \\ y'e_x^2 - x'e_x e_y - e_y e_z \end{bmatrix} - \mathbf{x}, \quad (3)$$

where $[e_x, e_y, e_z]^T = F[\mathbf{x}, 1]^T$ and $[x', y']^T = \mathbf{x} + \mathbf{w}_{up}^{i+1}(\mathbf{x})$.

Since the corresponding pixels are constrained on epipolar lines, it is not necessary to match pixels far from the lines. Also, the aperture problem, where the pixel correspondences cannot be determined due to the ambiguous matchings, can be reduced by incorporating the epipolar line constraint. However, the epipolar lines, which are determined by the estimated camera poses, may not be accurate enough to rule out all pixels off the lines. Here, we gradually decrease the search space from coarse pyramid levels to fine levels. In the i -th level, the matching pixels of pixel \mathbf{x}_s is parameterized as

$$\mathbf{x}_t \in \left\{ \mathbf{x}_s + \mathbf{w}_{up,r}^{i+1}(\mathbf{x}_s) + \frac{h(e_y, -e_x)^T + v(e_x, e_y)^T}{e_x^2 + e_y^2} \mid \right. \\ \left. h \in [-h_{max}^i, h_{max}^i], v \in [-v_{max}^i, v_{max}^i] \right\}, \quad (4)$$

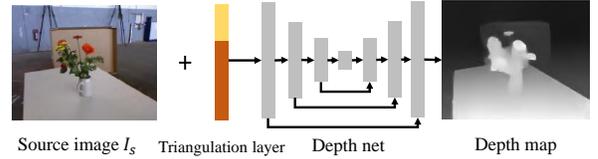
where h_{max}^i denotes the search range along the epipolar lines and v_{max}^i is the search range vertical to the lines. In total, $(2h_{max}^i + 1)(2v_{max}^i + 1)$ pixels are matched for each pixel at the i -th level.

Figure 3 illustrates the difference between the cost volume computation in PWC-Net and the proposed flow-motion net. With the static scene prior and the estimated motion, the estimated optical flow can be regularized, and the size of the cost volume is reduced, leading to efficient estimation.

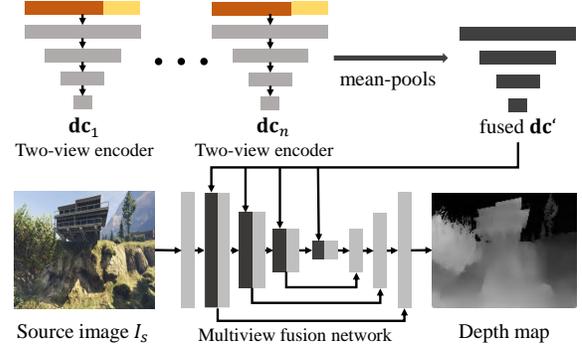
B. Depth Network

Given the estimated optical flow \mathbf{x} and camera pose R, \mathbf{t} , the pixel depth d can be easily triangulated by solving,

$$\mathbf{w}(\mathbf{x}) + \mathbf{x} = \lambda(KRK^{-1}[\mathbf{x}, 1]^T \cdot d + K\mathbf{t}), \quad (5)$$



(a) Depth net to solve two-view problems



(b) Extended depth net to fuse multiview depth information

Fig. 5. Extending the depth net to fuse multiple depth information. (a) Two-view depth estimation network. (b) Multiple depth fusion extension. The *two-view encoder network* encodes the depth information of each image pair into depth codes \mathbf{dc}_i . Multiple codes are pooled into \mathbf{dc}' and the *multiview fusion network* takes \mathbf{dc}' to estimate the depth map.

where $\lambda([x, y, z]^T) = [x/z, y/z]^T$ is the dehomogenization function. However, two drawbacks exist in this triangulation step. First, the depth is solved independently for each pixel, thus the overall smoothness and scene priors are ignored. Second, pixels around the epipolar (the projection of the target frame's optical center on the source image) cannot be triangulated reliably. Figure 4 illustrates the potential numerical issues in different camera motions.

To solve the above problems, DeMoN uses networks to refine the triangulated depth maps (with affected depth set to 0). Here, instead of refining the triangulated depth maps, we propose an eight-channel layer that encodes all the information for triangulation. The layer is called triangulation layer \mathbf{tri} , and for each pixel \mathbf{x} ,

$$\mathbf{tri}(\mathbf{x}) = [\mathbf{w}(\mathbf{x}) + \mathbf{x}, KRK^{-1}[\mathbf{x}, 1]^T, K\mathbf{t}]^T. \quad (6)$$

The depth network is an encoder-decoder network that takes the triangulation layer \mathbf{tri} , source image I_s , estimated optical flow \mathbf{w} and the last layer of the flow-motion network as input to estimate the depth map of the source image.

C. Multiview Depth Fusion

In real-world applications (e.g. robot navigation), the depth of the source image can be solved by multiple target images. Here, we extend the proposed two-view monocular stereo networks to fuse multiview information. Compared with two-view image pairs, multiview images bring more information about the environment structure, thus the fused depth maps can be more robust and accurate. However, fusing depth information from multiview images is non-trivial due to the arbitrary number of image pairs and different depth scales. Different from CodeSLAM, which fuses the information by

optimization methods, we propose to fuse the multiview information by a learned network.

Figure 5 shows how the two-view depth net is extended. The two-view depth net introduced in Sec. III-B is divided into two parts: two-view encoder and multiview fusion. The first part independently encodes the triangulation layer tri of each image pair into multi-resolution depth codes dc . Depth codes from multiple image pairs, $\{\text{dc}_0, \dots, \text{dc}_{N-1}\}$, are fused by mean-pooling layers. The fused code of each pixel $\text{dc}'(\mathbf{x})$ is calculated as,

$$\text{dc}'(\mathbf{x}) = \frac{1}{N} \sum_{i=0}^{N-1} \text{dc}_i(\mathbf{x}). \quad (7)$$

Using pooling layers to fuse information has been used in many multiview stereo works (e.g., DeepMVS [10]). Different from these works, we use multiple pooling layers to fuse the depth codes at different resolutions such that both the global information and fine details are preserved. The fusion network takes the fused depth code dc' and the source image I_s to estimate the corresponding depth map.

IV. NETWORK DETAILS

A. Optical Flow and Camera Motion

The search space of the cost volume calculation is reduced gradually from coarse to fine. The flow-motion network estimates the optical flow from level 5 to level 1. From the 5-th level to the 1-st level, the search steps h_{max} and v_{max} are set to $\{4, 4, 4, 4, 3\}$ and $\{4, 4, 4, 2, 1\}$, respectively. In the 1-st level, only 21 pixels are matched (81 pixels are used in PWC-Net). The optical flow loss is defined as,

$$L_{flow} = \sum_{l=1}^5 \sum_{\mathbf{x}} \|\mathbf{w}^l(\mathbf{x}) - \hat{\mathbf{w}}^l(\mathbf{x})\|_2, \quad (8)$$

where $\hat{\mathbf{w}}^l$ is the corresponding ground truth optical flow at the l -th level.

The camera rotation \mathbf{r} is parameterized as the three-dimension rotation vector: $\mathbf{r} = \theta \mathbf{v}$, where θ is the rotation angle and \mathbf{v} is the rotation axis. Similar to DeMoN, camera translation \mathbf{t} is normalized as a unit vector due to the unobservable scale. Since the optical flow on coarse resolutions cannot provide accurate pixel correspondences, the camera motion is estimated from level 3 to level 1. With the ground truth camera motion $\hat{\mathbf{r}}$ and $\hat{\mathbf{t}}$, the motion loss is,

$$L_{motion} = \sum_{l=1}^3 \|\mathbf{r}^l - \hat{\mathbf{r}}\|_2 + \sum_{l=1}^3 \|\mathbf{t}^l - \hat{\mathbf{t}}\|_2. \quad (9)$$

B. Depth Estimation

Multiple depth maps are estimated by the depth network at different resolutions (from level 3 to level 1). We adopt the depth parameterization from Eigen et al. [21] that the output of the network is the log depth: $\log(d) \in \mathcal{R}$. Due to the scale ambiguity in SfM problems, the scale-invariant depth error for each pixel \mathbf{x} is calculated as,

$$d_e^l(\mathbf{x}) = \log(d^l(\mathbf{x})) + \alpha^l - \log(\hat{d}^l(\mathbf{x})) \quad (10)$$

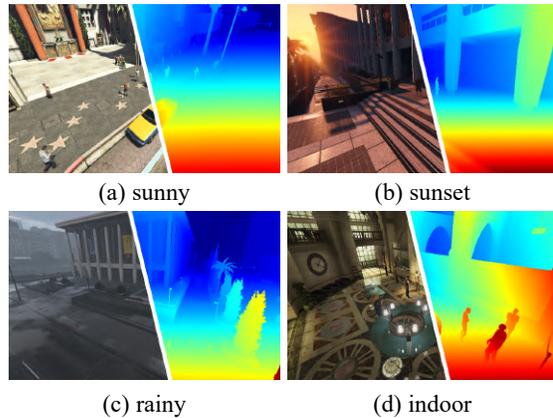


Fig. 6. Samples from the GTA-SfM dataset including different weather, time, and scenes. The flexibility to change the environment and camera settings improves the usability of the dataset in deep learning research.

where \hat{d} is the ground truth depth map, and $\alpha^l = \frac{1}{N} \sum_{\mathbf{x}} \log(\hat{d}^l(\mathbf{x})) - \log(d^l(\mathbf{x}))$ scales the estimated depth maps. Both the depth error L_d and gradient error L_g are calculated to train the triangulation network,

$$L_d = \sum_{l=1}^3 \sum_{\mathbf{x}} \|d_e^l(\mathbf{x})\|_{berHu}, \quad (11)$$

$$L_g = \sum_{l=1}^3 \sum_{\mathbf{x}} |\nabla_x d_e^l(\mathbf{x})| + |\nabla_y d_e^l(\mathbf{x})|, \quad (12)$$

where $\|\cdot\|_{berHu}$ is the reverse Huber [22], [23]:

$$\|x\|_{berHu} = \begin{cases} |x| & \text{if } |x| \leq 1 \\ x^2 & \text{if } |x| > 1 \end{cases}. \quad (13)$$

Using the berHu norm, large depth errors are punished by the L2 norm and small depth errors can also be effectively optimized by the L1 norm.

V. DATASETS

A. DeMoN Dataset

DeMoN proposes a collection of datasets to train and evaluate deep networks. The dataset contains images from multiple sources, such as RGB-D cameras [7], [24], multiview SfM results [1], [2], [25], [26], and synthetic images [3]. In total, the DeMoN dataset contains 57k image pairs for training and 354 pairs for testing.

Although the DeMoN dataset has been widely used in previous works [3]–[5], it contains several limitations. First, depth maps from RGB-D cameras are not synchronized with the color images and only provides less than 10 meters depth measurements. Second, most of the camera poses of the real-world images are calculated by optimization-based methods which can be affected by image noises or outlier features. Lastly, the rendered synthetic images in the dataset are not photorealistic. All these aspects limit the performance of the trained networks.

B. GTA-SfM Dataset

To overcome the limitations in the DeMoN dataset, we propose the GTA-SfM dataset as a supplement. The dataset is rendered from GTA-V, an open-world game with large-scale city models. Thanks to the active community, we develop

tools to extract unlimited photorealistic images with depth maps and camera poses. The extracted depth maps provide depth measurements for all objects in the images, including fine structures or reflective surfaces. We extracted 71k pairs of images for training and 5k pairs for testing. Training and testing dataset do not share common scenes. Different from the DeMoN dataset, one source image can have multiple target images, thus the multiview depth fusion can be tested.

A similar dataset, MVS-SYNTH, is released by DeepMVS [10] using graphics debugging tools. Compared with MVS-SYNTH, GTA-SfM tools can freely set the camera FOV, weather, and daytime such that the dataset diversity and usability are improved. Also, the camera trajectory is manually annotated that cameras move with large translation and rotation. Figure 6 shows samples from the proposed dataset.

VI. EXPERIMENTS

In this section, we extensively evaluate the performance of the proposed flow-motion network and depth network. We first compare the proposed network with the previous works [3]–[5] on two-view image pairs using the DeMoN dataset. Then, the depth fusion performance is evaluated using the proposed GTA-SfM dataset. The effectiveness of the proposed flow-motion joint estimation and the triangulation layer *tri* is also demonstrated in the ablation study. We further demonstrate the generalization ability of the method using real-world images and Google Earth images.

A. Evaluation Metrics

Different metrics are used to evaluate the estimated camera motion and depth maps. We follow the evaluation method used in DeMoN. The rotation error is defined by the relative angle between the estimated camera rotation and the ground truth rotation. Due to the scale ambiguity in SfM problems, the translation error is defined by the angle between normalized translation vectors. For the depth evaluation, estimated depth d is first optimally scaled [3], then three depth metrics are calculated,

$$\text{L1-inv}(d, \hat{d}) = \frac{1}{N} \sum_{\mathbf{x}} \left| 1/d(\mathbf{x}) - 1/\hat{d}(\mathbf{x}) \right|, \quad (14)$$

$$\text{sc-inv}(d, \hat{d}) = \sqrt{\frac{1}{N} \sum_{\mathbf{x}} z(\mathbf{x})^2 - \frac{1}{N^2} (\sum_{\mathbf{x}} z(\mathbf{x}))^2}, \quad (15)$$

$$\text{L1-rel}(d, \hat{d}) = \frac{1}{N} \sum_{\mathbf{x}} \left| d(\mathbf{x}) - \hat{d}(\mathbf{x}) \right| / \hat{d}(\mathbf{x}), \quad (16)$$

where $z(\mathbf{x}) = \log(d(\mathbf{x})) - \log(\hat{d}(\mathbf{x}))$, and N is the pixel number.

B. Two-view Evaluation

We train the flow-motion network and the depth network using *only* the DeMoN dataset for a fair comparison. Note that DeMoN is trained with a larger dataset including other synthetic images. Images are resized to 320×256 in the experiments. The flow-motion network was trained for 750k steps with the Adam optimizer [27]. With the trained flow-motion network, the depth network is trained for 260k steps. According to the model size, the mini-batch size is set to 16 for the flow-motion network and 24 for the triangulation

TABLE I
COMPARISON ON TWO-VIEW PROBLEMS

	Method	Depth			Motion	
		L1-inv	sc-inv	L1-rel	Rotation	Translation
MVS	SIFT	0.056	0.309	0.361	21.180	60.516
	DeMoN	0.047	0.202	0.305	5.156	14.447
	LS-Net	0.051	0.221	0.311	4.653	11.221
	BA-Net	0.030	0.150	0.080	3.499	11.238
	Ours	0.027	0.177	0.116	3.093	7.207
Scenes11	SIFT	0.051	0.900	1.027	6.179	56.650
	DeMoN	0.019	0.315	0.248	0.809	8.918
	LS-Net	0.010	0.410	0.210	0.910	8.210
	BA-Net	0.080	0.210	0.130	1.298	10.370
	Ours	0.015	0.268	0.179	0.615	5.331
RGB-D	SIFT	0.050	0.577	0.703	12.010	56.021
	DeMoN	0.028	0.130	0.212	2.641	20.585
	LS-Net	0.019	0.090	0.301	1.010	22.100
	BA-Net	0.008	0.087	0.050	2.459	14.900
	Ours	0.010	0.158	0.107	1.570	11.163
SUN3D	SIFT	0.029	0.290	0.286	7.702	41.825
	DeMoN	0.019	0.114	0.172	1.801	18.811
	LS-Net	0.015	0.189	0.650	1.521	14.347
	BA-Net	0.015	0.110	0.060	1.729	13.260
	Ours	0.009	0.105	0.076	1.494	12.049

network. Both learning-based methods (DeMoN, LS-Net, and BA-Net) and a classic method are compared in the experiment. The classic method is proposed and evaluated in DeMoN that solves camera poses by the normalized 8-point algorithm [28] (using SIFT features) followed by a reprojection error minimization. The depth maps are estimated using plane sweep stereo and semi-global matching [29].

Table I and Figure 7 show the results of both depth and motion comparison. Due to the flow-motion joint estimation, the proposed method achieves the best camera motion estimation in most of the cases. The proposed depth network also achieves consistently better performance compared with DeMoN. Compared with BA-Net which iteratively refines the results (95ms in total), our method generates consistently better camera poses, and competitive depth maps without any iterations (42ms in total). As shown in Figure 7, due to the triangulation layer that encodes the geometric information, both near and distant objects are reconstructed accurately.

C. Depth Fusion Evaluation

Since the DeMoN dataset only provides two-view image pairs, we use the proposed GTA-SfM dataset to train and evaluate the multiview depth fusion performance. We first train the flow-motion network using two-view image pairs for 210k steps and then train the extended multiview fusion network for 130k steps.

We first evaluate the quality of estimated depth maps using different numbers of target images. We also compare the depth net with DeepMVS [10] which is also trained using images from GTA5. DeepMVS takes ground truth camera poses as input. For each number of target images, we *randomly* sample 300 pairs and compute the mean depth error.

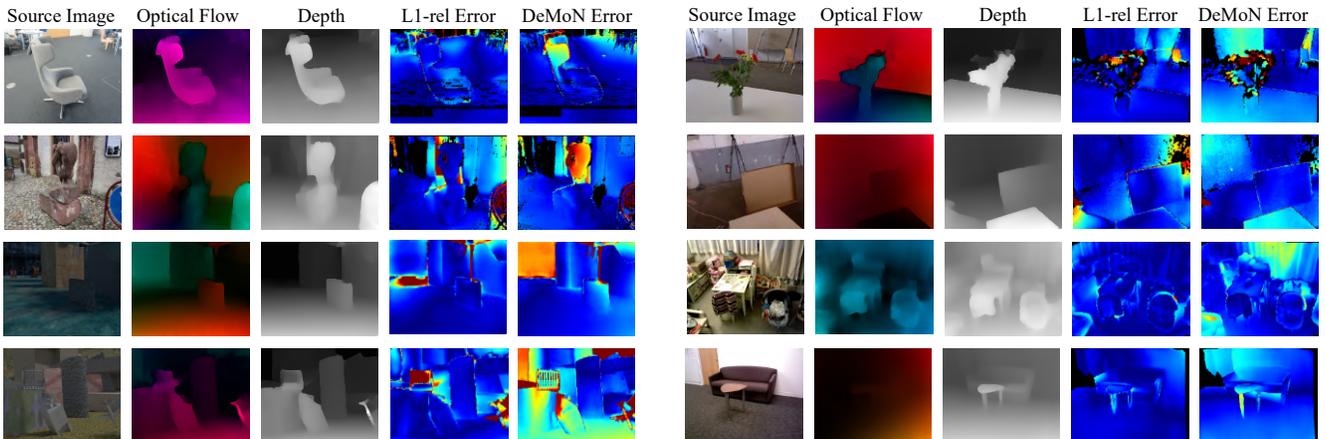


Fig. 7. Qualitative results on the DeMoN database. From left to right: source image, estimated optical flow, estimated depth map, L1-rel error map, and L1-rel error of DeMoN estimated depth map. The error map is JET color coded. As shown, our method generates high-quality optical flow and depth maps. With the proposed triangulation layer, the depth maps have less L1-rel errors.

Table II shows the depth quality given different numbers of target images. Clearly, the depth quality improves when more images are observed, which shows the effectiveness of the multiview fusion and matches the experience from classic SfM methods. We also visualize estimated depth maps for qualitative comparison in Figure 8. Our method estimates smooth and detailed depth maps and DeepMVS estimates discrete depth maps with outliers.

TABLE II
DEPTH MAP ERROR ON GTA-SfM DATASET.

View Num	Depth Error					
	L1-inv (1e-3)		sc-inv		L1-rel	
	Ours	DeepMVS	Ours	DeepMVS	Ours	DeepMVS
2	6.19	16.6	0.213	0.526	0.145	0.766
3	6.07	15.6	0.207	0.496	0.137	0.753
4	5.36	15.1	0.192	0.475	0.124	0.735
5	5.68	14.8	0.192	0.465	0.123	0.723
6	4.86	14.8	0.181	0.464	0.114	0.729

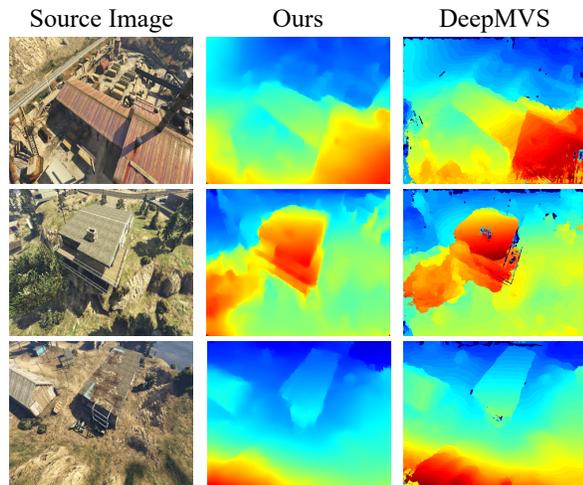


Fig. 8. Quality comparison of generated depth maps by the proposed method and DeepMVS. Each source image is observed by 6 target images, and DeepMVS is provided with ground truth camera poses.

D. Ablation Study

Here, we study the effectiveness of the contributions: the flow-motion joint estimation and the triangulation layer.

Joint Estimation To evaluate the importance of the epipolar line constraint and search space reduce, we remove the camera pose estimation in middle levels and the camera motion is estimated using the final flow estimation. Without the epipolar line constraint, 81 pixels (the same as PWC-Net) are searched at each level. As shown in Table III, the joint estimation improves *both* the optical flow and camera pose estimation.

TABLE III
EFFECTIVENESS OF THE JOINT FLOW-MOTION ESTIMATION

	Rotation Error	Translation Error	Flow Error
original	1.879	10.307	3.472
w/o joint	2.043	11.703	3.567

Triangulation Layer The triangulation layer is proposed to encode the estimated optical flow and camera motion without any numerical instability. To demonstrate the effectiveness, we replace the triangulation layer with a directly triangulated depth map. Similar to DeMoN [3], NaN values are set to 0. Both the networks are trained with the same flow-motion network as the front-end for 50 epochs. The comparison is shown in Table IV. With the proposed **tri**, depth network can better exploit estimated optical flow and camera poses.

TABLE IV
EFFECTIVENESS OF THE TRIANGULATION LAYER.

	L1-inv	sc-inv	L1-rel
original	0.015	0.195	0.134
w/o tri	0.017	0.200	0.140

E. Generalization Ability

To test the generalization ability of the proposed method, we further use the method to estimate depth maps of images from different sources. Figure 9 shows estimated depth maps of images taken with DJI Phantom 4 (outdoor) or a handheld camera (indoor). Figure 10 shows estimated depth maps of images from Google Earth. The depth map of each source image is fused from 5 or 6 target images. Because the proposed method first builds high-quality pixel correspondences



Fig. 9. Generate the proposed method to real-world images.

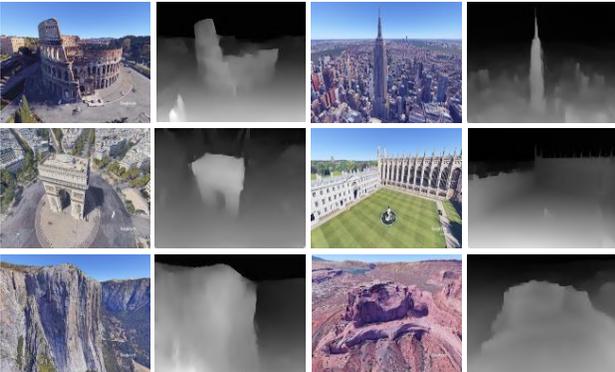


Fig. 10. Generate the proposed method to Google Earth images.

and then triangulate the depth of each pixel, it can effectively utilize multiview observations and generalizes well to other images. More details are in the supplementary material.

VII. CONCLUSION AND FUTURE WORK

In this letter, we propose a flow-motion network and a depth network that can estimate the camera motion and depth map given multiple motion stereo images. Both the networks are designed carefully to exploit the multiview geometric constraints among optical flow, camera motion and depth maps. We further extend the depth network to fuse multiple depth information into a depth map. To enlarge the available datasets, an open-source tool is proposed to extract unlimited photorealistic images with ground truth camera poses and depth maps. In the future, we plan to further develop the method by incorporating graph networks so that it can simultaneously estimate all camera poses and depth maps given a set of images.

REFERENCES

- [1] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [2] J. Schonberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [3] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [4] R. Clark, M. Bloesch, J. Czarowski, S. Leutenegger, and A. J. Davison. Learning to solve nonlinear least squares for monocular stereo. In *European Conference on Computer Vision (ECCV)*, 2018.

- [5] C. Tang and P. Tan. BA-Net: Dense bundle adjustment network. In *The International Conference on Learning Representations (ICLR)*, 2019.
- [6] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.
- [7] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using sfm and object labels. In *European Conference on Computer Vision (ECCV)*, 2013.
- [8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, and V. Golkov. FlowNet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [9] M. Bloesch, J. Czarowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM learning a compact, optimisable representation for dense visual SLAM. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [10] P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang. DeepMVS: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [11] K. Wang and S. Shen. MVDepthNet: real-time multiview depth estimation neural network. In *International Conference on 3D Vision (3DV)*, 2018.
- [12] S. Im, H. Jeon, S. Lin, and I. S. Kweon. DPSNet: End-to-end deep plane sweep stereo. In *The International Conference on Learning Representations (ICLR)*, 2019.
- [13] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018.
- [14] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [15] H. Zhou, B. Ummenhofer, and T. Brox. DeepTAM: Deep tracking and mapping. In *European Conference on Computer Vision (ECCV)*, 2018.
- [16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2004.
- [17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [18] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [19] D. Sun, X. Yang, M. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [20] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [21] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems (NIPS)*, 2014.
- [22] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision (3DV)*, 2016.
- [23] L. Zwald and S. Lambert-Lacroix. The BerHu penalty and the grouped effect. *arXiv preprint arXiv:1207.6868*, 2012.
- [24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2012.
- [25] S. Fuhrmann, F. Langguth, and M. Goesele. MVE a multi-view reconstruction environment. In *The Eurographics Workshop on Graphics and Cultural Heritage (GCH)*, 2014.
- [26] B. Ummenhofer and T. Brox. Global, dense multiscale reconstruction for a billion points. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [27] D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2015.
- [28] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:580593, 1997.
- [29] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.