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 realistic 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.**

***Index Terms*—SLAM, visual learning, aerial systems: perception and autonomy.**

1. INTRODUCTION

**D**

UE 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. Tradi- tional 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

Manuscript received September 10, 2019; accepted February 2, 2020. Date of

publication February 21, 2020; date of current version March 9, 2020. This letter was recommended for publication by Associate Editor L. Paull and Editor S. Behnke upon evaluation of the reviewers’ comments. This work was supported by the Hong Kong PhD Fellowship Scheme. *(Corresponding author: Kaixuan Wang.)*

Kaixuan Wang and Shaojie Shen are with the ECE, HKUST, Hong Kong, China (e-mail: [kwangap@ust.hk;](mailto:kwangap@ust.hk) [eeshaojie@ust.hk).](mailto:eeshaojie@ust.hk)

1https://github.com/HKUST-Aerial-Robotics/Flow-Motion-Depth. Digital Object Identifier 10.1109/LRA.2020.2975750

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 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 com- pared 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 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 move- ments. 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 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. Fig. 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-realistic synthetic images. To train and evaluate our pro- posed networks, we develop tools that can generate unlimited high-quality realistic images with ground truth depth maps and

2377-3766 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See h[ttps://www.ieee.org/publications/rights/inde](http://www.ieee.org/publications/rights/index.html)x.html for more information.

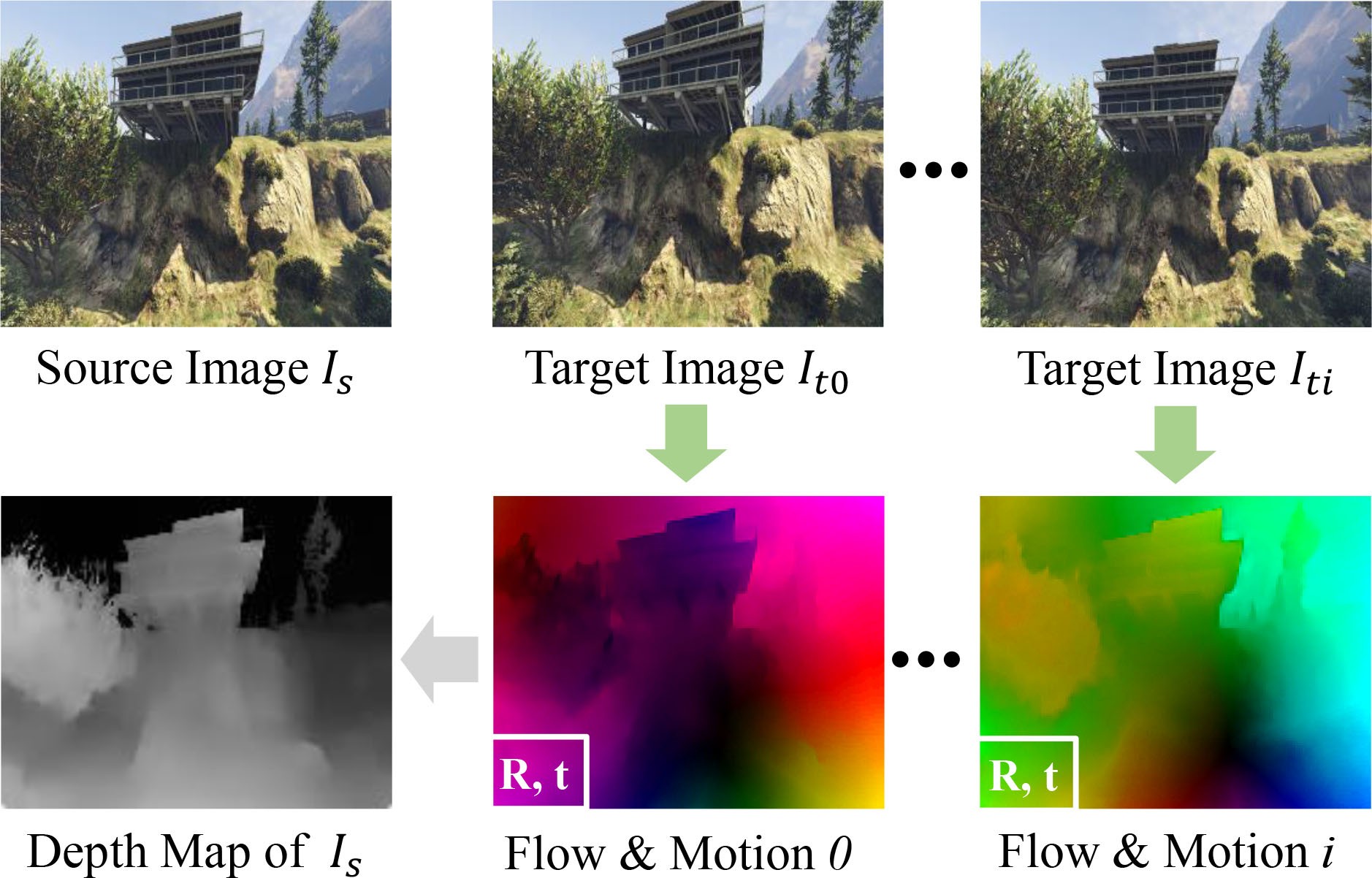


Fig. 1. Illustration of the proposed method. Given multiple images from a moving monocular camera, the flow-motion network (green in the figure) first estimates the optical flow and camera pose between the source image and each target image. The estimated flow and motion are further fused by the depth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

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 infor-

mation (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 realistic syn-

thetic images with different daytime, intrinsic parameters, etc. The extracted images serve as a supplementary dataset to train and evaluate learning-based SfM methods.

1. 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 fur- ther 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 meth- ods. 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 differen- tiable 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 learning-based approaches have been proposed to solve multiview stereo with known extrinsic and intrinsic param- eters. 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 struc- ture of networks. The cost volume calculation that explicitly measures the similarity between pixels generate high-quality depth maps. In the proposed method, we adopt the similar strategy that uses estimated camera poses for accurate 1D pixel correspondences.

In this letter, 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.

1. NETWORK ARCHITECTURE

As shown in Fig. 2, the proposed method consists of two networks: one flow-motion network and one depth network. Given a source image *Is* and a target image *It* 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 epipo- lar 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 [15]. 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 *Is*. The depth network can also be extended to fuse the information from multiple target images. When the source image is observed by multiple 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 processes 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.

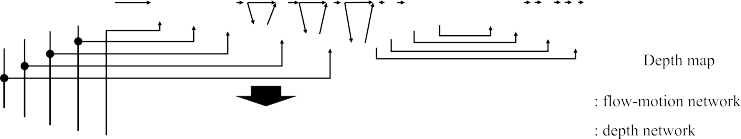
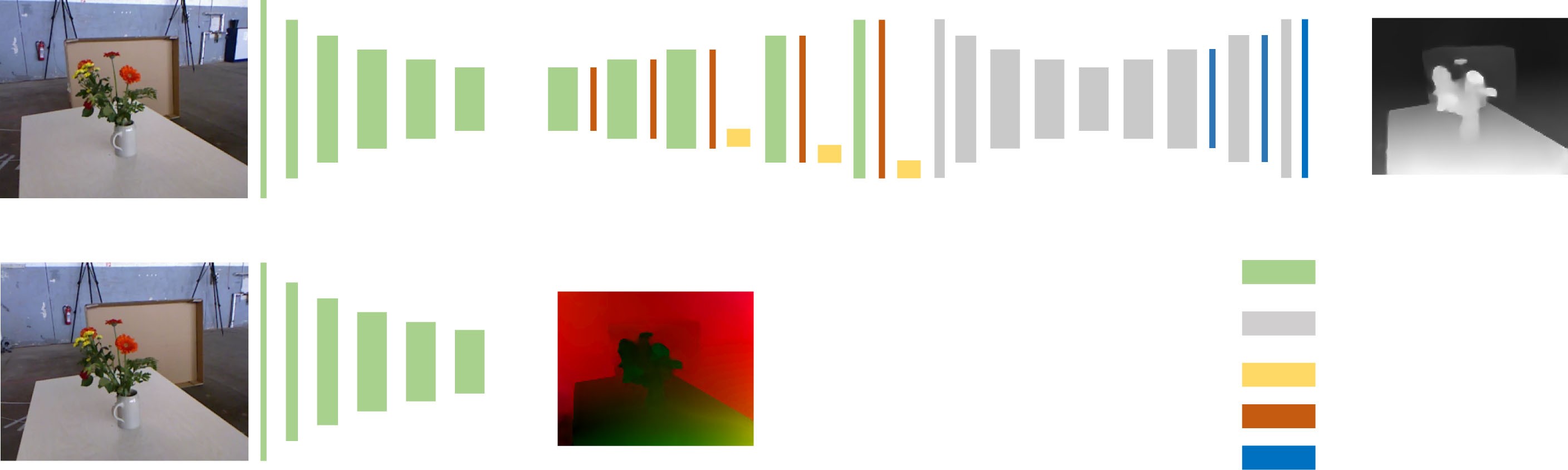


Fig. 2. The architecture of the proposed flow-motion network and depth network. Here, only the two-view architecture is shown for simplicity. The extension to fuse multiple depth information is shown in Fig. 5. The flow-motion network jointly estimates the optical flow and camera poses, and the depth 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.

1. *Flow-Motion Network*

A number of works [8], [16]–[18] 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 [18] but is tailored for static scenes

relationship,

**x** + **w**(**x**) *T*

1

*F* **x**

= 0*,* (2)

1

and jointly estimates camera poses.

To be robust to lighting and viewing angle changes, input im- ages are converted into L-level feature pyramids using a simple CNN. The feature map at the *i*-th level, **f** *i*, is processed by three

*x*

*z*

where *F* = *K*−*T* **t** *RK*−1 is the fundamental matrix. With the

estimated camera pose, the upsampled optical flow of each pixel

×

**w***i*+1(**x**) can be regularized by projecting the corresponding point to the epipolar line,

*up*

simple convolutional layers to generate the next level feature *x*×*e*2 − *y*×*e e*

*y*

*x*

*y*

— *e e*

pyramid levels are used, with **f** 0 being the original 3-channel

map **f** *i*+1 with the size downsampled by 2. In this work, *L* = 6

*up,r*

*e*2 + *e*2

*y*×*e*2 − *x*×*e e*

— *e e*

image. **f** *i* and **f** *i* are used to denote the *i*-th level feature maps

**w***i*+1 (**x**) = 1

— **x***,* (3)

*x y x*

*x y y z*

*s t*

of *Is* and *It*, respectively.

The optical flow **w** is estimated from coarse to fine to handle large pixel displacement. At the *i*-th level, the optical flow **w***i*+1 from the *i+*1-th level is firstly bilinear upsampled into **w***i*+1 as

*up*

an initialization of **w***i*. A cost volume **c***i* is constructed using **f** *i* and **f** *i*. Each element in the cost volume represents the feature similarity between a pixel **x***s* in **f** *i* and a pixel **x***t* in **f** *i*,

*s*

*t*

*s t*

**c***i*(**x** *,* **x** ) = 1 *i T i*

where [*ex, ey, ez*]*T* = *F* [**x***,* 1]*T* and [*x,*× *y*×]*T* = **x** + **w***i*+1(**x**).

Since the corresponding pixels are constrained on epipolar

*up*

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

*s t* (**f***s*(**x***s*))

*N*

*i*

**f***t* (**x***t*)*,* (1)

levels to fine levels. In the *i*-th level, the matching pixels of

and *Ni* is the feature dimension of **f** *i*. Due to the coarse-to-fine manner, only a subset of pixels in **f** *i* is needed to calculate the cost volume. The cost volume **c** , upsampled optical flow **w***up* , and

*s*

*i*

*t*

*i*+1

*y*

*s*

pixel **x***s* is parameterized as

**x***t* ∈ **x***s* + **w***up,r* (**x***s*)+

*i*+1

*h*(*e ,* −*e*

)*T* + *v*(*e ,e* )*T*

*x*

*x*

*y*

.

*e*2 + *e*2 .

**f** *i* are used to predict the optical flow **w***i* using the DenseNet [19]

structure.

The above cost volume construction and optical flow estima- tion are repeated from coarse to fine until the optical flow of

max

max

*h* ∈ [−*hi*

*, hi*

]*,v* ∈ [−*vi*

*x*

*, vi*

max

max

*y*

] *,* (4)

the desired resolution is estimated. In this work, we adapt and

where *hi*

denotes the search range along the epipolar lines and

improve the above processes by incorporating the static scene *vi*

max

*i*

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 **t**. Although simple CNNs generates accurate pose estimations (shown in experiments), more advanced networks (e.g., DFE [20]) can also be adopted. 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 **x** in the source image and its optical flow vector **w**(**x**) to the target image have the following

max is the search range vertical to the lines. In total, (2*h*max + 1)(2*vi* + 1) pixels are matched for each pixel at the *i*-th level.

Fig. 3 illustrates the difference between the cost volume com- putation 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.

max

1. *Depth Network*

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

**w**(**x**)+ **x** = *λ*(*KRK*−1[**x***,* 1]*T* · *d* + *K***t**)*,* (5)

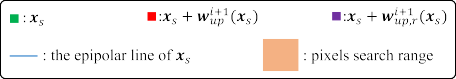
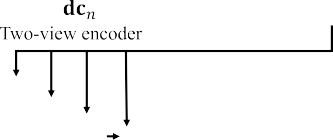
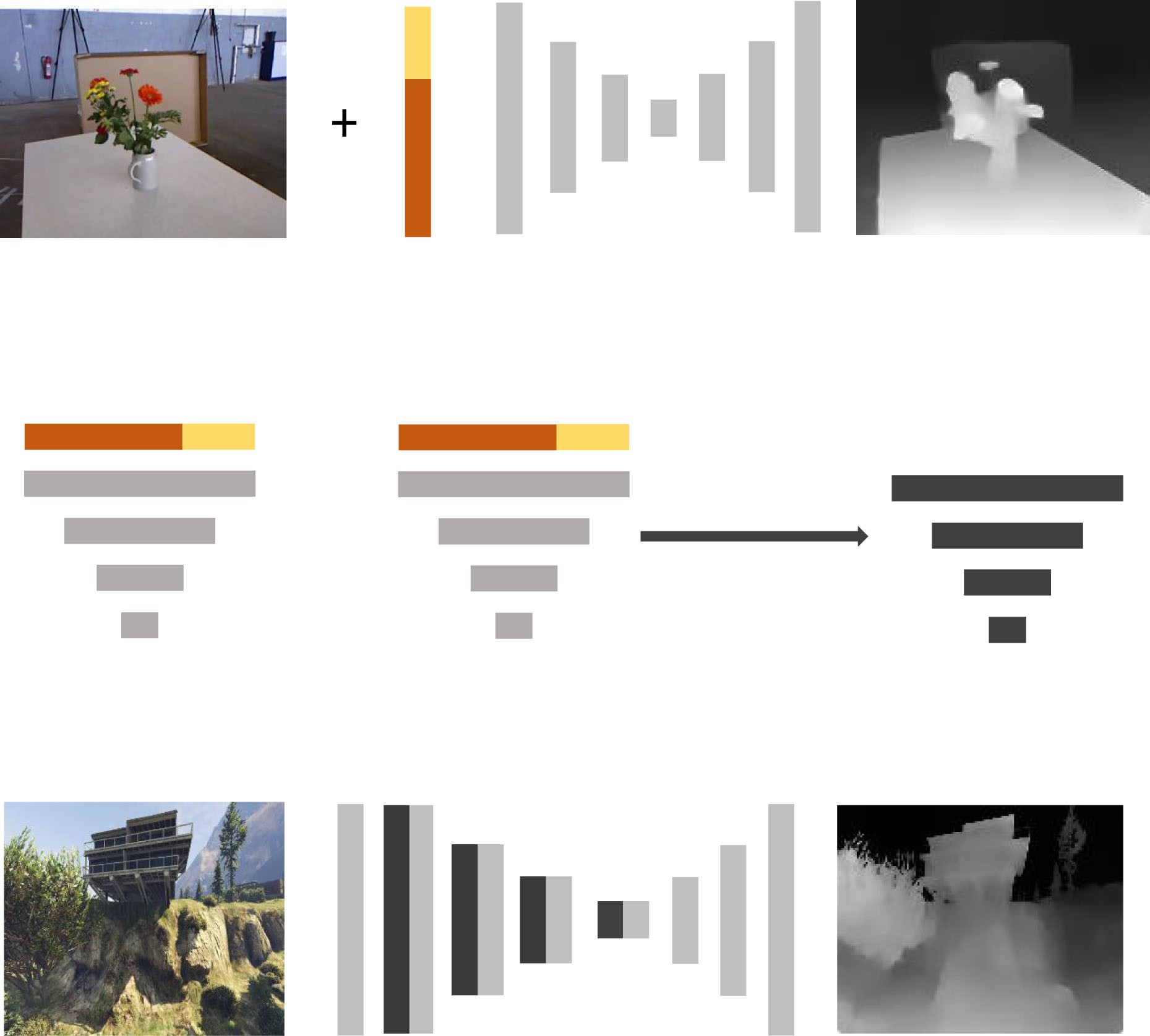




Fig. 3. Differences between the cost volume computation in PWC-Net (left) and the proposed flow-motion network (right). For each pixel **x***s* in **f** *i*, PWC-Net matches a fixed set of pixels (colored in orange) around **x***s* + **w***i*+1(**x***s*) to generate the cost volume. On the other hand, the proposed flow-motion network



*s*

*up*

first regularizes the initial flow **w***i*+1(**x***s*) into **w***i*+1 (**x***s*) and matches pixels

*up*

around the epipolar line.

*up,r*



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 **dc***i*. Multiple codes are pooled into **dc**× and the *multiview fusion network* takes **dc**× to estimate the depth map.

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

Fig. 4. Example to show the numerical stability in triangulation. *Os* and *Ot* are the optical centers of *Is* and *It*, respectively. *d*max and *dmin* are the maximum and minimum depth of the scene. **e***i* is the corresponding epipolar line of pixel **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 of **x**1 (*the epipolar point*) degenerates into a point, thus the depth is unobservable. For pixels near the epipolar point, such as **x**2, the epipolar line **e**2 is very short, and the result is noise-prone.

where *λ*([*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 reli- ably. Fig. 4 illustrates the potential numerical issues in different camera motions.

To solve the above problems, DeMoN uses networks to refine

1. *Multiview Depth Fusion*

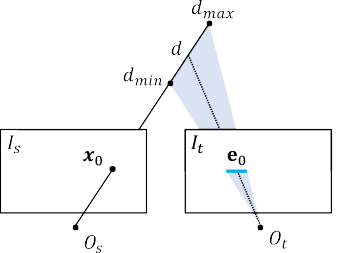
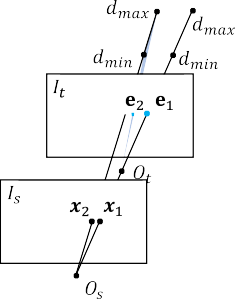
In real-world applications (e.g. robot navigation), the depth of the source image can be solved by multiple target images. Here, we extend the proposed two-view monocular stereo net- works to fuse multiview information. Compared with two-view image pairs, multiview images bring more information about 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.

Fig. 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, {**dc**0, ..., **dc***N* 1}, are fused by mean- pooling layers. The fused code of each pixel **dc** (**x**) is calculated as,

— ×

*N* −1

Σ

the triangulated depth maps (with affected depth set to 0). Here, instead of refining the triangulated depth maps, we propose an

**dc**×(**x**) = 1

*N*

**dc***i*(**x**)*.* (7)

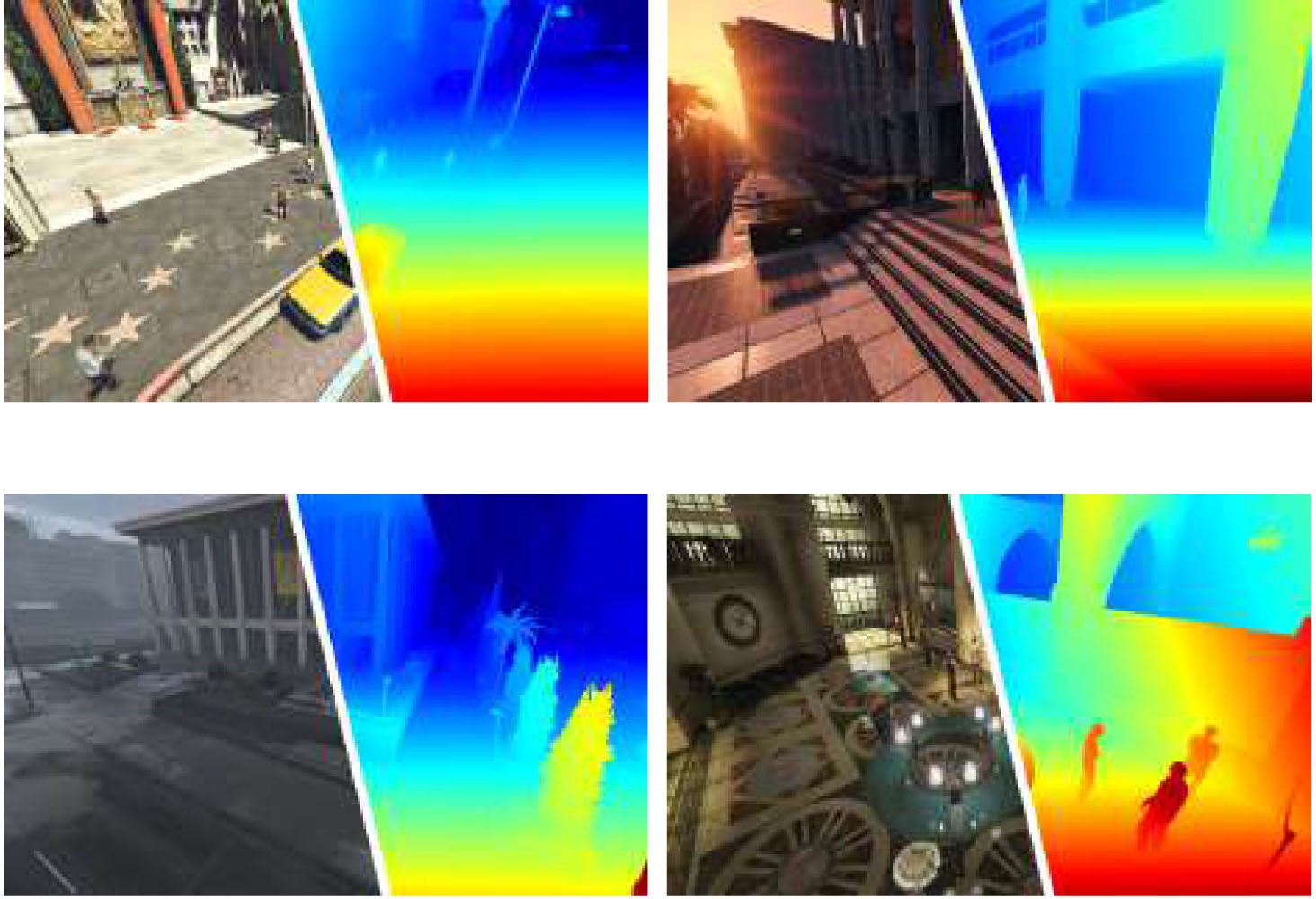
*i*=0

eight-channel layer that encodes all the information for triangu- lation. The layer is called triangulation layer **tri**, and for each pixel **x**,

**tri**(**x**) = [**w**(**x**)+ **x***,KRK*−1[**x***,* 1]*T ,K***t**]*T .* (6)

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 *Is* to estimate the corresponding depth map.



1. 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 esti- mates 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,

{ } { }

*Lflow* = Σ Σ ǁ**w***l*(**x**) − **w**ˆ *l*(**x**)ǁ2*,* (8) where **w**ˆ *l* is the corresponding ground truth optical flow at the

5

*l*=1

x

*l*-th level.

The camera rotation **r** is parameterized as the three-dimension rotation vector: **r** = *θ***v**, where *θ* is the rotation angle and **v** is the rotation axis. Similar to DeMoN, camera translation **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 ˆ**r** and ˆ**t**, the motion loss is,



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.

1. DATASETS
2. *DeMoN Dataset*

DeMoN proposes a collection of datasets to train and evalu- ate 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 57 k image pairs for training and 354 pairs for testing.

Although the DeMoN dataset has been widely used in pre- vious works [3]–[5], it contains several limitations. First, depth

3

Σ

*Lmotion* = ǁ**r***l* − ˆ**r**ǁ2

*l*=1

1. *Depth Estimation*

3

+ ǁ**t***l* − ˆ**t**ǁ2*.* (9)

Σ

*l*=1

maps from RGB-D cameras are not synchronized with the color images and only provides less than 10 meters depth measure- ments. 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

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*) *R*. Due to the scale ambiguity in SfM problems, the scale-invariant depth error for each pixel **x** is calculated as,

∈

*dl* (**x**) = log(*dl*)(**x**)+ *αl* − log(*d*ˆ*l*)(**x**) (10) where *d*ˆ is the ground truth depth map, and *αl* =

*e*

Σ log(*d* )(**x**) − log(*d* )(**x**) scales the estimated depth

1 ˆ*l l N* x

maps. Besides the l1 norm of the depth error *Ld*, we also follow

DeMoN to penalizes high frequence errors using the gradient error *Lg*,

3

rendered synthetic images in the dataset are not realistic. All these aspects limit the performance of the trained networks.

*B. GTA-SfM Dataset*

To overcome the limitations in the DeMoN dataset, we pro- pose 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 realistic 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 71 k pairs of images for training and 5 k pairs for testing. Training and testing datasets do not share common scenes. Different from the DeMoN dataset, one source

*Ld* = Σ Σ ǁ*dl* (**x**)ǁ

*e*

*berHu*

*,* (11)

image can have multiple target images, thus the multiview depth fusion can be tested.

*l*=1 x

*Lg* = Σ Σ .∇*xdl* (**x**). + .∇*ydl* (**x**). *,* (12)

*e*

*e*

3

*l*=1

x

A similar dataset, MVS-SYNTH, is released by Deep-

MVS [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

where ǁ· ǁ*berHu* is the reverse Huber [22], [23]:

are improved. Also, the camera trajectory is manually annotated that cameras move with large translation and rotation. Fig. 6

ǁ*x*ǁ*berHu*

= |*x*| if |*x*|≤ 1

*x*2 if |*x*| *>* 1*.*

(13)

shows samples from the proposed dataset.

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.

1. EXPERIMENTS

In this section, we extensively evaluate the performance of the proposed flow-motion network and depth network. We first

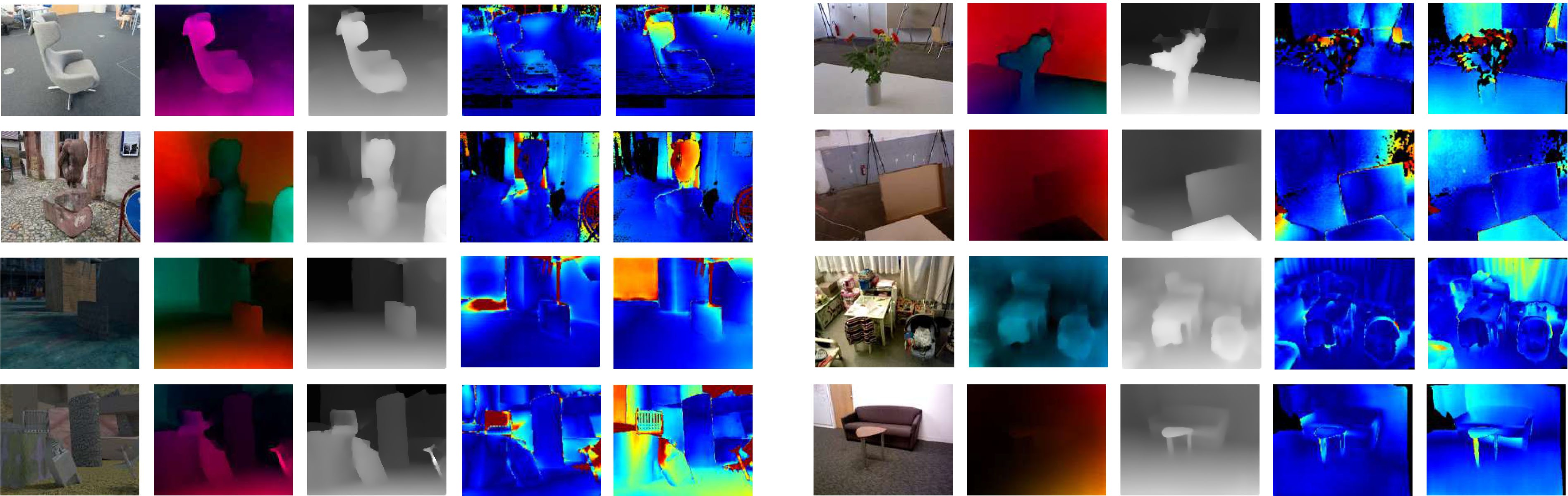


Fig. 7. Qualitative results on the DeMoN database. From left to right: source image, estimated optical flow, estimated depth map, L1-rel error map, and L1-rel error of DeMoN estimated depth map. The error map is JET color coded. As shown, our method generates high-quality optical flow and depth maps.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

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.

1. *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,

L1-inv(*d, d*ˆ) = 1 Σ .1*/d*(**x**) − 1*/d*ˆ(**x**). *,* (14)

*N*

TABLE I

COMPARISON ON TWO-VIEW PROBLEMS

x



sc-inv(*d, d*ˆ) = 1 Σ *z*(**x**)2 − 1 Σ *z*(**x**) 2*,* (15)

*N*

*N* 2

x

x

L1-rel(*d, d*ˆ) = 1 Σ .*d*(**x**) − *d*ˆ(**x**). ,*d*ˆ(**x**)*,* (16)

*N*

x

where *z*(**x**) = log(*d*(**x**)) log(*d*ˆ(**x**)), and *N* is the pixel number.

−

1. *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 750 k steps with the Adam optimizer [27]. With the trained flow-motion network, the depth network is trained for 260 k steps. According to the model size, the mini-batch size is set to 16 for the flow-motion network and 24 for the triangulation network. Both learning-based methods

×

(DeMoN, LS-Net, and BA-Net) and a classic method are com- pared 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 Fig. 7 show the results of both depth and mo- tion 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. Com- pared with BA-Net which iteratively refines the results (95 ms in total), our method generates consistently better camera poses, and competitive depth maps without any iterations (47 ms in

TABLE II

DEPTH MAP ERROR ON GTA-SFM DATASET

TABLE III

EFFECTIVENESS OF THE JOINT FLOW-MOTION ESTIMATION



TABLE IV



EFFECTIVENESS OF THE TRIANGULATION LAYER

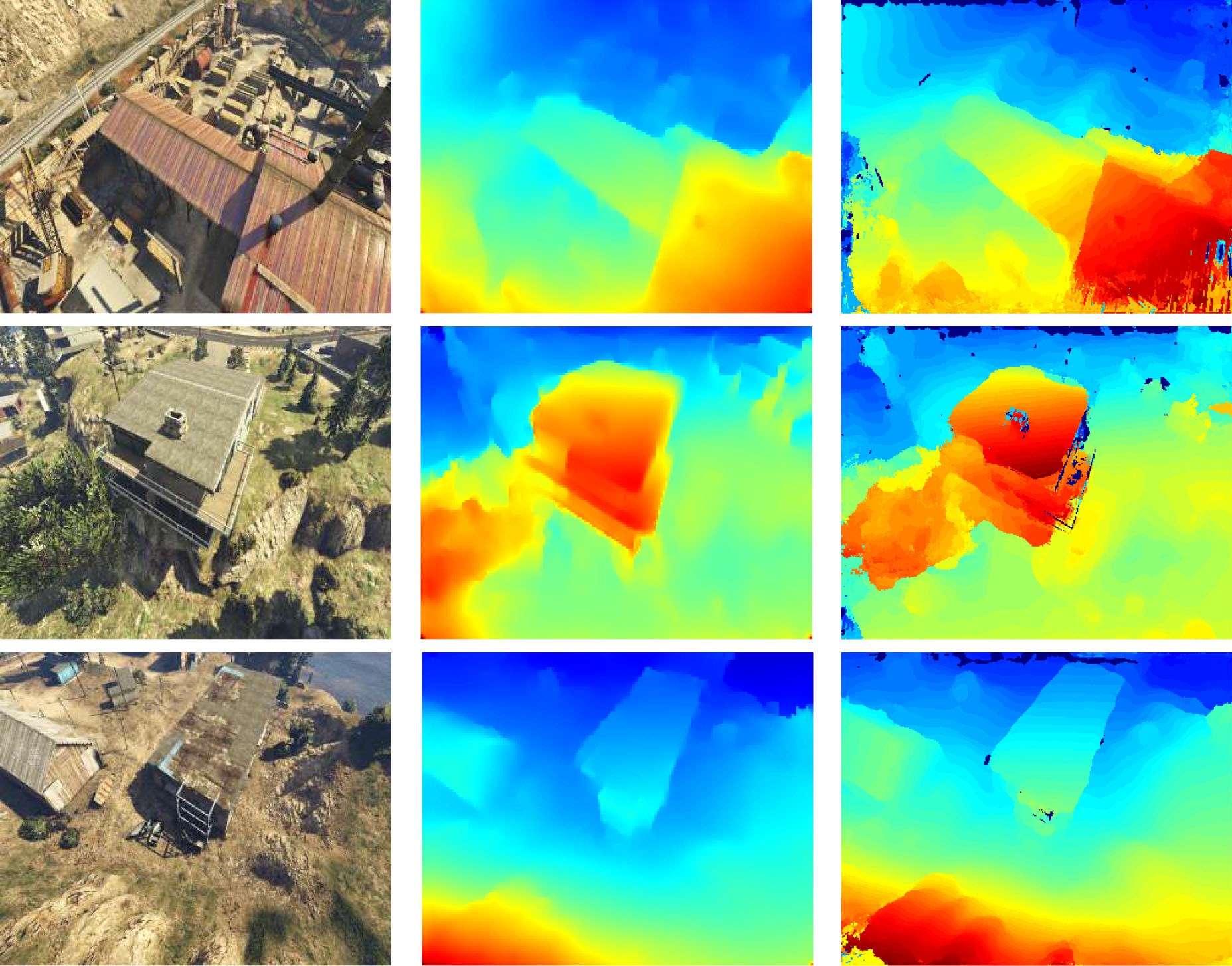


Fig. 8. Quality comparison of generated depth maps by the proposed method and DeepMVS. Each source image is observed by 6 target images, and Deep- MVS is provided with ground truth camera poses.

total). As shown in Fig. 7, due to the triangulation layer that encodes the geometric information, both near and distant objects are reconstructed accurately.

1. *Depth Fusion Evaluation*

Since the DeMoN dataset only provides two-view image pairs, we use the proposed GTA-SfM dataset to train and eval- uate the multiview depth fusion performance. We first train the flow-motion network using two-view image pairs for 210 k steps and then train the extended multiview fusion network for 130 k steps. The code sizes for depth fusion is set to 128, 128, 128, 64, and 64 from coarse to fine, respectively.

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 images with ground truth camera poses as input and our method use images only. For each number of target images, we *randomly* sample 300 pairs and compute the mean depth error. 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 clas- sic SfM methods. We also visualize estimated depth maps for qualitative comparison in Fig. 8. Our method estimates smooth and detailed depth maps and DeepMVS estimates discrete depth maps with outliers.

1. *Ablation Study*

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

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

**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.

**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. We also test the performance of the depth estimation using optical flow and camera pose without the joint optimization. The comparison is shown in Table IV. In the table, we also evaluate the performance of direct triangulated depth from optical flow in the first row. With the proposed **tri**, depth network can better exploit estimated optical flow and camera poses.

1. *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. Fig. 9 shows estimated depth maps of images taken with DJI Phantom 4 (outdoor) or a handheld camera (indoor). Fig. 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 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.



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

1. 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 realistic images with ground truth camera poses and depth maps.

REFERENCES

* 1. J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in

*Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4104–4113.

* 1. J. Schonberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 501–518.
  2. B. Ummenhofer *et al.*, “DeMoN: Depth and motion network for learning monocular stereo,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 5622–5631.
  3. R. Clark, M. Bloesch, J. Czarnowski, S. Leutenegger, and A. J. Davison, “Learning to solve nonlinear least squares for monocular stereo,” in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 284–299.
  4. C. Tang and P. Tan, “BA-Net: Dense bundle adjustment network,” in *Proc. Int. Conf. Learn. Representations*, 2019.
  5. J. Nocedal and S. J. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 2006.
  6. J. Xiao, A. Owens, and A. Torralba, “SUN3D: A database of big spaces reconstructed using sfm and object labels,” in *Proc. IEEE Eur. Conf. Comput. Vision*, 2013, pp. 1625–1632.
  7. A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, and V. Golkov, “Flownet: Learning optical flow with convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 2758–2766.
  8. M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, “CodeSLAM learning a compact, optimisable representation for dense visual SLAM,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2560–2568.
  9. P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang, “DeepMVS: Learning multi-view stereopsis,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2821–2830.
  10. K. Wang and S. Shen, “MVDepthNet: Real-time multiview depth estima- tion neural network,” in *Proc. IEEE Int. Conf. 3D Vision (3DV)*, 2018, pp. 248–257.
  11. S. Im, H. Jeon, S. Lin, and I. S. Kweon, “DPSNet: End-to-end deep plane sweep stereo,” in *Proc. Int. Conf. Learn. Representations*, 2019.
  12. Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, “MVSNet: Depth inference for unstructured multi-view stereo,” in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 767–783.
  13. Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, “Recurrent MVSNet for high-resolution multi-view stereo depth inference,” in *IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 5520–5529.
  14. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, MA, USA: Cambridge Univ. Press, 2004.
  15. E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Net- works,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1647–1655.
  16. A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2720–2729.
  17. D. Sun, X. Yang, M. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 8934–8943.
  18. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2261–2269.
  19. R. Ranftl and V. Koltun, “Deep fundamental matrix estimation,” in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 284–299.
  20. D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
  21. I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *Proc. Int. Conf. 3D Vision*, 2016, pp. 239–248.
  22. L. Zwald and S. Lambert-Lacroix, “The BerHu penalty and the grouped effect,” 2012, *arXiv:1207.6868*.
  23. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A bench- mark for the evaluation of RGB-D SLAM systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.
  24. S. Fuhrmann, F. Langguth, and M. Goesele, “MVE a multi-view recon- struction environment,” in *Proc. Eurographics Workshop Graph. Cultural Heritage*, 2014, pp. 11–18.
  25. B. Ummenhofer and T. Brox, “Global, dense multiscale reconstruction for a billion points,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1341–1349.
  26. D. P. Kingma and J. L. Ba, “ADAM: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015.
  27. R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.
  28. H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.