NETWORK-BASED STRUCTURE FLOW ESTIMATION

Shu Liu[1], Nick Barnes[2], Robert Mahony[2], Haolei Ye[1]
[1]College of Engineering & Computer Science, Australian National University
[2]Research School of Engineering, Australian National University
{shu.liu, nick.barnes, robert.mahony, haolei.ye}@anu.edu.au

***Abstract* — Structure flow is a novel three-dimensional motion representation that differs from scene flow in that it is directly associated with image change. Due to its close connection with both optical flow and divergence in images, it is well suited to estimation from monocular vision. To acquire an accurate measurement of structure flow, we design a method that employs the spatial pyramid structure and the network-based method. We investigate the current motion field datasets and validate the performance of our method by comparing its two-dimensional component of motion field with the previous works. In general, we experimentally show two conclusions: 1. Our motion estimator employs only RGB images and outperforms the previous work that utilizes RGB-D images. 2. The estimated structure flow map is a more effective representation for demonstrating the motion field compared with the widely-accepted scene flow via monocular vision.**

I. INTRODUCTION

Motion estimation is a core computer vision problem. Optical flow, predicting the 2D motion of each pixel across an image sequence, has been extensively studied over many years. It has many applications such as the autonomous control in the fields of robotics and driving [1, 2]. Traditional approaches adopt the approach of optimizing an energy function such as Horn and Schunk [3] or Lukas and Kanade [4]. However, recent work has shown effective performance from the network-based methods [14, 15, 16, 17, 18].

A more recent topic is the estimation of 3D motion field, where both the two objects moving by different distances from A and B to A' and B' respectively. Their 3D motions (scene flow) are denoted by V_A and V_B, while the projected 2D motions in the image plane are denoted by ϕ_A and ϕ_B. Notably, a disadvantage of scene flow is that V_A and V_B are not directly coupled to the optical flow ϕ_A and ϕ_B in the image plane.

Structure flow has been proposed as an alternative representation of the three-dimensional motion field [7]. Following the convention that l denotes the distance between the camera centre $\{C\}$ and the projected object, λ denotes the distance between the projected object and the actual object, structure flow can be expressed as

$$\omega = \frac{l}{l+\lambda} V \tag{1}$$

The motion in z-axis is correlated with looming changes in continuous images, and its rate at which objects expand or contract is proportional to structure flow. It is the same way that optical flow encodes 2D motion field across images. Due to its close dependence on actual image change, structure flow can be estimated with less uncertainty than scene flow with a monocular camera. In Fig.1, we can see that the length of the structure flow of both objects ω_A and ω_B matches the pattern on the image plane, while scene flow does not.

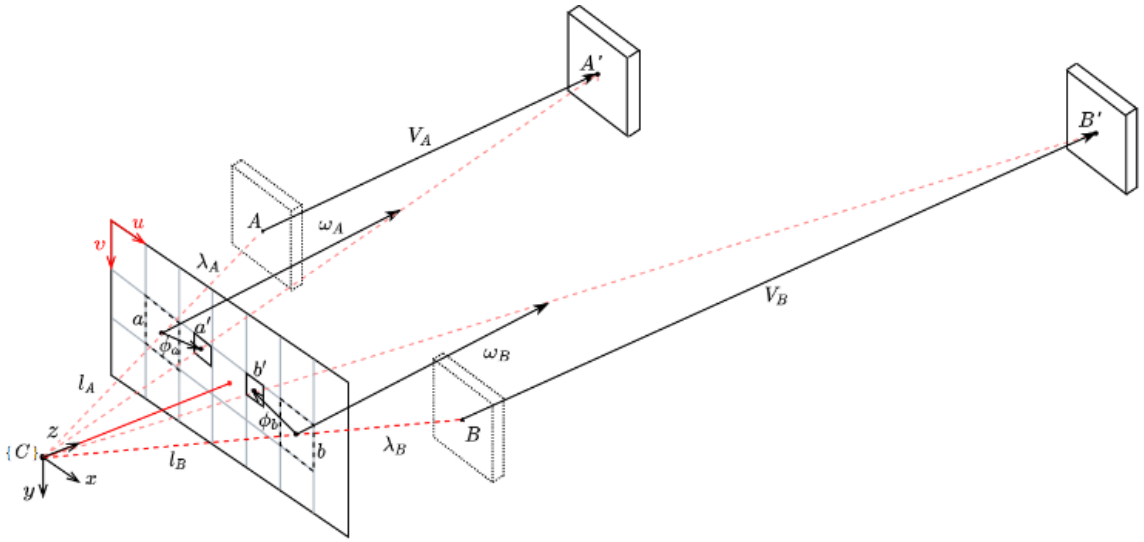


Fig. 1: Given two moving objects A and B, their structure flow ω_A and ω_B are proportional to the the projected two-dimensional optical flow ϕ_A and ϕ_B, while their scene flow V_A and V_B are not.

In this paper, we explore a deep convolutional neural network (DCNN) approach to the estimation of structure flow. We hope that this work can lead to methods that provide robust and accurate estimation of structure flow, showing the improvements that have been done with optical flow. We also compare our novel structure flow technique with state-of-the-art scene flow methods.

The contribution is threefold. By taking monocular images as the input, we propose (1) the first deep convolutional neural network approach for the structure flow estimation. (2) We propose and investigate a suitable approach, and datasets for learning structure flow estimation. (3) We then experimentally show that our approach outperforms current scene flow approaches on the standard benchmark dataset, and introduce mesh plots as an intuitive visualization for three-dimensional motion fields.

II. Related works

A. Representation of visual motion field

Visual motion fields were introduced to describe the apparent motion in a visual environment [8]. Optical flow is widely used in computer vision and has been exploited in areas such as robot control and helped to develop autonomous driving in recent years [1, 2]. Classical approaches such as Horn and Schunk [1] or Lukas and Kanade [2] optimize energy functions to estimate this motion representation.

B. Three-dimensional motion field

Similar to the two-dimensional optical flow, three-dimensional scene flow was defined for each pixel in a reference image, but consists of both the motion fields that are parallel and perpendicular to the image plane [11].

Vedula et al. [6] formulated the solution of generating scene flow without assuming rigidity of the observed scene. They applied two-dimensional optical flow to estimate the scene flow. Meanwhile, since three-dimensional motion cannot be completely observed from the change in the image plane, Patras et al. combined optical flow and the change of disparity for motion estimation. Such an idea is followed by Waxman et al. [10], who utilized multi-view images and a global smoothness constraint for the motion estimation. Subsequent work by Gong et al. estimated disparity flow with stereoscopic images [13].

Structure flow explored another path for describing 3D motion. It was practically formulated as scene flow scaled by the inverse depth and was implemented with a filter-based solution by Adarve et al. [7]. Meanwhile, the advantage of structure flow over scene flow was not discussed.

C. Network-based motion estimation

Deep convolutional neural networks (DCNN) were firstly applied for estimating the two-dimensional optical flow by Dosovitskiy et al. [14]. They proposed FlowNet that employed an hourglass-style architecture with encoder and decoder modules. FlowNet has been extended with improved modules and had more robust performance against occlusion [15, 16, 17]. SpyNet developed from FlowNet with the idea of the spatial pyramid [18]; it is a novel light-weighted architecture for optical flow estimation. Some traditional computer vision techniques, such as warping and cost volume, were combined with the network-based method by Sun et al., the integrated approach achieves state-of-the-art performance [19].

The estimation of the three-dimensional scene flow with a network-based method was explored by Mayer et al. [21]. They predicted dense disparity maps with continuous stereo frames. Subsequent work fused the independently estimated depth maps and optical flow maps for generating dense three-dimensional scene flow maps [20]. SF-Net estimated optical flow from RGBD images [32].

D. Dataset for motion field

The Middlebury dataset contains RGB image pairs and corresponding optical flow maps that are calculated with high-resolution UV images. It is widely used as a benchmark for evaluating the performance of optical flow estimation [21]. The Kitti dataset is a large dataset with optical flow and depth annotations, focused on the application of automated driving and was collected with autonomous driving platforms [9, 23, 24].

The development of the virtual motion field dataset started from McCane's work that applied synthetic sequences with optical flow and provided a metric with angular error [25]. Since the early-stage synthetic datasets were not realistic, MPI-Sintel was created to simulate spatial features that are semantically similar with the real-life scene [26]. Mayer's work produced multiple datasets that have a larger amount of annotated images compared with previous datasets [35]. In addition, [35] contains equally accurate motion measurements in all three axes.

III. Methods

Theoretically, structure flow can be estimated directly from a sequence of monocular images as it is insensitive to the scale of looming motion, whereas this is ambiguous for scene flow. Hence, the aim of this paper is to investigate whether structure flow can show this advantage over scene flow. Learnt from the mature techniques of scene flow estimation, we introduce our network-based approach to fuse optical flow and depth change for estimating the structure flow.

A. Recap of motion estimation

Correlation and Warping are two key operations to quantitatively evaluate motion from continuing image frames.

[12] introduces the correlation operation for capturing the pixel-wise motion between two feature maps. Denoting two feature maps by f_1 and f_2, the correlation operation $c(f1(x_1, y_1), f2(x_2, y_2))$ computes the inner product of square patches that are centred in (x_1, y_1) on f_1 and centred in (x_2, y_2) on f_2 :

$$c(x_1, x_2) = \sum_{o_x, o_y \in [-k,k]^2} \langle f_1(x_1 + o_x, y_1 + o_y), f_2(x_1 + o_x, y_1 + o_y) \rangle \quad (2)$$

where o_x and o_y represent the pixel-wise shift for x and y axes and the dimension of the patch is $(2k+1)^2$. In this case, $(2k+1)^2$ correlation maps are generated to represent the motion in all directions, which requires a high storage cost. Further, correlation computation is time-consuming if the distance between (x_1, y_1) and (x_2, y_2) is not restricted. We

take d to denote the maximal Euclidean distance between the two patch centres on different feature maps, that is

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} < d \quad (3)$$

The size of patch k and the maximal distance d are dataset specific parameters that depend on motion and feature size. So that correlation operation is a time-consuming operation as its parameter have to be determined experimentally.

Due to these high costs, [27] applied image warping to estimate two-dimensional motion flow maps. Denoting $warp(f, \phi)$ as an operation for warping feature map f with optical flow ϕ, the error of ϕ (denoted by e_ϕ) can be reflected by the Euclidean norm between the warped first feature map and the second feature map.

$$e_\phi = \|\text{warp}(f_1, \phi) - f_2\|_2 \quad (4)$$

In this case, the warping operation bridges the corresponding spatial features in two features, so that minimizing e_ϕ can be considered as an implicit method to acquire an optimal ϕ. Compared with correlation, warping significantly saves computational cost, and so is used in our technique.

B. Spatial pyramid structure

A spatial pyramid structure called hierarchical correlation for the motion estimation in [27] and was reintroduced by [18]. It utilized a CNN to generate optical flow maps. Following that, we aim to design a coarse-to-fine method for dealing with multi-scale motion estimation.

Given an initial feature map f_0 with size $a \times b$, the coarser features can be generated iteratively by pooling and the size of the generated features f_n is $(\frac{a}{2^n} \times \frac{b}{2^n})$. It leads to a set of features $f = \{f_0, \dots, f_n\}$ whose elements have decreasing size.

For the inverse operation that generates feature maps with a larger size, we applied the image interpolation from [36] to up-sample features.

C. Estimation of three-dimensional motion field

Given a pair of images (I, I') as input, we aim to estimate a three-dimensional motion field ψ that describes the pixel-wise motion from I and I'. In general, ψ can be defined as either scene flow or structure flow.

Based on section *B*, a set of image pairs $\boldsymbol{I} = \{(I_0, I_0'), \dots (I_n, I_n')\}$ is generated and the estimation starts from the coarsest image pair (I_n, I_n'). Following the idea of [18], the corresponding two-dimensional optical flow ϕ_n is initialized as a zero map to conduct the warping operation on I_n'. In this case, the warped image is denoted by $\tilde{I}_i$.

Assume i (in the range 0, …, n) denotes the sequence of processing, we take the concatenated features $F = \{I_i, \tilde{I}_i, \phi_i\}$ as input to estimate the motion field ψ_i that has the same size as I_i. The estimation function is expressed by $P()$ in the following paragraph, therefore:

$$\psi_i = P(\{I_i, \tilde{I}_i, \phi_i\}) \quad (5)$$

Such a process is conducted iteratively until ψ_0 is generated. Generally, the algorithm of three-dimensional optical flow estimation is expressed as below.

Algorithm 1: three-dimensional motion field estimation

Input: A set of image pairs $\boldsymbol{I} = \{(I_0, I_0'), \dots (I_n, I_n')\}$ in decreasing sizes generated by spatial pyramid
Input: Initialized zero optical flow ϕ_n
Input: maximal level of structure spatial pyramid n
Output: Estimated 3D motion field ψ_0 in the original size
For i in range from n to 0:
Warped image $\tilde{I}_i \leftarrow warp\ (I_i', \phi_i\ )$
Feature map $F \leftarrow$ Concatenate $\{I_i, \tilde{I}_i, \phi_i\ \}$ in the channel dimension
Estimate 3D motion field $\psi_i \leftarrow P(F)$
Extracted 2D motion field ϕ_i from ψ_i
Up-sampled optical flow $\phi_i \leftarrow$ interpolation (ϕ_i)

D. Network architecture of the motion estimator

We implement the motion flow estimators $P()$ from the previous section with a 6-layer CNN. Such a network takes stacked features as input and outputs a three-dimensional motion field.

Following section B, although the spatial pyramid structure is able to handle multi-scale features, two problems still exist. Firstly, the error of the estimated motion field in different sizes is accumulated through backpropagation, which leads to a huge numerical updating of the network parameters. Such updating will cause overflow if the pyramid structure has too many layers. Further, small objects cannot be visible on the coarsest images. Since the coarsest image is the reference for the motion estimation on finer images, it drives the network to neglect these objects when estimating the finer motion field. For avoiding these problems, we empirically set the levels of the spatial pyramid to be four (n = 4).

The network architecture is shown in Fig.2. The input image pairs are down-sampled for 4 times from left to right and transmitted to the red processor module.

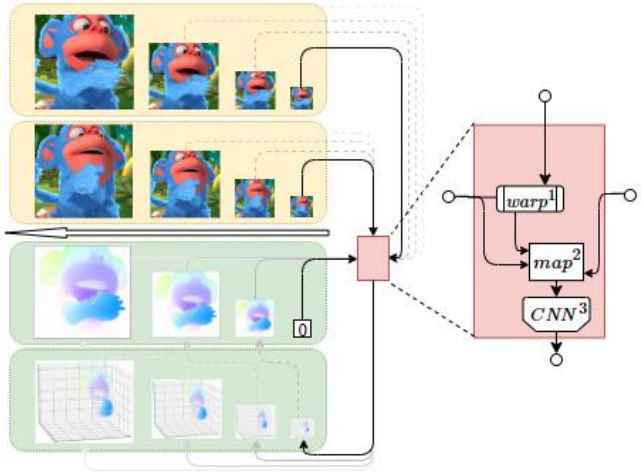


Fig. 2. The architecture of the network for estimating three-dimensional motion.

The coarsest image pair (I_4, I_4') and a zero optical flow are taken for conducting the operation shown in Algorithm1. The first image I_4 is warped with the initialized zero map to generated $\tilde{I}_4$ (in $\boldsymbol{warp^1}$ block). The warped second image $\tilde{I}_4$, the second image $\tilde{I}_4$ and the optical flow are concatenated to be a feature map with six channels (in $\boldsymbol{map^2}$ block). The $\boldsymbol{CNN^3}$ module processes the feature map and generates a 3D

motion field ψ_4 as the output of the initial iteration. After four iterations, the final output, ψ_0, has the same size as the original images.

E. Loss function design for scene flow and structure flow

We generate structure flow based on the optical flow and disparity as they are commonly provided in public motion datasets.

Given continuous input images I and I', the optical flow from I' to I is denoted by ϕ and the corresponding pixel-wise disparity maps are denoted by $\mathcal{D}$ and $\mathcal{D}'$. Since our method is designed to predict an arbitrary three-dimensional motion field ψ, both the scene flow and the structure can be estimated.

We formulate the ground truth scene flow V_{ground} and structure optical flow ω_{ground} as:

$$V_{ground} = \{[\phi_x, \phi_y, \mathcal{D}' - warp(\mathcal{D}, \phi)]\} \quad (6)$$

$$\omega_{ground} = \left\{\left[\phi_x, \phi_y, \frac{\mathcal{D}' - warp(\mathcal{D}, \phi)}{\mathcal{D}}\right]\right\} \quad (7)$$

In practise, we ignore focal length, as it is a uniform scale. Therefore, different from the definition in (1), we only scale the motion in the z-axis by the inverse depth in (7). Also note that the motion in the x- and y-axes are in pixel units, while the motion in the z-axis is dimensionless.

Given the predicted motion field V_{pred} and w_{pred}, the loss functions for both three-dimensional motion fields are defined by the Euclidean norm:

$$loss_V = \left|V_{pred} - V_{ground}\right|_2 \quad (8)$$

$$loss_\omega = \left|\omega_{pred} - \omega_{ground}\right|_2 \quad (9)$$

F. Dataset selection

The mainstream datasets for learned motion field estimation are summarized as below.

Table I: Statistics of the public datasets of motion field

	Frames	Virtual	Precision	Proportion[1]
Monkaa [19]	8640	True	Good	60: 0.8: 1
Driving [19]	1098	True	Good	4.7: 1.1: 1
Sintel [24]	1041	True	Bad[2]	1.5: 1: 2.6e6
Kitti [21, 22]	191136	False	Bad[2]	/
Virtual Kitti [32]	8640	True	Bad[2]	3.5: 4.2: 1

[1]The proportion of motion is in the sequence of x, y and z, while the positive values indicate leftward, upward and looming (i.e., moving towards the camera) direction respectively.

[2]Sintel has different precision of optical flow and depth measurement, while the annotation maps in Kitti and Virtual Kitti are sparse.

Most of datasets in Table I are not suitable for estimating three-dimensional structure flow for two reasons. The low-accuracy of the measurement of the motion field is the first reason (e.g., Kitti, Virtual Kitti, and Sintel). Such low-accuracy measurement causes missing objects and erroneous depth annotations (as shown in Fig. 3 and Fig. 4).

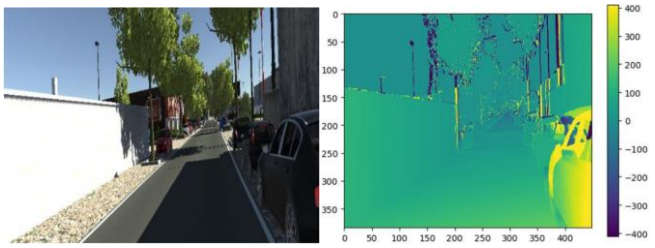
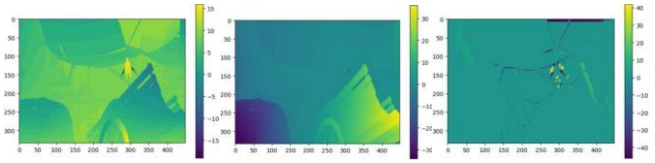


Fig. 3: A sample image and its ground truth of depth change in the Virtual Kitti. The tree trunks and poles have abnormally large or small values due to their depth being missing in the annotation map.



(a) x-axis (b) y-axis (c) z-axis

Fig. 4: Structure flow ground truth of a sample image in the Sintel dataset. The annotation in the z-axis (c) has a worse precision. It leads to a unsmooth pattern that is inconsistent with (a) and (b).

Another reason is the imbalanced motion in different axes. Proportion in Table I records the summed motion in each axis, and it is defined as:

$$r_x : r_y : r_z = \frac{\Sigma\phi_x}{\boldsymbol{\phi}} : \frac{\Sigma\phi_y}{\boldsymbol{\phi}} : \frac{\Sigma\phi_z}{\boldsymbol{\phi}} \quad (10)$$

where $\boldsymbol{\phi} = \Sigma\phi_x + \Sigma\phi_y + \Sigma\phi_z$. We compute the motion proportion of each dataset in Table I except Kitti as that has too many frames. This result shows the imbalance of different motion components among three axes commonly occurs in public datasets.

The influence of the imbalance is not vital for the method that estimates the optical flow and disparity independently. However, since our model estimates the motion in all three dimensions, large errors in the motion in one or two axes will dominate the loss.

G. Motion visualization

The current method of motion visualization displays different directions of motion with the colour cycle in HSV space. Since HSV space can only fit a two-dimensional vector space, the optical flow and the depth change are generally presented separately.

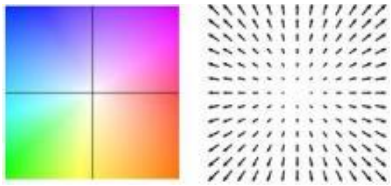


Fig. 5: The correlation between the optical flow and the colour in HSV space

Although such visualization displays the motion field, we cannot intuitively compare the magnitude of optical flow and depth change. In this case, we suggest a three-dimensional visualization of motion with a mesh plot that combines optical flow and depth change. Since generating a mesh plot is time-consuming, the three-dimensional motion map is down-sampled to a sparse manner for visualization.

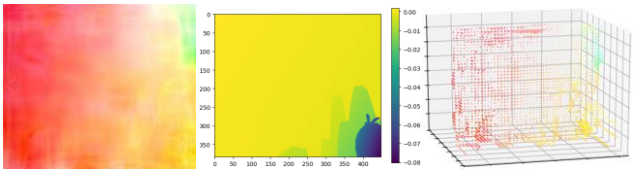


Fig. 6: Separate visualization (left: optical flow, mid: depth change) verse the rightmost down-sampled combined visualization

IV. Experiments

A. Network training

Based on the discussion in section *F* of the *Method*. Our structure optical flow estimation technique is applied to the Monkaa and Driving datasets as these two datasets have precise motion measurements in all three axes [21]. The estimation of the scene flow is produced for comparing with structure flow. We also compare the scene flow estimation of our method with previous works.

As shown in section *C* of *Method*, our method includes the estimation of the motion field at four different scales and we assume the input RGB image has size 448 × 334. The network-based estimator model is implemented with Pytorch [28].

Based on Pytorch framework, our network is trained with learning rate $a_1 = 0.001$ and SGD optimization in the first 100 epochs, the learning rate is then updated to $a_2 = 0.0001$ for converging. The entire process requires 180 epochs in 41 hours on a device with i7-6700k CPU and GTX1080 GPU.

The separation of training sets and testing sets is based on the scenes in datasets. For each scene, we randomly select 10% image pairs for testing, while the rest images pairs are treated as the training set.

For image augmentation, we conduct image rotation and random cropping for enhancing the robustness of our model. We follow [18] and set rotation range within $[-5°, 5°]$, followed by random cropping, then and resizing to original dimension. These processed images are then normalized using a calculated mean and standard deviation from ImageNet [29].

B. Benchmark result demonstration

We train our model for estimating structure flow and scene flow independently and the evaluation is based on average end-point-error between the estimated motion and ground truth in three dimensions. The experimental results of Monkaa and Driving are shown in Table II.

Table II: Comparison of the average end-point-error of results on Monkaa benchmark

	Structure flow		Scene flow	
	3D	2D	3D	2D
Monkaa	3.41(7.14)	(5.62)	2.74(5.33)	(4.32)
Driving	1.74(2.94)	(2.23)	1.63(2.81)	(2.22)

Training loss (testing loss)

Table II shows training and testing loss (in brackets). Our model is trained by minimalizing estimation error in three dimensions. The two-dimensional estimations have only testing loss as the predictions are extracted from the corresponding three-dimensional motion estimations. Without fine-tuning, Table II indicates that the training on only Monkaa is inadequate. Further, it also exhibits consistency of errors between the three-dimensional and two-dimensional motion estimation.

Table III: Comparison of the average end-point-error with previous works on Monkaa benchmark

	Two-dimensional optical flow	Three-dimensional Scene flow
PD-flow[28]*	43.62	-
SRSF[1]*	21.81	-
Sun et al.[29]*	19.54	-
SF-Net [30]*	4.91	-
Ours	**4.32**	5.33

*Take RGB-D images as the input

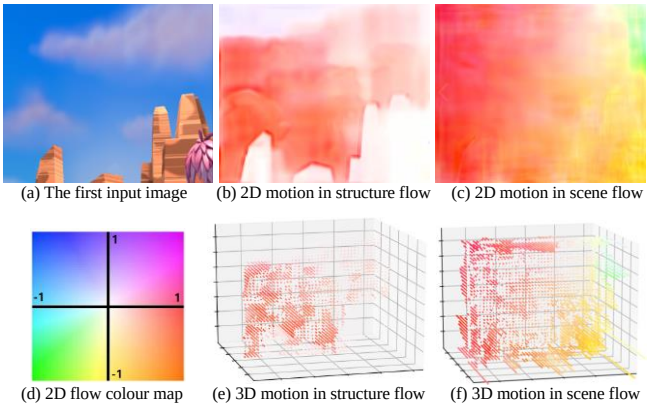
Comparison with previous works is based on two-dimensional optical flow since there is no current public benchmark for both scene flow and structure flow (Table. III). Different from previous works that use the images as well as their corresponding depth maps as input, our method focuses on monocular image and only employs RGB images as input.

Table III shows the superiority of our method on motion estimation, where our method outperforms the previous works with no depth information involved.

C. Comparison between scene flow and structure flow

Although numerically error is slightly improved for scene flow, we demonstrate the advantage of the structure flow over the scene flow with the visual estimation result.

Structure flow is theoretically less sensitive to depth change as its magnitude is proportional to pixel-wise motion in the image plane. One of the representative examples is shown below.

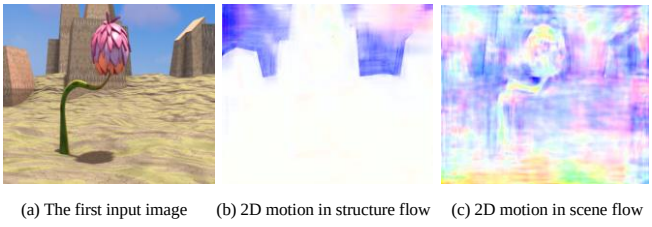


(a) The first input image (b) 2D motion in structure flow (c) 2D motion in scene flow

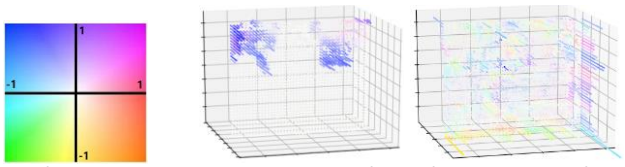
(d) 2D flow colour map (e) 3D motion in structure flow (f) 3D motion in scene flow

Fig. 7: The demonstration of the estimated result, structure flow captures the precious boundary of objects while scene flow does not.

With camera motion rotating around the mountain, the objects (mountains and a flower) are almost static compared with the moving clouds (Fig. 7). In this case, scene flow fails to identify the boundary of the front objects in the motion field, as the subtle motion of front objects and the large motion of the background are mixed. On the contrary, the structure flow shows recognizable front object motion as their motion fields are scaled by their magnitude of looming motion.



(a) The first input image (b) 2D motion in structure flow (c) 2D motion in scene flow



(d) 2D flow colour map (e) 3D motion in structure flow (f) 3D motion in scene flow

Fig. 8: The demonstration of the estimated result, scene flow map is orderless against the background with uniform textures.

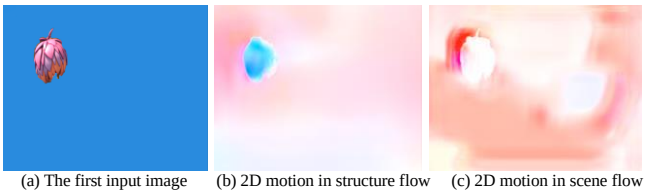
Fig. 8 shows camera motion with small magnitude with respect to front objects, which is correctly observed in the structure flow map. Meanwhile, the pixel-wise motion in the scene flow map lacks order. This is caused by the uniform texture of the background. Without depth scaling, the pixel-wise motion of the background between two images is not separated, leading to a cluttered estimated map.

In general, as an alternative representation of three-dimensional motion, structure flow demonstrates advantages over scene flow in foreground background motion cases whereas otherwise distant motion will dominate.

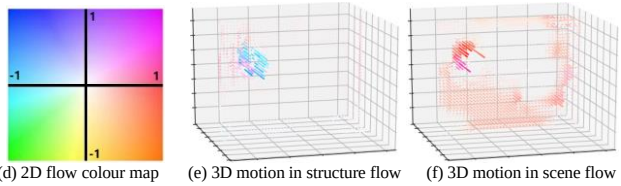
D. Discussion of color influence and bias

We observed consistency between the colour and object semantics. For an instance of the Monkaa dataset, the sky is the only object that indicates blue in all the scenes. In this case, we further validate the performance of the model in a pure background scene.

We extracted the colour of the sky in the Monkaa as the background for the experiment shown in Fig. 9. By moving the snipped pink flower leftward for five pixels, we observe the difference between structure flow and scene flow estimations.



(a) The first input image (b) 2D motion in structure flow (c) 2D motion in scene flow



(d) 2D flow colour map (e) 3D motion in structure flow (f) 3D motion in scene flow

Fig. 9: The demonstration of the estimated result with a blue background. Structure flow is more robust than scene flow.

Fig. 9 shows that the structure flow map correctly unveils the boundary and motion of the flower, while scene flow map cannot reflect the motion of the flower. As for the static background, we observe that the scene flow shows an unexpected motion in the mesh plot, but such motion is less significant in the structure flow map.

Further, the blue sky is usually moving in the Monkaa, which shows the consistency between the motion and the colour. With the recorded proportion of motion (-60:0.8:1) in Table2, we can find the rightward motion is the most frequent, which explains why the estimated structure flow shows the static background moving rightward.

V. Conclusions

In this paper, we introduced a network-based model for estimating structure flow. We investigated existing image motion datasets and discussed the feasibility of fitting a structure flow model based on the existing datasets. Our quantitative experiments show that our method has superior performance compared to previous works without employing RGB-D images. Based on the effective motion estimation, we show the robustness of structure flow against a background with ambiguous looming motion via monocular vision .

From another perspective, our experiment shows the unexpected correlation between the color and the direction of motion in the Monkaa dataset, which unveils over-fitting problem. Applying proper regularization for solving this problem is one of our future areas of investigation. Besides, as shown in Table I, the current motion field datasets do not consider balancing the motion components in different axes. The unbalanced magnitude of motion in different axes as well as the imbalanced colours result in biased estimation of the motion field. In this case, it is worth creating a larger and more balanced dataset for avoiding these biases.

Although structure flow and scene flow have a similar definition, we find that their estimated maps reflect different behaviours with the same input image (Fig. 9). This observation leads us to a hypothesis that a model may be able to estimate a better three-dimensional motion field if it has learned both structure flow and scene flow. This hypothesis suggests a multi-task learning task like [31] as future work.

In conclusion, the structure flow brings a novel view of three-dimensional motion and our work shows the advantage of the structure flow compared with traditional scene flow. We believe the idea of structure flow should benefit the development of motion estimation in the future.

References

[1] W. Zhao, R. Chellappa, PJ. Phillips, A. Rosenfeld, “Face recognition: A literature survey.”, ACM computing surveys (CSUR), 2003.

[2] Z. Sun, G. Bebis, R. Miller, “On-road vehicle detection: A review”, IEEE transactions on pattern analysis and machine intelligence, 2006, 28(5): pp.694-711.

[3] BKP. Horn, BG. Schunck, “Determining optical flow”, Techniques and Applications of Image Understanding. International Society for Optics and Photonics, 1981, 281, pp. 319-331.

[4] BD. Lucas, T. Kanade, “An iterative image registration technique with an application to stereo vision”, 1981.

[5] J. Quiroga, T. Brox, F. Devernay, J. Crowley, et al, “Dense semi-rigid scene flow estimation from rgbd images”, in European Conference on Computer Vision. Springer, Cham, 2014, pp.567-582.

[6] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade, "Three-dimensional scene flow." Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2. IEEE, 1999.

[7] J. Adarve. “Real-time Visual Flow Algorithms for Robotic Applications”. Australian National University, PhD dissertation, 2017.

[8] S. Ullman. “The interpretation of visual motion”, Massachusetts Inst of Technology Pr, 1979.

[9] A. Geiger, .P Lenz, R. Urtasun, “Are we ready for autonomous driving”, Computer Vision and Pattern Recognition, 2012.

[10] M. Rubinstein, C. Liu, WT. Freeman, “Towards longer longrange motion trajectories”, 2012

[11] Y. Zhang, C. Kambhamettu, “On 3D scene flow and structure estimation”, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[12] A. M. Waxman, JH. Duncan, “Binocular image flows: Steps toward stereo-motion fusion.”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986 (6), pp.715-729.

[13] M. Gong, YH. Yang, "Disparity flow estimation using orthogonal reliability-based dynamic programming." 18th International Conference on Pattern Recognition Vol. 2. IEEE, 2006.

[14] A. Dosovitskiy, P. Fischer, E. Ilg, et al, "Flownet: Learning optical flow with convolutional networks." Proceedings of the IEEE international conference on computer vision, 2015.

[15] E. Ilg, T. Saikia, M. Keuper, T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation." Proceedings of the European Conference on Computer Vision, 2018.

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, et al. "Flownet 2.0: Evolution of optical flow estimation with deep networks." Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

[17] X. Li, C. Loy, "Video object segmentation with joint reidentification and attention-aware mask propagation", Proceedings of the European Conference on Computer Vision, 2018.

[18] A. Ranjan, MJ. Black, "Optical flow estimation using a spatial pyramid network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[19] D. Sun, X. Yang, MY. Liu, J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[20] Z. Yin, J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[21] N. Mayer, E. Ilg, P. Hausser, P. Fischer, et al, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[22] S. Baker, D. Scharstein, JP. Lewis, et al, "A database and evaluation methodology for optical flow." International Journal of Computer Vision 92.1, 2011, pp.1-31.

[23] M. Menze, A. Geiger, "Object scene flow for autonomous vehicles." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[24] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets robotics: The KITTI dataset." The International Journal of Robotics Research32.11, 2013, pp.1231-1237.

[25] B. McCane, K. Novins, D. Crannitch, B. Galvin, "On benchmarking optical flow." Computer Vision and Image Understanding 84.1, 2001, pp.126-143.

[26] A. Bhoi, "Monocular Depth Estimation: A Survey.”, 2019.

[27] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, “High accuracy optical flow estimation based on a theory for warping.” European conference on computer vision. Springer, Berlin, Heidelberg, 2004.

[28] A. Paszke, S. Gross, S. Chintala, et al. "Automatic differentiation in pytorch.”, 2017.

[29] K. He, X. Zhang, S. Ren, J. Sun, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[30] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, D. Cremers, "A primal-dual framework for real-time dense RGB-D scene flow." 2015 IEEE international conference on robotics and automation. IEEE, 2015.

[31] D. Sun, EB. Sudderth, H. Pfister, "Layered RGBD scene flow estimation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[32] Y. L. Qiao, L. Gao, Y. Lai, et al, "SF-Net: Learning scene flow from RGB-D images with CNNs.", 2018.

[33] M. Siam, H. Mahgoub, M. Zahran, et al. "Motion and Appearance Based Multi-Task Learning Network for Autonomous Driving.", 2017.

[34] A. Gaidon, Q. Wang, Y. Cabon, “Virtual Worlds as Proxy for Multi-Object Tracking Analysis”, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp.4340-4349.

[35] D. J Butler, J. Wulff, G. B Stanley, M. J Black, “A naturalistic open source movie for optical flow evaluation”, European conference on computer vision. Springer, Berlin, Heidelberg, 2012, pp.611-625.