Deep 360° Optical Flow Estimation Based on Multi-Projection Fusion

Yiheng Li[1], Connelly Barnes[2], Kun Huang[1], and Fang-Lue Zhang[1,*]

[1] School of Engineering and Computer Science, Victoria University of Wellington
{Yiheng.Li,Kun.Huang,Fanglue.Zhang}@vuw.ac.nz
[2] Adobe Research, Seattle, US
ConnellyBarnes@gmail.com

Abstract. Optical flow computation is essential in the early stages of the video processing pipeline. This paper focuses on a less explored problem in this area, the 360° optical flow estimation using deep neural networks to support increasingly popular VR applications. To address the distortions of panoramic representations when applying convolutional neural networks, we propose a novel multi-projection fusion framework that fuses the optical flow predicted by the models trained using different projection methods. It learns to combine the complementary information in the optical flow results under different projections. We also build the first large-scale panoramic optical flow dataset to support the training of neural networks and the evaluation of panoramic optical flow estimation methods. The experimental results on our dataset demonstrate that our method outperforms the existing methods and other alternative deep networks that were developed for processing 360° content.

Keywords: Optical Flow, 360° Video, Spherical Image Projection

1 Introduction

If we take a sphere representing all viewing directions for 360° images and remove one point from it, then there is no projection of the resulting surface to a 2D plane that is both an equal area map and angle-preserving (conformal). This is because the sphere is not developable [10], as is well-known from cartography. This mathematical fact led to classical research into mesh parameterizations such as [28,30] that trade off between competing goals like area and angle-preservation when mapping textures on spherical surfaces. In this research, we were initially motivated to discover good projections for optical flow learning for 360° videos, which remains an interesting and open question in this field. We investigated three different projections from the sphere to the plane: cylindrical and cube-map, which are conformal within each chart, and equirectangular, which is neither area- nor angle-preserving. However, to our surprise, our experiments did *not* seem to indicate that any of our projections is always

[*] Corresponding author

better than the others, but rather that these projections make *complex tradeoffs* in optical flow accuracy that appear to depend jointly on the input image and the mathematical properties of each projection.

Since empirically no single projection appears to always be best, we therefore propose to learn optical flow using multiple projections: we learn to fuse the complementary information into a single final optical flow on the sphere. We find that such learned fusion of multi-projection optical flows consistently outperforms the single-projection optical flows that were fused. We thus propose that multi-projection optical flow with fusion can be an important principle for 360-degree optical flow, even though more remains to be discovered in future work about e.g. the desired mathematical properties of good projections. Since there are no existing large-scale datasets with ground truth panoramic optical flow to learn the full field-of-view (FOV) features, we also build the first 360° optical flow dataset that contains various scenes and dynamic objects, including around 6000 panoramic frames for training our deep models.

Besides the above issue due to the mathematical properties of projections for 360° optical flow, there are common challenges with narrow FoV videos such as robustness to outliers, illumination changes, and possible large displacements, which have been well addressed in existing non-360° deep optical flow deep models [31,7,14,37,40]. Although such motion models are formulated on 2D regular grids with convolutions that do not inherently deform according to distortion or area changes in the projection, we can still utilize them as strong backbone models to generate accurate complementary regions for different projections. To make the maximum use of their pre-trained models that can handle complex dynamic objects, we did not use spherical kernels [6] to modify their convolutional layers, as we found the modified receptive field of such kernels led to worse results when finetuning the backbone models using our datasets.

In summary, this paper makes the following contributions: 1. We introduce a fusion framework for learning 360° optical flow, which learns to fuse the complementary optical flow prediction results generated using different projections; 2. We apply common map projection techniques to make two spherical image projection methods, namely tri-cylinder projection and cube padding projection that are complementary to equirectangular projection in terms of distorted regions and performance under fusion; 3. We build a novel 360° optical flow dataset with ground truth optical flow data with various camera paths and dynamic objects.

2 Related Work

This section reviews prior work in the relevant domains of optical flow estimation, datasets for training deep models, and 360° image and video processing.

Optical Flow Estimation Optical flow shows the motion of objects in a video by a dense vector map pointing to the corresponding pixels between frames. The traditional methods mostly rely on feature constancy and local spatial coherence to solve the problem in an optimization framework [22,11,36,21,20]. In the era of end-to-end deep learning, the advances in optical flow estimation are driven by deep models [38]. The work of FlowNet [7] built a solid foundation for optical flow prediction using deep networks. They use FlowNetSimple to learn optical flow directly from adjacent frame pairs and use FlowNetCorrelation to learn the correlation between the patch-level features of the pair of input frames. FlowNet2 [14] presents some improvements based on FlowNet by proposing a stacked architecture and training schemes that benefit the convergence of the training process. PWC-Net [37] proposed to use Pyramid, Warping, and Cost volume to simplify the network structure and achieve high performance in optical flow estimation. RAFT [40] uses Recurrent All-Pairs Field Transforms that apply all-pairs correlation information into the neural network. In contrast, the above models only use nearby patches for correlation information for saving memory. Some other works focus on unsupervised learning schemes [24,42,19,18], fusing flows from multiple frames [32], and joint learning of optical flow and occlusions [13], aiming to train networks with the objective from view synthesis [45]. Despite promising results, generalizing these methods beyond narrow FOV videos remains challenging.

Optical Flow Dataset Current optical flow prediction methods treat optical flow estimation as a supervised learning task that requires a considerable amount of data for training. As the real-world optical flow data is hard to access, most datasets are synthetic and rendered by Blender, while few of them are collected by camera and LIDAR. Sintel [3] is derived from a 3D movie to provide thousands of image pairs with dense ground truth optical flow. Wulff et al. [43] further extend it to two versions, Clean and Final, which contains RGB albedo and visual effects such as motion blur and fog, respectively. FlyingChairs [7] is generated by a set of 3D chairs models and applies the random 2D affine transformation to the scene. The idea was then expanded to generate FlyingThings3D [23] from the assets published in Stanford's ShapeNet, where random motions were applied on the dynamic objects. KITTI [8] [25] dataset was collected from real-world by cameras and LIDAR installed on a car. However, due to the device limitation, their optical flow is sparse, especially for the natural background where the LIDAR could not provide valid depth information. To address the sparsity issue of the KITTI dataset, Mayer et al. released a dataset called Driving [23], which renders a naturalistic city from the perspective of a driving car to generate dense optical flow maps. Moreover, for the convenience of nonrigid and softly articulated motion analysis, Monkaa [23] is designed to resemble Sintel using cartoon animations. A small scale dataset of 360° optical flow was proposed recently by Yuan and Richardt [46], which can only be used for evaluation due to its limited size. To support the training of 360° optical flow estimation networks, a new dataset is needed in this domain.

360° Image and Video Processing 360° images and videos have been attracting increasing attention from computer vision and graphics researchers. Due to the special representation of 360° images, dedicated methods were developed for the analysis and processing tasks, such as depth estimation [41], saliency detection [27,48], edit propagation [47], and video stabilization [16,39]. With the recent development of immersive technologies, researchers have been working on the specific problems of panoramic video-based applications, such as automatic 360° navigation/piloting [15,12], 360° video assessment [17], and immersive video editing [29]. The lack of reliable temporal correspondences that are globally consistent hinders many potential applications of 360° video generation. Our method is proposed to fill that gap.

The majority of current convolutional networks were developed for coping with perspective projection onto a planar camera sensor. Researchers proposed several solutions to adopt convolutions on spherical images to facilitate the feature extraction and interpretation by the deep networks. A representative method, namely SphereNet [6,44], changes the perceptual field of their convolutional kernels according to the latitude on the equirectangular domain. However, changing the kernels of 2D optical flow networks to spherical kernels may degrade the effect of the pre-trained models, as shown in our experiments. The kernel transform technique [35] was also considered in this field to let the convolutional layers learn how to transform the spherical kernels to the pre-trained weights on regular kernels that were initially trained by perspective images. However, the current optical flow networks all have a large number of layers, causing an impractical computation and memory cost to learn to transform all the layers. Projecting the spherical images to cube maps can avoid the distortion issues when utilizing deep neural networks, as in 360° saliency detection [5]. However, after testing with cubemap-based projection, we found the inconsistency of the predicted results along the cube face boundaries can not be trivially addressed, which also inspires us that using any single projections would meet more or fewer issues in certain regions. We thus propose to fuse optical flow results predicted using different projections to utilize their complementary information.

3 Dataset Generation

Due to the lack of ground truth data [1] of 360° optical flow, existing optical flow estimation networks could not be appropriately trained for 360° videos. We thus build a dataset with ground truth panoramic optical flow maps by rendering synthetic 360° videos of dynamic 3D scenes.

3.1 Optical Flow Generation

Two assumptions are made when generating panoramic optical flows when rendering a 3D scene, which are commonly followed by other optical flow datasets [23]. Firstly, all the visible points that appear on a frame can ideally be seen on the next frame. Secondly, the geometry of rendered objects is not deformed. Denote a pixel position as p, its position on the corresponding 3D object O [34] can be calculated using its model transformation matrix M that represents the object motion and the camera projection matrix P by $O = M^{-1}P^{-1}I$. For the successive frame, assume M' and P' represent its model matrix and camera matrix, respectively, the corresponding image pixel positions for the point O on the 3D object can be obtained by $I' = P'M'O$. In equirectangular projection, optical flow needs to be transformed into spherical coordinates from a 3D point $(x, y, z)^T$ on the unit sphere in the camera space by:

$$\theta' = \text{atan}\left(\frac{z'}{x'}\right), \quad \phi' = \text{atan}\left(\frac{y'}{\sqrt{(x')^2(z')^2}}\right) \tag{1}$$

From this, we can obtain the optical flow vectors $(\triangle\theta, \triangle\phi)$ in spherical coordinates. We can easily convert and store these in an equirectangular projection-based representation. More details are included in the supplementary materials.

3.2 3D Scenes

We built our panoramic RGB and optical flow frames using two types of scenes: **City Scene** We designed the City scene to resemble the data with similar properties as KITTI [8] and Driving [23]. The city's layout is automatically procedurally generated, and over 300 vehicles are randomly moving in the city. We enable collision detection for all the objects in the scene to ensure that it can avoid the situations of object-object and object-camera interpenetration. For the pre-training and preliminary experiments, we built two subsets named City1000UR and City100UR, which contain 1011 and 102 up-right frames for fine-tuning and testing. We divide the whole 3D city into two parts, where half is used for generating training data and the other half for the validation and testing data. We then generated City2000, City200, and City100 datasets, where the camera is rotated within 45° , containing 2000, 217, and 138 frames, respectively. **Equirectangular FlyingThings Scene** We designed the EquirectFlyingThings scene to resemble FlyingChairs [7] and FlyingThings3D [23]. In this scene, objects are randomly distributed around the camera. In our implementation, we choose car models from Stanford's ShapeNet dataset[4], which contains complex mesh shapes and textures. Compared to FlyingThings3D, the EquirectFlyingThings dataset contains more objects in the scene because the camera takes panoramic photos. We rendered 300 objects to fill in the content and scale them to a suitable size. We also apply random translation and rotation to those objects. Furthermore, we change the camera rotation direction every 20 frames to keep randomness without camera shake. We built EFT2000, EFT200, and EFT100 datasets, and each of them contains 2211, 199, and 99 image pairs.

For our final models, we take the City2000 and EFT2000 as training datasets, while the City200 and EFT200 are used for validation. The final testing is performed on the City100 and EFT100 datasets.

4 Multi-Projection Fusion

To maximally leverage existing narrow FOV pre-trained models while minimizing the impact of various distortions that occur in spherical projections, we propose to use different methods to project the spherical images to planar images and

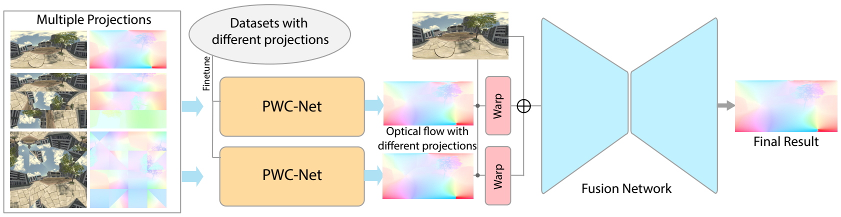


Fig. 1. Our multi-projection fusion method for estimating 360° optical flow

feed them to finetune the models. A fusion layer is finally built to combine the transformed optical flow results for the final 360° optical flow.

We first perform preliminary experiments to select the best baseline model to finetune and predict the optical flow with equirectangular projections. We adopt the widely used models FlowNet2 [14] and PWC [37], and the most recent state-of-the-art method RAFT [40] to finetune on our 360° dataset using equirectangular projection. We found that PWC achieved the highest performance and the equirectangular representation is the best single projection for optical flow estimation. Because our experiments indicate that fused models always perform no worse (and in fact, better) than single projections, we choose PWC as our baseline model and use it to finetune the optical flow prediction model for each projection. In addition, we also modified all the PWC-Net's convolutional layers to Spherical convolutional kernels as in [6]. However, the experimental results show that the finetuning with spherical convolutions cannot achieve a good performance due to kernel shape changes from the original pre-trained PWC model. More results are demonstrated in later sections.

4.1 Projections for Spherical Images

Here, we investigate the performance of the deep models when feeding 2D images projected from the spherical domain using different kinds of projections. To utilize the pre-trained 2D deep models, ideally we would wish for the panoramic view to be an isometry to the original spherical surface, after making at least one cut. This ideal projection does not exist as the sphere is not developable, so distortion is necessary. In our experiments, we thus explore different mapping methods. It is also worth noting that we take the optical flow C^0 and C^1 continuity as another important consideration because we want the neural network to keep the integrity of moving objects. Some violations of these e.g. C^0 everywhere and C^1 almost everywhere may be acceptable given the tolerance of convolutional networks to various local transformations.

Equirectangular Projection For 360° video, equirectangular projection is the most popular way to map a spherical surface to a 2D rectilinear plane. It can keep both C^0 and C^1 continuity in the optical flow map except for the boundary areas, which can be addressed by the circular padding scheme of CNNs. However, it is not conformal and introduces large distortions due to excessively sampling the top and bottom of the sphere. When using equirectangular frame pairs and optical flow maps to train the baseline model, we further modify the loss terms to compensate for distortion by applying solid angle weights onto their pixel-wise loss terms. The solid angle weight is generated by projecting the spherical grid θ and ϕ to the corresponding x, y, and z at the unit sphere presented in the Cartesian coordinate system. The pixel position we calculated is represented as the center pixel P_c. Then, we generate its adjacent right and downside pixel position P_r and P_d. Next, we calculate the area covered by each pixel S by:

$$P = (P_r - P_c) \times (P_d - P_c),\ S = \sqrt{P_x^2 + P_y^2 + P_z^2} \tag{2}$$

The covered area is used as the pixel-wise weight term in the loss function to reduce the effect caused by oversampling in training and testing.

Tri-Cylindrical Projection Unlike equirectangular projection, cylindrical projections preserve angle but are not equal-area. We use the Mercator projection to transform our equirectangular images to cylindrical images by converting the coordinate (θ_e, ϕ_e) to cylindrical coordinate (θ_c, h_c) by:

$$\theta_e = \theta_c, \quad \phi_e = 2\tan^{-1}(e^l) - \frac{\pi}{2} \tag{3}$$

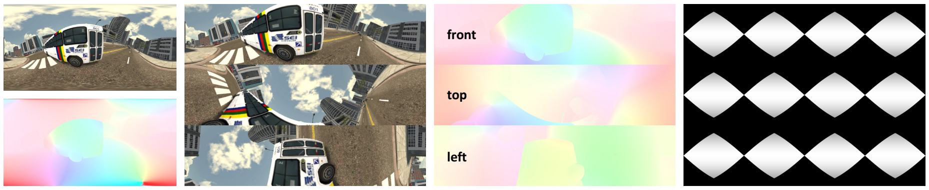
Here, l is a value related to latitude in cylindrical coordinate. The inverse conversion between the two coordinates can be done using the following formula:

$$\theta_c = \theta_e, \quad h_c = \ln(\tan(\phi_c) + \sec(\phi_c)) \tag{4}$$

For a given cylindrical coordinate (θ_c, h_c), we use its corresponding equirectangular pixel to find the target pixel using its flow vector and convert the target pixel to cylindrical coordinates by Eq. 4 for obtaining the cylindrical optical flow. Since the Mercator projection generates a much larger area for the regions of high latitude, causing an even more serious distortion and oversampling problem, we just convert a limited vertical field of view to cylinders where the area change is less extreme (90 degrees in our experiment). To ensure that every part of the spherical surface has an equal contribution, we propose to stack three cylindrical projection images together, where each cylinder to project is aligned with one of the X, Y, and Z-axis, as shown in Fig. 2.

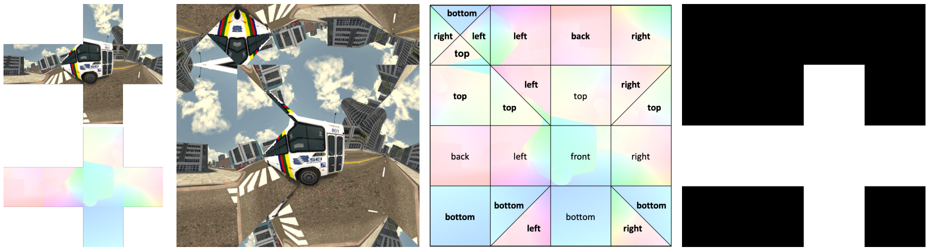
To ensure a fair comparison, the solid angle weights for all the pixels are again generated using the same approach as in the equirectangular image. However, as we stack some pixels three times in the same image, we have to ensure every pixel is calculated only once. We use the distances between the spherical pixels and the equator of each cylinder to decide which view to use for that point. Intuitively, if a pixel is closer to one equator of a cylinder, the distortion should be less, leading to a more accurate result. After applying such a mask, the normalized weight map for our tri-cylindrical projection is shown in Fig. 2.

Cube-Padding Projection Cube maps are another common representation of 360° images, with the advantages of using perspective images to represent a panoramic image. Within each face of the cube, the projection is angle-preserving but not area preserving, and is suitable for the pre-trained model to be further



(a) ERP and optical-flow (b) Tri-cylindrical projection (c) Tri-cylindrical optical-flow (d) Tri-cylindrical weight map

Fig. 2. A tri-cylindrical projection example. Note that different projections provide similar appearance with perspective images for different regions. The weight map is masked after applying solid angle weights to ensure every pixel is just calculated once.



(a) Cube-map projection and optical flow (b) Cube-padding projection (c) Cube-padding optical-flow (d) Cube-padding weight map

Fig. 3. The 360° image and its optical flow projected by our cube-padding approach and the weight map applied on the loss functions.

finetuned. A drawback of the original cube map is that adjacency information is lost if the zero values fill the space out of the cross formed by the six connected cube faces. To ensure global C^0 continuity of the optical flow, i.e., connecting the outer boundary of the six faces to their neighboring faces in the cube, we repeatedly pad the faces to stitch the six faces into one picture, keeping the spatial continuity based on the original cross layout.

The layout of the faces is shown in Fig. 3, where some faces are rotated and trimmed to ensure the C^0 continuity along the boundaries. The drawback of this projection is that the optical flow can not keep C^1 continuity because the optical flow is projected onto image planes of different cameras, leading to the direction changes along the face boundaries. Thus, although using this kind of projection to finetune the baseline model can give a certain level of improvement, it does not outperform the equirectangular-based model in our later experiments. We use perspective projections to convert the spherical pixels to cube map faces for producing cube-padding projection results.

4.2 Fusion from Multiple Projections

We observe that the models trained by different projections provide complementary results in different regions. The equirectangular projection model can better predict the optical flow of pixels around the equator. The cylindrical projection

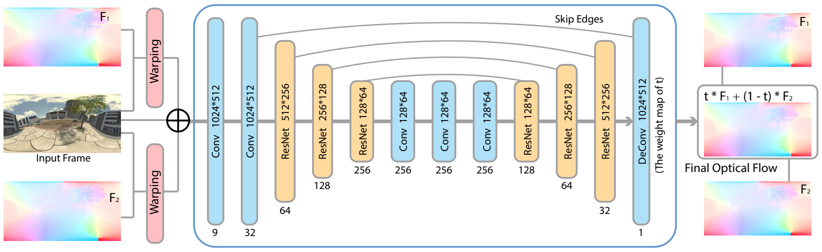


Fig. 4. Fusion network structure.

Model	FlowNet2	PWC-Net	PWC-Sph	RAFT-12	RAFT-24	RAFT-48
EPE	4.15	**3.17**	3.80	5.12	4.93	4.72

Table 1. Performance on City100UR using the finetuned models.

model has advantages at the top and bottom regions using the cylinders aligned with X and Z axis. The cube-padding projection can produce high-quality results besides all the boundary regions of the cube faces. Here we use error heatmaps to visualize different error distributions in Fig. 5. The minimum EPE value presented here shows that the prediction can be improved if appropriately blended with these estimations.

To utilize the advantages of different projections, we develop a fusion network to blend the result. The fusion network is a U-Net [33] with ResNet layers [9] to learn a confidence parameter t. The model structure is shown in the following table. The ResEncoder and ResDecoder are based on the ResNet structure shown is the figure 4. In addition, we take the Mish [26] as our activation function. The learnt confidence parameter is used to blend two models' predictions by: $P_{final} = t * P_a + (1 - t) * P_b$, where a and b represent the predictions using different projection methods. More specifically, the prediction results are converted to equirectangular frame representations before fed to the network. Then the input frame will be concatenated with the warped frames by the two predicted optical flow maps before entering the convolutional layers. From the experiment results, fusing any two of the three models based on different projections can generate significantly better results than using an individual projection, showing the effectiveness of our fusion scheme for 360° optical flow estimation.

5 Experiments

We implement our method by utilizing state-of-the-art optical flow estimating models. In the preliminary experiments, we fine-tuned three state-of-the-art neural network structures: FlowNet2 [14], PWC [37], and RAFT [40] using our datasets. Our experiment found that the PWC structure with only supervising the final output layer can give us the best performance on our test datasets.

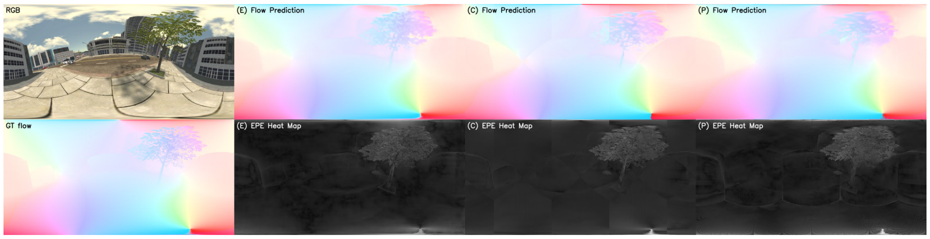


Fig. 5. EPE maps for different projections: (E)-Equirectangular, (C)-Cylindrical, and (P)-Cube-Padding. Here, the EPE values are normalized for visualization.

Thus, we take the PWC structure for our backbone model. Our experiment result for different models is shown in Table 1. We trained PWC-Net using different loss terms and tried applying the normally used solid-angle weights [49] to find out the best loss function as our backbone model. Finally, we chose the top-layer loss without solid-angle weights as the best approach (see our supplementary materials). We also tried modifying the PWC-Net architecture by changing their convolutional kernels to spherical kernels [6] to adapt to the equirectangular distortion. However, as the pre-trained weights are achieved using the regular convolutional kernel shape, the perceptual field changes led to unsatisfactory finetuning results as shown in Tab. 1.

5.1 Implementation

For the data pre-processing and the projection conversion, we implement OpenCL programs to convert the equirectangular image into cylindrical projection and the planar padding format, respectively. Our model utilizes the PWC structure as the backbone, while the correlation layer is from the FlowNet2's implementation. Our model size is 9.4MB, and the parameter number is 2.46M. All the components have an O(N) complexity. The running time of each component on an RTX3090 is reported in Tab. 2. Note that our implementations of the projection operations can satisfy real-time requirements.

Data scheduling Panoramic videos contain irregular movements of objects and cameras, making the scene too complex to analyze. The experiment presented in FlowNet2 shows that training the network with simple features can help the learning process [14]. Thus, we fine-tune models with City1000UR where the camera does not rotate about the x and z axis during data generation.

Fusion network with extra parameters We tested two versions of fusion layers. The first design takes the predictions P_e, P_c and the source images I_s.

Component	PWC-Net	Fusion Model	Projection-E2C	Projection-E2P
Time (ms)	25.6	25.9	8.8	6.8

Table 2. Running time of each component and projection.

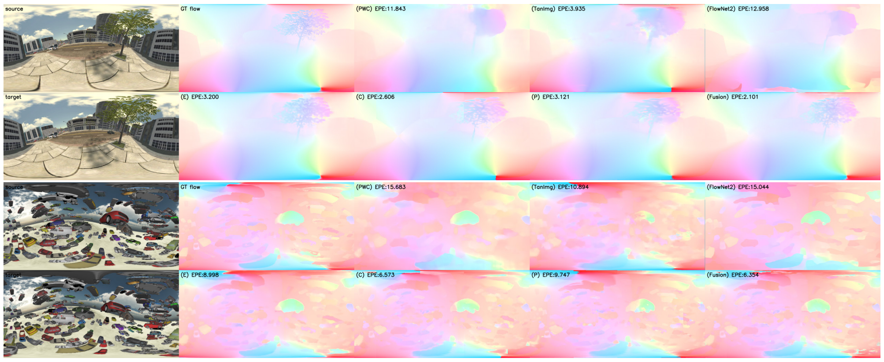


Fig. 6. Optical flow estimation results using PWC, tangent image-based method (TanImg), FlowNet2, single projections, and our final fusion model.

SD/EPE	Equi (E)	Pad (P)	Cyl (C)	C+P	E+P	E+C
City100	0.90/2.41	0.92/3.32	1.34/2.64	0.82/2.15	**0.75**/1.95	0.83/**1.79**
EFT100	2.98/6.65	2.98/6.25	2.82/5.2	2.77/5.06	2.95/5.84	**2.64**/**5.01**
Average	1.94/4.91	1.95/5.68	2.08/5.06	1.80/3.60	1.85/3.89	**1.74**/**3.40**

Table 3. Spherical Distance (SD) / End point errors (EPE) using equirectangular (Equi), cube-padding (Pad), cylindrical (Cyl) projections and their fusion models.

The other version also utilizes two channels of the latitude and longitude θ and ϕ of each pixel to enrich the information the network takes. Due to the possible overfitting to the spatial position, the second model has a higher EPE (2.65 v.s. 2.61). Thus, we discard the spatial position channels in our final network.

5.2 Fusion from Different Projections

The fusion module takes two branches of different projection results. For convenience when training the networks, we firstly trained models for equirectangular, cylindrical, and planar cube-padding, respectively. Thus, we can utilize these models to generate their estimation results and then warp all the flow estimations into the equirectangular projection for training the fusion network. The visualized optical flow results are shown in Fig. 6 and our supplementary materials. To quantitatively measure the quality of 360° optical flows, EPE may not be the best metric, especially for those high-altitude points. We added another metric spherical distance (SD) that is the geodesic distance on a sphere (also known as great-circle or orthodromic distance) between a point's ground truth and predicted optical flow. In Tab. 3, we report the SDs and EPEs of all the projection methods and their fusion models. Optical flow results of the models trained by the data with individual projections are all lower than the fusion model where they are involved. Note that equirectangular projection is the best single projection method, given its global continuity and subtle distortions for regions close

SD/EPE	E+C		E+P		C+P	
	CITY100	EFT100	CITY100	EFT100	CITY100	EFT100
Lower Bound	0.62/1.31	1.83/3.16	0.58/1.44	1.88/3.50	0.68/1.68	1.88/3.34
Upper Bound	1.62/3.74	3.96/8.69	1.23/4.29	4.08/9.40	1.58/4.27	3.91/8.10
Fusion Error	0.83/1.79	2.64/5.01	0.75/1.95	2.95/5.84	0.82/2.15	2.77/5.06

Table 4. Error bounds and the errors of our fused results (SD/EPE).

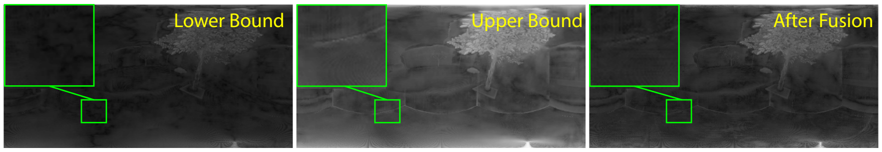


Fig. 7. Error maps. Green boxes show some fixed errors.

to the equator. The EPE is 13.7% lower than the tri-cylindrical projection and 3.1% lower than the cube-padding projection. The best model is the fusion model from equirectangular and tri-cylindrical projections, with a 5.7% lower EPE than the second-best model (E+P). The reason is that the cylindrical projection on the cylinders aligned with the z and x-axis can provide more complementary information to the equirectangular projection to improve the prediction quality further. We can also see that even without the best single equirectangular projection, the model trained by fusing tri-cylindrical and cube-padding projection can also get a significantly better result than equirectangular with a 26.6% lower EPE, demonstrating that our fusion strategy is effective. Note that *fused models always outperform* the associated single projection models, which indicates the importance of our fusion principle. To further validate the effectiveness of our fusion model, we calculate the lower/upper bound for fusion using an oracle to choose the worst/best flow rather than learning fusion for all our test data. We show in Tab. 4 and Fig. 7 the lower/upper bound for fusion and ours is closer to the lower bound, so fusion performs well, but further improvements are possible.

In our multi-projection process, equirectangular projection contains the entire scene and provides contextual information that can be learned by our PWC-based network. Although the other two projections change the layout of the scene, we maintain useful local contexts for objects by duplicating some content. From our results, we can see that for any given object, PWC's context network can still gather useful contextual information for that object since most pixels in the surrounding region are not crossing atlas chart boundaries. Also, the coarse-to-fine scheme of PWC works as long as there is relevant information in the local contextual area in the receptive field of the pyramid. Equirectangular projection always incorporates a full context in a coarse-to-fine scheme due to cyclic convolution operations. For the other two projections, PWC's context network with six levels of pyramids has a receptive field that is still relatively local and does not consider the entire image compared to our resolution of 1000 pixels wide, and is thus applicable to incorporating contextual information.

SD/EPE	PWC-Net	RAFT	TanImg	FlowNet2	Ours
City100	3.86/9.84	12.10/21.57	1.27/3.69	2.72/10.85	**0.82/1.79**
EFT100	6.26/15.64	20.21/29.64	4.61/8.06	4.91/14.88	**2.64/5.01**
Average	5.06/12.74	16.15/25.61	2.94/5.88	3.81/12.87	**1.73/3.40**

Table 5. Comparison with other methods using Spherical Distance (SD) / EPE.

5.3 Comparisons

Compare with 2D optical flow estimation methods We compare our method with the state-of-the-art optical flow estimation methods, FlowNet2, PWC-Net, RAFT, and the method delicately designed for 360° optical flow estimation [46]. We feed equirectangular frames to all the compared methods. The results are shown in Tab. 5 and Fig. 6. Here, the FlowNet2, PWC, and RAFT methods are all using their original method and model to demonstrate the benefits from both of our dataset and multi-projection fusion approach. The SD/EPE of our method is **65.8%/73.3%**, **89.3%/86.7%**, and **54.6%/73.5%** lower than PWC, RAFT, FlowNet2 respectively.

Comparison with tangent image-based approach [46] Although the tangent image-based method (TanImg) achieved a large margin of improvement on the panoramic optical flow quality in their own dataset of static objects with moving cameras, our SD and EPE are **41.1%** and **42.1%** lower than TanImg, respectively, as in Tab. 5. This shows that their method cannot adapt to the scenes with dynamic objects because they do not use a deep model to learn the useful cues for matching local textures of complex moving objects. We also choose the "circle" set of data in [46] to finetune our model since it is the only set with more than 600 frames (the other two sets both have only 180 frames). We use a 2:1 training and testing split. On the test set, we have an EPE of 4.73 if we directly use the original frames for training and an EPE of 3.8 if we use data augmentation using RGBShift and ChannelShuffle from Albumentations [2] to create a larger diversity. The result from our data-augmented model is slightly worse than their method (EPE: 3.5) when training by such a small scale dataset. But the significant improvement with only color-based data augmentation indicates that the amount of data is *not sufficient* for training a deep model, and it has a great potential to improve given more data.

5.4 Application for Video Editing

An accurate optical flow estimation is important for many tasks such as object tracking and video segmentation on 360° videos. To demonstrate the accuracy and the practical value of our method for real-world panoramic videos, we develop an application that utilizes the predicted optical flow to propagate edits across frames to evaluate our method visually. In our example shown in Fig. 8, we put a piece of graffiti at one frame and applied optical flow estimation to find

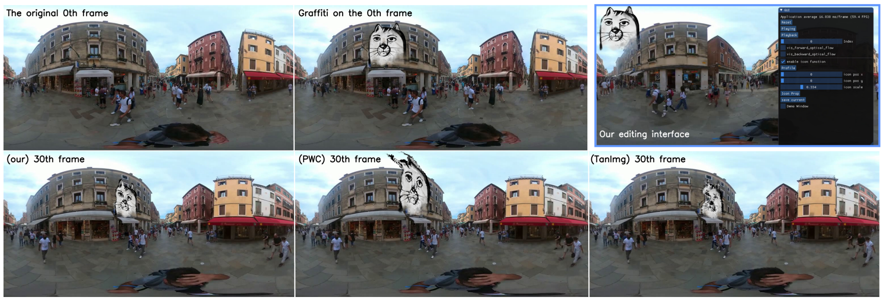


Fig. 8. Our video editing application on real-world 360° videos. The results are generated using optical flow results of our method, PWC, and TanImg respectively.

the corresponding affected pixels on the rest of the video frames. More specifically, we use the backward optical-flow to find whether a pixel in the following frames is affected, so that the edits can be propagated densely. We use different optical flow predictions from the PWC model, tangent image-based method, and our method for comparisons. The optical flow estimated by PWC fails to match the distorted corresponding pixels in the upper region. TanImg method is better, but it accumulates errors when the frame number is increasing, leading to obvious artifacts after 30 frames. In contrast, our method well adapts to the distortions of panoramic video representation, generating much more consistent editing results across all the frames. See our supplementary for the full video.

Limitation and Future Work Our dataset does not contain the real-world optical flow data. In the future, the 360° Lidar can be used for ground truth optical flow generation. We also consider adding modules to cope with occlusion regions and extending the work to 360° scene flow estimation for VR applications.

6 Conclusion

This paper presented a novel multi-projection fusion framework for 360° optical flow estimation. Besides the commonly used equirectangular projection, we proposed tri-cylindrical projection and circular cube-padding projection to provide complementary less distorted 2D panoramic representations to train the deep model. A fusion network is then applied to combine the warped frames based on the predicted panoramic optical flow using different projections for the final optical flow estimation. A novel dataset was also generated to enable the training of neural networks for 360° optical flow estimation. The experimental results demonstrate that our method has the best performance and is sufficiently practical to be applied to real-world use cases.

Acknowledgements This work is supported by Marsden Fund Council managed by Royal Society of New Zealand (No. MFP-20-VUW-180).

References

1. Bhandari, K., Zong, Z., Yan, Y.: Revisiting optical flow estimation in 360 videos. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8196–8203. IEEE (2021) 4
2. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: Fast and flexible image augmentations. Information **11**(2) (2020). https://doi.org/10.3390/info11020125, https://www.mdpi.com/2078-2489/11/2/125 13
3. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: A. Fitzgibbon et al. (Eds.) (ed.) European Conf. on Computer Vision (ECCV). pp. 611–625. Part IV, LNCS 7577, Springer-Verlag (Oct 2012) 3
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015) 5
5. Cheng, H.T., Chao, C.H., Dong, J.D., Wen, H.K., Liu, T.L., Sun, M.: Cube padding for weakly-supervised saliency prediction in 360 videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1420–1429 (2018) 4
6. Coors, B., Condurache, A.P., Geiger, A.: Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 518–533 (2018) 2, 4, 6, 10
7. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015), http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15 2, 3, 5
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 3, 5
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 9
10. Hilbert, D., Cohn-Vossen, S.: Geometry and the Imagination (2nd ed.). Chelsea (1952) 1
11. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981) 2
12. Hu, H.N., Lin, Y.C., Liu, M.Y., Cheng, H.T., Chang, Y.J., Sun, M.: Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1396–1405. IEEE (2017) 4
13. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5754–5763 (2019) 3
14. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017), http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17 2, 3, 6, 9, 10

15. Kang, K., Cho, S.: Interactive and automatic navigation for 360° video playback. ACM Trans. Graph. **38**(4) (jul 2019) 4
16. Kopf, J.: 360 video stabilization. ACM Transactions on Graphics (TOG) **35**(6), 1–9 (2016) 3
17. Li, C., Xu, M., Jiang, L., Zhang, S., Tao, X.: Viewport proposal cnn for 360° video quality assessment. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10169–10178. IEEE (2019) 4
18. Liu, L., Zhang, J., He, R., Liu, Y., Wang, Y., Tai, Y., Luo, D., Wang, C., Li, J., Huang, F.: Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6489–6498 (2020) 3
19. Liu, P., Lyu, M., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4571–4580 (2019) 3
20. Liu, S., Tan, P., Yuan, L., Sun, J., Zeng, B.: Meshflow: Minimum latency online video stabilization. In: European Conference on Computer Vision. pp. 800–815. Springer (2016) 2
21. Liu, S., Yuan, L., Tan, P., Sun, J.: Steadyflow: Spatially smooth optical flow for video stabilization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014) 2
22. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2. pp. 674–679. IJCAI'81, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1981) 2
23. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. CoRR **abs/1512.02134** (2015), http://arxiv.org/abs/1512.02134 3, 4, 5
24. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018) 3
25. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 3
26. Misra, D.: Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681 (2019) 9
27. Monroy, R., Lutz, S., Chalasani, T., Smolic, A.: Salnet360: Saliency maps for omnidirectional images with cnn. Signal Processing: Image Communication **69**, 26–34 (2018) 3
28. Nadeem, S., Su, Z., Zeng, W., Kaufman, A., Gu, X.: Spherical parameterization balancing angle and area distortions. IEEE transactions on visualization and computer graphics **23**(6), 1663–1676 (2016) 1
29. Nguyen, C., DiVerdi, S., Hertzmann, A., Liu, F.: Vremiere: In-headset virtual reality video editing. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. pp. 5428–5438 (2017) 4
30. Poranne, R., Tarini, M., Huber, S., Panozzo, D., Sorkine-Hornung, O.: Autocuts: simultaneous distortion and cut optimization for uv mapping. ACM Transactions on Graphics (TOG) **36**(6), 1–11 (2017) 1
31. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017) 2

32. Ren, Z., Gallo, O., Sun, D., Yang, M.H., Sudderth, E.B., Kautz, J.: A fusion approach for multi-frame optical flow estimation. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2077–2086. IEEE (2019) 3
33. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 9
34. Sterzentsenko, V., Saroglou, L., Chatzitofis, A., Thermos, S., Zioulis, N., Doumanoglou, A., Zarpalas, D., Daras, P.: Self-supervised deep depth denoising. In: ICCV (2019) 4
35. Su, Y., Grauman, K.: Kernel transformer networks for compact spherical convolution. CoRR **abs/1812.03115** (2018), http://arxiv.org/abs/1812.03115 4
36. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 2432–2439. IEEE (2010) 2
37. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume (2018) 2, 3, 6, 9
38. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE transactions on pattern analysis and machine intelligence **42**(6), 1408–1423 (2019) 3
39. Tang, C., Wang, O., Liu, F., Tan, P.: Joint stabilization and direction of 360° videos. ACM Transactions on Graphics (TOG) **38**(2), 1–13 (2019) 3
40. Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. CoRR **abs/2003.12039** (2020), https://arxiv.org/abs/2003.12039 2, 3, 6, 9
41. Wang, N.H., Solarte, B., Tsai, Y.H., Chiu, W.C., Sun, M.: 360sd-net: 360 stereo depth estimation with learnable cost volume. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 582–588. IEEE (2020) 3
42. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4884–4893 (2018) 3
43. Wulff, J., Butler, D.J., Stanley, G.B., Black, M.J.: Lessons and insights from creating a synthetic optical flow benchmark. In: A. Fusiello et al. (Eds.) (ed.) ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation. pp. 168–177. Part II, LNCS 7584, Springer-Verlag (Oct 2012) 3
44. Yang, J., Liu, T., Jiang, B., Lu, W., Meng, Q.: Panoramic video quality assessment based on non-local spherical cnn. IEEE Transactions on Multimedia **23**, 797–809 (2021). https://doi.org/10.1109/TMM.2020.2990075 4
45. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1983–1992 (2018) 3
46. Yuan, M., Richardt, C.: 360-degree optical flow using tangent images. arXiv preprint arXiv:2112.14331 (2021) 3, 13
47. Zhang, Y., Zhang, F.L., Lai, Y.K., Zhu, Z.: Efficient propagation of sparse edits on 360 panoramas. Computers and Graphics **96**, 61–70 (2021) 3
48. Zhang, Z., Xu, Y., Yu, J., Gao, S.: Saliency detection in 360 videos. In: Proceedings of the European conference on computer vision (ECCV). pp. 488–503 (2018) 3
49. Zhao, J., Chalmers, A., Rhee, T.: Adaptive light estimation using dynamic filtering for diverse lighting conditions. IEEE Transactions on Visualization and Computer Graphics **27**(11), 4097–4106 (2021) 10