Cloud Prediction Based on the Combination of Optical Flow and Deep Learning

Peng Muzi(✉), Zhao Kanglian, Dai Zheng, and Li Wenfeng

NanJing University, 163 Xianlin Street, Qixia Distirct,
Nanjing 210023, Jiangsu, China

Abstract. Satellite-to-ground laser communication has a problem of being susceptible to specific atmospheric environments, which will attenuate the laser transmission signal severely. To solve this problem, we have to know important prior information about whether the construction of a specific laser communication link is suitable. In this paper, in order to predict future images of cloud clusters around the laser links in advance, we propose a cloud prediction model based on the combination of optical flow and deep learning. Our model is based on Deep Voxel Flow (DVF), an end-to-end CNN designed for video frame synthesis. The 3D optical flow vector across space and time in the input cloud images is used to form an intermediate layer in DVF. By using DVF for multiple times to iterate the input cloud images at t second and t + 25 s, we can get the predicted cloud images during the next 100 s. Our experimental results show that, compared to the optical flow extrapolation method which is a typical method used for nowcast, our cloud prediction model can predict future cloud images with higher quality and accuracy.

Keywords: Laser communication · Cloud prediction model · Deep learning

1 Introduction

Laser communication is a kind of wireless communication which uses optical signal as the carrier to transmit information directly in the atmosphere. With the advantages of large communication capacity and strong confidentiality, laser communication has a wide application potential in satellite-to-ground communication [1].

However, satellite-to-ground laser communication has a problem of being susceptible to specific atmospheric environments [2]. Compared to satellite-to-satellite laser links, satellite-to-ground laser links have to pass through the atmosphere, during which the laser transmission signal would be easily affected by atmospheric environments such as clouds, fog, haze and so on. These complex atmospheric environments will attenuate the laser transmission signal severely, and even cause communication interruption. As a result, in satellite-to-ground laser communication, we must take the influence of different atmospheric environments into consideration.

In this case, if the condition of cloud clusters around the laser links can be obtained in advance, we will get to know important prior information about whether the

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 432–444, 2022.
https://doi.org/10.1007/978-3-030-93398-2_41

construction of a specific laser communication link is suitable and we will be able to predict the quality of the link, thus guaranteeing the stability of the uninterrupted satellite-to-ground laser communication. Such a task of extrapolating future cloud condition from the past trends of cloud change can be termed as cloud prediction.

2 Related Work

In weather forecast domain, radar echo image extrapolation [3] is a main approach of nowcast. Many typical methods of radar echo extrapolation have been proposed, including the centroid tracking method, the cross-correlation extrapolation method and the optical flow extrapolation method. The core idea of all these three typical methods is to find the corresponding relationship between the frames at the adjacent time.

The centroid tracking method tries to identify the monomers, and then scans the images at the adjacent time to match and track the target monomers [4]. However, this algorithm is only suitable for images with easily identified target monomers. As the target monomers on most of our cloud images are complex and hard to be identified, the centroid tracking method has limitations on our cloud prediction.

The cross-correlation extrapolation method [5] first divides the image region into several small regions, and then calculates the correlation coefficient between the small regions at the adjacent time. The corresponding relationship of the regions at the adjacent time is determined by the maximum correlation coefficient. However, the cross-correlation extrapolation method often fails when the image motion is fast. As the motion of cloud tends to be fast, this method also has limitations on our cloud prediction.

The optical flow extrapolation method makes use of the optical flow to track the image motion. Optical flow is the instantaneous velocity field of a moving object. It can be used to calculate the next position of the target point. It has been proved that the optical flow method has advantages over the other two methods above. Gunnar Farneback's algorithm [6] is used to calculate dense optical flow—the optical flow of all pixels in the image is calculated. Farneback optical flow method is a gradient based method. In this method, the image gradient is assumed to be constant and the local optical flow is assumed to be constant. Farneback optical flow method is a very suitable optical flow extrapolation method for weather radar echo extrapolation at present. However, the optical flow extrapolation method also has limitations on our cloud prediction. The optical flow method requires the image to follow the assumption of gray invariance, while the brightness of the target point is constantly changing because the actual cloud change with time is often accompanied by generation, development, weakening and dissipation. As a result, when the moving speed of clouds is fast and the time interval is long, the prediction error rate can still be large.

3 Our Cloud Prediction Model

In this paper, we decide to improve the optical flow extrapolation method by combining it with convolutional neural networks (CNN), which is one of the most representative networks in deep learning used for tasks of prediction. Our cloud prediction model is based on Deep Voxel Flow (DVF) [7]. DVF is an end-to-end CNN designed for video frame synthesis. Compared to optical flow extrapolation method and simple CNN-based model without using the optical flow [8], DVF can synthesize the next video frame with higher quality and accuracy with the input of former two consecutive frames. The structure of DVF is composed of two parts. The first part is the convolutional encoder-decoder used to predict voxel flow, as is show in Fig. 1. And the second part of DVF is the volume sampling layer used to synthesize the predicted frame by bilinear interpolation, with the predicted voxel flow and the previous two frames, as is shown in Fig. 2.

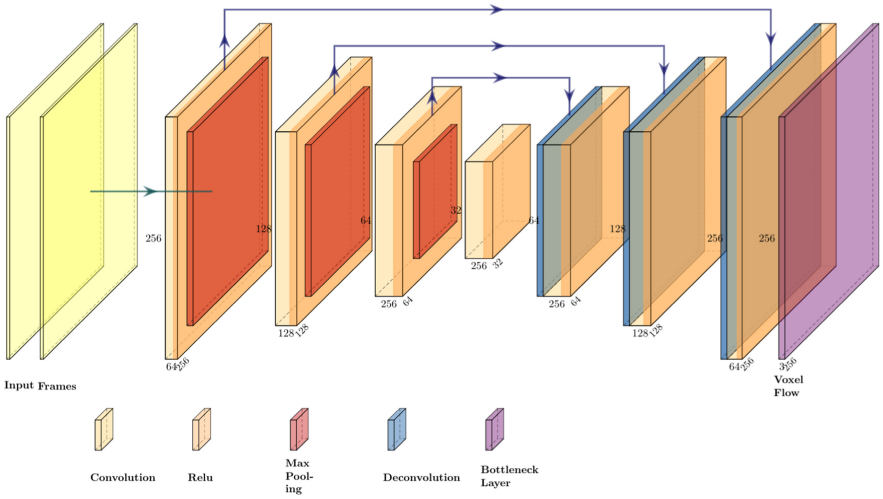

Fig. 1. The convolutional encoder-decoder in DVF

Instead of using CNN to predict and output the optical flow itself as FlowNet [9]do, the voxel flow vector across space and time in the input cloud images is used to form an intermediate layer in DVF, which means the correctness of the optical prediction will never be directly tested. The output of DVF is the predicted frame and we only have to directly consider the correctness of the predicted frame. Because of the superiority of DVF in the video prediction task, we adopt DVF to our prediction model. The structure of our cloud prediction model is shown in Fig. 3. By using DVF for multiple times to iterate the input cloud images at t second and t + 25 s, we can get the predicted cloud images during the next 100 s.

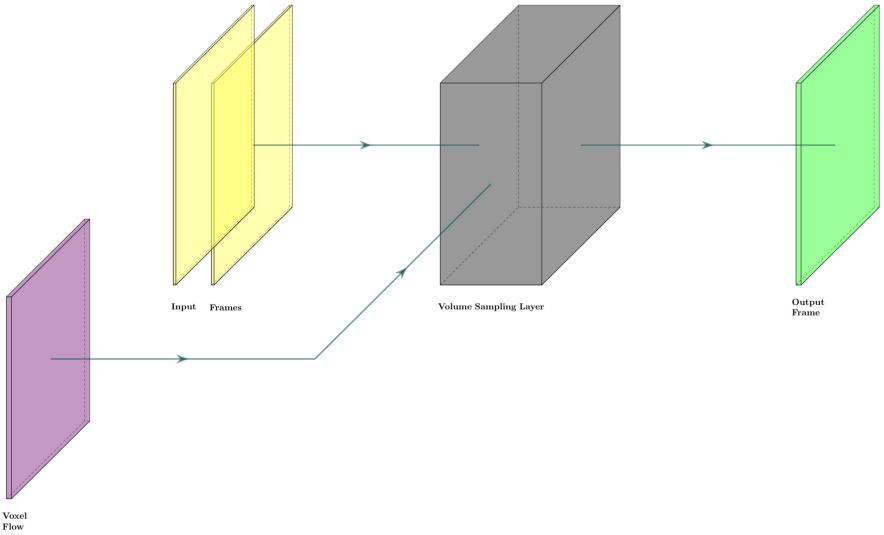

Fig. 2. The volume sampling layer in DVF

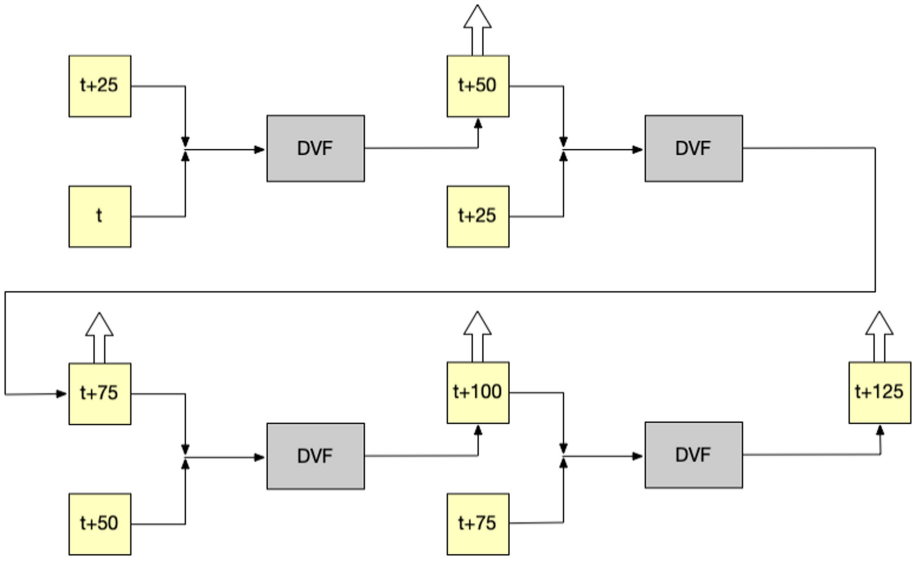

Fig. 3. Our cloud prediction model

3.1 Voxel Flow

The optical flow is a 2D vector describing instantaneous velocity of a point. Accordingly, the optical flow field is a 2D vector field, which reflects the gray change trend of each point in the image, and can be regarded as the instantaneous velocity field

generated by the pixel points with gray level moving on the image plane. In a word, the information contained in the 2D vector field is the instantaneous velocity vector information of each image point, and such information can be termed as spatial information.

However, the spatial information is not enough to measure spatiotemporal sequences, such as the video frames and our cloud images. Spatiotemporal sequences not only have the spatial information, but also have the temporal information. The task of spatiotemporal sequence prediction has to take both spatial information and temporal information into consideration.

The voxel flow adds the temporal dimension to the original two-dimensional optical flow. It is a per-pixel 3D vector across space and time. The voxel flow field is on a 2D grid of integer target pixel location. Let the two video frames as the input of DVF be $\mathbf{X} \in \mathbb{R}^{H \times W \times 2}$, and the predicted video frame be $\mathbf{Y} \in \mathbb{R}^{H \times W}$, where H and W are the height and width of the frame, then the voxel flow field $\mathbf{F}$ can be expressed as $\mathbf{F} = (\Delta x, \Delta y, \Delta t)$. The first and second dimension of $\mathbf{F}$ represents the optical flow from the target frame to the next frame. It can be understood as the spatial component of the voxel flow. Especially, the optical flow is assumed to be locally linear and temporally symmetric between two consecutive frames. In this case, we will be able to find the location of a target pixel point in previous two frames. Let the coordinates of the target pixel point be (x, y), then we can get its coordinates in previous two frames as $(x - 2\Delta x, y - 2\Delta y)$ and $(x - \Delta x, y - \Delta y)$. Moreover, based on the assumption of local linearity and temporal symmetry between the two consecutive frames, the temporal component of voxel flow F is a linear blend weight between the previous two frames. In computer vision, this temporal component is called mask. When we use the selected image to cover the processed image to control the region of image processing. The selected image is called a mask. The mask is a binary image composed of 0 and 1. When a mask is applied to a processed image, the 1-value region is processed, while the 0-value region which is covered will not be processed. The volume sampling layer which uses the voxel flow to synthesize the predicted frame by trilinear interpolation will help us to better understand the function of the mask. This will be discussed in Sect. 3.3.

In conclusion, the voxel flow is composed of the spatial information F_{motion} and the temporal information F_{mask}. To reflect the concept of the voxel flow more vividly, we visualize it on our cloud image dataset, as is shown in Fig. 4. Specifically, we visualize the ground truth cloud images which have different cloud forms and their F_{motion} and F_{mask} which predicted by DVF. Visualization of the optical flow is achieved by using flow field color coding [10]. In this method, flow direction is encoded with color and magnitude is encoded with color intensity, as is shown in Fig. 5. And we use heatmap to visualize F_{mask}, so the connection between F_{mask} and previous two frames can be vividly showed, as is shown in Fig. 6.

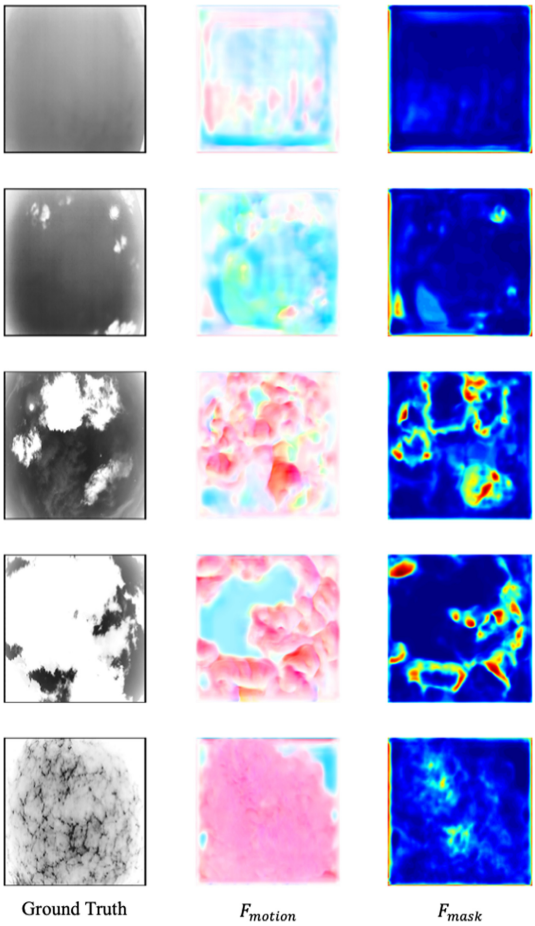

Fig. 4. Visualization of the voxel flow on our cloud image dataset

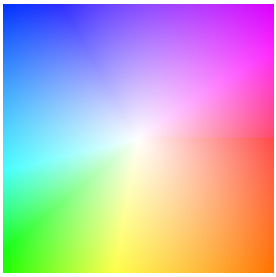

Fig. 5. The flow field color coding method

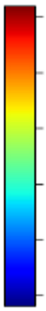

Fig. 6. The connection between F_{mask} and the previous two frames

3.2 Convolutional Encoder-Decoder

The convolutional encoder-decoder architecture is used to predict voxel flow with the input of two previous frames. The structure of this convolutional encoder-decoder is shown in Fig. 1. It is in fact a kind of U-Net [11] architecture. The U-Net was first proposed to solve the problem of medical image segmentation. U-Net is composed of two parts. The first part of U-Net is used for feature extraction and the second part is for up-sampling. As the whole structure of it is like the English letter U, it is termed as U-Net.

The feature extraction part of the network contains four convolution layers and three max pooling layers. We will get four feature maps of different sizes as 256×256, 128×128, 64×64 and 32×32. The convolution kernel sizes are 5×5, 5×5, 3×3 respectively. Then, in the up-sampling part, we first do deconvolution on the 32×32 feature map to get the 64×64 feature map. The 64×64 feature map is concatenated with the previous 64×64 feature map. Such concatenating action will better maintain the spatial information. Then we do convolution and up-sampling on the concatenated feature map to get the 128×128 feature map. And then we do the same thing to get the 256×256 feature map. Finally, through a bottleneck layer, we get the predicted voxel flow, which has a size of $3 \times 256 \times 256$.

3.3 Volume Sampling Layer

The volume sampling layer in DVF is used to synthesize the predicted frame by trilinear interpolation. The structure of it is shown in Fig. 2. The inputs of the volume sampling layer are the voxel flow generated by the convolutional encoder-decoder and the two previous frames, and the output is the predicted frame. The volume sampling function samples colors by interpolating within an optical-flow-aligned video volume computed from input $\mathbf{X}$.

In the paper Video Frame Synthesis using Deep Voxel Flow [7], the author has used mathematical expressions to explain the volume sampling function in detail. Here we use Python code to interpret the function of this volume sampling layer in a brief way, as is shown below. Firstly, we have to use F_{motion} to find the relationship between the output grids and their corresponding locations in the inputs. Then we fill the pixel

value of the corresponding position in the inputs into the output grid. As there are two input frames, so we will get two interpolation results. Finally, we can use the F_{mask} to combine two interpolation results. Specifically, we can multiply the interpolation result of the first frame with *mask* and multiply the interpolation result of the second frame with $(1 - mask)$.

```
coor_x_1 = grid_x – flow[0, :, :] * 2
coor_y_1 = grid_y – flow[1, :, :] * 2
coor_x_2 = grid_x – flow[0, :, :]
coor_y_2 = grid_y – flow[1, :, :]


output_1 = torch.nn.functional.grid_sample(
    input[0:3, :, :],
    torch.stack([coor_x_1, coor_y_1], dim=2),
    padding_mode='border')

output_2 = torch.nn.functional.grid_sample(
    input[3:6, :, :],
    torch.stack([coor_x_2, coor_y_2], dim=2),
    padding_mode='border')


prediction = mask * output_1 + (1.0 - mask) * output_2
```

4 Experiments

4.1 Our Cloud Image Dataset

We use the cloud image dataset collected by our laboratory. Using the infrared imager independently developed, our laboratory collected cloud image sequences over a specific area of Nanjing in March 2019. Cloud images were taken every 5 s and cloud images collected are all gray images with a resolution of 720×480. In the data preprocessing part, we resize them to 256×256.

In this experiment of cloud prediction, we sample the cloud sequences every 5 images, which means the time interval between two cloud images in our sampled image sequences is 25 s. We do this because the cloud motion change is relatively slow when it is compared to the image motion in many life scenes. By lengthening the time interval between sampled cloud images, we can get more obvious cloud motion changes for model training, although this will make it more challenging to train the DVF.

For sampled image sequences in the training set, every 3 consecutive images make up a training data. The first and second image serve as the input of DVF and the third one serves as the label. For sampled image sequences in the testing set, every 6 consecutive images make up a testing data. The first and the second cloud images serve as the input of our cloud prediction model and the following 4 cloud images serve as the ground truth. Finally, we get 42240 data for training and 7848 data for testing.

4.2 Model Training

We use MSE loss as our loss function. And we use Adam as the optimizer. The original learning rate is 0.0001, and we adjust it dynamically in the training process. The batch size is set to be 32.

4.3 Model Evaluation

We use MSE, PSNR, SSIM to evaluate the testing result of our cloud prediction model. These three indexes are the main indexes for image quality evaluation. We compare the results of our cloud prediction model with the optical extrapolating method using Gunnar Farneback's algorithm. Specifically, we make use of the calcOpticalFlowFarneback function in OpenCV library to implement the optical flow extrapolating method.

Average MSE, PSNR and SSIM on the testing set are shown in Table 1. The change trend of these three indices as the prediction time increases is shown in Fig. 7. From the results we can see that our cloud prediction model is better than the optical extrapolating method in all these three indices. As the prediction time step increases, our model keeps a higher prediction accuracy and quality than the optical flow extrapolation method. Moreover, we can see from the line charts that with the increase of the prediction time step, the advantage of our model becomes bigger and bigger compared to the optical flow extrapolation method.

Table 1. Average MSE, PSNR and SSIM on the testing set

Time step	Our cloud prediction model	Optical flow extrapolation method
Average MSE		
t + 50	0.010	0.022
t + 75	0.017	0.041
t + 100	0.024	0.057
t + 125	0.030	0.072
Average PSNR		
t + 50	23.673	20.830
t + 75	20.601	17.097
t + 100	18.801	15.086
t + 125	17.604	13.790
Average SSIM		
t + 50	0.874	0.844
t + 75	0.836	0.776
t + 100	0.814	0.728
t + 125	0.799	0.692

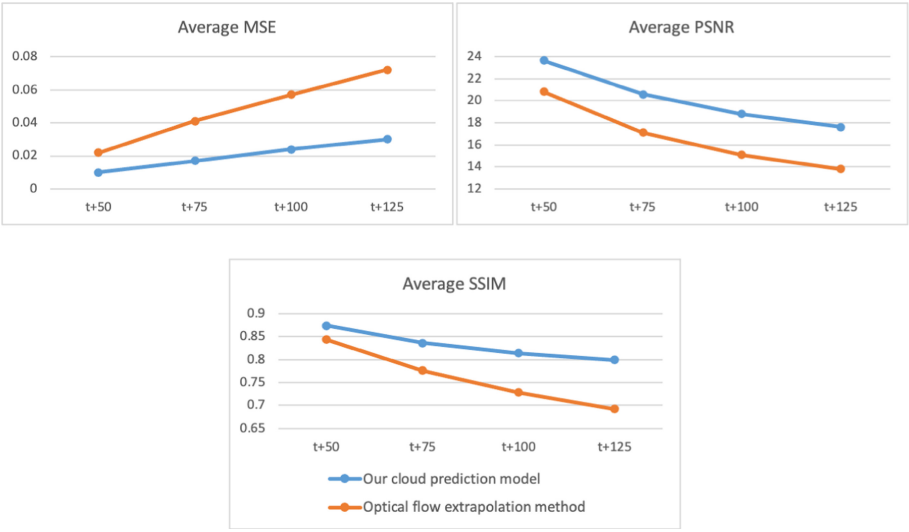

Fig. 7. The change trend of the three indices as the prediction time increases

Then we select prediction results of different cloud condition in the testing set and visualize them, as is shown in Fig. 8. In fact, the classification of cloud is a complex task. When a cloud is classified, cloud height, external characteristics, formation process and other principles are all needed to be considered. Here we mainly list several typical cloud conditions according to the amount and density of cloud.

As is shown in Fig. 8, prediction by our cloud prediction model is in the line of Prediction1, prediction by the optical flow extrapolation is in the line of Prediction2. Ground truth is in the third line. Also, to better show the difference between the prediction and ground truth, we visualize the difference images between the prediction result and ground truth in the fourth and fifth line. Difference1 is the difference between Prediction1 and the ground truth, and Difference2 is the difference between Prediction2 and the ground truth.

According to the visualized testing results, we can intuitively observe that our cloud prediction model can predict the location of cloud clusters with higher quality and accuracy than the optical flow extrapolation method. In particular, when the distribution of clouds is dense and messy, our cloud prediction model is obviously better than the optical flow extrapolation method. Our cloud prediction model is more accurate in the estimation of cloud location. Also, our model does better in the prediction of the edge information of future cloud images. In contrast, the optical flow extrapolation method brings much distortion. The position of the cloud cluster sometimes deviated greatly. Moreover, the optical flow tracking sometimes fails especially when the cloud condition changes rapidly, so we can see that there are obvious black spots on some cloud images predicted by the optical flow extrapolation method.

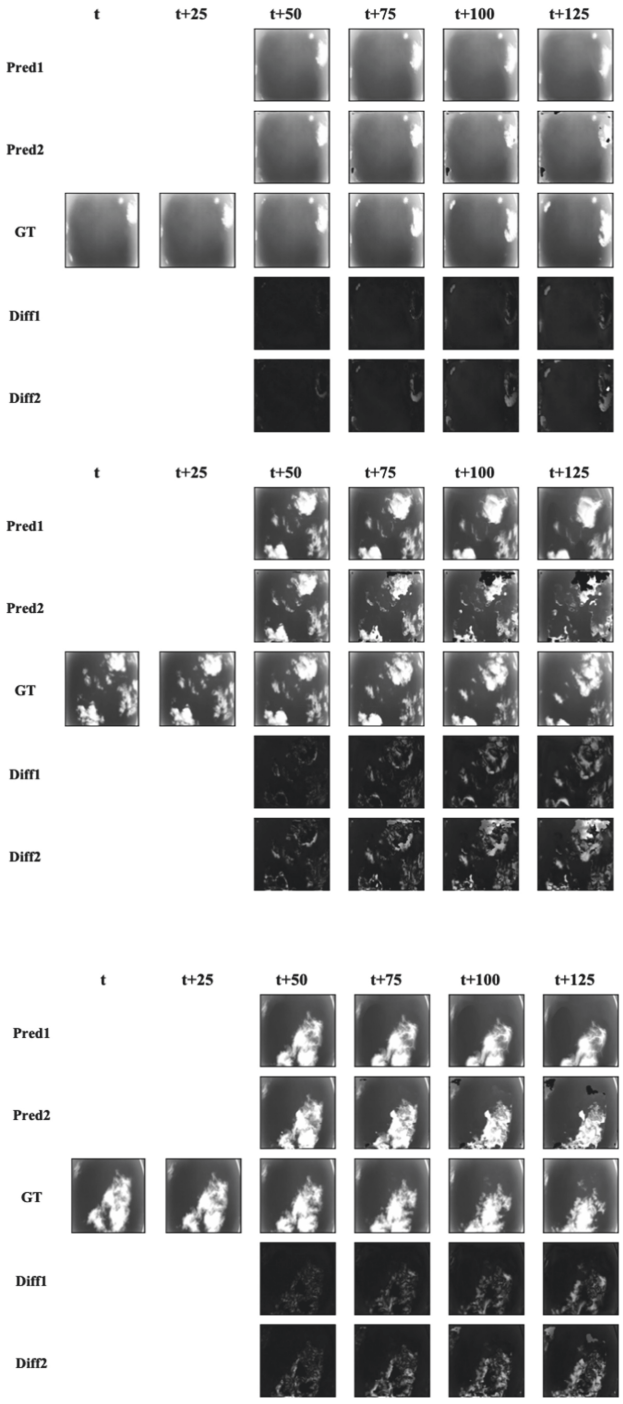

Fig. 8. Visualization of the testing results

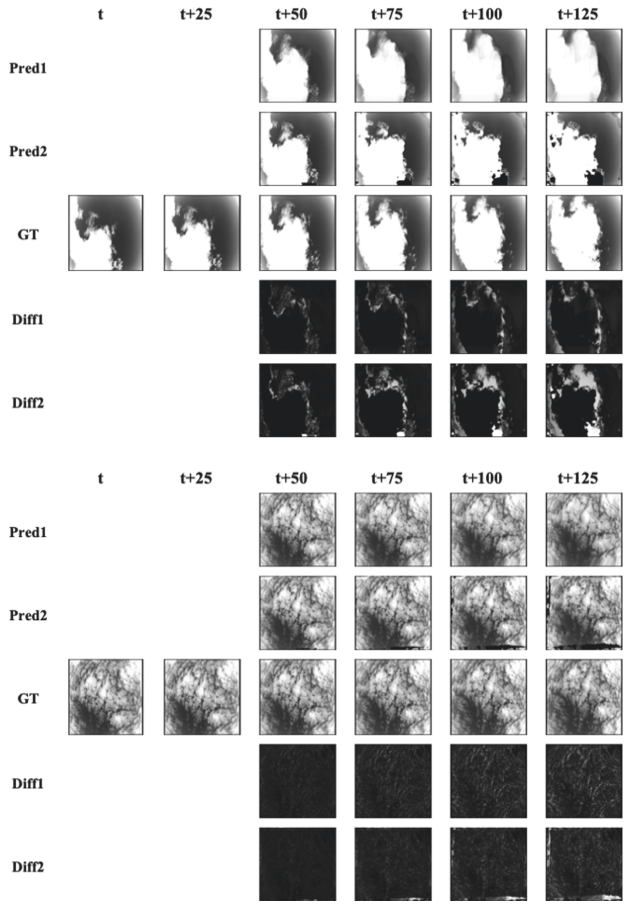

Fig. 8. (*continued*)

5 Conclusion and Future Work

In this paper, in order to deal with the problem of satellite-to-ground laser communication being susceptible to specific atmospheric environments, we propose a cloud prediction model based on the combination of optical flow and deep learning to predict future cloud images. Our model is based on Deep Voxel Flow (DVF), a CNN structure designed for video synthesis. By using DVF for multiple times to iterate the input cloud images at t second and t + 25 s, we can predict cloud images during the next 100 s. We train and test our cloud prediction model on the cloud image dataset collected by our laboratory. Our experimental results show that, compared to the optical flow extrapolation method which is a typical method used for nowcast, our cloud prediction model can predict future cloud images with higher quality and accuracy.

For future work, firstly we consider adding a cloud image pre-classification model. As the cloud conditions are complex, actually it will be challenging to train a single

model to apply to all types of cloud condition. As a result, it will be sensible to add a cloud image pre-classification model and then train the cloud prediction model for different types of cloud conditions respectively. Moreover, we will do more research to improve the structure of the present cloud prediction model. We expect that our model can predict cloud images for a longer time span and with a higher accuracy in the future.

Acknowledgement. This work is supported by the 13th Five-Year Civil Aerospace Technology Pre Research Project, the Fundamental Research Funds for the Central Univesities under Grant 021014380187 and the National Natural Sciences Foundation of China under Grant 62131012.

References

1. Arnon, S., Kopeika, N.S.: Laser satellite communication network-vibration effect and possible solutions. Proc. IEEE **85**(10), 1646–1661 (1997)
2. Ricklin, J.C., et al.: Atmospheric channel effects on free-space laser communication. J. Opt. Fiber Commun. Rep. (2006)
3. Austin, K.: Nowcasting precipitation—a proposal for a way forward. J. Hydrol. (2000)
4. Lei, H.: Review on Development of Radar-based Storm Identification, Tracking and Forecasting. Meteorological Monthly (2007)
5. Hamill, T.M., Nehrkorn, T.: A short-term cloud forecast scheme using cross correlations. Weather Forecast. **8**(4), 401–411 (1993)
6. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) Image Analysis. SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/3-540-45103-X_50
7. Liu, Z., et al.: Video frame synthesis using deep voxel flow. IEEE (2017)
8. Goodfellow, I.J., et al.: Generative Adversarial Nets. MIT Press, Cambridge (2014)
9. Fischer, P., et al.: FlowNet: learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE (2016)
10. Butler, D.J., et al.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision – ECCV 2012. ECCV 2012. LNCS, vol. 7577, pp. 611–625. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_44
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28