ISBN 978-981-14-1455-8
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
WCSE_2019_SPRING
Yangon, Myanmar, February 27-March 1, 2019, pp. 17-21
doi:10.18178/wcse.2019.03.004

View Synthesis from Silhouette Using Deep Convolutional Neural Network

Samer JAMMAL[1], Tammam TILLO[2+], and Jimin XIAO[3]
[1]Liverpool University, UK
[2]University of Bozen-Bolzano, Italy
[3]X'ian Jiaotong Liverpool University, China

Abstract. Multiview video could be the basis to support various applications, such as Three-Dimensional video (3DV), Virtual Reality (VR), and Free Viewpoint Video (FVV). Multiview data is intrinsically redundant, in fact, the semantic contents of different views are almost similar, and this is especially true for small baseline views. Obviously, wide baseline views might significantly differ in their contents, where some objects might be completely absent in some views. However, the current approaches of representing multiview data, even if they exploit inter-view correlation, require large bandwidth for transmission. This bandwidth is almost linear with the number of transmitted views. Thus, in this paper we propose to address this problem by representing lateral views solely using their edges, while dropping their texture content. The texture content is synthesized, at the receiver side, by a convolutional neural network (CNN) exploiting the edges and the information in the central view. The edges of the lateral views represent the location of the "objects" in their corresponding views, whereas, we assume that their texture in other views does not change significantly, consequently there is no need to represent them in the lateral views. In this work, in addition to the proposed paradigm of representing multiview data, we also propose a training framework for the CNN network. Experimental results demonstrate the effectiveness of the proposed framework and demonstrate that the network is able to synthesis accurate and reliable lateral views starting from their edges.

Keywords: View synthesis, silhouette, edge map, stereo video, convolutional neural network.

1. Introduction

Multiview video technologies have attracted attention in recent years as it either provides the depth perception of 3D scene, or it allows the viewer to observe a scene from different viewpoints and angles such as Free Viewpoint Video (FVV) [1] where a viewer is not restricted to be in a particular position. The main problem for multiview videos applications is that a large number of different views, with a small baseline, are recorded by multiple synchronized cameras. In fact, increasing the number of recording cameras provides users with a more realistic 3D viewing experience. On the other hand, this imposes a huge load on the storage, compression, and transmission of multiview video data. A solution is to use view-interpolation to generate these additional views which reduces the transmission bitrate, such systems would transmit a limited set of viewpoint videos. At the receiver side, any required intermediate viewpoints will be generated by view synthesis techniques.

View synthesis techniques are tools used for rendering new views from existing views. However, view synthesis is a challenging task, in particular for wide baseline views, where synthesized views might be significantly different from existing views. Several approaches have been introduced to solve this problem and to improve the accuracy and reduce the computational cost.

Recently, deep learning based methods have been utilized in novel view synthesis. Ji et al. [2] present a CNN based method. First, it rectifies the two input view images and uses the rectified images to estimate

+ Corresponding author. Tel.: + 39 0471 016 026; fax: +39 0471 016 009.
E-mail address: ttillo@unibz.it.

homograph by deep networks, and then generate intermediate views using another deep networks. The recent work by Flynn et al. [3] uses a deep CNN to learn the geometry of a scene to synthesize an intermediate view of a scene from a set of input images. This method can produce good quality views for small baseline and has difficulties for large baseline and moving objects. The work in [4] tries to synthesize stereo image pairs from a single input image. It just generates the corresponding right image from the left image. Srinivasan et al. [5] train a network to reconstruct the light field images from one single image.

In contrast, to previous works the aim of this paper is to present an accurate, fast, practical method for the synthesis of novel view for both small and large baselines. In this work, we represent the required views using their edges, while dropping their texture content. The texture content gets synthesized by a Convolutional Neural Network (CNN) from the central view. Generating the edges of images is more reliable than generating the disparity map, and most importantly, compressing the edge images does not induce geometric distortion to the reconstructed scene. In fact, it is worth noting that compression-induced-distortion on disparity (or depth map) causes geometric displacement which might results in false occlusion or dis-occlusions, with a severe impact on the perceived quality of the synthesized view. On the other hand, the compression of the edge map will avoid such problem. A deep CNN will be employed to exploit the edges of the lateral view with the texture in the central view and synthesize the new view. The used network was trained end-to-end without a post-processing phase. Meanwhile, the proposed method is fast with 0.06 s foreword run time.

2. Proposed View Synthesis Network

It is speculated that the used deep convolutional neural network learns the geometry structure between the central view and the edges of the lateral views. The edge information is used to facilitate the matching between the central and lateral edge map to generate a realistic rendering of the lateral view.

2.1. Network Architecture

We introduce a Silhouette based View Synthesis framework (SVSNet), which uses the architecture of DispNet [6]. Given a central view and the edge map of a lateral view, the goal of the proposed framework is to synthesis the lateral view. The inputs of the used network include an edge map and the central view.

In particular, the used network includes a contracting part that progressively decreases the spatial size of the convolutional features, providing large receptive fields for higher-level convolutional layers, which in turn, enables the network to capture more global information. The large receptive field is an important aspect of the used network, as it allows it to deal with large objects displacements between the two views. The texture content of the lateral view gets synthesized by the network using the texture from the central view. Thus, the edges of the lateral views represent the location of the "objects" in their corresponding views. However, the pooling in the contracting part reduces the resolution of the synthesized view. Consequentially, an upsampling part gradually and nonlinearly expands the feature maps (using a set of upconvolutional layers), taking into account the features from the contractive part.

The main goal of the proposed framework is to in-paint and fills the edge map of the lateral view based on the central view. In other words, while view synthesis needs precise per-pixel localization, it also requires finding correspondences between the texture view and the edge map. This involves learning to match the feature generated from the central view with the feature generated from the edge map at different locations, and for various objects and displacements in the two images. This matching process is achieved in the correlation layer of DispNet between two feature blocks of size $1 \times 1 \times 128$. The first block is copied from the left feature map for a given centre position of correlation, while the second block is copied from the second feature with a displacement d, and an element-wise multiplication between the two blocks is conducted. The obtained vector is then summed and filled in the d-th feature map that represents the matching score for displacement d.

The network is end-to-end trained using a dataset that includes the lateral edge map, texture central view and ground truth view (label). The goal is to train the DispNet using the Mean Squared Error (MSE) as the loss function. The network weights are initialized with from a random Gaussian distribution with zero mean.

3. Experimental Results

The validity and effectiveness of the proposed approach were checked on several datasets, and several comparisons were conducted in this section.

3.1. Computer Graphic Training Set

The network is trained using the FlyingThings3D synthetic training set [6]. This dataset contains 22,740 stereo images for training, and 4,760 stereo images for testing. Each image contains 5-20 everyday objects flying in the scene. The parameters of the objects (size, type, positions, texture, and rotation) are randomly sampled, which makes this dataset suitable for training a deep network.

To do the training each training dataset was pre-processed to generate the silhouette images of the right views, thus, each training instance is composed of the left texture image, the silhouette image extracted from the right texture images (using Canny Edge detector), and the ground truth is the right texture image.

In the fine-tuning phase, the network is trained using the MPI Sintel dataset. Sintel is a synthetic dataset derived from a short open source animated 3D movie. It contains sufficiently dense and realistic scenes including natural image degradations such as fog and motion blur. This makes the dataset a very reliable training set for CNNs. This dataset contains 1,064 stereo images for training, which makes it suitable for fine-tuning the network.

Initially, the training was performed using patches of size 768×384 which were cropped from the original images of size 960×540 from the FlyingThings3D training set, with a batch size of 4 per iteration. During the experiment, we found that the training loss exhibits a considerable amount of noise as a result of the small batch size employed in the training process. This noise can be reduced by using a larger batch size. Increasing the batch size increases the demand on memory. Training the network with a batch size of 8 images takes 3 days to run 200k iterations using one Titan X GPU.

3.2. Framework Evaluation

The proposed approach was evaluated on both real-world dataset KITTI 2015 and synthetic datasets including FlyingThings3D which contains 4760 frames for testing, Sintel dataset which contains 1064 image pairs, Driving dataset with 4392 frames, and Monkaa dataset with 8591 frames.

Fig. 1 shows that proposed method learned to efficiently exploit the central view texture information to generate the texture of the lateral view based on its edges. In general, it could be observed that the network was able to synthesis the lateral view keeping small details and texture content for both small and large displacement and the occluded areas. For visual results, Fig. 2 depicts some examples using the proposed network for out-door scenes KITTI 2015 where illumination and edges are more complex. The results shows that the synthesized view is robust and accurate for both close objects with large displacements and far objects with smaller displacements.

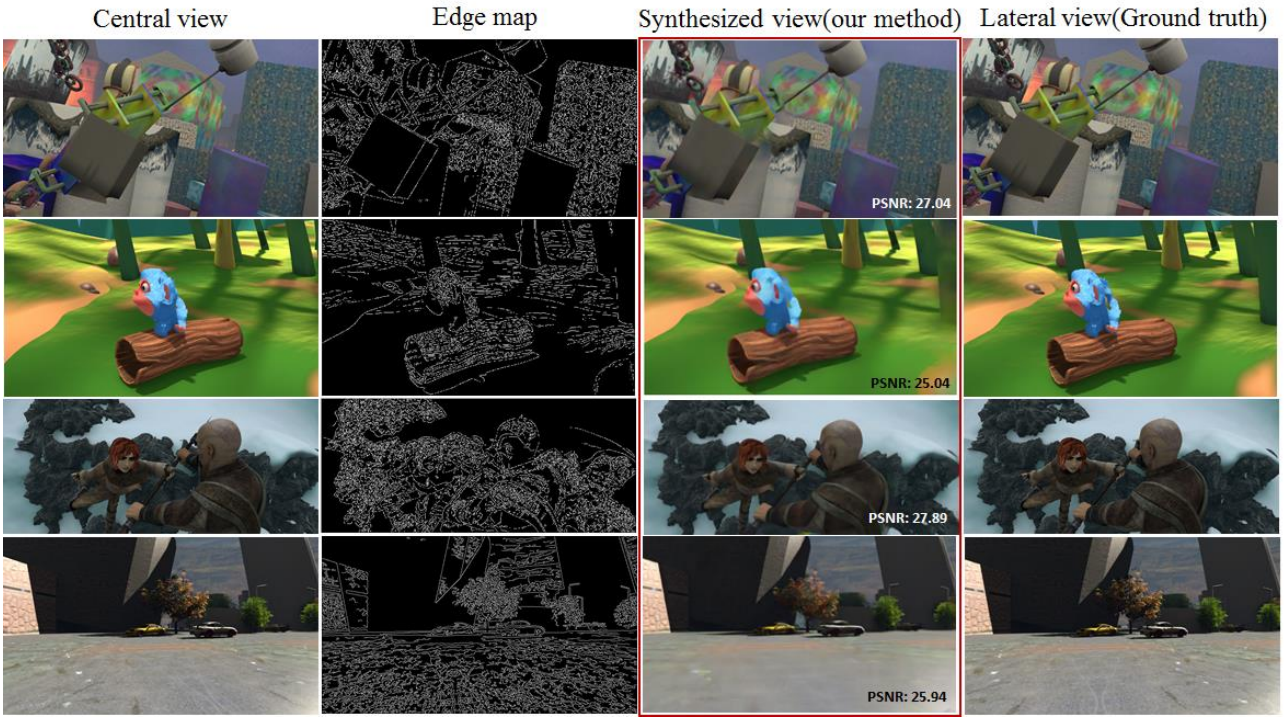

Fig. 1: Examples of view synthesis results from silhouette using SVSNet (proposed framework) on computer graphic datasets. Each column from left to right: central view (input 1), lateral edge map (input 2), synthesized view and the ground truth image. Rows from top to down: FlyingThings3D (clean), Monkaa and Sintel (clean).

Fig. 2: Examples of view synthesis results from silhouette using SVSNet (proposed framework) on real outdoor scenes KITTI 2015 dataset. Each column from left to right: central view (input 1), synthesized view and the ground truth label.

3.3. Various Edge Detector Thresholds

This paper proposes to address the view synthesis problem by representing lateral views solely using their edges, while dropping their texture content. Thus, edge information are crucial for the correct work of the proposed approach. Therefore, it is normal to expect, that the more accurate the edge information the better the obtained results, however, the richer in details the silhouette map the more bit rate is required to transmit this map. In this section, the impact of different amount of extracted edge was investigated, by using various edge detector thresholds on the quality of view synthesis from silhouette.

The first experiment was conducted using the automatic threshold of Canny's method. This threshold provides rich edge map with details about texture objects, this information allow the proposed framework to provide accurate results, with fine details especially at the objects' boundaries. The second set of experiments uses various thresholds related to the automatically generated threshold. This approach allows analyzing the effect of the threshold value while using the same mechanism to generate the threshold. The experimental results using these thresholds values: *Auto-threshold* $\times \{1, 2, 3, 4\}$ are reported in Table 1. As it is expected, this table demonstrates that using rich edge map helps to synthesize more accurate views, where the objects are clearly distinguished, whereas, using less complex edge maps reduces the quality of the rendered view. It is worth reporting that for the Driving dataset, which has naturalistic scenery with streets from driver viewpoint (this dataset is made to resemble the KITTI datasets), the *Auto-threshold* $\times 1$ did not provide the best performance.

Table. 1: PSNR results for the proposed view synthesis from silhouette approach, using different edge detector thresholds (Canny Edge detector) on Driving, FlyingThings3D test, Monkaa and Sintel datasets.

Threshold	Driving	Flyingthings3D	Monkaa	Sintel
Auto-threshold	19.61 dB	**25.13** dB	**23.11** dB	**21.89** dB
Auto-threshold×2	19.72 dB	24.50 dB	22.67 dB	19.16 dB
Auto-threshold×3	19.68 dB	24.35 dB	22.38 dB	20.60 dB
Auto-threshold×4	**19.76** dB	24.03 dB	21.38 dB	20.26 dB

4. Conclusions

In this paper we proposed silhouette-based view synthesis framework by representing lateral views using their edges, while dropping their texture content. These texture contents get synthesized by a CNN exploiting the edges and the information in the central view. The edges of the lateral views represent the location of the "objects" in their corresponding views. Moreover, in this work only information from neighbour's views is exploited, however for multiview videos applications further performance improvement can be expected by exploiting temporal information from other frames. We believe that this is an interesting direction for future research.

5. References

[1] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto. 2009. View Generation with 3D Warping Using Depth Information for FTV. Image Commun. 24, 1-2 (Jan.2009), 65–72.

[2] D. Ji, J. Kwon, M. McFarland, and Silvio Savarese. 2017. Deep View Morphing. CoRR abs/1703.02168 (2017).

[3] N. K. Kalantari, T. Wang, and R. Ramamoorthi. 2016. Learning-Based View Synthesis for Light Field Cameras. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016) 35, 6 (2016).

[4] J. Xie, R. B. Girshick, and A. Farhadi. 2016. Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks. CoRR abs/1604.03650 (2016).

[5] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. 2017. Learning to Synthesize a 4D RGBD Light Field from a Single Image. CoRR abs/1708.03292 (2017).

[6] N.Mayer, E.Ilg, P.H ̈ausser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).