University of Kentucky
UKnowledge

Computer Science Faculty Publications

Computer Science

6-2020

Interactive Free-Viewpoint Video Generation

Yanru Wang
Nanjing University, China

Zhihao Huang
Nanjing University, China

Hao Zhu
Nanjing University, China

Wei Li
Peking University, China

Xun Cao
Nanjing University, China

See next page for additional authors

Repository Citation

Wang, Yanru; Huang, Zhihao; Zhu, Hao; Li, Wei; Cao, Xun; and Yang, Ruigang, "Interactive Free-Viewpoint Video Generation" (2020). *Computer Science Faculty Publications*. 36.
https://uknowledge.uky.edu/cs_facpub/36

•Article•

Interactive free-viewpoint video generation

Yanru WANG[1], Zhihao HUANG[1], Hao ZHU[1*], Wei LI[2], Xun CAO[1], Ruigang YANG[3*]

1. *School of Electronic Science and Engineering, Nanjing University, Nanjing* 210023, *China*

2. *Center on Frontiers of Computing Studies, Peking University, Beijing* 100871, *China*

3. *Department of Computer Science, University of Kentucky, Lexington, KY* 40506, *USA*

*** Corresponding author,** zhuhaoese@nju.edu.cn; r.yang@uky.edu

Received: 25 February 2020 **Accepted:** 7 April 2020

Supported by USDA 2018−67021−27416; NSFJS BK20190292.

Citation: Yanru WANG, Zhihao HUANG, Hao ZHU, Wei LI, Xun CAO, Ruigang YANG. Interactive free-viewpoint video generation. Virtual Reality & Intelligent Hardware, 2020, 2(3): 247—260
DOI: 10.1016/j.vrih.2020.04.004

Abstract **Background** Free-viewpoint video (FVV) is processed video content in which viewers can freely select the viewing position and angle. FVV delivers an improved visual experience and can also help synthesize special effects and virtual reality content. In this paper, a complete FVV system is proposed to interactively control the viewpoints of video relay programs through multimedia terminals such as computers and tablets. **Methods** The hardware of the FVV generation system is a set of synchronously controlled cameras, and the software generates videos in novel viewpoints from the captured video using view interpolation. The interactive interface is designed to visualize the generated video in novel viewpoints and enable the viewpoint to be changed interactively. **Results** Experiments show that our system can synthesize plausible videos in intermediate viewpoints with a view range of up to 180°.

Keywords Free-viewpoint video; View interpolation; Interactive interface

1 Introduction

In recent years, with the rapid development and enhancement of visual computing technologies and the television and internet video markets, free-viewpoint videos (FVVs) have gained in popularity. These allow viewers to freely select the viewing position and angle[1]. Compared to traditional video, FVV delivers more 3D information and a sense of stereoscopic viewing, which significantly improves the visual experience. FVV can help synthesize special effects (such as "bullet time"), and it can be readily transformed into virtual reality (VR) assets. Therefore, an interactive system to generate FVV has a large utility value.

Generating FVVs requires a multi-viewpoint system (consisting of a camera array) to capture video simultaneously from multiple viewpoints. One major problem in generating FVV from captured multi-viewpoint video is synthesizing the video in the intermediate viewpoints, a process referred to as "view interpolation." Initial studies reconstructed a 3D model of the whole scene from multi-viewpoint images[2,3] and then rendered the model to yield FVV[4–6]. In later works, image-based rendering (IBR) techniques were proposed to render novel viewpoints directly from input images[7–15]. With the rapid development of deep learning methods, numerous works[16–18] have demonstrated that neural networks can greatly improve the speed and accuracy of FVV generation, compared to traditional approaches.

In this paper, we propose a complete system whereby a user can interactively control the viewpoints of video relay programs through multimedia terminals such as computers and tablets. The system comprises a hardware setup, an FVV-generation algorithm, and an interactive interface design. The core module of the software is the novel-viewpoint generation algorithm, through which the number of the available viewpoints is dramatically increased using a neural network. With the proposed FVV-generation system, users can control the viewing angle of the video program and switch smoothly between perspectives. In our experiment, the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) result of our system outperform the traditional 3D-rendering method. The processing frame-rate achieved is 30fps with a resolution of up to 720P, which is sufficiently visually pleasant for applications. Our system is tested in various environments, such as basketball stadiums and indoor scenes.

2 Related works

The key process in generating FVV is that of synthesizing images in a novel viewpoint from one or more reference images. Existing methods for synthesizing images/videos from novel viewpoints can be divided into two categories: traditional 3D-rendering methods and image-synthesis methods.

Traditional 3D-rendering methods can be further divided into model-based, depth-based, and image-based methods[19]. In initial studies, researchers explicitly modeled the scene or object into 3D structures[4–6,15,20–26], with the aim of recovering the geometric information to render the novel perspective. Although these methods were successful with sufficient input images, they were unable to recover the desired target viewpoint with a limited quantity of images, due to the ambiguity of 3D models. Subsequently, researchers focused on image-based rendering. IBR techniques render novel viewpoints directly from input images[7]. IBR typically uses proxy geometry to synthesize viewpoints. Initial works using IBR methods considered plenoptic modeling[8], light field rendering[9], super-pixel segmentation[12], alpha matting[17] and depth-based rendering[27]. A recent work[28] employed more than two reference perspectives, to obtain more color/depth information and interpolate the novel views for wide-baseline camera arrays. Several recent works[9,10,13,29] have used the IBR method to obtain reasonably high-quality synthesis results, employing powerful deep learning techniques to predict the depth map[10,14,29], blending weights[11] or multi-plane images[15].

Image-synthesis methods usually employ an end-to-end framework. For example, GAN networks[30,31] directly synthesize images in the novel viewpoint from narrow-baseline videos[32–35]. Zhou et al. proposed to sample from input images by predicting the appearance flow between the input and output for both multi-view syntheses and view interpolation[36]. Park et al. further introduced an image-generation network based on the appearance flow-prediction network, to construct the unseen region[37]. Many studies have attempted to solve the decomposition problems of viewpoint synthesis by using multiple networks. For example, Kalantari et al. segmented the viewpoint synthesis process into disparity and color-estimation components, which were solved by two sequential convolutional neural networks[38]. Ji et al. proposed to rectify the two viewpoint images through estimated homography in the first a convolutional network, and then synthesize the images in the middle viewpoint using the second convolutional network[39]. Another strategy[14,40] synthesized the images onto a range of different planes (at different depth levels) and then selected one plane or blended all planes for each pixel at factually different depths.

3 Capturing system

We build a complete system to generate interactive FVVs. It consists of a hardware system to capture multi-viewpoint videos and a software to synthesize videos from novel viewpoints. We detail the capturing system in this section, then explain the algorithm and user interface in the next.

3.1 Camera array

We used a workstation to receive video streams from the camera array (the main component of the hardware system) (Figure 1). Considering the limitations of bandwidth and data-transfer rates, two kinds of setups were adopted to strike a balance between frame-rate and image resolution. To ensure the pixel utilization of captured frames, the cameras are placed in similar horizontal planes.

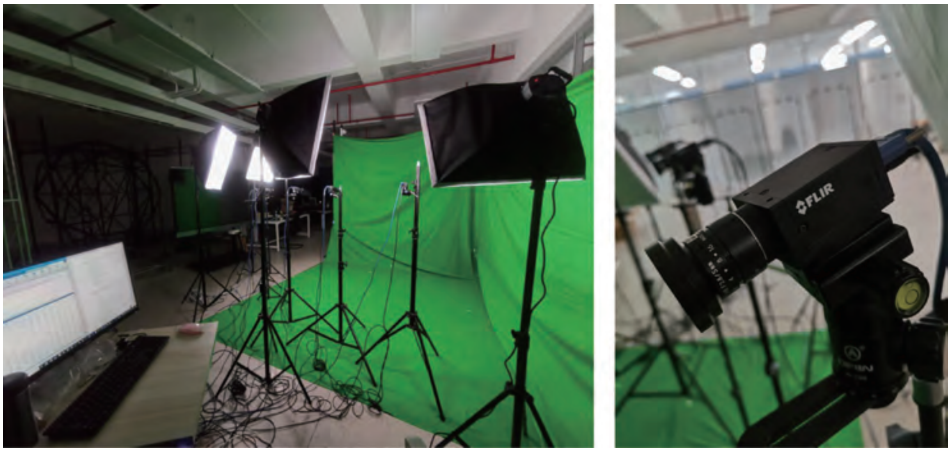
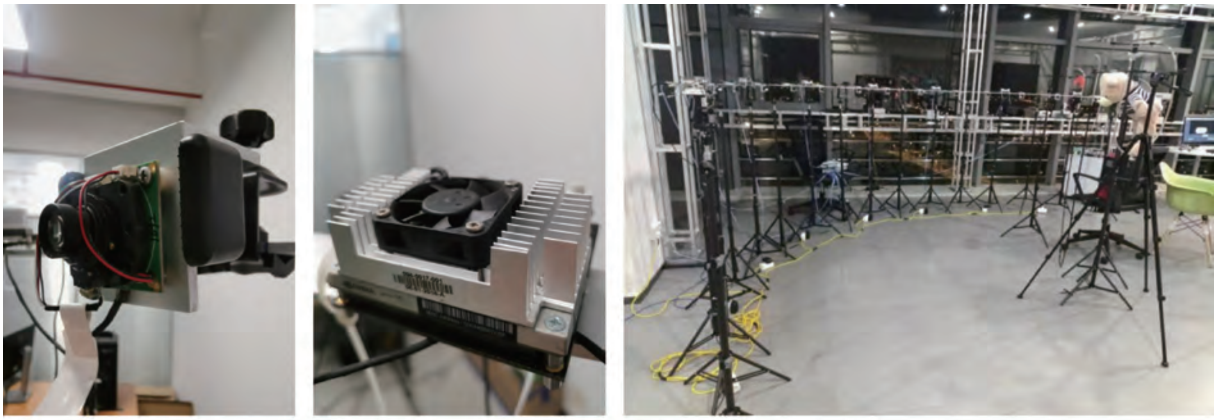


Figure 1 A high-frame-rate, six-camera synchronous acquisition system (frame-rate up to 120fps), the camera unit is an FILR industrial camera, and the system is synchronized through either a software or hardware trigger mode.

The first setup focuses on a high frame-rate capture; it consists of six FILR industrial cameras with a 1280×720 resolution and a 60fps frame rate. This setup supports synchronous data acquisition in both software and hardware trigger modes. The hardware trigger mode is set to send trigger signals to the camera from the synchronous triggering box. We set the trigger frequency to 60ps and the duty ratio to 50%. While the system is also compatible with a software trigger, we preferred to use the hardware trigger because it had a higher synchronous accuracy.

The second setup focused on high-resolution capture; it consisted of 16 cameras using Sony IMX274 CMOS sensors, eight Nvidia Jetson TX1 modules, and a switch (Figure 2). The resolution was set to 3864×2174 and the frame-rate was set to 30fps. This system was synchronized through a software trigger (using the Mantis software developed by Aqueti). The synchronization accuracy meets the requirements with a synchronous error of 10−20ms.



Sony IMX 274 CMOS sensor

Nvidia Jetson TX1

Whole 16 cameras system

Figure 2 High-resolution, 16-camera synchronous acquisition system with a resolution of 4K (3864×2174) and synchronized by software trigger.

3.2 Synchronization

Frame synchronization of the numerous captured videos is critical for the subsequent process of view interpolation. In the six-FILR industrial camera array, all cameras were synchronized using the signal generator via a video-data trigger line, and the trigger signal frequency was set to 60Hz. Regarding the second setup (which used 16 industrial cameras), 2 cameras were connected to each TX1 and the synchronized signal was controlled by the local server, as shown in Figure 3.

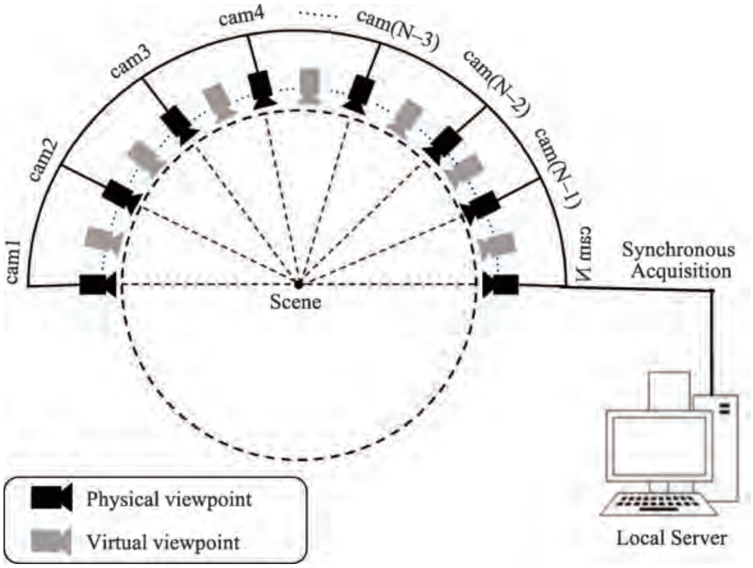


Figure 3 Industrial camera array for acquiring synchronized video stream.

4 Free-viewpoint video generation

With the captured multi-viewpoint video as an input, three modules were used to generate videos in novel viewpoints: data preprocessing, view interpolation, and interactive interface design. We detail each module in the following subsections.

4.1 Data preprocessing

The preprocessing stage aimed to render the input multi-viewpoint videos suitable for the learning-based view-interpolation framework. The first step was post-rectification, which ensured that (a) most of the feature points were aligned horizontally and (b) the viewpoint rotation axis was aligned vertically. A superior alignment can reduce ghost effects in the final view interpolation results. All 16 synchronized frames were extracted from the 16 videos captured by our multi-camera system. A suitable frame was selected as the reference image from the first group of 16 synchronized frames; then, we chose a viewpoint rotation axis for this frame, which was typically taken as the middle vertical line to reduce unnecessary image cutting. The affine transformation matrix is defined as:

$$M = \begin{pmatrix} \alpha & \beta & (1-\alpha)\cdot t_x - \beta\cdot t_y \\ -\beta & \alpha & \beta\cdot t_x + (1-\alpha)\cdot t_y \\ 0 & 0 & 1 \end{pmatrix}, \qquad (1)$$

where $\left(t_x, t_y\right)$ is the translation term, $\alpha = k\cdot cos(\theta)$, $\beta = k\cdot sin(\theta)$, and k and θ are the scale and rotation parameters, respectively.

Images in other views are warped to align with the reference frame; this is achieved by interactively adjusting the parameters t_x, t_y, k, and θ of the affine transformation, as defined in Equation (1). During the adjustment process, the crucial task is to align the vertical viewpoint rotation axis via translation and then ensure that most feature points are aligned horizontally on the same horizontal line. As shown in Figure 4, most feature points are aligned horizontally, and the viewpoint rotation axis is kept stable by adjusting image rotation, zoom, and pan parameters. Blue lines represent the horizontal calibration lines and the green line represents the viewpoint rotation axis, which is typically the vertical line in the middle of the image. This is possible because all cameras are arranged on a plane. In summary, we estimate the affine matrix by optimizing an objective function consisting of two terms, where the first term denotes a horizontal alignment and the second term denotes a vertical alignment. The objective function is formulated as.

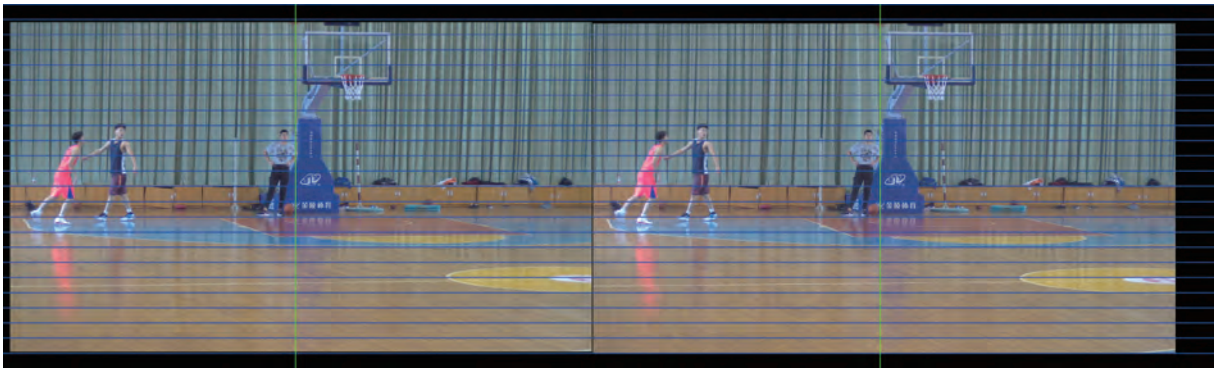


Figure 4 Example of post-rectification.

$$E_{total} = \sum_{p_i \in H}^{i} \left\| p_i(y) - p_{i\_reference}(y) \right\|_2^2 + \alpha \cdot \sum_{p_j \in V}^{j} \left\| p_j(x) - p_{j_{reference}}(x) \right\|_2^2,$$

where H and V are feature point sets on the horizontal calibration lines and vertical calibration line, respectively; and $p(x)$ and $p\left(y\right)$ are the x and y coordinates of the feature points of the images. After we obtained the affine matrix for each view, all remaining frames were processed using the aforementioned affine warping relation.

The second step is color calibration. Although we implemented camera white-balance calibration and color compensation for all cameras, it was still necessary to postprocess the white balance, to ensure that the final FVV maintains strict color consistency when users switch between viewpoints. The gray-world algorithm[41] was used to calibrate color in our system.

4.2 View interpolation

4.2.1 Learning-based method

To design our network, we referred to the work of Niklaus et al., who considered frame interpolation in the time domain. The core idea of their CNN network was to estimate using 2D convolution kernels, which simultaneously consider motion estimation and re-sampling.

As shown in Figure 5, we fed two images of adjacent views into an encoder-decoder network, to interpolate the middle-viewpoint image. To detect large motions between two input images, we increased the convolutional kernel size to 50, which resulted in an over-sized model. To reduce the model size, we introduced a pooling layer to the encoder network and replaced the 2D kernels in the decoder network with two pairs of 1D kernels. All blocks in the network contained three convolution/deconvolution layers and one pooling/up-sampling layer. The tail of the decoder was designed to contain four subnets to predict the

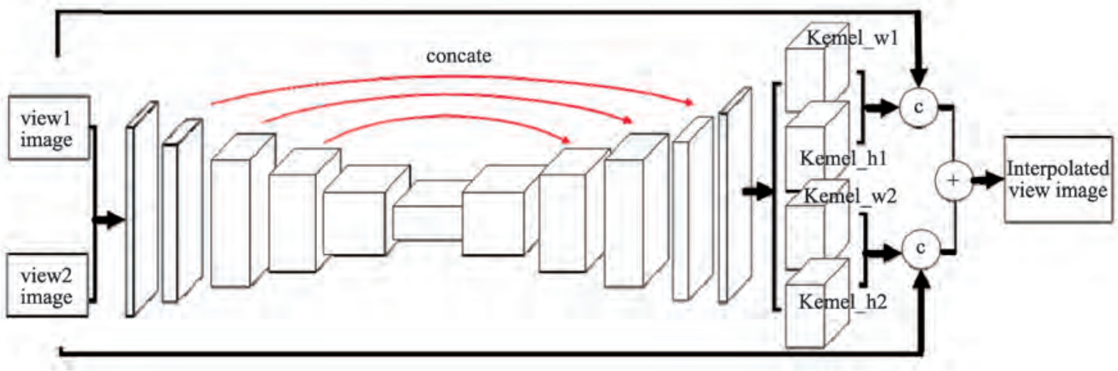


Figure 5 The main structure of our network.

four 1D kernels separately, instead of predicting the entire group of 1D kernels directly. This is because training with the former structure can result in faster convergence. To synthesize the images in novel viewpoints besides the middle one, we ran the network interactively and interpolated for the target viewpoint. The pairs of 1D kernel can be used for depth-wise convolutions and point-wise convolutions, respectively, for the same size of input image; this kind of separable convolution can significantly reduce the model size. To obtain the virtual, interpolated-viewpoint image, the two original input frames are convolved with the 1D kernel pairs and summed, as

$$V_{virtual}(x,y) = K_1(x,y) * P_1(x,y) + K_2(x,y) * P_2(x,y), \quad (2)$$

where $P(x,y)$ indicates a patch centering on (x,y) in the original input image, $K_1(x,y)$ is the output of a subnet, and * is the convolution operation.

There are two terms in the loss function. The first term is the pixel-level loss, defined as the L2-norm of pixel values between the predicted image and ground truth; the second term is the perceptual loss, which is a feature-level loss measuring the feature similarity[42]. Thus,

$$L_{pixel} = \left\|R - R_{GT}\right\|_2^2, \ L_{perceptual} = \left\|S(R) - S(R_{GT)}\right\|_2^2, \quad (3)$$

where S is a form of feature-extraction function, which is the output of relu4_4 in the VGG-19 network. The total loss is defined as

$$L_{total} = L_{pixel} + \alpha \cdot L_{perceptual} \quad (4)$$

4.2.2 3D reconstruction-based method

We used the 3D reconstruction and rendering method to generate view interpolation results and evaluated the performance. First, the captured multi-viewpoint frames were calibrated to obtain the intrinsic and extrinsic matrices. An arc was constructed based on the calibrated camera locations (Figure 6); then, we synthesized new viewpoints based on the relative camera parameters. Next, the 3D scene was reconstructed using a traditional multi-viewpoint reconstruction pipeline, which generated the textured triangle mesh from calibrated multi-viewpoint images[43]. Poisson blending[44] was then used to fill the holes and cracks. Finally, new virtual-viewpoint images were rendered with the virtual camera parameters.

4.3 Interactive interface

To generate visual and interactive results, we stitched the synchronous video frames from all perspectives (in order of physical placement) into a matrix-shaped "moment frame", then we recomposed the stitched frames of all moments into an FVV.

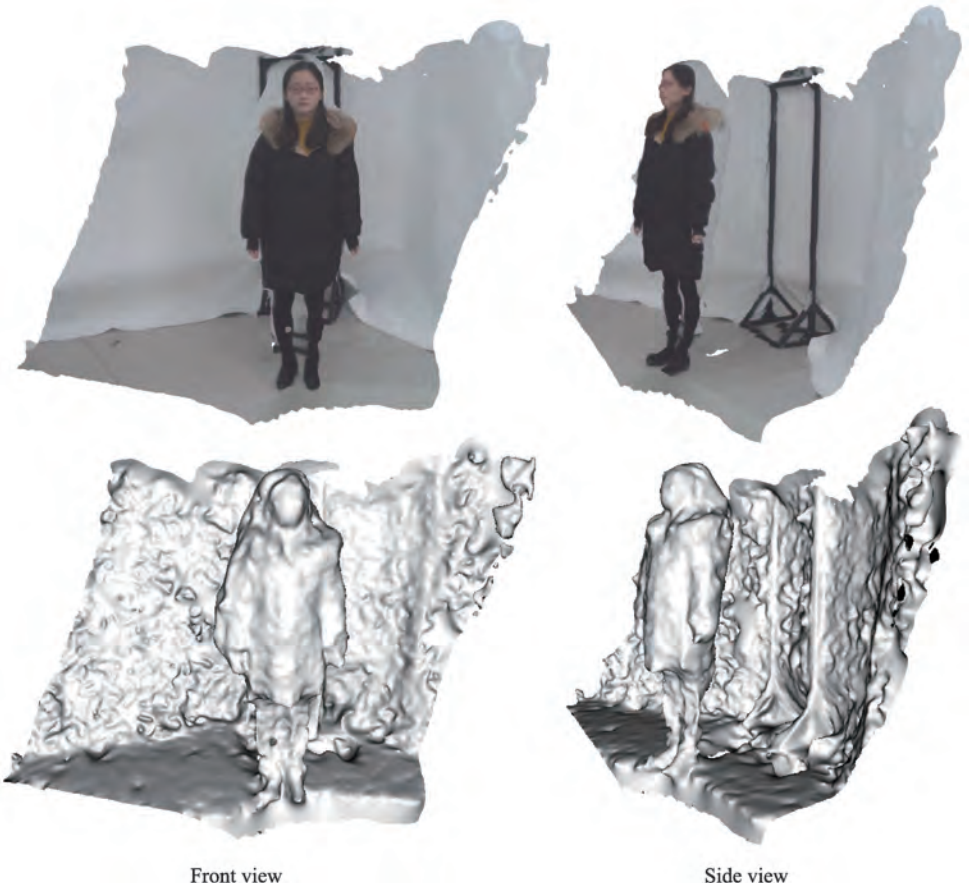
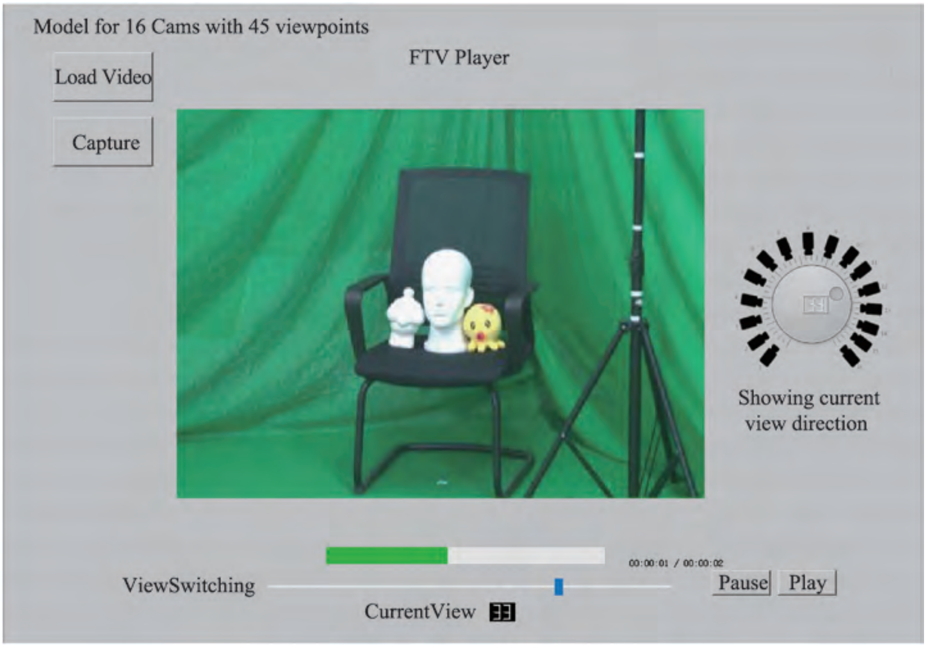


Figure 6 An example of the intermediate 3D-reconstruction result.

As shown in Figure 7, the interactive software was designed around QT, and the interactive video had a maximum resolution of up to 720P (1280×720). On the user interface, a dynamic video is presented, in which the user can select any viewpoint within the view range. By dragging the slider at the bottom (or by switching the dial on the right-hand side), different viewpoints can be switched between with a visually


Model for 16 Cams with 45 viewpoints
FTV Player
Load Video
Capture
Showing current view direction
ViewSwitching
Pause
Play

00:00:01 / 00:00:02

CurrentView 33

Figure 7 The interactive software designed around QT.

pleasant smoothness. Users can stop the video at any time and change the viewpoint to realize the "bullet time" effect.

5 Experiments

5.1 Experimental setup

We tested our systems on two setups, as explained in Section 3.1; the input videos were captured in laboratory scenes and a basketball stadium. The visual results are displayed in Figures 8 and 9, and a quantitative evaluation is given in Table 1. In Figure 8, the baselines of two cameras (which captured the input viewpoints of Test 1, Test 2, Test 4, and Test 5) were relatively narrow, with viewpoint direction intersection angles in the range of 1° to 4° ; in this range, our network can stably generate high-

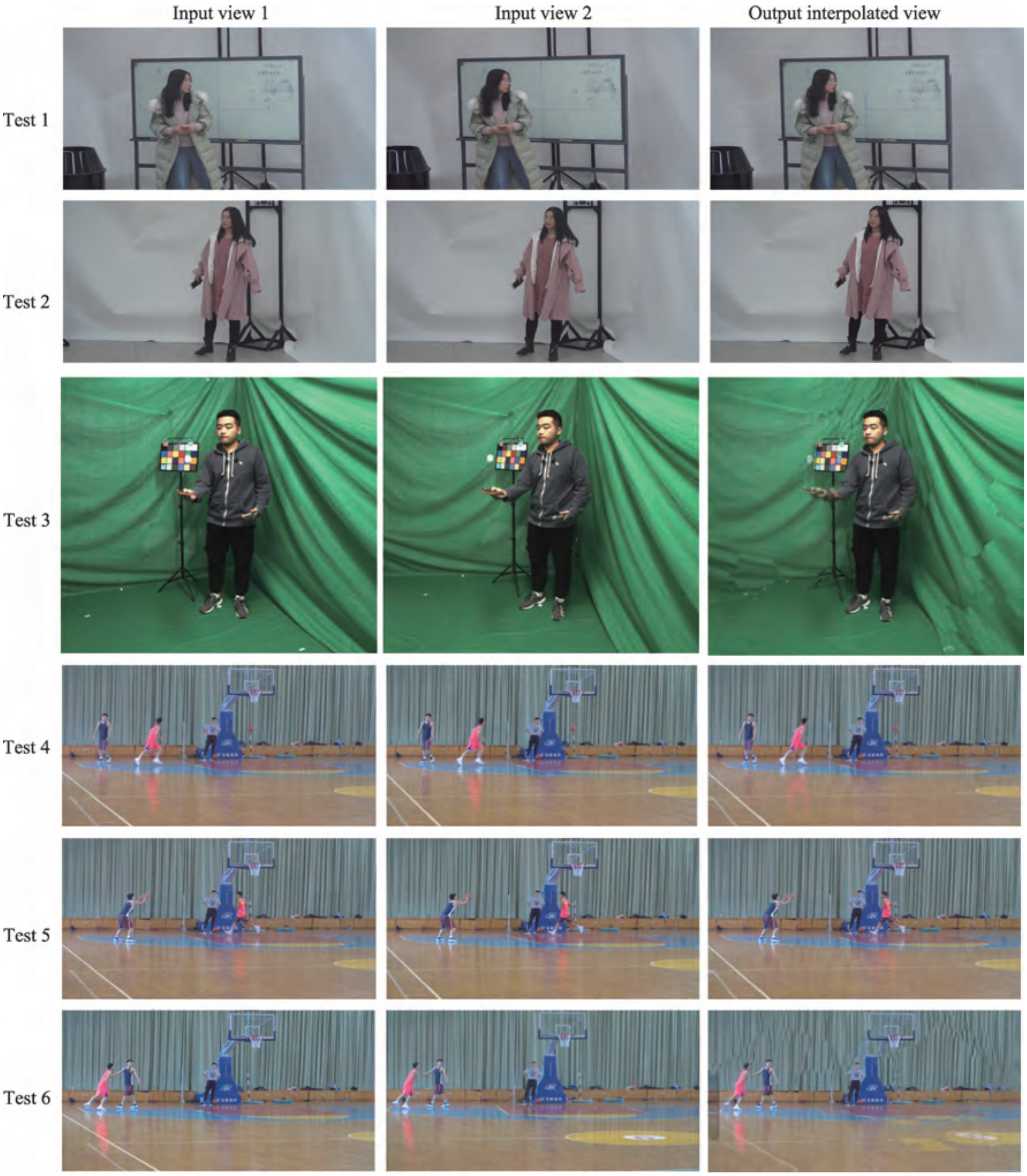


Figure 8 Results of interpolated viewpoint system using different scenes and baselines.

performance results from our system. For Test 3 and Test 6, the camera baselines were much wider, with viewpoint direction intersection angles exceeding 15°. The results of Test 3 still perform well owing to the simple background; however, the results of Test 6 suffer from ghost effects, due to the complicated background. Our method can be used to synthesize VR material, as shown in Figure 10.

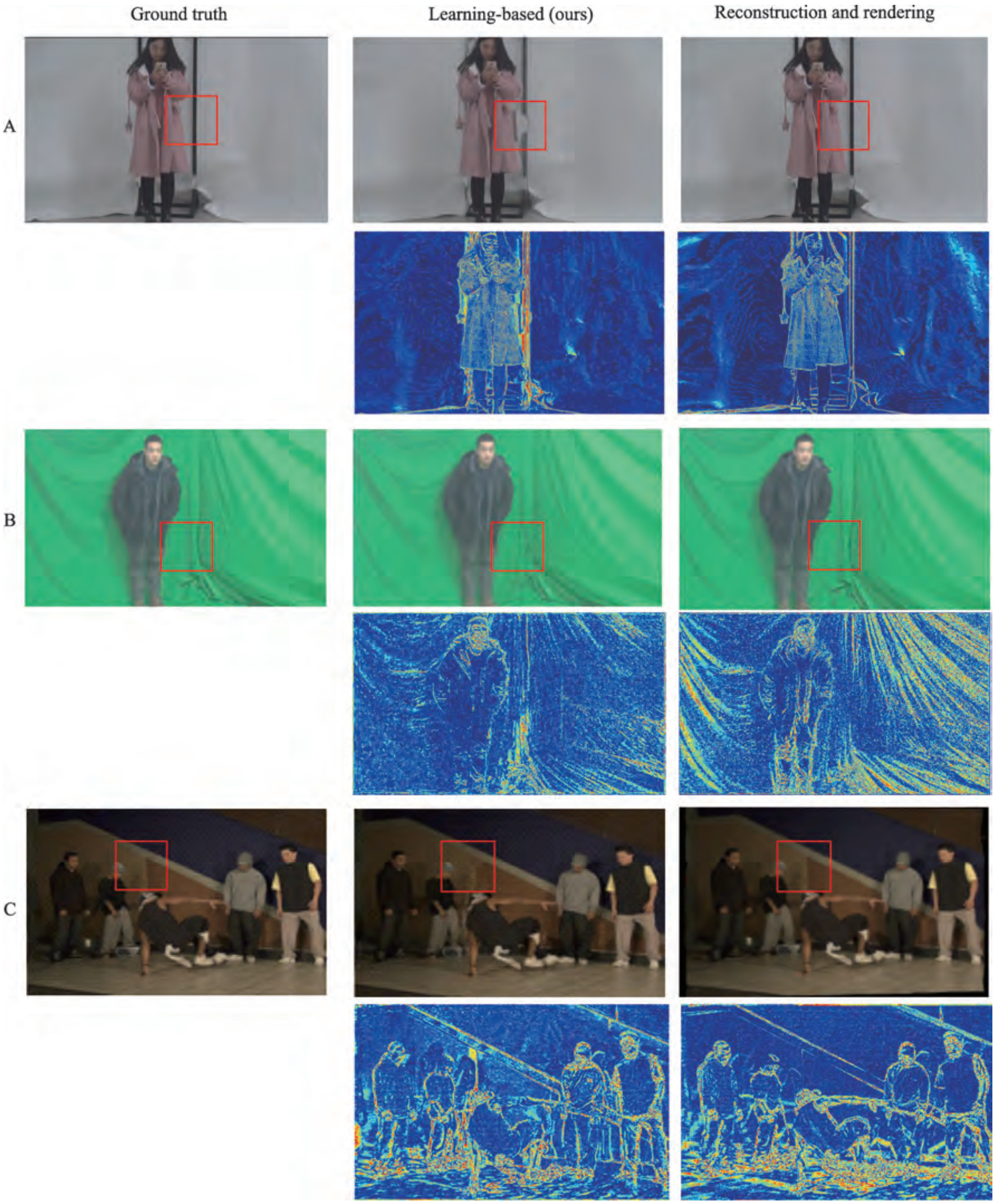


Figure 9 Comparison of the generated virtual viewpoint images and error maps with different methods. A and B contain images captured using our system, C contains images from Zitnick et al.[14].

To compare the generated results with ground truths, we chose the first- and the third-camera viewpoints (from three successive cameras in our array) as input viewpoints. The images captured by the middle one are defined as the ground truth. We used the images captured with our system for evaluation (Figure 9A and Figure 9B), and used a third-party dataset[14] for comparison (Figure 9C). The viewpoint direction

Table 1 Evaluation of SSIM and PSNR for the three samples

	PSNR / SSIM				
	Ours	3DRR	MVA	OF	DAIN
A	27.9264 / 0.9167	26.5868 / 0.9156	23.7966 / 0.7794	24.3421 / 0.8419	28.2023 / 0.9189
B	29.9093 / 0.9382	22.4726 / 0.8222	22.1897 / 0.7492	22.0877 / 0.7035	29.9818 / 0.9392
C	25.7507 / 0.9049	25.2853 / 0.9011	21.7692 / 0.8031	24.0143 / 0.8754	27.3504 / 0.9187

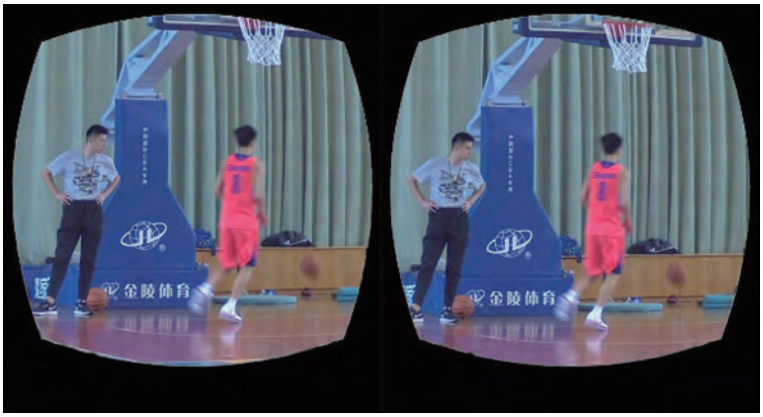


Figure 10 Synthesized VR material.

intersection angles slightly exceeded the optimal angle range of our system. As shown in Figure 9, ghosts or cracks appear in some zones where the pixel intensity changes sharply; for instance, at the borders of objects in scenes. The values in the image error maps are defined by pixel-wise L1 distance.

5.2 Comparison with reconstruction and rendering

We compare a learning-based method (ours) and a reconstruction and rendering method visually in Figure 9 and quantitatively in Table 1. The quantitative evaluation shows that the learning-based method outperforms the 3D-reconstruction and rendering method, as indicated by the measured parameters SSIM and PSNR; the superiority was greater when the background was simple. The high performances measured by the SSIM and PSNR metrics demonstrate that the viewing experience is more pleasant when switching between viewpoints. By visual comparison, it can be seen that though the results of reconstruction and rendering methods are visually inferior in term of similarity to the ground truth, they perform better in partial regions with complex structures or textures, as shown by the red boxes in Figure 9. This indicates that the multi-viewpoint inputs generate a more accurate reconstructed structure under the reconstruction and rendering methods; however, the learning-based method cannot predict accurate structures from only two viewpoints.

5.3 Comparison with previous methods

We compared our method with previous methods using the metrics of PSNR, SSIM, and run-time. MVA[45] and OF[46] are non-learning based methods that synthesize the target image via multi-plane images and optical flow, respectively. DAIN[47] is a learning-based method that takes advantage of depth-aware optical flow. As shown in Tables 1 and 2, our method outperforms MVA and OF in terms of PSNR/SSIM/run-time for all three samples. Although our method scores slightly lower in PSNR and SSIM in comparison to DAIN, our method has a distinct advantage in its speed, because its runtime is one-seventh of DAIN's. As shown in Figure 11, the synthesized images of MVA and OF contain obvious defects at the edge of the human body, whereas DAIN produces the visually plausible results comparable to those of our method.

Table 2 Run-time testing for different methods

	Our Method	3DRR	MVA	OF	DAIN
Runtime (s)	3.347	38.672	45.106	9.191	23.208

Notes: We computed the mean run-times of the A, B, C samples mentioned above.

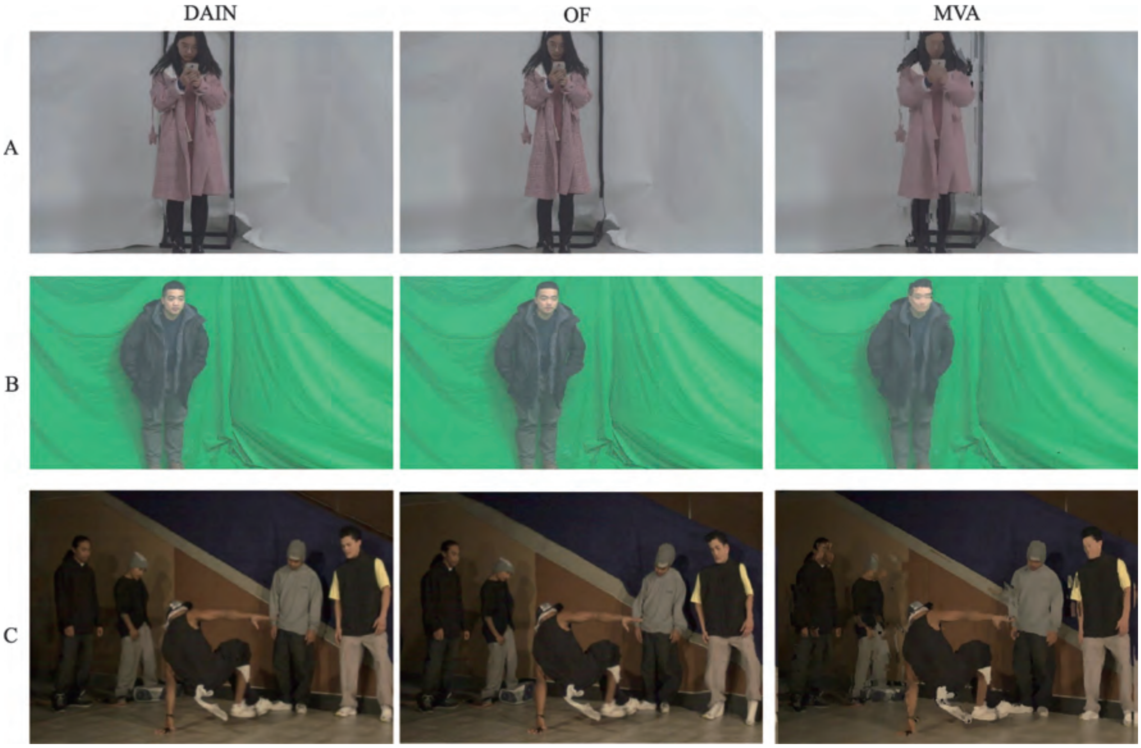


Figure 11 Synthesized images of other methods for our testing sample.

5.4 Traditional method vs learning method

The learning-based method and the reconstruction and rendering method (traditional method) have their own strengths and weaknesses. The learning-based method only requires two input images as input and the processing is quick. The quantitative evaluation shows that the learning-based method performs better, especially when the background is simple. However, it must also be noted that the learning-based method may fail in some regions, such as a slender rod. By comparison, the reconstruction and rendering method requires more images as inputs to ensure the quality of multi-viewpoint reconstruction. The 3D-reconstruction process is a long pipeline with many steps, including structure from motion, dense stereo matching, optimization of disparity, and meshing. This long pipeline tends to be fragile because it requires all steps to function correctly; moreover, it has a long processing time. The reconstruction and rendering method can generate accurate results in a complex structure when there are sufficient textures between overlapping views. In the experiment, it was observed that the learning-based method for view interpolation has an excellent performance in maintaining fidelity, even for wide-baseline input images when the background is relatively simple; furthermore, it is superior in speed and practicability.

6 Conclusion

In this paper, we proposed a complete FVV-generation system. We used two sets of equipment for the hardware system, which focused on high resolution and high frame-rate, respectively. The software consisted of the FVV-generation algorithm and an interactive interface. The core module of the software was that of novel viewpoint generation, in which the number of the available viewpoints was increased using a neural network. With the proposed FVV-generation system, the user is able to control the viewing angle of the video program and can also switch between perspectives smoothly. We tested our system on a basketball stadium and indoor scenes; the PSNR and SSIM results showed that our method outperformed the traditional 3D-rendering method. The processing frame-rate and resolution of raw data were as high as 30fps, 4K (3864×2174) and 60fps, 720P (1280×720) for the two setups, respectively; and 30−60fps, 720P for the final FVV in our interactive software, which is visually pleasant for users.

Some problems remain to be solved in the future. For example, the images generated by the CNN typically feature blurred boundaries for wide-baseline pairs of input images. Although reconstruction and rendering methods produce clear boundaries, they are too time-consuming and suffer from model cracks and holes.

References

1 Tanimoto M, Tehrani M, Fujii T, Yendo T. Free-viewpoint TV. IEEE Signal Processing Magazine, 2011, 28(1): 67−76
DOI:10.1109/msp.2010.939077

2 Seitz S M, Curless B, Diebel J, Scharstein D, Szeliski R. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York, NY, USA, IEEE, 2006, 519−528
DOI:10.1109/cvpr.2006.19

3 Zhu H, Nie Y M, Yue T, Cao X. The role of prior in image based 3D modeling: a survey. Frontiers of Computer Science, 2017, 11(2): 175−191
DOI:10.1007/s11704-016-5520-8

4 Seitz S M. Photorealistic scene reconstruction by voxel coloring. IEEE Conference on Computer Vision and Pattern Recognition Conference, 1997

5 Chen J, Watanabe R, Nonaka K, Konno T, Sankoh H, Naito S. Fast free-viewpoint video synthesis algorithm for sports scenes. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Macau, China, IEEE, 2019
DOI:10.1109/iros40897.2019.8967584

6 Miller G, Hilton A, Starck J. Interactive free-viewpoint video. IEEE European Conference on Visual Media Production, 2005

7 Shum H, Kang S B. Review of image-based rendering techniques. Visual Communications and Image Processing, 2000

8 McMillan L, Bishop G. Plenoptic modeling. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. New York, USA, ACM Press, 1995
DOI:10.1145/218380.218398

9 Levoy M, Hanrahan P. Light field rendering. ACM Transactions on Graphics, 1996

10 Hedman P, Kopf J. Instant 3D photography. ACM Transactions on Graphics, 2018, 37(4): 1−12
DOI:10.1145/3197517.3201384

11 Hedman P, Philip J, Price T, Frahm J M, Drettakis G, Brostow G. Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics, 2019, 37(6): 1−15
DOI:10.1145/3272127.3275084

12 Chaurasia G, Duchene S, Sorkine-Hornung O, Drettakis G. Depth synthesis and local warps for plausible image-based navigation. ACM Transactions on Graphics, 2013, 32(3): 1−12
DOI:10.1145/2487228.2487238

13 Hedman P, Ritschel T, Drettakis G, Brostow G. Scalable inside-out image-based rendering. ACM Transactions on Graphics, 2016, 35(6): 1−11
DOI:10.1145/2980179.2982420

14 Zitnick C L, Kang S B, Uyttendaele M, Winder S, Szeliski R. High-quality video view interpolation using a layered representation. ACM Transactions on Graphics, 2004, 23(3): 600
DOI:10.1145/1015706.1015766

15 Zhu H, Zuo X X, Wang S, Cao X, Yang R G. Detailed human shape estimation from a single image by hierarchical mesh deformation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA, IEEE, 2019
DOI:10.1109/cvpr.2019.00462

16 Flynn J, Neulander I, Philbin J, Snavely N. Deep stereo: learning to predict new views from the world's imagery. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, IEEE, 2016
DOI:10.1109/cvpr.2016.595

17 Zhou T H, Tucker R, Flynn J, Fyffe G, Snavely N. Stereo magnification: learning view synthesis using multiplane images. 2018

18 Penner E, Zhang L. Soft 3D reconstruction for view synthesis. ACM Transactions on Graphics, 2017, 36(6): 1–11
DOI:10.1145/3130800.3130855

19 Smolic A. 3D video and free viewpoint video: From capture to display. Pattern Recognition, 2011, 44(9): 1958–1968
DOI:10.1016/j.patcog.2010.09.005

20 Zhu H, Su H, Wang P, Cao X, Yang R G. View extrapolation of human body from a single image. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, IEEE, 2018
DOI:10.1109/cvpr.2018.00468

21 Collet A, Chuang M, Sweeney P, Gillett D, Evseev D, Calabrese D, Hoppe H, Kirk A, Sullivan S. High-quality streamable free-viewpoint video. ACM Transactions on Graphics, 2015, 34(4): 1–13
DOI:10.1145/2766945

22 Debevec P E, Taylor C J, Malik J. Modeling and rendering architecture from photographs. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. New York, USA, ACM Press, 1996
DOI:10.1145/237170.237191

23 Montemerlo M, Thrun S, Koller D, Wegbreit B. A factored solution to the simultaneous localization and mapping problem. Conference on Artificial Intelligence, 2002

24 Sturm P, Triggs B. A factorization based algorithm for multi-image projective structure and motion//Lecture Notes in Computer Science. Berlin, Heidelberg, Springer Berlin Heidelberg, 1996, 709–720
DOI:10.1007/3-540-61123-1_183

25 Tan F, Zhu H, Cui Z. Self-supervised human depth estimation from monocular videos. IEEE Conference on Computer Vision and Pattern Recognition, 2020

26 Yang H, Zhu H, Wang Y, Huang M, Shen Q, Yang R, Cao X. FaceScape: a large-scale high quality 3D face dataset and detailed riggable 3D face prediction. IEEE Conference on Computer Vision and Pattern Recognition, 2020

27 Bosc E, Pepion R, Le Callet P, Koppel M, Ndjiki-Nya P, Pressigout M, Morin L. Towards a new quality metric for 3-D synthesized view assessment. IEEE Journal of Selected Topics in Signal Processing, 2011, 5(7): 1332–1343
DOI:10.1109/jstsp.2011.2166245

28 Ceulemans B, Lu S P, Lafruit G, Munteanu A. Robust multiview synthesis for wide-baseline camera arrays. IEEE Transactions on Multimedia, 2018, 20(9): 2235–2248
DOI:10.1109/tmm.2018.2802646

29 Niklaus S, Mai L, Yang J M, Liu F. 3D Ken Burns effect from a single image. ACM Transactions on Graphics, 2019, 38 (6): 1–15
DOI:10.1145/3355089.3356528

30 Regmi K, Borji A. Cross-view image synthesis using conditional GANs. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, IEEE, 2018, 3501–3510
DOI:10.1109/cvpr.2018.00369

31 Lu Y L, Sun T F, Jiang X H, Xu K, Zhu B. Frontal view synthesis based on a novel GAN with global and local discriminators. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics. Suzhou, China, IEEE, 2019, 1–5

DOI:10.1109/cisp-bmei48845.2019.8965829

32 Wang Y L, Liu F, Wang Z L, Hou G Q, Sun Z N, Tan T N. End-to-end view synthesis for light field imaging with pseudo 4DCNN//Computer Vision−ECCV 2018. Cham: Springer International Publishing, 2018, 340−355
DOI:10.1007/978-3-030-01216-8_21

33 Niklaus S, Mai L, Liu F. Video frame interpolation via adaptive separable convolution. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, IEEE, 2017, 261−270
DOI:10.1109/iccv.2017.37

34 Liu Z W, Yeh R A, Tang X O, Liu Y M, Agarwala A. Video frame synthesis using deep voxel flow. In: 2017 IEEE International Conference on Computer Vision. Venice, IEEE, 2017, 4463−4471
DOI:10.1109/iccv.2017.478

35 Niklaus S, Liu F. Context-aware synthesis for video frame interpolation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, IEEE, 2018, 1701−1710
DOI:10.1109/cvpr.2018.00183

36 Zhou T H, Tulsiani S, Sun W L, Malik J, Efros A A. View synthesis by appearance flow//Computer Vision−ECCV 2016. Cham: Springer International Publishing, 2016, 286−301
DOI:10.1007/978-3-319-46493-0_18

37 Park E, Yang J M, Yumer E, Ceylan D, Berg A C. Transformation-grounded image generation network for novel 3D view synthesis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, IEEE, 2017
DOI:10.1109/cvpr.2017.82

38 Kalantari N K, Wang T C, Ramamoorthi R. Learning-based view synthesis for light field cameras. ACM Transactions on Graphics, 2016, 35(6): 1−10
DOI:10.1145/2980179.2980251

39 Ji D H, Kwon J, McFarland M, Savarese S. Deep view morphing. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, IEEE, 2017
DOI:10.1109/cvpr.2017.750

40 Zhou T H, Tucker R, Flynn J, Fyffe G, Snavely N. Stereo magnification: learning view synthesis using multiplane images. 2018

41 Lam E Y. Combining gray world and retinex theory for automatic white balance in digital photography. In: Proceedings of the Ninth International Symposium on Consumer Electronics. Macau SAR, IEEE, 2005
DOI:10.1109/isce.2005.1502356

42 Johnson J, Alahi A, Li F F. Perceptual losses for real-time style transfer and super-resolution//Computer Vision−ECCV 2016. Cham: Springer International Publishing, 2016, 694−711
DOI:10.1007/978-3-319-46475-6_43

43 Furukawa Y, Ponce J. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(8): 1362−1376
DOI:10.1109/tpami.2009.161

44 Pérez P, Gangnet M, Blake A. Poisson image editing. ACM Transactions on Graphics, 2003, 22(3): 313
DOI:10.1145/882262.882269

45 Bleyer M, Gelautz M, Rother C, Rhemann C. A stereo approach that handles the matting problem via image warping. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, IEEE, 2009
DOI:10.1109/cvpr.2009.5206656

46 Sun D Q, 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. San Francisco, CA, USA, IEEE, 2010, 2432−2439
DOI:10.1109/cvpr.2010.5539939

47 Bao W B, Lai W S, Ma C, Zhang X Y, Gao Z Y, Yang M H. Depth-aware video frame interpolation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA, IEEE, 2019, 3703−3712
DOI:10.1109/cvpr.2019.00382