HYBRID MOTION/DEPTH-ORIENTED INPAINTING FOR VIRTUAL VIEW SYNTHESIS IN MULTIVIEW APPLICATIONS

Kuan-Yu Chen, Pei-Kuei Tsung, Pin-Chih Lin, Hsin-Jung Yang, and Liang-Gee Chen

DSP/IC Design Lab, Graduated Institute of Electronics Engineering
National Taiwan University, Taipei, Taiwan

ABSTRACT

Multiview video can provide users a 3D and virtual reality perception by its multiple viewing angles. To improve the quality of virtual view synthesized frame and remove the disocclusion region, hole filling technique is required. By classifying the different image artifact and applying proper hybrid motion/depth-oriented inpainting algorithm, the image quality will be closer to the real image. In addition, the simulation result and comparison with previous works on inpainting show that both objective evaluation estimates and subjective perceptual vision come to a better result by the system proposed in this paper.

Index Terms **—** virtual view synthesis, multiview video, 3D TV, FTV, inpainting

1. INTRODUCTION

In recent years, 3D video and related applications, including 3DTV [1] and free view-point TV (FTV) [2], are getting more and more attention. By capturing different view point, multiview video can provide a complete 3D scene perception to users and thus is a popular 3D video format. However, due to the physical limitation of camera volume and the bandwidth limit from the communication system, a multiview video sequence contains only limited view numbers and cannot switch the viewpoint continuously. In order to support the free view-point characteristic, virtual view synthesis is used to generate frames from the view point which is not captured by a real camera. Under the matrix-based coordinate translation, pixels in the virtual view frames can be filled by pixels from the neighboring real views based on camera matrices and the calibrated depth maps. In order to make the view synthesis and the corresponding multiview applications practical, MPEG-FTV group is working on virtual view synthesis since 2007. [3] The view synthesis reference software (VSRS) is released by the MPEG-FTV group as the reference software and the research platform. [3] In virtual view synthesis, the main design challenges are on the artifact removing and occlusion filling. Although the virtual view can be generated by the neighboring views, there may be still some hiding background regions with no reference pixels. In VSRS, two methods are provided to fill these hole regions. The first is the inpainting function from OpenCV [4] and the other is the background padding. These two methods are both filling the occlusion by the neighboring pixels. However, both of the algorithms cannot solve the information loss problem in the pixel domain. Therefore, holes with hidden textures cannot be reconstructed with good perceptual quality. Furthermore, not all the hole regions are occluded backgrounds. Therefore, only a single algorithm is not enough to deal with all the situations. In order to provide virtual views with better quality, a hybrid motion/depth-oriented inpainting scheme is proposed in this paper. First, by the proposed virtual view motion vector calibration, the motion information between the virtual view frames is generated. Therefore, texture information from temporal domain can be used in hole filling. Second, the depth-oriented inpainting is proposed to deal with the hidden background region. Finally, by the artifact profiling and boundary detection, occlusions and artifacts from different sources are classified to different categories and solved by the corresponding algorithms. Comparing with the prior-arts, the proposed algorithm provides better visual quality based on the enhancement from the temporal domain. In the objective comparison, the proposed algorithm also has 0.14 to 0.3 PSNR gains higher than the prior-arts.

The remaining of this paper is organized as follows: First, the artifact profiling in the virtual view synthesis and the problem definition is introduced in Sec. 2. Second, the proposed hybrid motion/depth-oriented inpainting scheme is described in Sec. 3. Then, the simulation result is shown in Sec. 4. Finally, Sec. 5 concludes this paper.

2. ARTIFACT PROFILING IN VIRTUAL VIEW SYNTHESIS

Conventional view synthesis flow is shown in Fig. 1 (a). First of all, real-captured views are regarded as reference views in the view synthesis process. By using calibration parameters, depth map, and the real texture image, the reference views are warped to the target virtual view point. After warping, the background region covered by foreground objects in the reference views may be exposed since the relative location is changing. Therefore, each warped virtual view may contain holes. The occluded region is shown as the green part in Fig. 1 (b). To complete the synthesized frame, these virtual view frames are intermixed together to reduce the occluded region.

Following the flow shown in Fig. 1(a), the intermediate virtual view is generated. However, besides the occluded regions that are not covered by each reference view, there are still some unwanted effects. As shown in Fig. 1 (c) and (d), some artifacts, such as small cracks and ghost effects are introduced. According to our previous work in [5], small cracks result from truncation gaps during the floating-point warp. To solve the artifacts, the running interpolation with z-buffer scheme is proposed in [5]. Figure 2 reveals the result of running the interpolation with z-buffer. The other type of artifact is the ghost effect, also known as boundary effect, which is due to the imperfect depth map. That is, sometimes the object's boundary on the depth map is not so clear. The mismatch problem emerges between the boundary of the

978-1-4244-6379-4/10/$26.00 ©2010 IEEE

depth maps and the texture frame. Then, the foreground boundary will be judged as the background and be warped to the wrong position. Figure 3 (a) depicts the case of the ghost effect. In the previous work [5], background erosion method is proposed to remove this defect. Figure 3 shows the result of executing the background erosion. Based on the previous work, most of the artifacts can be eliminated accordingly. Nevertheless, the fully-occluded regions, as shown in Fig. 1 (a) and (e), are still problems.

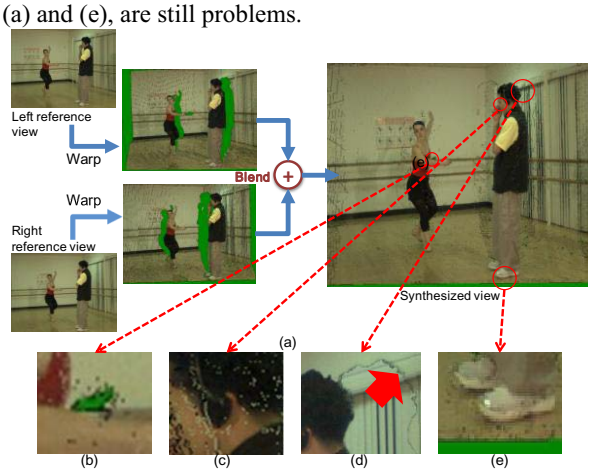
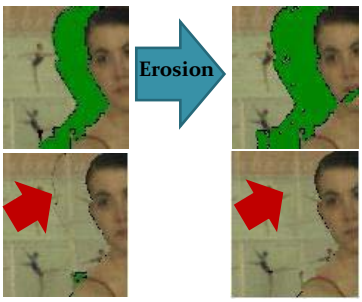


Figure 1. (a) Virtual view synthesis flow and (b)-(e) synthesis artifact classification, (b) disoccluded region, (c) small cracks, (d) ghost effect and (e) out-of-boundary region

(a) (b)

Figure 2. Small cracks removing (a) before and (b) after running interpolation [5]



(a) (b)

Figure 3. Boundary effect removing (a) before and (b) after the background erosion [5]

3. PROPOSED ALGORITHM

3.1 Proposed Virtual View Motion Vector Calibration

Owing to the lack of reference information, occluded holes still exist after view blending. These holes may be caused from out of the reference frame boundary or the uncovered background region. In order to recover these non-reference areas, additional information from both the spatial and temporal domain is used. As working on the video applications, the available data from virtual view sequences includes not only the spatial reference pixels but also temporal reference data. In other words, the holes in current virtual view frame may be fixed by the existing content in the previous frames. In spite of this fact, we can intelligibly just replace the hole region by the same content from the previous reference frame, the area originally occupied by dynamic objects may result in further problems. In case of the background holes in the current frame removed by the moving foreground objects such as people or animals, the wrong texture will be filled to the hole which ends up with inappropriate results. Accordingly, finding the corresponding position is essential in the motion-oriented inpainting.

The concept of the proposed motion-oriented inpainting flow is illustrated in Fig. 4. To acquire the relative image data, the motion vector information is taken into consideration. However, the motion vectors only exist in the real captured and encoded views. There are no motion vectors in the virtual view and the motion vectors in the reference view cannot be directly applied to virtual view owing to the warping displacement between views. Thus, the objective is to find out the precise motion vector in the virtual views. While the hole pixel is not contained in the real view, every time the occluded pixel is detected, its nearest non-hole pixel is located and warped back to get the spatial reference view for gathering the motion information. To get a reasonable virtual view motion vector, the two endpoints of the motion vector in the reference view are warped to the virtual view. After that, two new points are generated. Then, the new motion vector calibrated to the virtual view position can be generated by these two new points. Finally, the corresponding texture image in the temporal reference view is picked out to be the replacing material. However, this method cannot provide a precise result since the motion vector from compression may be unreal. Therefore, some limitations are set to avoid the false-alarms. For example, since the occluded region always emerges at the background region, as mentioned before, the background holes should not be substituted by the foreground objects. For this reason, before inpainting with motion data, the corresponding depth is verified. If the depth surpasses the pre-defined threshold of foreground region, the temporal reference data will not replace the hole region and the hole remains still.

In this paper, the encoding platform is the joint multi-view video model (JMVM) released by joint video team. [7] In JMVM, the H.264/AVC is used as the baseline profile. Thus, the multiple-block-size is used. In the proposed motion-oriented inpainting, it means that pixels in the same block share the same motion vector. Aiming to simplify the progress, the reference data is also reused in block-based format to reduce the operating time. Whenever one hole pixel is detected and is qualified to repair by previous corresponding texture, other hole pixels in the same block which sorted by JMVM are inpainting simultaneously.

3.2 Out-of-Boundary Region Detection

In the virtual view, there is another type of holes which is not caused by the uncovered background. After warping, the virtual view frame may not maintain parallel rectangular and may also have an angular rotation. For this reason, the virtual view frame will contain area that is out of the range of the reference view which we called the out-of-boundary (OOB) region as shown in Fig. 1 (e). While the OOB region occurred, there are no reference texture data in the temporal domain since this occlusion is defined by the camera setting.

Therefore, the motion-aware inpainting is not appropriate to OOB region. To deal with this kind of occlusions, a proper out-of-boundary detection scheme is required first. Intending to distinguish the OOB region, the pixel locations of the four corners are recorded during warping. With this information, the out-of-boundary region can be marked out by connecting the four corner pixel. Hence, OOB regions can be separated from the other occluded part. Then, since the OOB regions are not due to the depth difference, the general case inpainting in OpenCV is used in these regions.

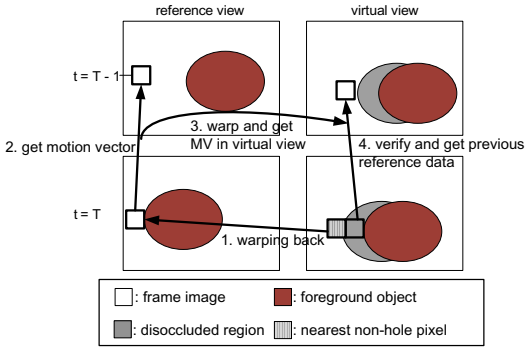


Figure 4. Proposed virtual view motion information gathering scheme

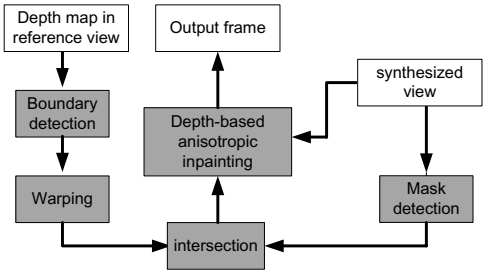


Figure 5. Proposed depth-based inpainting flow

3.3 Proposed Depth-Oriented Inpainting

After motion-oriented inpainting, occluded region with valid temporal reference texture data are removed. However, there are still some holes which cannot be filled. For example, blocks under intra mode with no motion vectors. However, these regions are also caused by the uncovered background which is blocked by the foreground objects in the reference real view but appeared in virtual view. The conventional OpenCV inpainting is not suitable. As we mentioned before, this type of occluded region is the extension of background. Intuitively, the inpainting data propagated into the holes should come from the background only. Therefore, the depth-oriented inpainting is proposed as follows.

In the proposed depth-oriented inpainting, the anisotropic filter used in conventional inpainting is also adopted as the base layer [4]. First, the image to be processed is defined as I_{MxN} , and Ω stands for the inpainting region. To get the inpainting result more realistic, the isophote line prolongs into Ω instead propagate data perpendicular to the inpainting boundary δΩ. The main inpainting equation is as follow:

$$I^{n+1}(i,j) = I^{n}(i,j) + \Delta t I_t^{n}(i,j), \forall (i,j) \in \Omega$$

where n denotes the iteration count of the inpainting process; (i , j) is the coordinate of the pixel; and Δt is the rate of improvement. The representation of $I_t^{n}(i,j)$ stands for the update of $I^{n}(i,j)$.

To enhance the ability of dealing with the hole generated by virtual view synthesis, some depth-oriented constraint is used. The depth information in virtual view is also deficient like motion vector data. However, our purpose is not to obtain the exact depth value but to propagate data form background content into the inpainted regions. In other words, our goal is to indicate the boundary between foreground and holes. Hence, we first detect the image boundary of the reference view frame. Afterward, the boundary map is warped to the virtual view. It is obvious that the boundary position after warping is still boundary in the virtual view frame and no redundant boundary pixel will be produced. On the other hand, the mask points out the inpainting region are easily generated from the virtual view frame. To mark out the foreground and hole boundary, the detected edge map and the mask are intersected. During inpainting process, we ignore any data that cross the intersection, which is avoiding foreground data prolong inward to the background region. Finally, the occluded region is fully repaired and generates a virtual view. The detailed flow of the proposed depth-oriented inpainting is introduced in Fig. 5.

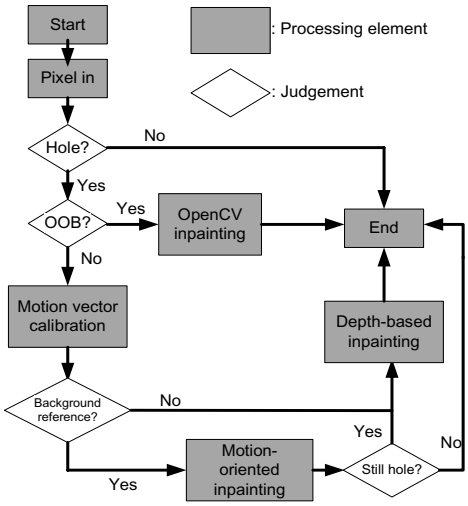


Figure 6. Proposed inpainting algorithm flow

3.4 Proposed Hybrid Motion/Depth-Oriented Inpainting Flow

Figure 6 summarizes the overall process of the proposed hybrid motion/depth-oriented inpainting. In the beginning, the synthesized view data comes in and the hole detection process is triggered to determine whether it belongs to hole region or not. When the hole is located in the out-of-boundary (OOB) region, it will be processed by OpenCV inpainting function. On the other hand, the occluded region due to the uncovered background is first passed through the motion vector calibration. As long as the temporal reference data is in background region, the motion-oriented inpainting is used. Last but not least, if the reference data is not qualified for the background condition or there is still hole after motion-oriented hole filling, the depth-based anisotropic diffusion is applied. After that, the synthesis virtual view is fully repaired.

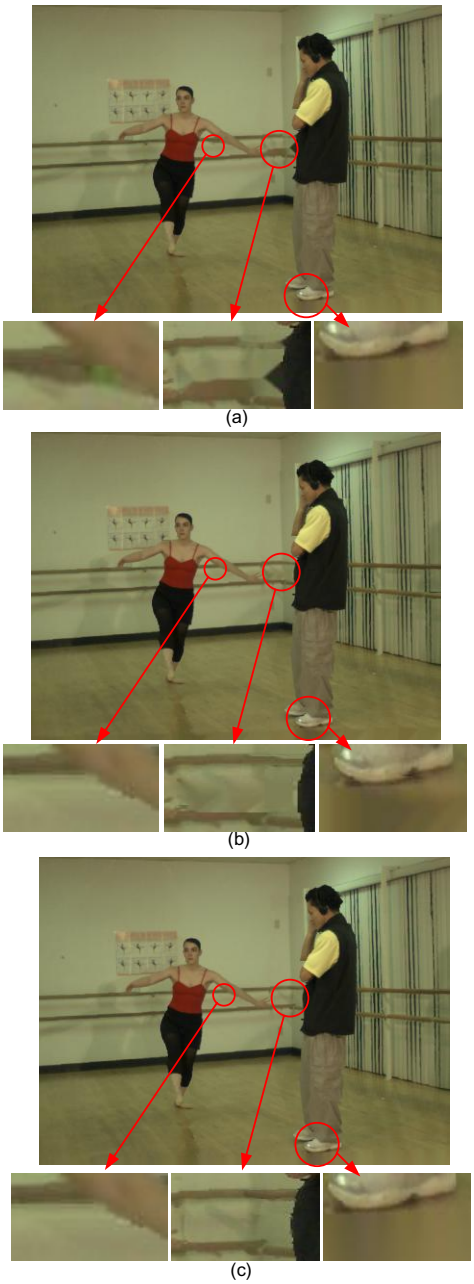


Figure 7. Subjective virtual view inpainting comparison (a) OpenCV inpainting function [4], (b) only apply depth-based inpainting [6] and (c) hybrid motion/depth-oriented inpainting (the contrast of figures are modified to highlight differences)

4. SIMULATION RESULT

In this paper, the multiview sequence "ballet" and "breakdancers" published by Microsoft are taken as the test sequences. The objective result is reveal in Table 1. Both PSNR and the structural similarity (SSIM) index are measured to perform as the comparing conditions. In PSNR evaluation, our proposed algorithm produced 0.14 dB and 0.3 dB gain better than only depth-oriented inpainting, like the prior-art [6], and the OpenCV inpainting. When it comes to SSIM, the proposed algorithm and OpenCV inpainting have very close performance and both outperform the depth-oriented inpainting. However, since there is no "real" data of the "virtual" view synthesis, only the objective result is not convincible enough. Figure 7 displays the inpainting results of the synthesized frame. As presented in Fig. 7, the OpenCV function is short of dealing with the occluded region at the fore/background boundary (Fig. 7 (a)) because of the foreground data propagating to background region. In Fig. 7 (b), depth information is considered. As we take a look at the region beside the woman's arm, the fore/background disoccluded region is modified well. However, the OOB region and texture in large hole may come into deficiency. Finally, the hybrid motion/depth-oriented inpainting algorithm proposed in the paper is applied and the result can be observed in Fig. 7 (c). Both large area texture and fore/background separation outperforms in comparison of two previous results.

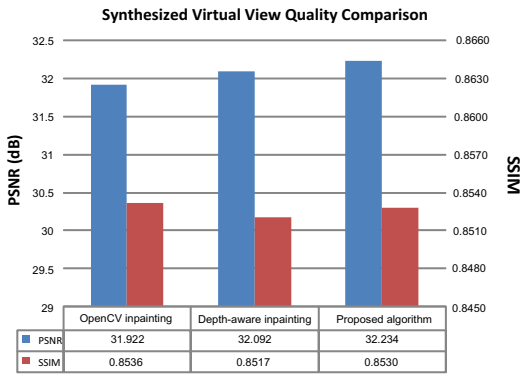


Table 1. Objective experimental result

5. CONCLUSION

This paper has presented a hybrid inpainting algorithm which collaborates with motion-oriented, depth-based, and conventional diffusion manners. To aim for better quality of virtual view synthesis in multi-view video applications, the appropriate solution is found with good variety for dealing with different types of image artifacts. The simulation result shows superior PSNR gain is obtained in average and an average progress is also revealed with SSIM metric while comparing with previous works. The proposed hybrid inpainting algorithm outperforms by both perceptual quality and the objective metric measure.

6. REFERENCES

[1] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proceedings of the IEEE, vol. 93, no. 1, pp 33-36, Jan. 2005*

[2] M. Tanimoto, "Free viewpoint television - FTV," *in Proceedings of Picture Coding Symposium, 2004*

[3] MPEG-FTV Group, "LDV Virtual View Rendering Software" *ISO/IEC JTC1/SC29/WG11 MPEG2008/M16040, Feb. 2009*

[4] Marcelo Bertalmio and Guillermo Sapiro, "Image Inpainting," *SIGGRAPH 2000*, pages 417 - 424

[5] P. K. Tsung et al, "Single iteration view interpolation for multiview video applications," *in Proceedings of 3DTV conference 2009, May 2009, pp. 1-4*

[6] Kwan-Jung Oh; Sehoon Yea; Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," *Picture Coding Symposium, 2009. PCS 2009* , vol., no., pp.1-4, 6-8 May 2009

[7] MPEG-4 Video Group, "Joint Multiview Video Model (JMVM) 1.0," *Number ISO/IEC JTC1/SC29/WG11 N8244*, July 2006