A View Synthesis Approach for Free-navigation TV Applications

Ilya Ganelin
Electrical & Computer Engineering Department, University of British Columbia
ICICS, University of British Columbia
Vancouver, BC, Canada
iganelin@ece.ubc.ca

Panos Nasiopoulos
Electrical & Computer Engineering Department, University of British Columbia
ICICS, University of British Columbia
Vancouver, BC, Canada
panos@ece.ubc.ca

Mahsa T. Pourazad
TELUS Communications Incorporation
Vancouver, Canada,
pourazad@ece.ubc.ca

***Abstract*—The need for multiview content is more pronounced with the emergence of Free-viewpoint Television (FTV), Super Multiview (SMV), and Free Navigation (FN) technologies. For multiview content creation, it is not practical to capture all of the views required by different multiview display technologies. Instead, a limited number of views are captured and the remaining views are synthesized using the available views. The efficiency of the view synthesizing process has a high impact on the quality of the generated multiview content. In this paper, we present a novel view synthesizing scheme, which utilizes unique techniques such as background decomposition in three layers, background edge dilation, vertical interpolation and edge aware warping, to generate high quality virtual views. Subjective evaluations confirm that our approach outperforms the state-of-the-art interpolation-based view synthesizing.**

Keywords-Free Navigation TV; Multiview TV; view synthesis; hole filling; Multimedia communication; Image generation; Image reconstruction.

I. INTRODUCTION

Free-viewpoint Television (FTV) provides the viewers with realistic impression of a scene by allowing them to freely navigate through the scene in Free Navigation (FN) applications or perceive scene depth in the case of Super Multiview (SMV) applications [1]. There are a number of hurdles in the proliferation of these technologies, such as availability, production, and transmission of multiview content to the end user. Multiview content production is expensive and highly demanding in terms of camera configuration and post processing [1]. As FTV technology evolves, manufacturers attempt to provide viewers with a larger number of views to improve transition between sweet spots. As a result, the number of views of the preliminary multiview content will no longer be enough, and thus synthesizing virtual views becomes essential. In the case of FN, where captured views are further apart, the quality of the synthesized view is even more important as there is no additional information from a neighboring view as in the case of SMV.

The main challenge with view synthesis is estimating the information of the occluded areas [1]. A common solution is to apply inter pixel interpolation to estimate the missing texture. This approach has been utilized in the state-of-the-art View Synthesis Reference Software (VSRS) [4], which has been adopted by the MPEG-3DV group to synthesize test sequences for 3D video compression standardization activities [2][4]. Unfortunately, the downfall of all interpolation-based hole-filling methods is that the interpolated texture does not resemble the true structure of the occluded areas.

In a different approach presented in [3], warping is utilized for synthesizing additional views. In this method, first a sparse saliency map is created. This map helps to separate foreground from background and then the saliency map information is used to stretch background parts of the picture and cover occluded areas. This technique minimizes the visible distortions of the image, but due to stretching it can also distort some human visual cues such as vertical lines and shapes of known objects, e.g., faces.

To overcome shortcomings of the existing schemes, we propose a new and unique view synthesizing scheme which uses background-to-foreground warping and background separation for accurate filling of occluded regions. Our method improves on the inter pixel interpolation idea of the VSRS approach as well as the Disney's warping technique [2][3][4]. The performance of the proposed technique is compared with that of the state-of-the-art VSRS [2][4], for both view extrapolation and view interpolation scenarios (see Scheme below) subjectively.

The remainder of this paper is structured as follows: Section 2 describes our method, Section 3 presents the experimental results, and conclusions are drawn in Section 4.

 ISBN: 978-1-61208-533-3

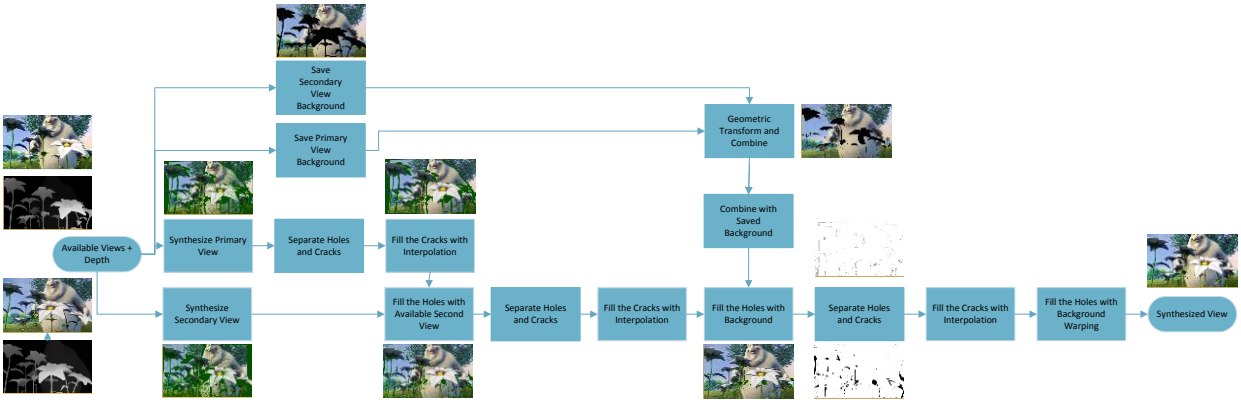


Figure 1: Block diagram of our method

II. OUR VIEW SYNTHESIS APPROACH

A. *Solving background leakage*

For FTV applications, it is common practice that several different views are captured with multiple cameras (usually in a parallel or arch setup) and additional views are synthesized, as if there were more cameras in the original setup. The resulting original views are spread apart from each other, so that many views need to be synthesized in between to generate free navigation or SMV (3D) content. Figure 1 shows the block diagram of our approach and the different stages of our hole filling process. In our approach, we create a primary synthesized view using the closest available real camera view position to the location of the synthesized view.

The translation process of pixels from the original view to the synthesized one creates holes (pixels with missing pixel values) in a way similar to a disparity-based synthesizing approach [4].

Due to the fact that the depth map of a foreground object might have different values, because of its volumetric nature, transitioning these pixels from one camera plane to another may result in consecutive pixels of the same object being space-separated. The background pixels might be shifted into these spaces and no hole would be identified in these locations. Therefore, these "empty" locations will not be included in the hole filling process. This effect is called background leakage. Figure 2a shows the leakage caused by the VSRS approach (artifacts on the flower). In our approach, in order to address this issue we separate each frame into three non-overlapping depth layers using two threshold values. We start the transformation of the pixels to the virtual camera plain from the foreground layer, followed by middle ground, and, lastly, the background. We limit the pixel transition such that the pixels from lower layers cannot be shifted within the upper layer object's boundaries. As a result, we do not end up with the background leaking to the occluded areas (Figure 2b). The next step in our process involves filling the occluded areas.

B. *Holes Filling Using Interpolation*

At the beginning of this stage, the occluded regions are classified as cracks (region sizes less than 0.3% of the frames width) and holes (rest of the occluded regions). The cracks are filled using nearest neighbor interpolation (similar to the VSRS approach) as described by following equations 3 and 4:

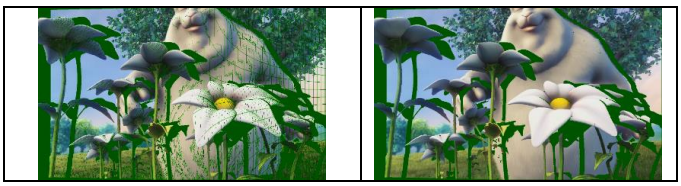
(a) (b)

Figure 2: Background Leakage in (a) VSRS, (b) our method.

$$Horizontal: new\_image(x, y) = image(x - 1, y) \quad (3)$$

$$Vertical: new\_image(x, y) = image(x, y - 1) \quad (4)$$

Figure 4a shows the primary synthesized view with cracks and holes shown in green. Figure 4b shows the same image after the cracks are filled by interpolation. A secondary synthesized view is generated using the further away captured view in a similar manner. At this stage, information from the secondary view, if available, is used to fill the existing holes in the primary synthesized view. As a result, some holes may be completely filled and some others may become smaller falling into the previously defined crack

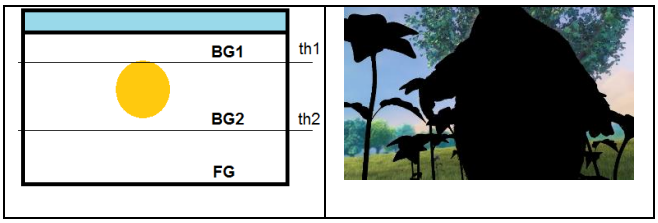


(a) (b)

Figure 3: (a) Primary synthesized view with cracks and holes shown in green; (b) cracks are filled by interpolation.

 ISBN: 978-1-61208-533-3

Figure 4: Image resulted after the remaining holes were filled using information from the secondary synthesized view.



(a) (b)

Figure 5: (a) Simple Parallel Background (BG) Composition, (b) background 1 of the scene using simple parallel BG.

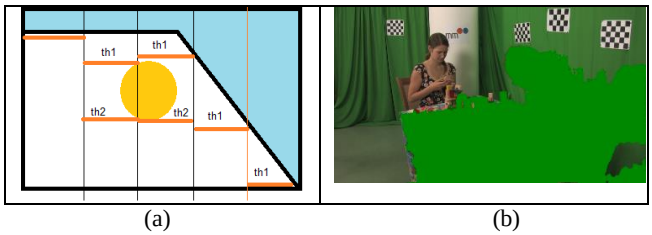
category. Figure 4 shows the resulting frame. Unlike VSRS, which fills all the holes using interpolation, we use temporal background information to partially fill the remaining holes.

C. Hole Filling Using Background Information

As background tends to remain unchanged within a scene, we decided to use it for filling holes at this stage. Background separation is a challenging task, since defining what is background, depends on the scene composition and the subjective opinion of the viewer.

A simple approach is to define a single threshold based on the depth map of the whole scene. This approach can successfully handle the majority of outdoor scenes where the background is at the horizon and parallel to the camera's plane and the foreground objects are much closer to the viewer than the background. A more accurate approach, which we chose for our implementation, is Otsu's method, which chooses the threshold to minimize the intra class variance of the black and white pixels of the provided depth map [10]. Figure 5a shows the two thresholds (th1 and th2) used to define background 1, background 2 while Figure 5b shows background 1 for the outdoor scene with all the objects in front of th1 removed (dark regions).

For more complex, usually indoor scenes, the background may not be parallel to the camera's plane and its distance can vary from frame to frame (such as the green curtain in the Poznan Blocks sequence, Figure 6b). For such scenes we developed a different approach, where we separate each frame into five vertical slices, as shown in Figure 6a, compute the depth thresholds for each of them using the same Otsu's method, and merge them into a single



(a) (b)

Figure 6: (a) Slicing Complex Background into 5 vertical regions, (b) resulting background using vertical slicing.

Figure 7: (a) Misalignment of the color image and depth map (b) part of the hair on the right of the hole is identified as a background.

background image (image shown in Figure 6b). The reason for choosing five slices for the frame was the fact that the average width of the foreground objects in our test set was approximately one fifth of the frame's width (Figure 6a).

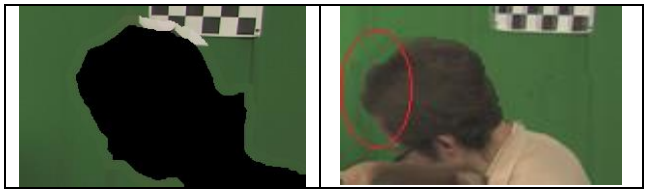
The complex scene approach can be used for all sequences, but since it is computationally more demanding, an automated classification into "simple" and "complex" background scenes is preferable.

For all concurrent frames we separated the background with the mentioned above approach and used SURF [5] to geometrically translate previously saved background image to the current camera's physical plane. After translation we filled the holes in the saved background image with the new available information from newly extracted background image, effectively increasing the coverage for future hole filling process.

There are cases where due to inaccuracies of the depth map, the foreground object's edges in the depth map are not aligned with those in the color image (Figure 7a), leading to various unwanted artifacts such as edge deformation of the foreground body (Figure 7b). Figure 7b shows how this mismatch will end up with object information on the other side of the hole as part of the background (see circled area which is part of the hair). Any effort to use the background information to interpolate or warp in order to fill the hole will result in foreground object information to be used as a "fill" for the hole as shown in Figure 8. In the interest of reducing edge artifacts and increasing the background coverage for further hole filling, we found through the tests that deleting 5 edge pixels from the background image (see Figure 9a) and then interpolating pixels from the holes' edges using 16 background pixels, effectively increases background coverage as shown in Figure 9b.

 ISBN: 978-1-61208-533-3

Figure 8: Hole's Edge Deformation



(a) (b)

Figure 9: (a) Shows our background interpolation step expanding the hole and (b) shows the resulting artifacts using VSRS.

We take advantage of temporal redundancies in the background by tracking all the frames in a scene and identifying newly exposed background areas due to movement of foreground objects. As one would expect, this "extending" background allows us to cover more holes for the later frames, and it is efficient if we have relatively constant background and moving foreground objects.

The next stage involves separating the remaining occluded regions into cracks and holes once more and filling the cracks with the previously described interpolation process. The remaining holes are filled using the warping method presented in [1].

D. Warping

The main difference in our warping process is the use of the edge mask. Since the human visual system is very sensitive to the vertical line distortions, we used the mask in order to stop the warping from deforming these lines. In order to compute the background edge mask we used Sobel edge detector on Luma component of the YUV frame. The warping process does not warp the background pass the detected edges and thus not destroying the background structure.

The second difference in our approach from the aforementioned one, is that in our implementation we warp/use background information that corresponds only to 85% of the hole's width, resulting in a better visual quality as we avoid excessive background warping. In the case where background is not available, such as the regions close to the edge of the image, we interpolate existing pixel values to fill in the hole.

E. Second View Interpolation

As already mentioned, for view interpolation we use information from the second closest from the virtual camera

(a) (b)

Figure 10: (a) Final image synthesized using VSRS, (b) final image using our method that correctly copies foreground information from the second view.

position view was utilized as most reliable technique for holes filling, since it contains information from the real camera view. To this end, a secondary synthesized view was generated solely based on the further view by following the same procedure as creation of the primary synthesized view. Once the secondary synthesized view is generated, the holes in the primary synthesized view are filled by corresponding available areas in the secondary synthesized view. In order to make sure that the objects closer to the new camera plane are not obscured by background, the hole filling is performed from background towards foreground (using depth map information).

As we can see from Figure 10a, in the case of VSRS the background information from second real view was copied over the foreground leave. Figure 10b shows the result of our method, where the flower and the leaves are complete, and there is no ghosting effect on the rabbit's hand.

III. METHODOLOGY AND TEST RESULTS

To evaluate the performance of our algorithm, we

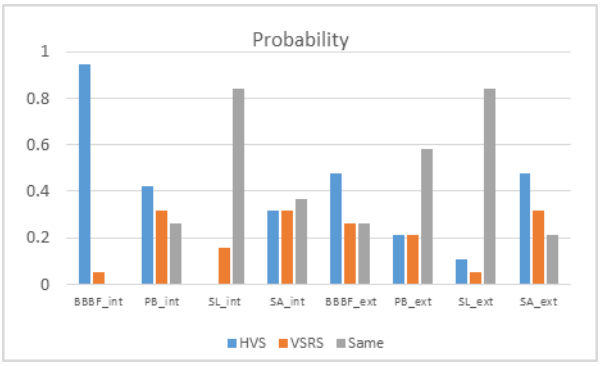


Figure 11: Probability for one of the method in each sequence to be chosen by the viewer.

conduct subjective tests and compare our synthesized views with those generated by the state-of-the-art VSRS [4]. The full paired comparison evaluation methodology is used for

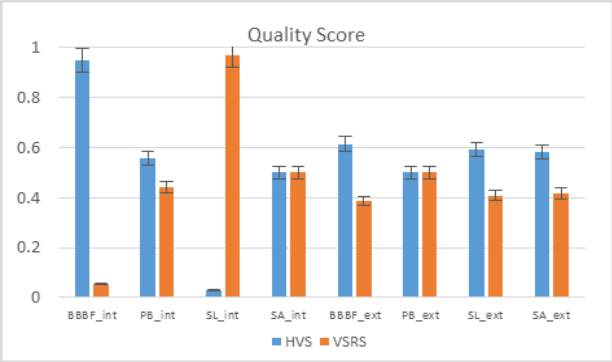


Figure 12: Quality Scores

our subjective tests [8]. A pair of the images is compared, with the subject asked to choose if either the "Left" or "Right" image is of better quality, or both are the "Same". For this evaluation we use four sequences recommended by MPEG [9]: "Soccer Linear2", "Soccer Arch1", "Poznan Blocks", and "Big Buck Bunny Flowers". We synthesize the required number of the virtual views between the provided real ones at the specified virtual points in space according to [9] using our approach and VSRS. We synthesized views using the two closest real views in the case of interpolation and using single closest view for extrapolation as the most appropriate case for FN. All sequences have Arch camera arrangement with different angle of convergence to the scene, except "Soccer Linear 2", which has linear camera arrangement. 19 subjects participated in the test. All the subjects are screened for the color blindness and vision acuity (Snellen and Ishihara charts) before conducting the test. Also to make them familiar with the test process, there was a training session using two test sequences ("Balloons" and "LoveBird1" [9]). After collecting test results, outliers were detected using circular triads method with defined threshold [8].

We use the Bradley-Terry model (BT) [7] combined with the maximum likelihood criterion as described in [8] to convert the results into the quality score metric. The pair ties are incorporated where they are available [6].

Figures 11 and 12 illustrate the subjective test results of our proposed method with those of VSRS for interpolation (marked as "int" for interpolation and "ext" for extrapolation in Figures 11 and 12) and extrapolation for the test sequences with 95% confidence interval.

As it can be observed, the "Big Buck Bunny" (BBBF) sequence shows significant improvement in both extrapolation and interpolation tests. The main reason for that is the fact that this sequence has color image perfectly aligned with the generated depth map and that the movement of the flowers and the rabbit exposed additional background that was stored and later used for the holes filling. The temporal background hole filling process bundled with the threshold based view synthesis handles this very well, improving overall quality of the synthesized view.

The "Poznan Block video" (PB) sequence shows small improvement, due to the overall low quality depth map, that does not align with the color image. "Soccer Arch's" (SA) modest gain, on the other hand, comes from the fact that the cameras' locations were far away apart and the camera calibration parameters were off. The misalignment of the left and right views is obvious on the synthesized views, making it a hard task to fill in the large holes. Our warping technic helps to slightly improve over VSRS.

In the case of "Soccer Line2" (SL), although it looks like the videos have completely different quality score, the results a priori show no statistically significant preference for HVS or VSRS, as the "same" option was selected in 84% of the cases (Figure 11). Video produced by our method, does not show any significant improvement over VSRS, since the scene's objects are located far from the camera plane and both foreground and background have insignificant differences in depth values. There are no artifacts due to the small shift of the objects in the scene.

IV. CONCLUSIONS

We present a novel view synthesizing scheme which utilizes unique techniques, such as background decomposition in three layers, background edge dilation, vertical interpolation and edge aware warping, to improve the overall visual quality. Performance evaluations have shown that our method yields a significant visual improvement over VSRS for FTV.

REFERENCES

[1] I. Koreshev, M. T. Pourazad, and P. Nasiopoulos, "Hybrid view-synthesizing approach for multiview applications," 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012, pp. 1-4.

[2] ISO/IEC JTC1/SC29/WG11 MPEG Document N11631, "Report on Experimental Framework for 3D Video Coding," Guangzhou, China, October 2010.

[3] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross, "Nonlinear disparity mapping for stereoscopic 3D," ACM SIGGRAPH, 2010, pp. 75.

[4] ISO/IEC JTC1/SC29/WG11, MPEG, "View Synthesis Software Manual," Sept. 2009, release 3.5.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Computer vision and image understanding, 2008, pp. 346-359.

[6] http://www.stats.ox.ac.uk/~caron/code/bayesbt/ , 2017

[7] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," Biometrika, vol. 39, 1952, pp. 324-345.

[8] J. S. Lee, L. Goldmann, T. Ebrahimi, "A new analysis method for paired comparison and its application to 3D quality assessment," Proceedings of ACM Multimedia, 2011, pp. 1281-1284.

[9] V. Baroncini, M. Tanimoto, O. Stankiewicz, "Summary of the results of the Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation," MPEG2016, Geneva, June 2016.

[10] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, No. 1, 1979, pp. 62-66.

 ISBN: 978-1-61208-533-3