HIGH QUALITY VIRTUAL VIEW SYNTHESIS FOR IMMERSIVE VIDEO APPLICATIONS

by

Ilya Ganelin

B.S. in Electrical and Electronics Engineering, Tel Aviv University, 2008

M.Eng. in Electrical Engineering, University of British Columbia, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2018

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, a thesis/dissertation entitled:

---

submitted by Ilya Ganelin in partial fulfillment of the requirements for

the degree of Master of Applied Science

in Electrical and Computer Engineering

Examining Committee:

Dr. Jane Wang
Supervisor

Dr. Panos Nasiopoulos
Supervisory Committee Member

Dr. Victor Leung
Supervisory Committee Member

Additional Examiner

Additional Supervisory Committee Members:

Supervisory Committee Member

Supervisory Committee Member

Abstract

Advances in image and video capturing technologies, coupled with the introduction of innovative Multiview displays, present new opportunities and challenges to content providers and broadcasters. New technologies that allow multiple views to be displayed to the end-user, such as Super Multiview (SMV) and Free Viewpoint Navigation (FN), aim at creating an immersive experience by offering additional degrees of freedom to the user. Since transmission bitrates are proportional to the number of the cameras used, reducing the number of capturing devices and synthesizing/generating intermediate views at the receiver end is necessary for decreasing the required bandwidth and paving the path toward practical implementation.

View synthesis is the common approach for creating new virtual views either for expanding the coverage or closing the gap between existing real camera views, depending on the type of Free Viewpoint TV application, i.e., SMV or 2D walk-around-scene-like (FN) immersive experience. In these implementations, it is common for the majority of the cameras to have dissimilar characteristics and different viewpoints often yielding significant luminance and chrominance discrepancies among the captured views. As a result, synthesized views may have visual artifacts, caused by incorrect estimation of missing texture in occluded areas and possible brightness and color differences between the original real views.

In this thesis, we propose unique view synthesis methods that address the inefficiencies of conventional view synthesis approaches by eliminating background leakage and using edge-aware background warping and inter-pixel color interpolation techniques to avoid deformation of

foreground objects. Improved occlusion filling is achieved by using information from a temporally constructed background. We also propose a new view synthesis method specifically designed for FN applications, addressing the challenge of brightness and color transition between consecutive virtual views. Subjective and objective evaluations showed that our methods significantly improve the overall objective and subjective quality of the synthesized videos.

Lay Summary

New immersive video technologies such as Free Viewpoint TV and its subcategories Super Multiview and Free Navigation require huge amount of information to be transmitted to the end-user. View synthesis creates virtual views from the real ones and is the best way to address the bandwidth challenges of these technologies as well as the large variety of Multiview displaying technologies at the receiver end. Synthesized views, however, may suffer from visual artifacts, mainly caused by “occluded” regions did not exist in the real views and their color and intensity information have to be predicted through some kind of interpolation. In this work, we introduce a new view synthesis method that eliminates most of the problems of existing view synthesis approaches for Free Viewpoint TV applications, yielding better overall visual quality.

Preface

All of the work presented in this thesis was conducted in the Digital Multimedia Laboratory at the University of British Columbia, Vancouver campus.

A version of Chapter 2 has been published as a conference paper by Ilya Ganelin, Mahsa Pourazad, and Panos Nasiopoulos, “A View Synthesis Approach for Free-navigation TV Applications,“ at The Eighth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, IARIA 2017. M. T. Pourazad and P. Nasiopoulos were the supervisor on this project and were involved with research concept formation and manuscript edits. An extended version of this chapter was submitted as a journal paper authored by Ilya Ganelin and Panos Nasiopoulos, "An Efficient View Synthesis Scheme Based on Temporal and Spatial Information for Free Viewpoint TV Applications" to the IEEE Transactions on Multimedia in March 2018. I was the lead investigator responsible for all areas of research, data collection, as well as the manuscript edit. P. Nasiopoulos was the supervisor of this project and was involved with research concept formation and manuscript edits. M. T. Pourazad provided invaluable technical support and guidance as well.

A version of Chapter 3 has been submitted to an IEEE conference as Ilya Ganelin and Panos Nasiopoulos, “Color Matching for Free Viewpoint TV Applications using Gaussian Mixture Model”. I was the lead investigator responsible for all areas of research, data collection, as well as the manuscript composition. P. Nasiopoulos was the supervisor on this project and was involved with research concept formation, and manuscript edits.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BL	Background Leakage
BT	Bradley-Terry Model
CfE	Call for Evidence
DIBR	Depth Image Based Rendering
DM	Depth Map
FN	Free Navigation
fps	Frames Per Second
FTV	Free Viewpoint TV
GD	Geometrical Distortions
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
HVS	Human Visual System
MPEG	Motion Picture Experts Group
MVD	Multiview Video and Depth
PSNR	Peak Signal-to-Noise Ratio
SMV	Super Multi View
SURF	Speed Up Robust Features
VS	View Synthesis
VSRS	View Synthesis Reference Software
Y-PSNR	Peak Signal-to-Noise Ratio of Luma channel only

Acknowledgements

I would like to start by expressing my most sincere gratitude to my supervisor and mentor **Dr. Panos Nasiopoulos** for his support through the past three years and patient guidance and inspiration through the thesis.

I would also like to thank **Dr. Mahsa T. Pourazad** for her constant help and support during different stages of this thesis.

I am also grateful to my lab mates, Dr. Ronan Boitard, Stelios Ploumis, Dr. Hamid Palangi, Dr. Hamid Tohidypour, Ahmad Khaldieh, Joseph Khoury, Fujun Xie, Anahita Shojai, Maryam Azimi, Dr. Sima Valizadeh, Abrar Wafa, Pedram Mohammadi, Basak Oztas, Dr. Davood Karimi, and Dr. Hossein Bashashati who helped me through feedback and support during the different stages of my research.

Dedication

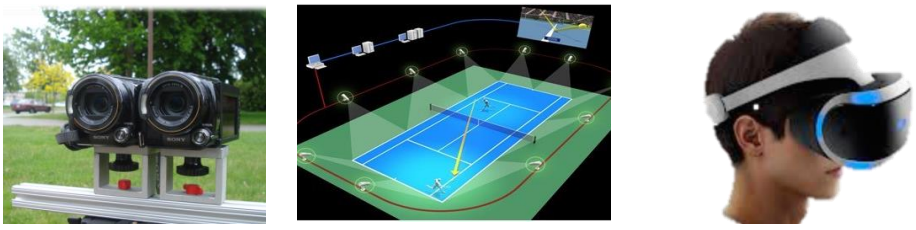
To my beloved family.

1. Chapter 1: Introduction

Nowadays, multiple capturing and display technologies (Free Viewpoint TV), such as Super Multiview (SMV) (Figure 1.1a), Free Navigation (FN) [1] (Figure 1.1b), and Virtual Reality (VR) [2] (Figure 1.1c), are trying to provide viewers with a more realistic impression of camera captured or computer-generated scenes. Each one of the above-mentioned technologies is trying to create an immersive experience by offering additional degrees of freedom to the user. Life-like impression of the perceived surroundings can be achieved using different technological approaches and different physical devices.

There are a few major differences between FN and SMV, both in capturing and displaying content. First, the physical distance amongst content capturing cameras differs. SMV provides more densely spaced views (eighty or more), whereas FN's goal is to provide smooth transition between views that are spaced further apart than those in SMV (up to ten meters between neighboring cameras). Figures 1.1a and 1.1b show a camera setup for Stereoscopic and Free Navigation respectively. The second difference is how that content is projected to the viewer. SMV is 3D content displayed on 3D enabled monitors (see Figure 1.2) that can show multiple number of the views at once, whereas FN is 2D content that the viewer can sweep through.

VR provide the viewer with partially or fully generated images to replicate a real environment and simulate the user's physical presence in this environment. That requires multiple images to be



(a) (b) (c)

Figure 1.1. Camera setups: (a) Super Multiview, (b) Free Navigation, (c) VR.

(a) (b)

Figure 1.2. Super Multiview 3D (a) Alioscopy 8 views display, (b) Dimenco 28 views display.

transmitted to the user's display device of choice at any given time based on his/her position in space. Depth maps, usually, are not part of these technologies, due to absence of depth capturing devices in the current setups, so the need for their synthesis still exists.

1.1 Motivation

Advances in the image and video capturing technologies, coupled with the introduction of more affordable Multiview and 3D displays, present new opportunities and challenges to content providers and broadcasters. New technologies that require multiple camera views to be displayed to the end client, such as Super Multiview (SMV) and Free Viewpoint Navigation (FN), were recently introduced to the market [5]. All these technologies provide viewers with a realistic impression of the scene by allowing them to freely navigate through the scene and perceive the scene's depth [3].

View Synthesis (Figure 1.3) plays an important role in the above-mentioned technologies. The need for view synthesis arises from the fact that its usage can significantly reduce the amount of content that should be captured, stored, and transmitted. The transmission bitrate is proportional to the number of the cameras used to capture the scene; hence, by reducing the number of capturing

devices used in the setup and synthesizing/generating intermediate views at the end device instead, has the potential to decrease the required bandwidth.

As the quality of the synthesized views should be such that yields the best possible visual experience to the user regardless of the application in hand, the view synthesis process should try to match the quality of the "missing" views to that of the real ones.

View synthesis may affect differently each of these technologies, as the challenges are different for each application. In case of Super Multiview, equipment manufacturers and content providers supply viewers with a large number of views (ten to eighty), in order to improve transition between sweet spots. As the number of views varies from one display to another, synthesizing virtual views becomes an essential task, beyond the obvious necessity for bandwidth savings.

Free Navigation (Figure 1.1b) enables the viewer to seamlessly transition between adjacent cameras that surround the scene. Camera arrays arranged around the scene, usually in parallel or arch converging mode, are used for offering an immersive experience; examples may be watching a soccer game from different angles or watching and listening a symphony from different positions in a theatre.

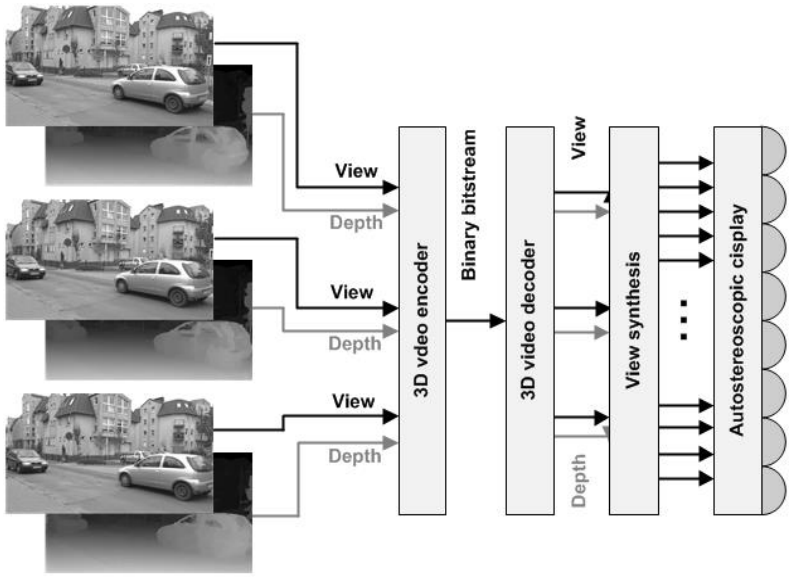


Figure 1.3. View Synthesis process model, from transmission to representation.

There are a few major differences between FN and SMV in both content capturing and displaying aspects. First, the physical distance amongst content capturing cameras varies. SMV provides more densely spaced views, whereas FN's goal is to smooth transition between views that are spaced further apart than those in SMV (up to ten meters between neighboring cameras). In such cases, where captured content is further apart due to the physical distance between cameras (see Figure 1.1b), the quality of the synthesized views is even more important, as there is no adequate information from neighboring views that can be used for filling the occluded regions in rendered views [3].

The second difference is how that content is projected to the viewer. In SMV, 3D content is displayed on 3D enabled monitors that can show multiple number of views at once, whereas in FN 2D the viewer can sweep through the content.

In SMV and FN implementations, it is common for the majority of the cameras to have dissimilar radiometric lens characteristics and be pointed to the scene from a different viewpoint, often yielding significant luminance and chrominance discrepancies among the captured views [20]. As a result, synthesized views may have visual artifacts, caused by incorrect estimation of missing texture in occluded areas and possible brightness and color differences between the original real views.

Although several methods have been designed for improving the visual quality of the synthesized views for 3D and Multiview applications, unfortunately they do not directly apply to FN implementations. This is much more evident in the case of color and discrepancies between views. In Multiview applications, the camera base is narrow and color differences between the real views are due to the optical differences of the cameras and lenses. However, in case of FN, the cameras are far apart, and the inconsistencies in brightness levels and colors are not due to miss-calibration

or the optical differences between the cameras, but in fact, they represent the actual scene viewed from different angles.

Existing view synthesis methods [4] fail to tackle the problem of color prediction for synthesized views in this case, thus creating visible color, shade, and brightness related artifacts. Color and brightness estimation of the objects in the scene is one of the big challenges that we addressed.

1.2 Thesis Organization

The rest of the thesis is structured as follows. Chapter 2 gives an overview of existing view synthesis and color matching works. Chapter 3 presents in detail our first proposed view synthesis and occlusion filling approaches. Chapter 4 introduces our color estimation method designed for FN applications. Finally, conclusions, discussions, and future work are drawn in Chapter 5.

2. Chapter 2: Background

2.1 View Synthesis

Depth Image Based Rendering (DIBR) [6] has been adopted by industry as the most efficient approach for view synthesis, since it provides the depth information of the scene that can be used for intermediate view generation. DIBR still suffers from artifacts arising from the fact that some regions of the synthesized views are not visible in the original images, which results in occlusions (pixels without information) that are filled using varies techniques. The detailed explanation of the origin of the occlusions can be found in recent work by Zhu et al [7].

The DIBR view synthesis model uses point cloud representation of the objects in the scene. Each pixel location can be derived from the depth map and the camera's known position in space. Based on that data, each color pixel is translated to the virtual camera plane separately. The translation process creates occlusions in a way similar to a disparity-based synthesizing approach that was dealing only with horizontal disparity of the objects in the scene, which suitable for capturing the scene with parallel camera arrangement [8]. Additionally, neighboring pixels can be mapped to the non-integer location and rounded to the same pixel, effectively creating small occlusions. Due to the sparse camera locations, synthesized content of FN will have bigger occlusions in comparison with SMV. An alternative to the single pixel transition approach is using triangular meshes in order to reduce the number of the occlusions [9].

Recently, view synthesis has been generalized for the circular camera arrangement described in [8] as part of the 3D-HEVC standard based on the 3D in Multiview Video and Depth format (MVD). For MVD, the depth information of the pixel determines its location in the virtual plane based on its coordinates in space, and camera's extrinsic, intrinsic, and translation parameters.

A series of color cameras bundled with depth cameras can capture the scene from multiple viewpoints located parallel, in arch mode, or around the scene. In order to reduce the amount of the information that should be captured and transmitted, the FN and SMV technologies use a depth map (DM) plus a color image pair for generating all the transitional views between them (DIBR approach similar to [4]). The displaying device determines the number of the synthesized views in Multiview that can vary from eight to over a hundred. For FN, the number of the virtual views in between the real views defines how smooth the transition is. The main difference between Free Viewpoint Navigation and the Super Multiview technologies is the distance between capturing devices. The wider disparity in FN will affect the size of the occlusions, thus affecting the quality and accuracy of the synthesized views.

In general, the occlusion filling methods can be divided into three categories depending on the source of information these occlusions will be filled with. The first category involves approaches that use temporal information from previous and future frames to obtain additional temporal predictions of the synthesized frame [10][11][12]. In this case, forward motion vectors are applied in the temporal sense, computed in the reference views and warped in the synthesized view to obtain up to four temporal predictions, which are blended together with the DIBR predictions using either an average or adaptive approach [12]. This helps to extract information on occluded areas. Luo et al. [13] utilized the random walker algorithm for foreground extraction combined with dynamic background reconstruction and used that background for occlusion filling.

In the second category, patches are extracted from available real views and used for occlusion filling in newly synthesized image [14][15][16]. The downfall of both categories is that not all the occlusions can be visible in space or temporal neighboring frames, leaving some of them not filled thus causing unwanted artifacts.

In the last category, missing pixel color information can be predicted based on the observation that the background color and texture should be continuous. Inter pixel color interpolation (inpainting) is the most commonly used approach where occlusions are filled by interpolating neighboring color pixel information with this process sometimes followed by a smoothing filter such as in the View Synthesis Reference Software (VSRS) [8]. Another approach involves warping the neighboring background into the occluded region [17][18]. Even though both approaches can cover all the occlusions, in doing so they can create unwanted visible artifacts. Warping will cause foreground stretching or background line distortions, whereas interpolation alters background's visible patterns.

In FTV, visual artifacts are Background Leakage (BL) and occluded regions. BL is caused by the fact that objects in a scene have volumetric nature, resulting in several different depth values for the same object. While transitioning color pixels from the real camera to the new virtual camera plane, depth map values will create discontinuities in solid objects, which will end up filled with background information, thus resulting in visual artifacts.

2.2 Color Matching

Color matching attracted a lot of attention over the years, with one of the first methods proposed by Reinhard et al. [21]. This method used statistical analysis to impose one image's color characteristics on another and achieved color correction by choosing an appropriate source image and applying its characteristics to another image. A representative color correction method for Multiview applications is presented in [22]. This method uses block-based disparity estimation to find matching points between all the views and efficiently estimate the average color that is used for color correction. In [23] an example-based color transfer algorithm (EBVCC) achieves color matching by preserving the gradient details of the source using Laplacian pyramids. A more recent

method designed to correct color differences in Multiview video sequences uses a dense matching-based global optimization framework [24]. Dong-Won et al. attempted to solve the color mismatch problem in Multiview applications by finding correspondences between the source and target viewpoints and calculating a translation matrix using a polynomial regression technique in CIELab color space [25].

All the above-mentioned methods focus on Multiview applications. However, in the case of FN, matching colors between views is not the "ideal" objective, as colors in real views may be different due to the fact that the actual scene is viewed from different angles and brightness. The actual color of the same object in two real images can be different due to the brightness, sparse camera location, and orientation in the scene. Thus, the objective in Free Navigation applications is to create a seamless transition between two real cameras through the scene using synthesized views whose brightness and color changes depend on their position with relation to the real views. To the best of our knowledge, there is no published work on color estimation for FN applications at the time of writing this Thesis.

3. Chapter 3: View Synthesis and Occlusion Filling

3.1 Introduction

In this chapter, we propose a unique view synthesis method, which addresses present shortcomings. First, a layer-based view synthesis step is introduced that eliminates background leakage. Second, an occlusions classification approach with spatial filling is applied, utilizing both extended edge-aware background warping and background inter-pixel color interpolation techniques to avoid deformation of foreground objects. Lastly, better hole filling is achieved by using information from a temporally constructed background over several frames. The following subsection describe our method in detail.

3.2 Our Proposed Method

Figure 3.1 shows the workflow of our view synthesis method. As a first step, we address the background leakage (Figure 3.2) by using our layer-based translation process to synthesize the virtual view. As a second step, we classify occlusions based on their size and fill-in the smaller ones using inter pixel color interpolation. After that, view blending is applied to fill-in larger occlusions with the information available from the second view. Since some occlusions will remain, the second step is repeated separating them into small and large holes. The next step involves the use of our temporally combined background information to fill the bigger occlusions, as the smaller ones were filled using inter pixel color interpolation. The final step of our method uses edge-aware background-only warping to fill the remaining occlusions. For the view extrapolation case, where only one camera view and the corresponding depth map are used, the above process is identical with the exception that the blending stage is not applicable.

In the following subsections, we describe the different parts of our approach in detail.

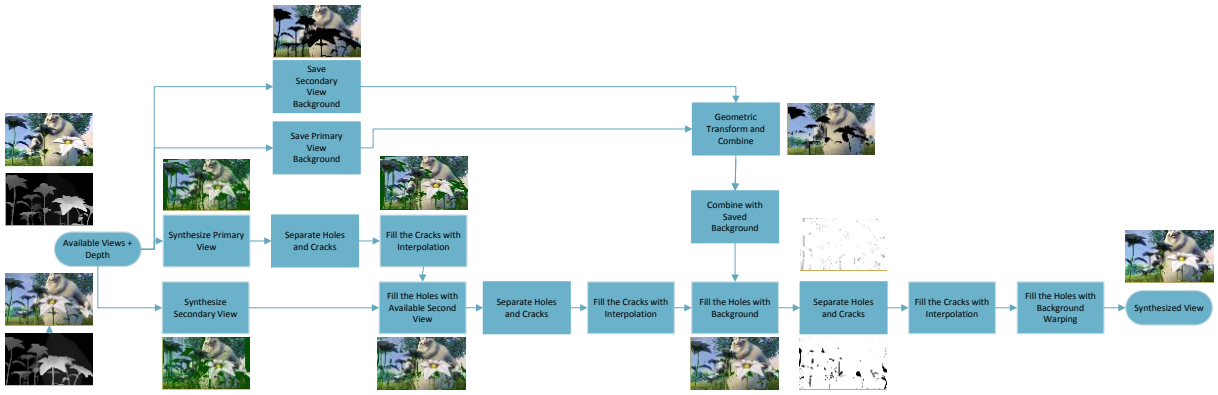


Figure 3.1. Block diagram of our view synthesis method.

3.2.1 Layer-based View Synthesis

Our first objective and first part of our approach is to address the background leakage artifacts (Figure 3.2). To this end, the depth information for each pixel and the point cloud method expressed in equations (3.1) and (3.2) were used to transition the color pixels from the available real camera plane *(r)* through the 3D position in space to the virtual camera plane.

$$\begin{bmatrix} Zr \cdot Xr \\ Zr \cdot Yr \\ Zr \\ 1 \end{bmatrix} = Pr \cdot P^{-1} \cdot \begin{bmatrix} Z \cdot X \\ Z \cdot Y \\ Z \\ 1 \end{bmatrix} \tag{3.1}$$

$$P = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R & -R \cdot T \\ O^T & 1 \end{bmatrix} \tag{3.2}$$

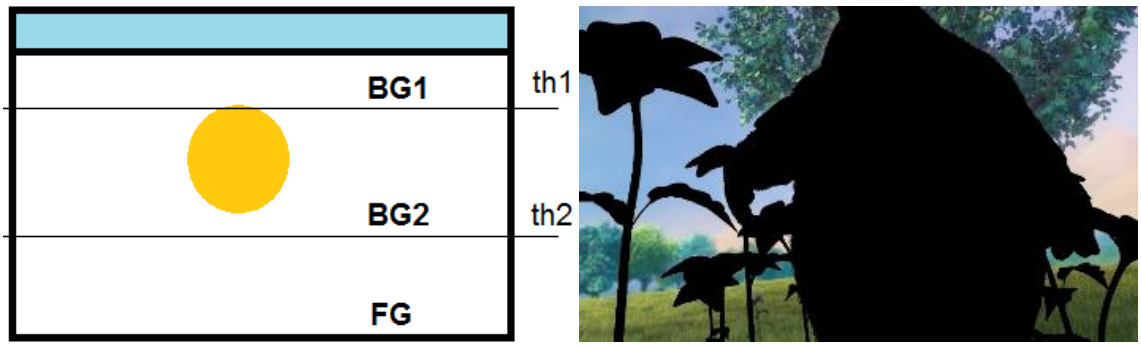
where *X*, *Y*, *Z* are coordinates of the pixel in space, *R* is a rotation matrix, *T* is a translation matrix, and *f* is the focus distance of the camera. Projection matrices *P* and *Pr* consist of the camera's extrinsic, intrinsic, and translation parameters.

(a) (b)

Figure 3.2. Background Leakage: (a) State-of-the-art view synthesis, (b) Our Method.

In our approach, instead of moving all the pixels from the captured camera plane to the virtual plane based on their depth map value one by one, we divide the depth map of the scene into three uncoupled layers: background, middle ground, and foreground using the Background Separation process.

The background separation is a challenging task, since defining what background is depends on the scene's composition and the subjective opinion of the viewer [3]. We start by classifying the background into two classes: parallel to the camera plane ("simple", Figure 3.3a), such as the sky behind the rabbit in the "Big Buck Bunny" sequence [26], and non-parallel ("complex", Figure 3.4a), such as the green curtain behind the table in the "Poznan Blocks" sequence [27]. Simple background separation begins with the declaration of two global thresholds (th_1 and th_2) using

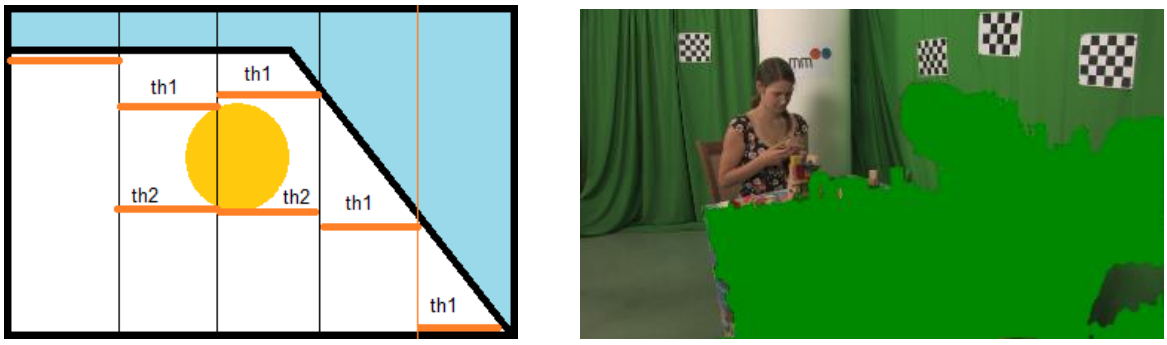


(a) (b)

Figure 3.3. (a) Simple Parallel Background Composition, (b) background 1 of the scene using simple parallel background model.

Otsu's method [28] (Figure 3.3b), which chooses the threshold to minimize the intra class variance of the black and white pixels of the provided gray level depth map [29]. The approach to the complex background scenes is more sophisticated, since the global threshold will inaccurately define the background of the image. For that case, we apply the threshold to each pixel-wide column in the depth map (Figure 3.3b). This step decreases the speed of the algorithm, while it yields excellent results. In our previous work, we used a number of vertical slices instead of pixel-sized columns, which gave us better speed performance, but the resulted background separation and visual results depended on the scene's content. The same background separation process is used later for the temporal background fill step.

After separating the depth map scene representation into three separate layers, we start the background leakage elimination process by first synthesizing the virtual view depth map (Figure 3.5). We observe that the foreground objects (brightest flowers) have dark lines (holes) which are caused by the volumetric nature of the objects that results in different disparity values for the same object. Since our goal is to remove the small holes inside the objects at each layer, we start a dilation process, which doubles each bright pixel in each direction. This step eliminates all the small holes inside the foreground the objects (Figure 3.6), but it also expands the edges by two



(a) (b)

Figure 3.4. (a) Slicing Complex Background into five vertical regions, (b) resulting background using vertical slicing model.

pixels. In order to bring the edges back to original position, we enlarge the black background pixels by two pixels in each direction (this process is known as erosion, Figure 3.7). We repeat this process for the middle-ground, separating accurately the three regions and eliminating background leakage.

We continue the transition of the background pixels in exactly the same manner as the middle ground (Figure 3.8). This approach insures that the color information from a lower layer (farther away from the viewer) will not be introduced within the borders of the upper layer object (Figure 3.2).

Currently, our layer based view synthesis step is limited to only three layers, since based on our observations; the majority of the scenes have objects of interest in only three space regions. Further studies on a larger set of video streams may be needed to determine if that is true for all the cases.

Figure 3.5. Virtual View Depth Map.

Figure 3.6. Virtual View Depth Map after dilation.

Figure 3.7. Virtual View Depth Map after erosion.

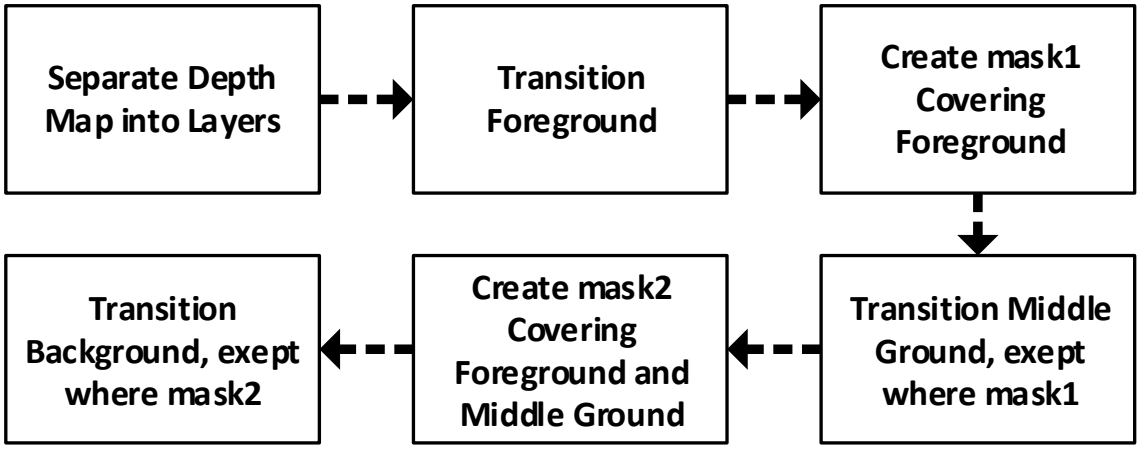


Figure 3.8. Pixel translation diagram to virtual camera plane.

3.2.2 Occlusion Classification and Inter Pixel Color Interpolation

Occlusions, which are areas of the synthesized image with missing color pixel information, can be classified into two categories: border and non-border [30]. Border occlusions occur due to the physical positioning of the cameras, their relative rotation to the scene, which results in border parts of the synthesized image being not visible. Since the majority of those can be filled from a secondary synthesized view, the blending stage of our model deals with this type of occlusions. If the part of the scene is not visible in both real views, we utilize inter pixel color interpolation.

Objects in the foreground that obscure parts of the background, or other objects behind them, which should be visible in the synthesized view, cause the large non-border image occlusions, called holes. Due to the motion of the foreground objects and camera, the occlusions' locations vary over time and produce different discontinuities at different time instances in the synthesized view. Thus, part of the missing information may be available in future or past frames [10] ("temporally combined background information"), and may be covered using information from the secondary image ("view blending"), and for the remaining occlusions we can apply edge-aware background warping.

Rounding errors in the pixel translation process and small depth discontinuities will cause small non-border image occlusions. Our empirical tests have shown that occlusions smaller than 3% of the frame's width can be efficiently and without any significant visual degradation filled using the inter pixel color interpolation algorithm [8]. These occlusions are defined as cracks in this work.

The inpainting technique (interpolating neighboring pixel's color into occluded region) has been widely used, as it is the fastest and most straightforward approach for filling occlusions [8]. Even when some of the big occlusions cannot be completely filled by spatial or temporal occlusion-

filling approaches, they can still be reduced in size, significantly reducing the size of artifacts in the final image.

Based on these findings, we use inter pixel color interpolation as an intermediate step between larger occlusion fillings steps (see Figure 3.9), interpolating background pixel value in order to fill small occlusions in both vertical and horizontal directions. Using only background pixels helps us preserve the edges and the silhouettes of the foreground objects. Equation 3.3 describes the decision process for choosing the start or the end of a crack as a source for the pixel's color to be filled in, based on the depth value of those pixels:

$$\exists p_c(x,y) = \begin{cases} p_c(x_s, y_s), & p_d(x_s, y_s) \leq p_d(x_e, y_e) \\ p_c(x_e, y_e), & p_d(x_s, y_s) > p_d(x_e, y_e) \end{cases} \tag{3.3}$$

where $p_c(x,y)$ is the color image pixel values at *(x, y)* locations and $p_d(x,y)$ is the sum of start/end (*s/e*) pixel of the depth map with its neighboring non zero depth value (*d)*. We use the sum of two pixels in order to eliminate the depth map inconsistencies around an object's edges, according to [31].

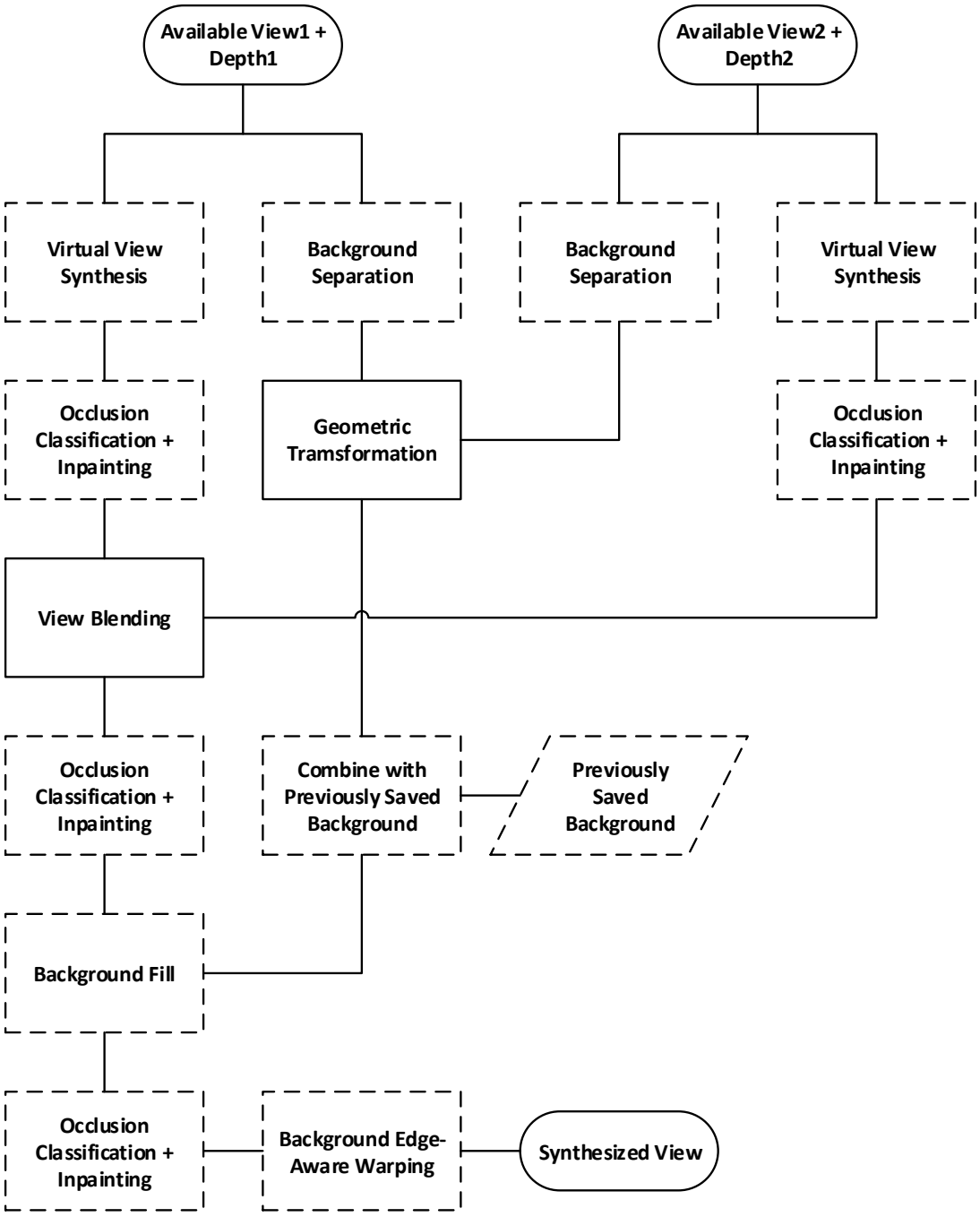


Figure 3.9. Block Diagram of our method where our contributions are highlighted in dashed line.

3.2.3 View Blending

Some parts of the scene that are not visible from a primary camera location can appear in the neighboring cameras due to the relative cameras' rotation to the scene. For this reason, after synthesizing the primary virtual view from the closest to the virtual point real camera frame, we fill all the occluded areas with available information from the secondary real view (the camera second closest to the virtual camera point). There are several view blending approaches, such as those described in [17] and [32], but both rely on color images only (without depth map) and perform image-based 3D warping. Since DIBR provides the renderer with information about the camera position and orientation in space (in configuration file format), we can easily calculate the distances from the real cameras to the virtual point in space. Based on this fact, in our approach we use the distance-based blending approach, where the closest to the virtual view real camera view is the primary view, upon which we synthesize the new image (Figure 3.10).

View blending will fill in the majority of the created occlusions with the real information from the available real camera views of the same scene.

(a) (b)

Figure 3.10. (a) Final image synthesized using VSRS, (b) final image using our method that correctly copies foreground information from the second view.

3.2.4 Hole filling using Temporally Combined Background Information

Background reconstruction has been widely used in [13] and [10][33], where temporally correlated information from both 2D video and its corresponding depth map are exploited to construct background video [13]. In contrast to the mentioned methods, our approach does not rely on motion flow to separate the background, but only on the provided depth maps. That helps us to classify even still foreground objects and achieve stable and accurate background reconstruction over time while using a rather fast and simple approach.

Extensive tests involving a large set of video streams have shown that a temporal window of eight frames yields an adequately complete background, which can be used for efficiently filling occluded areas in the current frame.

After separating the depth map into three layers, background (B), middle ground (M), and foreground (F) using 8 frames, we want to combine each layer from all the frames in to a single frame. In order to align the frames, we use the feature-matching algorithm SURF [34] to geometrically translate each background layer to the current virtual camera's coordinates. After translation, we fill the holes in the saved background frame with the newly available information using the recently extracted background frame [3]. The color pixel assignment decision process is described by the following equation:

$$\exists p_c(x,y) \in \begin{cases} F, \ p_d(x,y) > th_1 \\ M, th_1 > p_d(x,y) > th_2 \\ B, \ p_d(x,y) < th_2 \end{cases} \tag{3.5}$$

where th_1 with th_2 are two depth thresholds $(th_1 < th_2)$, *F*, *M*, and *B* are stand for foreground, middle ground and background respectively, $p_c(x,y)$ and $p_d(x,y)$ are the color and depth pixel values at location *(x, y).*

	VSRS (Y-PSNR, dB)				Our Approach (Y-PSNR, dB)			
QPs	**25**	**30**	**35**	**40**	**25**	**30**	**35**	**40**
Balloons	29.567	29.525	29.449	29.276	30.863	30.903	30.878	30.915
Kendo	27.975	27.935	27.858	27.733	29.790	29.757	29.678	29.46
Newspaper	26.451	25.858	25.751	25.644	25.496	25.547	25.499	25.496

Table 3.1. Y-PSNR objective tests results for Multiview content for three sequences and four QP levels.

Depth maps may be either captured during filming [35] or be synthesized using the real views by visual cue methods described in [36]. In both cases, they may have inaccuracies and misalignments with the texture counterpart, which will result in synthesized image objects' edge artifacts. In the interest of reducing edge artifacts and increasing the background coverage for better occlusion filling, we empirically found that deleting five edge pixels from the background image and then interpolating pixels from the holes' edges using sixteen background pixels significantly increases background coverage [3] as shown on Figure 3.13.

The saved background and middle ground are used for occlusion filling in the synthesized view at that stage. From our evaluations, we noticed that for the videos with foreground objects without significant spatial movement (such as the Rabbit from the "Big Buck Bunny" sequence) the background-filling step does not cover all the occlusions.

The separation of the remaining occlusions into large and small ones, namely holes and cracks, is the next step of our approach. The cracks are filled using the already mentioned inter pixel color interpolation technique, while background edge-aware warping, which is described in the following subsection, is used for filling the holes.

3.2.5 Background Edge-Aware Warping

For the remaining occlusions, we do not have any "real" pixel information from the neighboring views or from future and past frames. Since using interpolation at this stage has shown to yield artifacts, we apply an edge-aware background warping that integrates the nonlinear disparity mapping principles of [18] and [37] to efficiently fill the remaining holes.

Since the edges in the color image are important to the human visual system, our approach, unlike other methods, incorporates an edge mask to make sure that warping does not cross those edges throughout the process. In order to compute the background edge mask, we apply Sobel edge detector [38] on the Luma component (which represents the brightness of the image) of the YUV frames. Our occlusion warping process warps only up to 85% of the occlusion from the neighboring furthest area of the scene in the image, in contrast with [18] where the occlusion-size background is warped in fully. We found that usage of only 85% of the hole's size background pixels introduces a smaller stretching effect to the final image. Warping is performed by comparing the average of a group of pixels' depth values on both sides of the occluded region that we are about to fill (same approach as with the inter pixel color interpolation). If the area that we are going

(a) (b)

Figure 3.11. (a) Shows our background interpolation step expanding the hole and (b) shows the resulting artifacts using VSRS.

to warp does not have enough texture information, the approach will apply inter pixel color interpolation instead. Those cases include the border type occlusions at the edges of the image. This approach yields good results, as it does not create visual artifacts that are perceived by our visual system.

3.3 Objective Tests

For the objective tests, we employed the Luma-based Y-PSNR metric and compared our results with the methods presented in [12] and [39]. For VSRS, we used the linear mode with no blend option selected, as suggested by the methods we compare against. Table 3.1 shows the average Y-PSNR values in dB over 300 frames of the Balloons", "Newspaper", and "Kendo", sequences and for four compression levels, QP25, QP30, QP35, and QP40 for our method and VSRS. Using these results and results from [12] and [39], we calculate the gain in Y-PSNR using the VSRS values as a reference point. Table 3.2 shows the average gain for our method, Purica et al [12] method with average blending (P+Bavg) and the adaptive blending (P+Badapt), and motion compensated temporal interpolation (MCTI) method presented in [39]. We observe that, on average, our approach gains 1.9 dB in visual quality for the "Balloons" sequence, and 2.4 dB for "Kendo", while its performance drops to -0.55 dB for the "Newspaper" sequence. The first reason for the poor performance on the Newspaper sequence is that the depth map and real video are not aligned, affecting the accuracy of our background occlusion filling technique. Wrongly classified background, which is actually foreground in this case, is forced into the corresponding occlusions, creating artifacts. The second reason is the narrow overall depth in this sequence that interferes with the depth-layer separation process. Even though the number of visible artifacts is significantly smaller than VSRS, the layer based translational approach created a few big shift-based artifacts

that significantly reduce the performance of the Y-PSNR metric. Another depth level separation process, such as the Gaussian Mixture Model, might be able to address both of above-mentioned problems.

3.4 Subjective Tests

The first set of subjective tests compare our method against VSRS. For this implementation, the VSRS-General mode with quarter pixel precision has been chosen. We used the full-paired comparison evaluation methodology recommended in [40]. An LG 55EA9800 OLED TV in 2D mode TV set has been used for showing the sweep videos to the viewers.

We compared a pair of frames, with the subjects asked to choose if either the "Left" or "Right" image is of better quality, or both are the "Same". For these evaluations, we use four representative sequences recommended by MPEG [41] for Free Viewpoint Television: "Soccer Linear2",

	Balloons				**Kendo**				**Newspaper**			
QPs	**25**	**30**	**35**	**40**	**25**	**30**	**35**	**40**	**25**	**30**	**35**	**40**
Our Approach (Y-PSNR gain)	1.3	1.38	1.43	1.64	1.82	1.82	1.82	1.73	-0.95	-0.31	-0.25	-0.15
P+Bavg (Y-PSNR gain)	0.35	0.37	0.35	0.32	-0.12	-0.09	-0.02	0.04	0.71	0.75	0.72	0.64
P+Badapt (Y-PSNR gain)	0.37	0.38	0.37	0.31	0.39	0.36	0.36	0.31	0.65	0.69	0.66	0.59
MCTI (Y-PSNR gain)	0.19	0.18	0.16	0.012	0.19	0.17	0.15	0.14	0.09	0.08	0.07	0.06

Table 3.2. Average Y-PSNR difference across Multiview Sequences with 4 QP settings for our approach, Purica (P+Bavg and P+Badapt), and motion compensated temporal interpolation (MCTI).

(a) (b)

Figure 3.12. Final view synthesis result of (a) VSRS and (b) our method for Big Buck Bunny Flower sequence.

"Soccer Arch1", "Poznan Blocks", and "Big Buck Bunny Flowers". We synthesized the required number of the virtual views between the provided real ones at the specified virtual points in space according to [41] using our approach and VSRS. View generation has been done using the two closest real views in the case of interpolation and single closest view for extrapolation, as the most appropriate case for FN. All sequences have an arch camera arrangement with each camera having a different angle of convergence to the scene, except "Soccer Linear 2", which has a linear camera arrangement. Nineteen subjects participated in our test. We screened all the participants for color blindness and vision acuity (Snellen and Ishihara charts) before conducting the tests. In addition, there was a training session using two test sequences ("Balloons" and "LoveBird1" [41]) to make them familiar with the test process. After collecting test results, three outliers were detected using the circular triads method with defined threshold and were removed from our results [40].

We use the Bradley-Terry model (BT) [42] combined with the maximum likelihood criterion as described in [41] to convert the results into the quality score metric. The pair ties are incorporated where they are available.

Figure 3.12 shows one frame of the synthesized views by VSRS (Figure 3.12a) and our method (Figure 3.12b). We observe that our view synthesis approach produces a better overall picture, significantly reducing the visual artifacts. Figures 17, 18 illustrate the subjective test results of our

proposed method with those of VSRS for interpolation ("int") and extrapolation ("ext") for the test sequences with 95% confidence interval. Figure 3.13 shows us the probability of the viewer choosing videos produced by our method over VSRS. Figure 3.14 shows the quality score that has been calculated using Bradley-Terry model from the probability results (Figure 3.13).

As it can be observed, the "Big Buck Bunny" (BBBF) sequence shows significant improvement in both the extrapolation and interpolation tests (Figure 3.14). The main reason for that is the fact that in this case the video sequence is perfectly aligned with the corresponding depth map and that the movement of the flowers and the rabbit expose additional background over time that is efficiently used by our temporal background hole filling approach. The temporal background hole filling process bundled with the layer based view synthesis handles this very well, improving overall quality of the synthesized view.

The "Poznan Block" (PB) video sequence shows small improvement, due to the overall low quality depth map, that does not align with the video sequence.

The "Soccer Arch's" (SA) modest gain, on the other hand, comes from the fact that this represents a FN case with the cameras located far away from each other and the camera calibration parameters

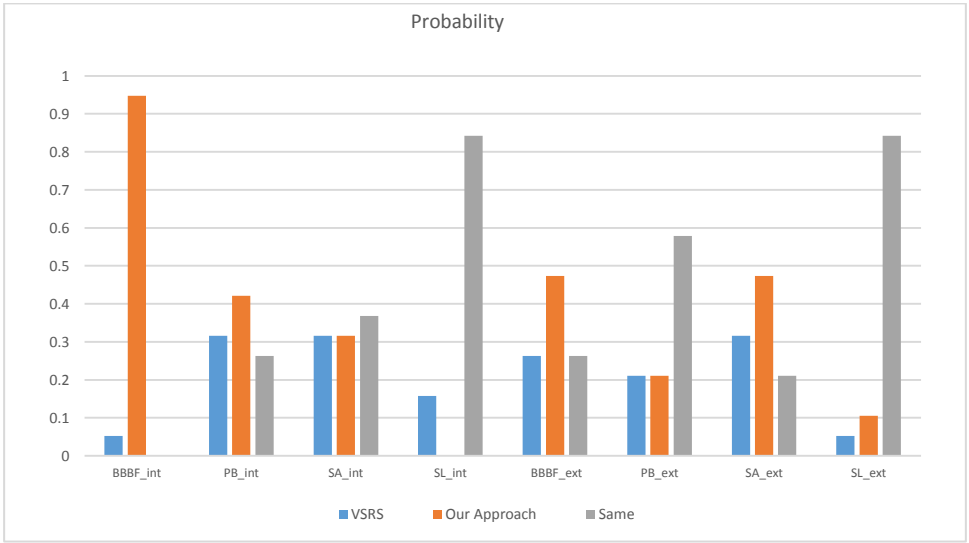


Figure 3.13. Probability for one of the method in each sequence to be chosen by the viewer.

(extrinsic and intrinsic matrices from equations 3.1 and 3.2) 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 border holes. Our warping technique provides slight improvement over VSRS.

In the case of "Soccer Line2" (SL), although it looks like the videos have a completely different quality score, the results a priori show no statistically significant preference for our method or VSRS, as the "same" option was selected in 84% of the cases (Figure 3.13, Figure 3.14). 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.

No.	Source	Seq. Name	Number of Views	Resolution (pel)	Frame rate (fps)	Length	Camera Arrangement	Views positions to be transmitted	Frame range to be transmitted
1	UHasselt	Soccer-Linear 2 (FN)	8	1392x1136	60	10 sec 600 frames	1D parallel	1-7	0-599
2	UHasselt	Soccer-Arc 1 (FN)	7	1920x1080	25	22 sec 550 frames	120 deg. Corner, arc	1-7	0-249
3	Poznan University of Technology	Poznan Blocks (FN & Multiview)	10	1920x1080	25	40 sec 1000 frames	100 deg. arc around the scene	2-8	0-249
4	Holografika	Big Buck Bunny Flowers noBlur (Multiview)	91	1920x1080	24	5 sec 121 frames	45 deg, arc	6,19,32,45, 58,71,84	0-120

Table 3.3. Summary of the sequences used in subjective test.

Figure 3.14. Probability converted to Quality Scores using Bradley-Terry model.

3.4.1 FTV Video Dataset for Subjective Evaluation

Our test included four sequences from the FTV video dataset provided in the MPEG CfE (Table 3.3). One of the sequences, "SoccerLinear2" uses linear/parallel camera arrangement, and three remaining "SoccerArc1", "Poznan Blocks", "Big Buck Bunny Flowers noBlur" use arch camera arrangements with various degree of convergence.

The materials for the subjective evaluation were prepared according to the methodology described in the CfE [1] as follows. Video clips of the rendered views were combined to create sweeps through all of the rendered and reconstructed views. The starting positions of the sweeps were selected randomly by the test chair (Table 3.4) [41].

The sweeps (Figure 3.15) were constructed at a speed of one frame per view. This has been performed for the anchor and the responses [41].

3.4.2 Display

LG 55EA9800 OLED TV in 2D mode has been used for showing the sweep videos to the viewers.

3.4.3 Viewers

Nineteen 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. In addition, to make them familiar with the test process, there was a training session using two test sequences ("Balloons" and "LoveBird1" [41]). After collecting test results, outliers were detected using circular triads method with defined threshold [40].

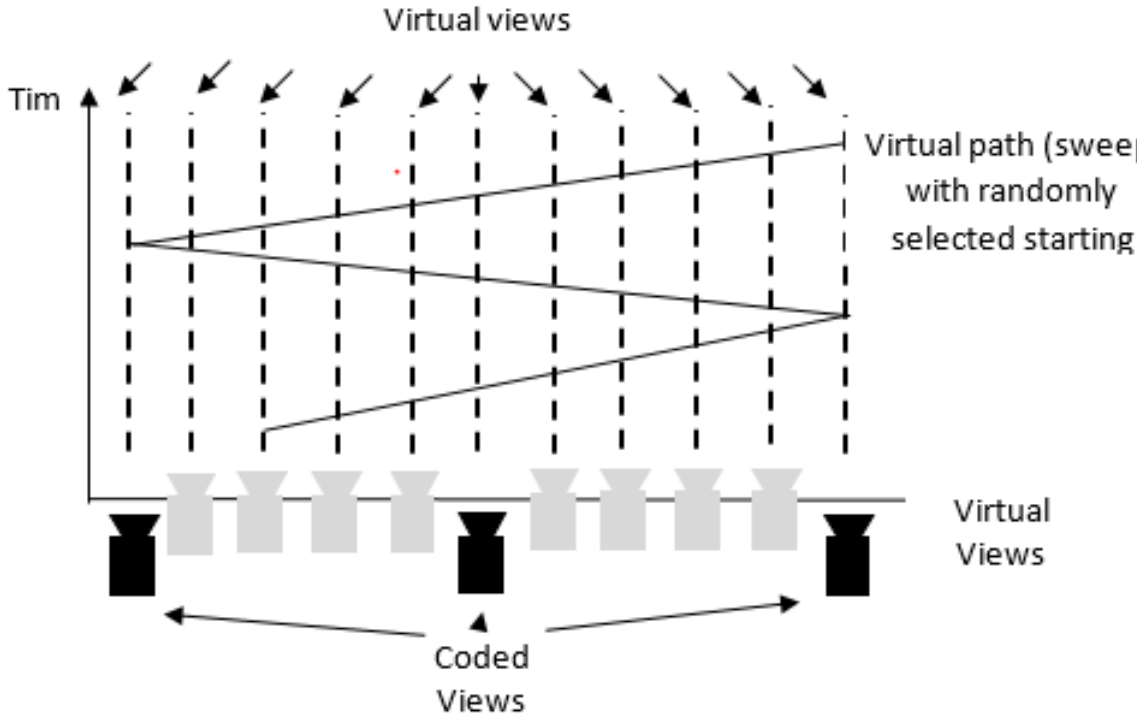


Figure 3.15. Free Navigation (FN) sweep evaluation procedure.

No.	Seq. Name	Starting position of the sweeps
1	Soccer Linear 2	60
2	Soccer Arc	110
3	Poznan Blocks	20
4	Big Buck Bunny Flowers	35

Table 3.4. Starting positions of the sweeps selected randomly by the test chair for FTV.

3.5 Subjective Tests against the Multiview Synthesis (MVS) approach and Discussions

In addition to the subjective evaluation against VSRS using sequences recommended by FTV, we also compered our results with the Multiview Synthesis (MVS) presented in [9].

Subjective quality evaluation against MVS was conducted using the same quality score metric presented in the paper. Each viewer rated the quality of synthesized views on the scale of 0 to 10, where 10 is the perfect quality – indistinguishable from the reference [9]. We chose four FTV sequences for the experiment and displayed them to 15 non-expert viewers.

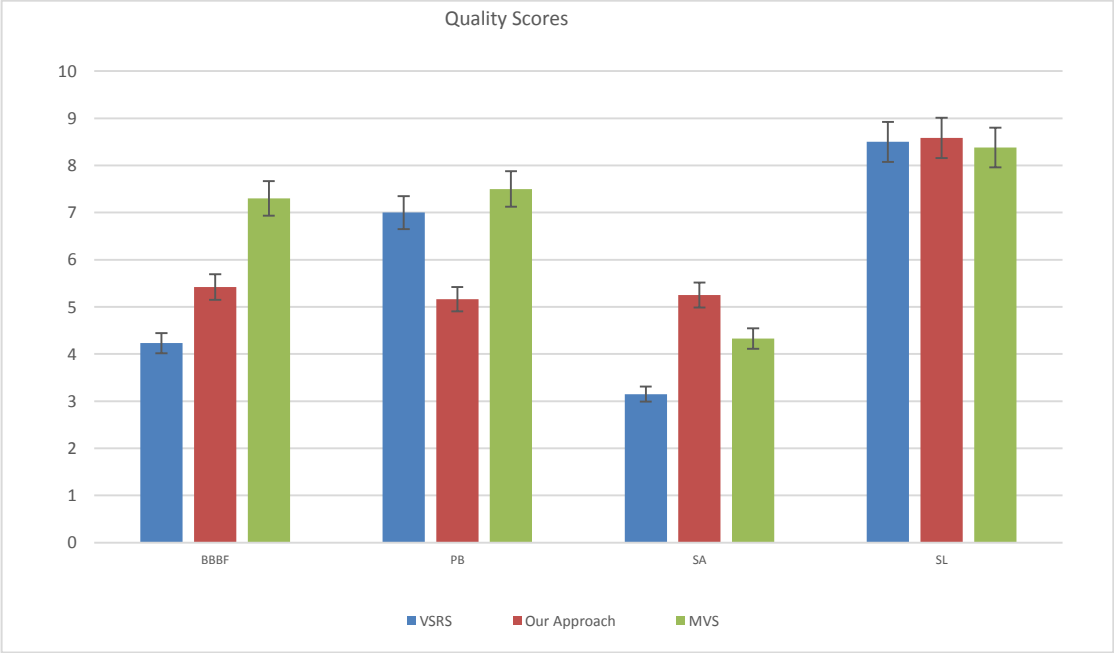


Figure 3.16. Comparison of subjective quality of VSRS, our approach, and MVS using 11 step scale.

The results of MVS [9] are informal and were conducted on expert viewers, whereas our results were conducted on non-expert level observers, which can explain the large deviation of the scores amongst our participants. The distance between the cameras is twice that recommended for the FTV tests [41] and falls under the FN category. Figure 4.1 shows the quality scores for our method, MVS and VSRS. As we can see for the SA and SL sequences, our method still outperforms both VSRS and MVS (by 11.8% on average), as the relative increase in distance between cameras is insignificant compared to the overall depth of the scene, while increasing the base line does not affect the amount of the artifacts in the same proportion as in BBBF or PB sequences. That indicates that the previously performed suggested subjective tests are still in line with these findings. The PB sequence suffers the most from the increased distance between cameras, since the overall depth of the scene is small (distance from the camera to the green curtain), creating multiple visible artifacts.

3.5.1 Conclusions

We presented a novel view-synthesizing scheme for Free Viewpoint TV applications. Objective comparisons against methods presented in [9] and [33] showed that our approach gains an average of 1.9 dB in visual quality in Y-PSNR. Subjective tests using Mean Opinion Score (MOS) have shown that our method yields significant visual improvement over VSRS for Multiview applications (over 30% in MOS). For Free Navigation applications, subjective tests showed that our method yields 4% in MOS over VSRS 11.85% against MVS.

4. Chapter 4: Color Estimation

4.1 Introduction

In this chapter, we introduce a new color estimation method specifically designed for FN view synthesis applications, addressing the challenge of brightness and color transition between consecutive virtual views. Our method uses a Gaussian Mixture Model for representing color histograms and then estimates the color of any virtual view based on its position in space relative to the real views. Subjective performance evaluations have shown that our method yields better visual quality than the existing methods.

4.2 Our Proposed Method

Our objective here is to create smooth and seamless transition of colors between FTV views. Figure 4.2 shows the view synthesis pipeline and the three components of our method: histogram approximation by the Gaussian mixture model (GMM), estimation of the virtual view's GMM

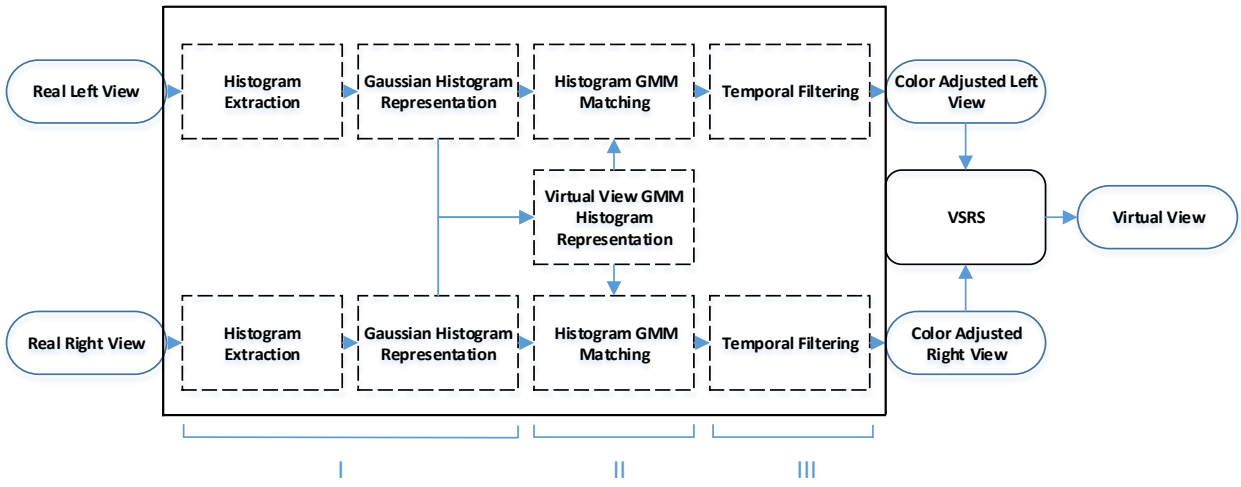


Figure 4.1. Flow chart of the entire view synthesis pipeline, including the three main components of our method: I. Histogram representation by Gaussian Mixture Model (GMM), II. Estimation of the virtual view's color histogram in GMM based on its relative position between two real views, and III. Temporal filtering.

color histogram based on its relative position between two real views, and temporal filtering that removes flickering caused by any abrupt color changes.

4.2.1 Gaussian Mixture Model Histogram Representation

First, we calculate the histogram of each color channel, which shows the pixel distribution according to their intensities in the entire frame. Although in the case of color matching between two views, it is easy to match one of the histograms to the other, in the case of estimating the histogram of the virtual view, based on naturally different color histograms of the real views and the physical location of this virtual point in space, the problem is not as trivial. For this reason, we decided to represent the histograms using the Gaussian Mixture Model (GMM) [43]. In our implementation, GMM allows us to represent the color histogram of a frame (Figure 4.3a) as a weighted sum of Gaussian functions or, in other words, by a set of scaled and shifted Gaussians (Figure 4.3b). Each Gaussian function represents a region of the frame with certain color intensities. Note that since we assume that the two real views used for virtual view synthesis share the majority of the scene's content, then those regions will be common in both histograms, meaning that the corresponding Gaussian functions in the two representations will be very similar and can be easily matched (Figure 4.4a, Figure 4.4c).

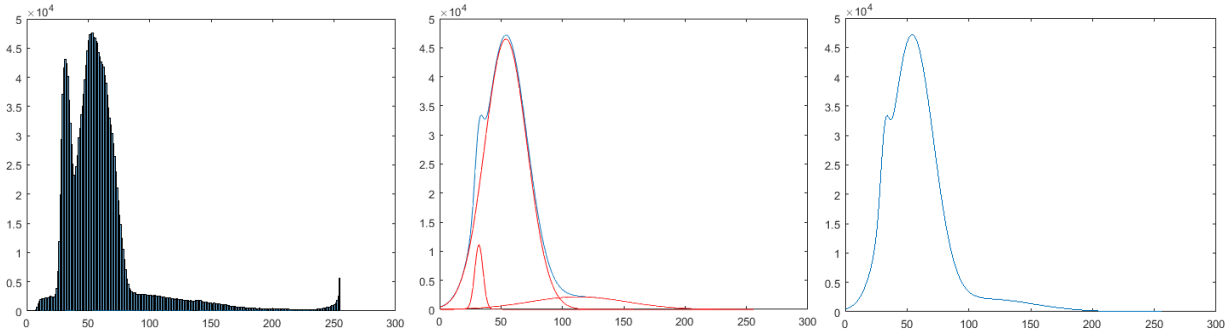
Each arbitrary histogram can be formulated as a weighted sum of k Gaussians, which requires to estimate three vectors of k parameters as follows [44]:

$$H \cong \sum_{i=1}^{k} s_i \cdot \mathbb{N}(\mu_i, \sigma_i) \qquad (4.1)$$

where H is a histogram of one of the color channels of the input frame, s_i is a scaling factor, and $\mathbb{N}(\mu_i, \sigma_i)$ is an i-th Gaussian function with mean μ_i and variance σ_i.

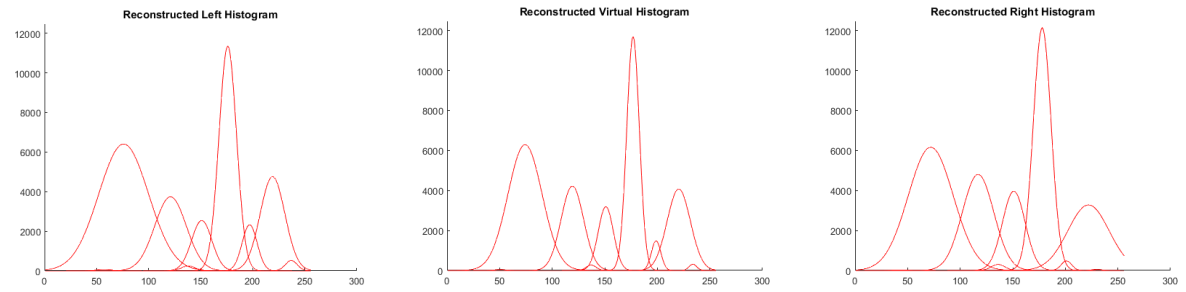
We start with extracting the histograms for each of the real views. After that, we calculate each color channel histogram GMM approximation. It is common practice when attempting to optimize the number of Gaussian functions that accurately represent a histogram to use the expectation maximization (EM) algorithm for the mean and variance calculations [45][46][47]. However, since the complexity of this algorithm is prohibitive for real-time implementations, we decided to determine the number of the Gaussian functions and their parameters by applying a technique introduced by Abdoli et al in [44]. In this approach, the mean, variance, and scaling factor are estimated by iterative greedy algorithms that after each iteration step exclude the estimated parameters from the estimation process of the next Gaussians [44]. We use least-square optimization as a cost function for determining the minimum number of Gaussians needed for a predefined error/difference between the estimated and original histograms.

During the process of GMM histogram approximation, a cost function ensures that the resulting GMM histogram representation is as close to the original histogram as possible. For that reason, our Gaussian model may yield a different number of Gaussian functions for each real view at the same instance and in time. In order to address this issue for the same time instance, we gradually



(a) (b) (c)

Figure 4.2. (a) Original histogram of a single channel, (b) Gaussian Mixture Model, (c) histogram's approximation by combining Gaussian functions.



(a) (b) (c)

Figure 4.3. Gaussian Mixture Histogram approximations of (a) Left, (b) Virtual, and (c) Right views.

adjust (decrease) the cost function of view with the smaller number of Gaussians functions until its number of Gaussian functions becomes equal to that of the other view.

4.2.2 Virtual View Histogram Creation and Real View Matching

Our next step focuses on estimating the GMM histogram of the virtual view, taking into consideration its physical location and the known GMM histograms of the neighboring real views. As previously discussed, each Gaussian function in the left real view is expected to have a matching function in the right view, due to the fact that both views cover the same scene. Based on this fact, we can estimate each Gaussian function in the virtual view histogram as a weighted sum of the two corresponding GMMs in the real views, with the weights calculated as a function of the virtual view's position in space. Although there are many ways of calculating these weights, extensive tests showed that the linear approach yields as good results as any other metric while offering the simplest implementation. Equation (4.2) shows the weight assignments based on the relative distance from a real view:

$$w_l = \frac{D_{vr}}{D_{lr}};\ w_r = \frac{D_{vl}}{D_{lr}} \tag{4.2}$$

where *wl* and *wr* are the weights for the left and right parameters, *Dvr* and *Dvl* are the distance of the virtual view from the right and left real views, and *Dlr* is the distance between the right and left real views.

Equations (4.3) to (4.5) show how the weights are used for calculating the mean, variance and scaling factor for a Gaussian function of a virtual view:

$$\mu_{virtual} = w_l \cdot \mu_l + w_r \cdot \mu_r \quad (4.3)$$

$$\sigma^2_{virtual} = w_l^2 \cdot \sigma_l^2 + w_r^2 \cdot \sigma_r^2 \quad (4.4)$$

$$s_{virtual} = w_l \cdot s_l + w_r \cdot s_r \quad (4.5)$$

where μ_l, μ_r, and $\mu_{virtual}$ are mean values of the left, right and the virtual view; σ_l, σ_r, and $\sigma_{virtual}$ are variances of the left, right, and the virtual view; $s_{virtual}$, s_l, and s_r are the scaling factors of the functions, *wl* and *wr* are the weights for the left and right parameters accordingly.

The GMM histogram representation for the left real view, the virtual view, and the right real-view are shown in Figure 4.4a, Figure 4.4b, and Figure 4.4c, respectively.

4.2.3 Temporal Filtering and Virtual View Synthesis

One challenge in this approach is that although the real and virtual views of any instance are represented by the same number of Gaussian functions that is not true for the frames of the views in time. Consequent views may end up with a different number of Gaussian functions, due to small variations in scene content, which will have an effect on our optimization process that tries to come up with the most accurate approximation of the original histogram. The result of assigning different number of Gaussians to consecutive frames in time means changes in some color intensities, which in turn may translate to flickering. To address this problem, we make sure that we have the smallest possible error when we optimize the number of Gaussian functions for the first frame of each scene

and then we force our algorithm to assign the same number of Gaussians to the rest of the frames in the scene.

Another cause for flickering may arise from the fact that the shape of the virtual Gaussian functions may change in time, as it is directly related to the content changes in the real views. Any drastic change in shape will translate to changes in color intensities, which will cause flickering. To address this challenge, we introduced a cumulative moving average temporal filter that is applied only on the Gaussian parameters of the mean, the variance and scaling. The general form of this equally weighted cumulative moving average filter (*CMA*) is as follows:

$$CMA_n = \frac{x_1 + \cdots + x_n}{n} \qquad (4.6)$$

where *n* is the current number of inputs and *x1 … xn* is the sequence of values of a given parameter (i.e., μ, σ, and *s*).

The advantage of this filter is that there is no need to store any information other than the previous CMA value of each parameter for the duration of the scene.

As a last step, we used the color-adjusted frames for the real views to synthesize the virtual view according to the FTV test recommendations [41]. We used the View Synthesis Reference Software (VSRS) for the view synthesis step, as it is the preferred choice of industry and the reference software for MPEG [8].

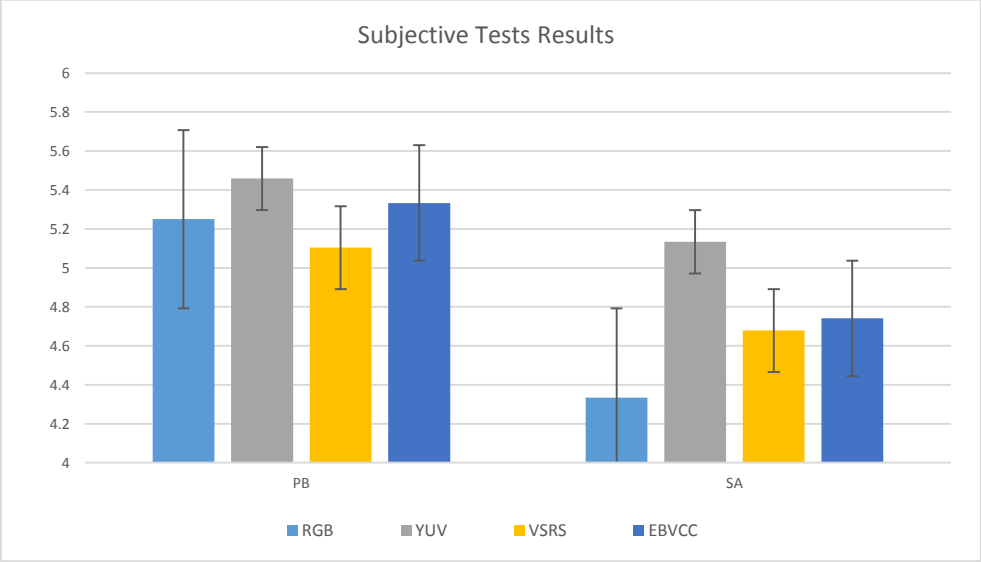


Figure 4.4. Subjective results (score 1 to 10 – only 4 - 6 shown for clarity) for our method, VSRS, and EBVCC.

4.3 Test Results and Discussions

We subjectively evaluated our method by comparing it with a no color correction VSRS and a representative color matching method (EBVCC presented in [23]). To this end, we used two FTV sequences namely the Poznan Blocks (PB) and Soccer Arch (SA) recommended by MPEG. We followed the ITU-R BT.500 recommendation for subjective assessments [48]. Twenty non-expert subjects were shown the sequences on an LG 55EA9800 OLED TV in 2D mode and were asked to rate the sequences on the scale from 1 to 10, with 1 being poor color matching and 10 meaning that no color-related artifacts were visible in the synthesized view. We also decided to apply our method to two different color spaces, RGB and the YCbCr to determine which will offer more accurate color representation. Four outliers were found and removed from our tests. As we can see in Figure 4.4, viewers clearly believed that our method provides better color accuracy than EBVCC (5.31% on average for FN sequences) and the no-color corrected VSRS methods (8.34% on average for FN sequences) when the YCbCr color space is used.

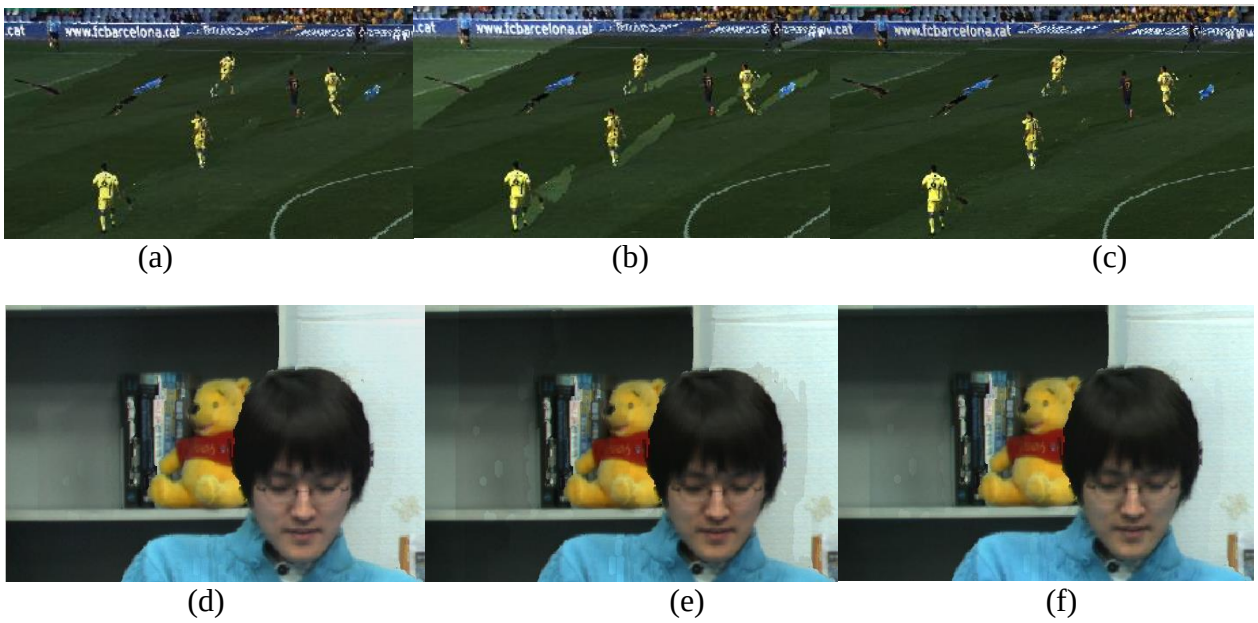


Figure 4.5. (a), (b) and (c): Frames from Soccer Arch sequence generated with our method, VSRS and EBVCC; (d), (e) and (f): Frames from Poznan Blocks generated with our method, VSRS, and EBVCC.

Figure 4.5 shows the one frame from the two sequences tested, generated by the original VSRS (Figs. 4.5b and 25e), EBVCC (Figs. 4.5c and 4.5f), and our method (Figs. 4.5a and 4.5d). We observe that our method manages to remove the shadow artifacts and the background artifacts on the left part of the frame and the ones on the right of the head, which apparently are caused by wrong color estimation in the virtual view.

4.3.1 Conclusions

We presented a color estimation method for Free Navigation applications. Subjective tests against EBVCC and VSRS showed that our method reduces color related artifacts and helps to create more immersive experience for the Free Navigation sequences by smoothing the viewpoint transition. We gain 5.3% in MOS from our subjective tests against EBVCC and 8.3% against VSRS rendered videos.

5. Chapter 5: Conclusions and Future Work

5.1 Conclusions

In this thesis, we addressed the challenges of view synthesis for Multiview and Free View navigation applications. Our first contribution involved the introduction of a novel approach for synthesizing virtual views for Multiview applications, where the real views are relatively close to each other. Our method uses background-to-foreground warping and background separation for accurately filling large occluded regions, while improving traditional inter pixel interpolation to efficiently generate the rest of missing texture in the virtual view.

Our second contribution is a new color estimation method for synthesizing views in a Free View environment, addressing the challenge of color transition between consecutive views for these applications. Our method uses a Gaussian Mixture Model for approximating color histograms and estimates the color of any virtual view based on its position in a space relative to the real views. A flickering reduction method designed for this scheme was also developed.

Subjective and objective tests have shown that our methods outperformed the existing ones, significantly improving the visual quality of the synthesized content.

5.2 Future Work

FN applications' distant content capturing setup requires new view synthesis approaches to address Geometrical Distortions (GD) that appear due to the different convergence angles, camera distance from the objects and flat background where the objects might appear. The reason for this is that GDs may not offer significant depth value representation, thus not enabling the existing algorithms to adjust their appearance at the view synthesis stage of the model. Proper detection of such features is needed as a first step, followed by transformation to the virtual view plane. A possible solution is to match these distortions in the neighbouring real images and create a geometrical

transformation matrix that will help us reconstruct the features for the virtual view, as close, to what it should be in reality, as possible.

Multiple light sources' and camera locations may not only cause the foreground objects shift, but also their shading to be different. Shade distortions create artifacts that are very pronounced to human visual system, as they are located in proximity to the foreground objects that we pay attention to. The problem seems to be similar to the previously stated GD, but it is also affected by the light sources present in the scene. Multiple light sources will create different shades of varying intensity, thus complicating the model.

A Free Navigation camera setup has a very wide baseline compared to Multiview applications. Cameras can be located up to ten meters away from each other and are pointing to the scene from different directions, thus affecting the amount of light that each camera will receive. The bright light sources can be pointed directly to the camera plain in some of the views, affecting the overall brightness of the captured frame. The amount of the light that hits a camera also affects the color perception. That means that the images captured from neighbouring cameras will have different colors for the same object. There are a number developments that can be used in order to define what is the "real" color of the current frame should be that can help to match colors for two images (for example, work presented in [49] and [22]). While synthesizing one of the transitional views (between neighbouring cameras), we have to take into account that the color of each object in the scene, for specific virtual view, should be the same from both cameras, for future occlusions filling from the second closest camera to the virtual camera position. In other words, the luminance and color, which have been retrieved from the original cameras, should be adjusted to match virtual view position. Appropriate color and brightness representation should be applied to the images, so it will be possible to generate perceptually acceptable transitional images. The virtual view color

and brightness are a function of the position of the virtual camera in space and the light sources in the scene, and even though the linear approximation can be applied, functions that are more appropriate should be investigated.

One suggestion is to explore the idea of histogram approximation using the Gaussian mixture model and wavelets to match two histograms. Initial results of preliminary research showed significant brightness and color matching artifact reduction in the synthesized views.

It is worth mentioning that the sweep through the synthesized views should not affect the quality of experience for the viewer, meaning that there should be no big “steps” in color and brightness differences between consecutive views in the resulted sweep video.

BIBLIOGRAPHY

[1] "Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation", ISO/IEC JTC1/SC29/WG11 MPEG2015/N15348, Poland, Warsaw, June 2015.

[2] H.G. Hoffman, et al. "Feasibility of articulated arm mounted Oculus Rift Virtual Reality goggles for adjunctive pain control during occupational therapy in pediatric burn patients," Cyberpsychology, Behavior, and Social Networking, Vol. 17.6, pp. 397-401, 2015.

[3] I. Ganelin, P. Nasiopoulos, M. T. Pourazad, "A View Synthesis Approach for Free-navigation TV Applications," The Eighth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, Computation Tools, 2017.

[4] JH. Cho, W. Song, H. Choi, T. Kim, "Hole Filling Method for Depth Image Based Rendering Based on Boundary Decision," IEEE Signal Processing Letters, Vol 25.3, pp. 329-333, 2017.

[5] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, "Emerging technologies for 3D video: creation, coding, transmission and rendering," John Wiley & Sons, 2014.

[6] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," Signal Processing Image Communication, Vol 23.2, pp. 217-234, 2007.

[7] Zhu, Ce, and Shuai Li "Depth image based view synthesis: New insights and perspectives on hole generation and filling," IEEE Transactions on Broadcasting, Vol 63.1, pp. 82-93, 2016.

[8] J. Stankowski, L. Kowalski, J. Samelak, M. Domański, T. Grajek, K. Wegner, "3D-HEVC

extension for circular camera arrangements," 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), IEEE, pp. 1-4, 2015.

[9] A. Dziembowsk, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domańsk, "Multiview synthesis—Improved view synthesis for virtual navigation," Picture Coding Symposium (PCS), IEEE, pp. 1-5, 2016.

[10] A.I. Purica, E.G. Mora, B. Pesquet-Popescu, M. Cagnazzo, and B. Ionescu. "Improved view synthesis by motion warping and temporal hole filling," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1191-1195, 2015.

[11] M. Salmistraro, LL. Raket, C. Brites, J. Ascenso, S. Forchhammer, "Joint disparity and motion estimation using optical flow for multiview Distributed Video Coding," 2014 Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), pp. 286-290, 2015.

[12] A. Purica, M. Cagnazzo, B. Pesquet-Popescu, F. Dufaux, B.Ionesc, "View synthesis based on temporal prediction via warped motion vector fields," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1150-1154, 2016.

[13] G. Luo, Y. Zhu, Z. Li, L. Zhang, "A Hole Filling Approach Based on Background Reconstruction for View Synthesis in 3D Video," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1781-1789, 2016.

[14] HR. Kaviani, H. Rezaee, S. Shirani, "An Adaptive Patch-based Reconstruction Scheme for View Synthesis by Disparity Estimation Using Optical Flow," IEEE Transactions on Circuits and Systems for Video Technology, pp. 1, 2017.

[15] NK. Kalantari, E. Shechtman, S. Darabi, DB. Goldman, P. Sen, "Improving patch-based

synthesis by learning patch masks," 2014 IEEE International Conference on Computational Photography (ICCP), pp. 1-8, 2015.

[16] A. Fitzgibbon, Y. Wexler, and A. Zisserman, "Image-based rendering using image-based priors," International Journal of Computer Vision, Vol. 64.2, pp. 141-151, 2005.

[17] S. Zinger, D. Luat, and P. H. N. de With, "Free-viewpoint depth image based rendering," Journal of visual communication and image representation, Vol. 23.5, pp. 533-541, 2010.

[18] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, M. Gross, "Nonlinear disparity mapping for stereoscopic 3D," ACM Transactions on Graphics (TOG), Vol. 29.4, pp. 75, 2010.

[19] G. Lafruit, M. Domański, K. Wegner, T. Grajek, T. Senoh, J. Jung, PT. Kovács, P. Goorts, L. Jorissen, A. Munteanu, B. Ceulemans, "New visual coding exploration in MPEG: Super-Multiview and Free Navigation in Free viewpoint TV," Electron. Imaging, Vol. 2016.5, pp. 1-9, 2016.

[20] S-A. Fezza, M-C. Larabi, "Color calibration of multi-view video plus depth for advanced 3D video," Signal, Image and Video Processing, Vol. 9.3. pp. 177-191, 2015.

[21] E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, “Color transfer between images,” IEEE Computer graphics and applications, pp. 34–41, September 2003.

[22] C. Doutre, P. Nasiopoulos, "Color correction preprocessing for multiview video coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19.9 (2009), pp. 1400-1406, 2009.

[23] C-H. Yao, C-Y Chang, S-I Chien, "Example-based video color transfer," Multimedia and Expo (ICME), IEEE International Conference on., pp. 1-6, 2016.

[24] B. Ceulemans, et al. "Globally optimized multiview video color correction using dense spatio-temporal matching," 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), pp. 1-4, 2015.

[25] Shin, Dong-Won, Y-S Ho, “Color correction using 3D Multiview geometry,” Color Imaging XX: Displaying, Processing, Hardcopy, and Applications, International Society for Optics and Photonics, Vol. 9395, pp. 93950O, 2015.

[26] S. Goedegebure, S. Goedegebure, A. Goralczyk, E. Valenza, N. Vegdahl, W. Reynish, BV. Lommel, C. Barton, J. Morgenstern, T. Roosendaal, "Big Buck Bunny," 2008.

[27] M. Domanski, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner,“Poznan Blocks - a multiview video test sequence and camera parameters for Free Viewpoint Television,” MPEG2014, M32243, 2015.

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

[29] C. Lipski, F. Klose, and M. Magnor, "Correspondence and depth-image based rendering a hybrid approach for free-viewpoint video," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 25.6, pp. 942-951, 2015.

[30] S. Huq, A. Koschan, and M. Abidi, "Occlusion filling in stereo: Theory and experiments," Computer Vision and Image Understanding, Vol. 117.6, pp. 688-704, 2014.

[31] MS. Farid, M. Shahid, M. Lucenteforte, and M. Grangetto, "Edge enhancement of depth based rendered images," 2014 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 5452-5456, 2015.

[32] K. Mueller, A. Smolic, K. Dix, P. Merkle, P. Kauff, T. Wiegand, "View synthesis for

advanced 3D video systems," EURASIP Journal on image and video processing, Vol 2008.1, pp. 1-11, 2009.

[33] I. Daribo, W. Milded, and B. Pesquet-Popescu, “Joint Depth-Motion Dense Estimation for Multiview Video Coding,” Journal of Visual Communication and Image Representation, Vol. 21, pp. 487–497, 2010.

[34] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded-up robust features (SURF)," Computer vision and image understanding, Vol. 110.3, pp. 346-359, 2008.

[35] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, MA. Magnor, "Markerless motion capture using multiple color-depth sensors," VMV, pp. 317-324, 2013.

[36] A. Wafa, “Automatic real-time 2D-to-3D video conversion,” Doctoral dissertation, University of British Columbia, 2016.

[37] O. Wang, M. Lang, N. Stefanoski, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, M. Gross, "Image Domain Warping for Stereoscopic 3D Applications," Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering, pp. 207-230, 2014.

[38] R.O. Duda, and E. H. Peter, "Pattern recognition and scene analysis," 1974.

[39] M. Le Dinh, et al. "Improving 3D-TV view synthesis using motion compensated temporal interpolation," International Conference on Advanced Technologies for Communications (ATC), IEEE, pp. 312-317, 2016.

[40] 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, pp. 1281-1284, 2011.

[39] M. Le Dinh, LV. Tung, XH. Van, DD. Trieu, TP. Thanh, H. Le Thanh, "Improving 3D-TV view synthesis using motion compensated temporal interpolation," 2016 International Conference on Advanced Technologies for Communications (ATC), pp. 312-317, 2016.

[41] 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, W16318, June 2016.

[42] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," Biometrika, Vol. 39, pp. 324-345, 1953.

[43] D. Reynolds, "Gaussian mixture models," Encyclopedia of biometrics, pp. 827-832, 2015.

[44] M. Abdoli, et al. "Gaussian mixture model-based contrast enhancement," IET image processing 9.7, pp. 569-577, 2015.

[45] Y.R. Lai, K.L. Chung, G.Y. Lin, C.H Chen, "Gaussian mixture modeling of histograms for contrast enhancement," Expert systems with applications, Vol. 39, (8), pp. 6720–6728, 2013.

[46] T. Celik, T. Tjahjadi "Automatic image equalization and contrast enhancement using Gaussian mixture modeling," IEEE Transactions on Image Processing, Vol. 23.1, pp. 145–156, 2013.

[47] A. Kapoor, et al., "MPI implementation of Expectation Maximization algorithm for Gaussian mixture models," Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI, Vol. 2, Springer International Publishing, pp. 517-523, 2015.

[48] Recommendation ITU-R BT.500-13, "Methodology for the subjective assessment of the

quality of television pictures," 2013.

[49] A. Shamir, O. Sorkine, "Visual media retargeting," In SIGGRAPH ASIA Courses, pp. 11, 2009.