QUALITY ANALYSIS OF VIRTUAL VIEWS ON STEREOSCOPIC VIDEO CONTENT

Mattias Eisenbarth[a], *Florian Seitner*[b], *Margrit Gelautz*[a]

[a]Institute for Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstr. 9-11/188/2, A-1040 Vienna, Austria;
[b]emotion3D GmbH, Phorusgasse 8/6, A-1040 Vienna, Austria

ABSTRACT

Depth Image-Based Rendering (DIBR) allows the creation of virtual camera viewpoints from a 2D image and its corresponding disparity map. This enables a variety of new applications in 3D film post-production where disparities can be computed in an automatic way from the stereoscopic content. Examples include scene depth correction, content remastering and multi-view generation for auto-stereoscopic displays. In this paper, a comparison of state-of-the-art DIBR techniques in the context of 3D video adaptation is presented. We first provide an evaluation method that enables subjective comparison of the visual quality of DIBR-generated 3D film sequences. Based on this, we then evaluate the impact of image artifacts on the visual comfort and the depth impression for four different DIBR approaches.

Index Terms— DIBR, evaluation, warping, in-filling

1. INTRODUCTION

The currently available stereoscopic and auto-stereoscopic displays allow the users to watch a scene in 3D and from one or multiple viewpoints. *Free Viewpoint Video* (FVV) enables displaying different views of the same film scene and allows the users to receive different stereoscopic images depending on their viewing position or the chosen viewpoint [6]. However, as the transmission of 3D movies for multiple viewing points requires a high amount of data and results in a high data redundancy during transmission, generation and interpolation techniques which can create new camera viewpoints (novel views) at the receiver side have evolved.

Starting from a pair of stereo images, a depth map that describes the depth of each pixel is extracted [9]. The video is transferred to the home user in a Multi-view Video plus Depth (MVD) format where only one or just a few views and the corresponding depth maps are broadcast [3]. The depth map and the original images are then used to generate an arbitrary number of novel views by means of Depth Image-Based Rendering (DIBR) [1, 2, 4, 5]. DIBR techniques exploit the characteristic of stereoscopic displays, which use the horizontal parallax of pixels to generate a 3D effect. A 2D image can be mapped to a new viewpoint position using the depth information (z-coordinates) of its visual content. A description of DIBR is provided in Section 2.

Despite the fact that DIBR can reduce the amount of the transferred video data significantly, various challenges have to be addressed to achieve visually satisfying results. First, an accurate warping of each pixel according to its depth and to the correct position in the Novel View (NV) image has to be done. Warping of fine image structures as well as of object border regions, where pixels are typically a mixture of foreground and background colors, must be addressed adequately. Second, the information gaps in the NV that result from this warping process must be filled in. For example, for regions that have not been visible in the original view (e.g. occluded by a foreground object) no image information is available in the NV and *in-painting* techniques for filling these regions are required.

This work evaluates the visual quality of state-of-the-art DIBR methods using subjective quality comparison and is structured in the following way: Section 2 provides an overview of DIBR and the methods considered in this study. Section 3 describes the test setup and Section 4 and 5 the results of our study and conclusions, respectively.

2. DEPTH IMAGE-BASED RENDERING METHODS

Typically, a DIBR method can be divided into two main steps: (1) *warping* and (2) *in-painting*. Both steps are introduced in this section.

2.1. Image warping

A NV is derived from an original image by shifting each pixel horizontally by a disparity value that represents the pixel's depth. For an original pixel at position p_0, its position p_s in the new view is computed by

$$p_s = p_0 + s\left(d_{p_0} - \delta\right) \tag{2.1}$$

with d_{p_0} representing the corresponding disparity value and s being a scaling factor that defines the distance between two virtual camera positions. A convergence parameter δ defines the disparity displacement [4]:

1. $d > \delta$: pixel is **behind** the screen plane,
2. $d = \delta$: pixel is **on** the screen plane,
3. $d < \delta$: pixel is **in front of** the screen plane.

The color value at pixel position p_s is set to the corresponding color value of the original pixel position p_0.

During the warping process occlusions can occur if two or more pixels of the original view are warped to the same position in the novel view. Since foreground (FG) objects occlude background (BG) objects, pixels with low disparity values (=FG) replace pixels with high disparity values (=BG), which may have been warped earlier to the same position.

This simple concept works well for most of the image parts. However, object boundaries lead to problems as disparity values around border regions are often a mixture of depth values of the neighboring foreground and background objects. Disparity morphing then leads to imprecise results and image artifacts. Layered-based novel view methods try to solve those problems by constructing reliability layers previous to the image warping step [2, 3].

2.2. Image in-painting

Disocclusions occur if parts of the scene become visible, which have been hidden by a foreground object in all of the original views. This results in 'holes' where no pixel information is available in the new virtual views. For filling these holes, various *in-painting* methods exist that range from simple pixel filling to complex structure/texture based methods [2, 3, 5, 7]. However, in this user study, we focus on fast algorithms that can address the real-time requirements of interactive FVV. These algorithms will be described more in detail in the following:

(A) Horizontal background extrapolation [7]: The hole is filled by horizontally copying the color of the one border pixel (in the same scan line) that lies in the background. It is assumed that the disocclusion reveals the nearest background object even if this object is located in the front area of the scene. Every pixel inside the hole H is filled by the horizontal background extrapolation according to:

$$I[n] = \begin{cases} I[m_l] & \text{if } d[m_l] > d[m_r] \\ I[m_r] & \text{if } d[m_l] \le d[m_r] \end{cases} \quad \forall n \in H \tag{2.2}$$

where m_l and m_r are the coordinates of the first pixels of the border of hole H in left and right direction in the scan line of pixel n, which has to be filled, and d denotes the disparity values of those pixels.

(B) Horizontal copy background [5]: As opposed to (A), the hole is now filled by completely copying the horizontally neighboring background into the hole, but not only using the color of the first border pixel:

$$I[(x,y)] = \begin{cases} I[(x-s,y)] & \text{if } d[m_l] > d[m_r] \\ I[(x+s,y)] & \text{if } d[m_l] \le d[m_r] \end{cases} \quad \forall (x,y) \in H \tag{2.3}$$

The parameter s represents the size of the hole (number of pixels) in horizontal direction [5].

(C) Mean background extrapolation [10]

Mean background extrapolation is applied within two steps. First, holes have to be detected and marked. A weighting mask W is applied on the hole. Every neighboring pixel, which borders the hole horizontally, vertically or diagonally, receives a weight as well. Pixels inside the hole are weighted 0.0. Background pixels are weighted 1.0 and foreground pixels receive a user defined initial value. If foreground pixels' color values are not to be included in the in-filling process, they have to be weighted 0.0 as well.

In a second step – after the weighting mask W is generated – the simultaneous in-painting of the hole and updating of the weights in W is applied. In our evaluation in-filling is done in top-down scan line order. The horizontal filling direction is always given by the video content: from the background to the foreground side of the hole.

For every pixel in the hole, the set N of its neighboring pixels is discovered, which includes all 2D-coordinate-tuples (i, j) of the virtual image I, which are direct neighbors (horizontal, vertical, diagonal) of the currently processed pixel. Missing color values are then processed by means of Equation 2.4.

$$I[(x,y)] = \frac{\sum_{(i,j)\in N} I[(i,j)] w_{i,j}}{\sum_{(i,j)\in N} w_{i,j}} \quad \forall (x,y) \in H,\ w_{i,j} \in W \tag{2.4}$$

H is the set of all of the pixels inside the hole. Together with the color value, the weight $w_{x,y}$ of the currently processed pixel in weighting mask W is updated using Equation 2.5.

$$w_{x,y} = \frac{\sum_{(i,j)\in N} w_{i,j}}{8}. \tag{2.5}$$

The denominator in Equation 2.5 results from the number of 8 direct neighboring pixels for every pixel in the hole.

3. TEST SETUP

The following subsections describe the test setup of our evaluation method step by step.

3.1. Measuring display boundaries

Prior to the generation of virtual camera positions, we have analyzed the depth boundaries of the displaying monitor. This enables us to generate novel 3D views within the physical limitations of the display and to assure that no virtual view exceeds the *depth budget* and hence the visual capabilities of the display.

The depth budget is typically defined by the minimal and maximal disparity D_{min} and D_{max} in pixels that can be displayed on the display, respectively. This defines the largest pop-out and pop-in-effect the display is capable of.

For measuring D_{min} and D_{max}, a synthetic test image and a corresponding disparity map are generated (Figure 1). This image contains simple elements such as rectangles. We generate two views by moving these elements horizontally by known disparity values. By displaying the 3D test image on the screen, we can determine which elements appear blurred and hence exceed the visual range of the display. The simple shapes and colors in the synthetic image support a clear decision for the 'blurredness' of each element.

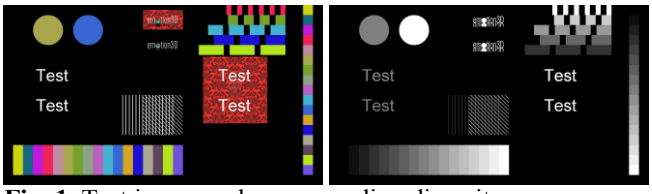

Fig. 1. Test image and corresponding disparity map

The display measurement itself is carried out in the following way. First, the **pop-out-effect** is tested by only shifting the test image's elements by negative disparity values. Beginning with a low depth, the views are displayed on a stereoscopic display. The depth is then increased step by step. Test objects must appear sharp and three-dimensional and have to be easily focusable all the time. If disparities get too big, the brain cannot fuse the stereo images any longer and therefore cannot generate a 3D-impression. The maximum foreground disparity of a display equals the maximum disparity of the last test image that was perceived three-dimensional **and** easily focusable. Second, the maximal **pop-in-effect** (positive disparities) is measured in the same way. At the end, when the largest and smallest possible disparity values have been identified, they can be tested together. Suitable novel view algorithms are able to position virtual cameras and move 3D scenes in z-direction to generate those 3D spaces that completely exploit the limits of the depth budget. The identified disparity values were tested on a single test scene.

3.2. Test runs

Prior to the evaluation, a pre study on still images was carried out to discover in-painting methods generating improper novel views. The mean background extrapolation algorithm led to comparatively bad results so that it was excluded from the main study [10].

Based on the test setup described above, a subjective quality evaluation on video sequences was carried out. Warping methods were used to produce external camera views. This means that only one of the two original stereo images and the corresponding disparity map were used to generate the novel views. Holes in the warped images were closed using a selection of in-painting methods.

For comparison, two videos – each produced with a different method – were displayed side-by-side on a stereoscopic screen. We used the stereoscopic Acer GD245HQ 24" TFT monitor for our temporal evaluation, since tests on auto-stereoscopic monitors showed that the results were influenced by the viewing position. We decided to use video content instead of still images for our evaluation as flickering artifacts only occur in temporal footage. On the other hand, very small visual artifacts, which can be perceived in spatial content, may not be noticeable in video clips.

When evaluating the algorithms, the users had to compare four different test scenes, each individually processed with the four chosen algorithms (Section 3.3.), and rank them with regard to their visual quality. The users always received two clips of the same test scene and had to decide whether the quality was equal or one of the two methods generated better results. The comparisons were made with algorithms I-II, II-III, III-IV, and IV-I (see 3.3). To avoid biasing effects, the position (left or right) of the applied algorithms on the display as well as the order of the test scenes were chosen randomly.

3.3. Evaluated algorithms

We combined and evaluated four combinations of warping and in-painting algorithms:

Algorithm I is a combination of a slightly modified version of the layered approach for warping (no fusion due to generation of only external views) and the horizontal background extrapolation for in-painting.

Algorithm II combines disparity morphing with the horizontal copy background in-painting. Copy background was added a threshold value in order not to copy foreground object's pixel information into the gap [10].

Algorithm III is a combination of disparity morphing and the horizontal background extrapolation. In-painting was done with a hierarchical approach. The original image's geometrical resolution was reduced half in size for two times. In-filling was then applied to the image with the lowest resolution. Then the image was upscaled two times again and afterwards fused with the primarily warped image. Image in-painting is therefore a combination of the original background extrapolation and the upscaling process.

Algorithm IV simply combines disparity morphing with the horizontal background extrapolation.

3.4. Test method

Based on the idea of Rajae-Joordens and Engel [8], the results of the study were analyzed using the Thurstone-Model. Based on this model, statistical evaluations with high statistical power can be achieved already for small sample sizes. Our study was carried out with 14 users (5 female, 9 male) aged between 22 and 61 years (mean age: 33.71 years, median age: 28.5 years).

First, a preference matrix, which shows the fraction of users that prefer one algorithm (1) over another (-1), was computed (Table 1). Second, z-values were derived by means of the Thurstone-Model (Equation 4.1) [8].

$$\Phi^{-1}(p) = a \times A + b \times B + c \times C \qquad (4.1)$$

In Equation 4.1, a, b and c denote the z-values, A, B and C are the columns of the regression matrix (A for algorithm I and so on) and Φ^{-1} is the inverse of the cumulated normal distribution. The z-value d for algorithm D (IV) is set to zero. Significance tests for the resulting z-vales are then applied according to Rajae-Joordens and Engel [8].

4. EVALUATION RESULTS

Table 1 shows the resulting preference matrix of our user study as a basis for the evaluation by means of the Thurstone-Model. The way to calculate CFractions (Table 1) is explained in [8].

Table 1. Preference (regression) matrix

Users	Algorithm				Fraction	CFraction
	I	II	III	IV		
14	1	-1	0	0	0,0000	0,0357
14	1	0	-1	0	0,7857	0,7857
14	1	0	0	-1	0,2143	0,2143
14	0	1	-1	0	1,0000	0,9643
14	0	1	0	-1	0,9286	0,9286
14	0	0	1	-1	0,0000	0,0357

Analysis of the Thurstone-Model resulted in model coefficients (estimated z-values) $a^{est} = -0.727$, $b^{est} = 1.1427$, $c^{est} = -1.4763$, and $d = 0$. The algorithm with the biggest z-value is rated as the best of the compared algorithms to produce novel views. The outcome of our study after performing significance tests is:

1. Algorithm II is significantly better than all the other methods and therefore the most preferred technique.

2. Algorithm IV is significantly better than algorithm III. Using the horizontal background extrapolation for the in-painting process leads to better results if the hierarchical approach is **not** applied.

3. There is no significant difference between algorithms I-IV and algorithms I-III. The advantages of the layered approach (algorithm I) therefore do not justify the longer computation time and can be substituted with simple disparity morphing when producing external camera views.

5. CONCLUSION

Our study introduces a method for comparing DIBR algorithms on temporal videos. A synthetic test image has been used to first measure the boundaries of our 3D test display and then provide insights into the warping and in-filling process of DIBR methods. We found that using a stereoscopic display to compare 3D video clips leads to more reliable results than comparisons on auto-stereoscopic monitors, where results are more influenced by the viewing position. The side-by-side displaying of test sequences allowed the users to concentrate on the same artifacts, while perceiving the same content simultaneously. The results show that a combination of simple disparity morphing [4] and the horizontal copy background in-filling algorithm [5] achieve the highest subjective user satisfaction amongst the evaluated algorithms. In future work, we want to develop methods for quality assessment of (i) stereoscopic and auto-stereoscopic 3D displays and (ii) DIBR-generated 3D content.

6. REFERENCES

[1] Y. Morvan, D. Farin, and P. H.N. de Wirth, "Design Considerations for View Interpolation in a 3D Video Coding Framework", 27th Symposium on Information Theory in the Benelux, Noordwijk, The Netherlands, 2006.

[2] A. Smolic, K. Müller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate View Interpolation based on Multiview Video plus Depth for Advanced 3D Video Systems", ICIP 2008, San Diego, USA, 2008.

[3] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View Synthesis for Advanced 3D Video Systems", EURASIP Journal on Image and Video Processing, pp. 1-11, 2008.

[4] D. Alessandrini, R. Balter, and S. Pateux, "Efficient and Automatic Stereoscopic Videos to N Views Conversion for Autostereoscopic Displays", SPIE Proc. on Stereoscopic Displays and Applications XX, San Jose, USA, 2009.

[5] J. Overes, "Occlusion Filling in Depth-Image-Based Rendering", Master thesis, Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Man-Machine Interaction Group, Delft, The Netherlands, 2009.

[6] A. Smolic, "An Overview of 3D Video and Free Viewpoint Video", Proc. of CAIP 2009, Münster, Germany, 2009.

[7] C. Vázquez, W.J. Tam, and F. Speranza, "Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering", SPIE Proc. on Three-Dimensional TV, Video, and Display V, Boston, 2006.

[8] R. Rajae-Joordens and J. Engel, "Paired Comparisons in Visual Perception Studies using Small Sample Sizes", Displays, Volume 26, Issue 1, pp. 1-7, 2005.

[9] M. Bleyer and M. Gelautz, "Temporally Consistent Disparity Maps from Uncalibrated Stereo Videos", Proc. of ISPA 2009, pp. 383-387, Salzburg, Austria, 2009.

[10] M. Eisenbarth, "Evaluierung von Algorithmen zur Novel-View-Generierung aus Stereobildern", Master thesis, Vienna University of Technology, Faculty of Informatics, Interactive Media Systems Group, Vienna, Austria, 2011.