AUTOMATIC DETECTION OF DEPTH JUMP CUTS AND BENT WINDOW EFFECTS IN STEREOSCOPIC VIDEOS

Sotirios Delis *Nikos Nikolaidis* *Ioannis Pitas*

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, GREECE
{sdelis, nikolaid, pitas}@aiia.csd.auth.gr

ABSTRACT

3DTV and 3D cinema witness a significant increase in their popularity nowadays. New movie titles are released in 3D and there are more than 35 TV channels in various countries that broadcast in 3D worldwide. It is well known today and becomes more obvious, as the 3D video content availability increases, that stereoscopy is associated with certain 3D video quality issues that may affect in a negative way the 3D viewing experience. In this paper, we propose two novel algorithms that exploit available disparity information to detect two disturbing stereoscopic issues, namely depth jump cuts and bent window effects. Representative examples are provided to assess the algorithms performance. The proposed algorithms can be helpful in the post-production stage, where, in most cases, the detected issues can be fixed, and also in assessing the overall quality of stereoscopic video content.

***Index Terms*—** 3D quality, stereo video, disparity

1. INTRODUCTION

A breakthrough in 3D cinema took place in 2009, when 24 movies premiered in cinemas in 3D. This trend continued in the following years and TV stations that broadcast in 3D started to appear. These developments created a considerable demand for stereo content production. Production of quality stereoscopic video content is a difficult process that has to combine technical, perceptual and artistic aspects [1]. It is nowadays well known that certain stereoscopic issues in the 3DTV/3D cinema content may confuse the human visual system and brain and affect the viewing experience in a negative way. Moreover, a prolonged exposure to such 3D content can cause symptoms such as eye strain, headaches and visual fatigue [2]. To cope with these problems, 3D cinematographers have established a number of cinematographic rules to be followed during the production process. Nevertheless, in many cases, time constraints, low budget and poor advance planning prevent that from happening. However, many of these problems can be fixed in a post processing stage, as long as they are detected.

A few assistance systems for stereo shooting and 3DTV production and postproduction have been proposed [3],[4],[5],[6]. Such systems can analyze stereoscopic properties of a video to detect and/or correct mismatches (color, key-stoning, vertical disparity) or other 3D quality issues, or change depth composition of stereoscopic content.

In this paper, we propose two algorithms that exploit the available disparity information to detect and characterize depth jump cuts and bent window effects in stereoscopic videos [1], so that they can be fixed in a post-processing stage. Depth jump cuts occur between two consecutive shots when their depth does not match in terms of depth whereas bent window effects happen when an object that appears significantly in front of the screen interferes with the upper and lower frame boundaries. The proposed algorithms can be useful in assessing the overall quality of stereoscopic video content. A couple of papers deal with the issue of correcting depth jump cuts. In [7] the authors propose a method for non linear disparity adaptation, in which they present an example of interpolating depth jump cuts (rapid scene cuts), in order to create smooth transitions from one shot to the next. Kopal et al [5] proposed a viewer-centric editor for stereo cinema that gives the ability to the system operator to fix depth jump cuts by applying a cross fading effect, while translating images to change the convergence point before and after the cut, to have the incoming shot depth match that of the outgoing one. However, as far as we know, no method for the automatic detection of depth jump cuts has been proposed so far. The same applies to bent window effect.

2. DETECTION OF DEPTH JUMP CUTS

During the editing process, which is part of the post-production phase, individually recorded shots are assembled in a sequential order. This process is more complex in 3D cinematography, compared to the 2D one, because the editor has to take into consideration, among other factors, the depth continuity rule. This rule states that one should not cut between two shots if their depth does not match [1].

There is no objective definition for the “matching depth” concept between two shots, because it depends on the viewer's preferences. Nevertheless, a cut from a wide shot where objects are positioned behind the screen to a close-up inside the theatre space is a good example of non-matching depth cut, as the eye vergence point in the close-up shot is too far away from the convergence point in the wide-shot. The viewer loses 3D perception, until his/her visual system adapts to the new convergence point and the left and right images are fused together to produce a proper 3D scene perception again. This phenomenon is called a depth jump cut. A forward jump cut is much more disturbing than a backward one. In a forward jump cut, the new convergence point is closer to the viewer. Thus, the viewer's eye has to squint to restore stereopsis. On the contrary, in a backward jump cut, the eye convergence point is farther away from the previous one and the viewer has to relax its eye muscles.

In 3D cinematography, there is another type of depth cut, the so-called active depth cut. It is used in cases where a cut between two shots with "non-matching" depth is absolutely necessary e.g., in a live concert, where shots depicting the band are interchanged with shots depicting the audience. In an active depth cut, the eye vergence point of the wide-shot is moved to the screen plane, the cut to the close-up shot is performed and the close-up shot vergence point keeps moving towards the viewer, till it takes its correct position. Other types of transitions, very common in 2D cinematography, like cross fades, wipes and split screens can be adapted to fit 3D cinematography, but their use is very limited, because their implementation is much more difficult than in the 2D case. Furthermore, they do not provide significant changes in the way a viewer perceives the transition.

The algorithm proposed for the detection of the depth jump cuts begins by calculating the mean positive and negative disparity values for the entire disparity map for every video frame $i = 1,..,N$. Given a set of disparity maps $\mathbf{d} = \{d_1, d_2, ..., d_N\}$, for every disparity map d_i, two subsets are defined, $\mathcal{A}_i^+ = \{(u,v)|d_i(u,v) > 0\}$ and $\mathcal{A}_i^- = \{(u,v)|d_i(u,v) < 0\}$ and the average positive and negative disparity values are calculated as:

$$\overline{d_i^+} = \frac{1}{|\mathcal{A}_i^+|} \sum_{(j,k)\in\mathcal{A}_i^+} d_i(j,k) \tag{1}$$

$$\overline{d_i^-} = \frac{1}{|\mathcal{A}_i^-|} \sum_{(j,k)\in\mathcal{A}_i^-} d_i(j,k) \tag{2}$$

where $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. This way, two mean disparity signals $\overline{d_i^+}$ and $\overline{d_i^-}$, $i = 1,..,N$ are created, as shown in Figure 1b. Since the disparity maps are generally noisy, a median filter is applied on both the positive and negative mean disparity signals. Positive disparities suffer from noise more severely than negative ones, since they usually refer to the background, which is customarily displayed behind the screen and is often blurred and covers a much bigger region than foreground. Thus, disparity estimation is harder on the background than on the foreground. Taking the above into account, we use median filter masks of length $M = 5$ and $M = 15$ to filter the negative/positive mean disparity signals, respectively. Thus, two filtered disparity signals are constructed, namely $\overline{d_i^+}'$, $\overline{d_i^-}'$. Then, their first derivative, which is related to the speed of change of the average positive/negative disparity, is estimated numerically by $V_i^+ = \overline{d_{i+1}^+}' - \overline{d_i^+}'$ and $V_i^- = \overline{d_{i+1}^-}' - \overline{d_i^-}'$. In order to determine whether a depth jump cut is present, we set thresholds for the negative disparity speed T_{dsn} and the positive disparity speed T_{dsp}. The above thresholds cannot be fused to one, because the range of positive disparity values in order those values to lie within the comfort zone ($1-2\%$ of the screen width), is smaller than negative ones ($2-3\%$ of the screen width). In our experiments T_{dsp} is set to $0.005W$, (W being the screen width) and T_{dsn} to $0.0078W$, which is translated into approximately 10 and 15 pixels respectively for a fullHD (1920×1080 p) video. We check every point in signals $|V_i^+|$ and $|V_i^-|$, against the T_{dsp}, T_{dsn} thresholds, respectively. If either of the thresholds is exceeded, at time instance i, then we label a depth jump cut in the background and in the foreground respectively, while the velocity sign indicates a positive or negative depth jump cut. Note that there are cases where both depth jump cuts in positive and in negative disparities are present at the same time instance, as is the case in Figure 1b.

It has been proven experimentally that rapid changes in depth, such as depth jump cuts are annoying for the human visual system and cause visual fatigue and discomfort [8], [9], [10]. Thus, in its final step, our algorithm tries to rate a depth jump cut, according to the stress it causes. We define three discomfort characterizations for depth jump cuts namely "mildly uncomfortable", "uncomfortable", "highly uncomfortable". A characterization is given to a depth jump cut according to the sign of positive and negative depth change speeds V_i^+ and V_i^- as shown in Table 1. Rows 1,3,6 and 8 in Table 1 refer to cases where a jump cut in positive disparities is combined with a jump cut in negative disparities. In rows 2, 4, 5 and 7, a jump cut only in negative or positive disparities happens. The "=" sign means no jump cut. The labelling criteria, have been established as follows. After a negative jump cut in the foreground (negative disparities), the viewer has to squint the eyes to adapt to the new viewing condition where the object is much closer to him. This is a painful process that needs much time. On the other hand, after a positive depth jump cut in the foreground, the viewer has to relax the eyes, which is a quicker and less painful process. When a positive jump cut happens in background (positive) disparities, the viewer has to diverge the eyes to see the background clearly, which is a tiring process for the human visual system. In a negative jump cut in background disparities, the viewer has to converge his eyes, which is a

Table 1. Depth jump cuts characterization according to depth speed sign. U: uncomfortable, MU: mildly uncomfortable, HU: highly uncomfortable.

$\mathbf{V^-}$	$\mathbf{V^+}$	**Label**
-	-	**U**
	=	**U**
	+	**HU**
=	-	**MU**
	+	**U**
+	-	**MU**
	=	**MU**
	+	**U**

more comfortable process. Moreover, a jump cut in negative disparities is much more annoying and uncomfortable for the human visual system than a jump cut in positive disparities. At this point, it must be stated that the perceived intensity of a depth jump cut depends on the viewer's point of attention at the time the jump cut happens. Thus, it is highly subjective. Nevertheless, usually the viewer's focus lies on the foreground objects, which simplifies things. Thus, the previously described characterization approach is valid to a large extent.

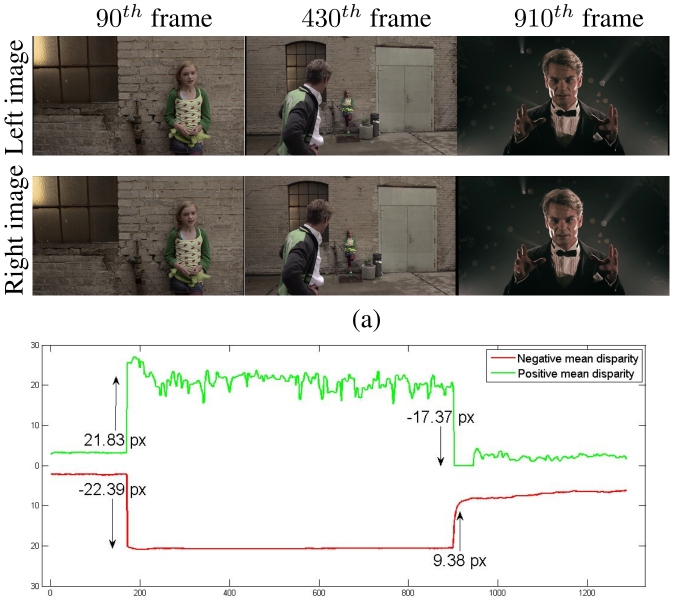
2.1. Depth Jump Cut Detection Example

In this section we provide an example of the performance of our algorithm in depth jump cut detection. The video segment is taken from the short movie "The Magician". The disparity maps are extracted using the Hybrid Recursive Matching (HRM) algorithm [11].

As shown in Figure 1, an on-the-screen medium close-up shot, is followed by a shot with large depth that starts at frame $n = 170$, which is followed by a shallow depth shot, beginning at frame $n = 902$. The disparity range in the first shot is $[-2, .., 3]$ pixels. In the second shot, the disparity range increases to $[-24, .., 25]$ pixels. The third shot, though, is a close-up and has shallow depth, since its disparity range is $[-8, .., 2]$ pixels.

The algorithm detects a negative depth jump cut of $V^-_{170} = -22.39$ pixels, at frame 170, as $V^-_{170} > T_{dsn}$. A positive depth jump cut is also detected, since the depth speed $V^+_{170} = 21.83$ pixels exceeds the positive depth change speed threshold T_{dsp}. The overall depth cut is labelled as "highly uncomfortable", because $V^-_{170} < 0$, and $V^+_{170} > 0$. Thus, viewer must squint his/her eyes considerably, to see the man appearing in front of screen plane, while he has to diverge his/her eyes too much to explore the background.

On the other hand, the depth cut at frame $n = 902$ is labelled as "mildly uncomfortable", as only the T_{dsp} is exceeded. As $V^+_{902} < 0$, viewer has to relax his eyes, so that they converge to a point closer to him to explore the background. It must be pointed out that a positive depth cut at frame 902 ($V^+_{902} = 9.56$ pixels) is not considered as a depth jump cut since it does not exceed the T_{dsn} threshold.



(a)

(b)

Fig. 1. (a) Sample frames from the three shots. (b) A "highly uncomfortable" depth jump cut detected at frame 170 and a "mildly uncomfortable" one at frame 902. Arrows show the first derivative in positive and negative mean disparities.

3. DETECTION OF THE BENT WINDOW EFFECT

The bent window effect occurs when an object with strong negative value in the (left) disparity map interferes with the top and bottom borders of the frame. In such a case, contradictory cues stem from the object position, since its top and bottom sides cannot be in front of the screen, as they are cut off by the frame's top and bottom borders, but the rest of the object is clearly in front of the screen, because it has strong negative disparity. The way the brain handles this conflict, in most cases, is to decide that the stereoscopic window is bent towards the viewer [1].

The proposed algorithm that detects a bent window effect takes as input the disparity map of a video frame (e.g., the left disparity map). Initially, we detect objects that have significantly negative disparity. To do so, we perform connected component analysis only on pixels with negative disparity lower than a threshold $-T_1$. A value of $T_1 = 0.025W$ (W being the frame width) is used to detect objects that appear clearly close to the viewer. Then we enclose every such object in a rectangular Region of Interest (ROI), whose upper-left and lower-right corner coordinates are $[X_{min}, Y_{min}]^T$ and $[X_{max}, Y_{max}]^T$. The final output of this step is a set of ROIs $\overline{R} = \{R_1, R_2, ..., R_N\}$. Then the algorithm checks if any of the objects R_i found in the previous step is in contact with the upper and lower frame boundary. If this is the case, the object is marked as the cause of a bent window effect. In other words, for every object R_i, if $Y_{min} = 0$ and $Y_{max} = H - 1$, the object R_i causes a bent window effect.

3.1. Bent window detection example

In the video frame shown in Figure 2 the metal pole has strong negative disparity of about -35 pixels and intersects the top and bottom edges of the frame. Thus, the algorithm labels the pole as the cause of a bent window effect. Disparity was estimated using the algorithm described in [12] which produces low quality disparity maps, showing that the proposed algorithm can operate on less accurate disparity values.

Fig. 2. A bent window effect example. The ROI that encloses the detected object is marked red. The yellow region has disparity values lower than the corresponding threshold.

4. CONCLUSIONS

Certain stereoscopic issues in the 3DTV content may confuse the human visual system and brain, affect the 3D viewing experience in a negative way and, eventually, cause symptoms such as eye strain, headaches and visual fatigue. In this paper algorithms for the automatic detection of depth jump cuts and bent window effects by exploiting disparity information are presented. Representative examples are provided proving the algorithms effectiveness. Our future plans include conducting perceptual tests to further verify the validity of the proposed algorithms, as well as developing algorithms for the detection of other such issues.

5. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287674 (3DTVS). This publication reflects only the author's views. The European Union is not liable for any use that may be made of the information contained herein. The authors wish to thank Camelot Broadcast and Max Hemmo for providing the movie "The Magician", used in the experimental evaluation of the jump depth cuts detection algorithm.

6. REFERENCES

[1] Bernard Mendiburu, *3D movie making. Stereoscopic digital cinema from script to screen*, Focal Press, 2009.

[2] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks, "Vergence – accommodation conflicts hinder visual performance and cause visual fatigue," *Journal of Vision*, 2008.

[3] Frederik Zilly, Marcus Mller, Peter Eisert, and Peter Kauff, "The stereoscopic analyzer an image-based assistance tool for stereo shooting and 3d production," in *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, September 2010.

[4] Simon Heinzle, Pierre Greisen, David Gallup, Christine Chen, Daniel Saner, Aljoscha Smolic, Andreas Burg, Wojciech Matusik, and Markus Gross, "Computational stereo camera system with programmable control loop," *ACM Transactions on Graphics*, vol. 30, no. 4, 2011.

[5] Sanjeev J. Koppal, C. Lawrence Zitnick, Michael F. Cohen, Sing Bing Kang, Bryan Ressler, and Alex Colburn, "A viewer-centric editor for stereoscopic cinema," *IEEE Computer Graphics and Applications*, vol. 31, no. 1, 2011.

[6] The Foundry, "Ocula stereoscopic post-production plug-ins for Nuke," https://www.thefoundry.co.uk/products/ocula/.

[7] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross, "Nonlinear disparity mapping for stereoscopic 3d," *ACM Transactions on Graphics*, vol. 29, no. 3, pp. 10, 2010.

[8] Sumio Yanoa, Masaki Emotoa, and Tetsuo Mitsuhashia, "Two factors in visual fatigue caused by stereoscopic HDTV images," *Displays*, vol. 25, no. 4, pp. 141–150, November 2004.

[9] Sumio Yanoa, Shinji Idea, Tetsuo Mitsuhashib, and Hal Thwaitesc, "A study of visual fatigue and visual comfort for 3d HDTV/HDTV images," *Displays*, vol. 23, pp. 191–201, October 2002.

[10] Kazuhiko Ukai and Peter A. Howarth, "Visual fatigue caused by viewing stereoscopic motion images: Background, theories and observations," *Displays*, vol. 29, pp. 106–116, October 2007.

[11] N. Atzpadin, P. Kauff, and O. Schreer, "Stereo analysis by hybrid recursive matching for real-time immersive video conferencing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 3, pp. 321–334, March 2004.

[12] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proceedings of IEEE Conference of Computer Vision*, 2001, vol. 1, pp. 508–515.