AUTOMATIC 3D DEFECTS IDENTIFICATION IN STEREOSCOPIC VIDEOS

Sotirios Delis *Nikos Nikolaidis* *Ioannis Pitas*

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

ABSTRACT

3DTV and 3D cinema have become quite popular during the last few years. It is now well understood that certain 3D video quality issues may have a negative effect in the 3D viewing experience. In this paper, we propose two novel algorithms that exploit available disparity information, in order to detect two disturbing stereoscopic issues, namely Stereoscopic Window Violations (SWV) and bent window effects. The algorithms' performance is tested on a number of examples. The proposed algorithms can be used for assessing the overall quality of stereoscopic video content or in order to enable fixing the detected issues in a post-production stage.

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

1. INTRODUCTION

A breakthrough in 3D cinema took place in 2009, sparked by the success of James Cameron's Avatar 3D movie. This trend continued in the following years and was one of the reasons, along with the availability of affordable 3D TV sets and the perfection of 3D video capturing and processing hardware and filming techniques, for the rise of 3D television (3DTV). Indeed, more than 35 3DTV stations broadcast today worldwide. These developments significantly increased the demand for stereo content production. Based on the accumulated experience, it is now known that certain stereoscopic issues in 3DTV (or 3D cinema) content may confuse the viewers visual system and ruin or negatively affect the 3D viewing experience. Even worse, viewing such content for a long period can cause eye strain, headaches and visual fatigue [1]. 3D cinematographers have already identified such issues and created a number of stereography rules which, if followed during the production process, can alleviate the problems. In addition, many of these problems can be fixed in a post processing stage, provided that they are detected.

In this paper, we propose two novel algorithms that utilize disparity information, in order to detect stereoscopic quality defects in videos. Particularly, we cope with detection of the Stereoscopic Window Violations (SWV) and bent window effects. SWVs happen when objects appearing in front of the screen touch the left or the right frame boundaries, thus resulting in retinal rivalry, whereas a bent window effect occurs when an object appearing significantly in front of the screen interferes with the upper and lower frame boundaries. The proposed algorithms can be used for assessing the overall quality of stereoscopic video content or for enabling fixing the detected defects in a post-processing stage.

Section 2 describes the related work. The two stereoscopic issues studied in this paper, the algorithms proposed for detecting them and some examples of the algorithms performance are provided in Section 3. Finally, conclusions are drawn in Section 4.

2. RELATED WORK

Production of quality stereoscopic video content is a demanding task that requires combining technical, perceptual and artistic aspects [2]. There are certain software and hardware devices nowadays, that support stereographers in their work. Among others, they help to avoid annoying phenomena, such as SWV in production stage or to eliminate them in post-production. Indeed, a small number of assistance systems for stereo shooting and 3DTV production have been proposed. Fraunhofer Heinrich-Hertz Institute [3] presented the stereoscopic analyzer (STAN) that detects SWV and gives a framing alert. The utilized approach works in real time but is of limited accuracy as it involves sparse disparity maps. Moreover, it involves special hardware and is embedded on a stereo camera system. Heinzle et al, refer to SWV (it is referred as framing violation) detection as a possible extension of the computational stereo camera system they propose [4], but such an algorithm is neither implemented, nor tested. The same team proposed a method that corrects the SWV, by pushing the object that violates the frame's border behind the screen with a disparity scaling. Kopal et al [5] proposed a viewer-centric editor for stereo cinema that gives the ability to the system operator to fix SWV, by adding a floating window mask (it is referred as proscenium arch) to the appropriate image. However, both the SWV detection and its correction are done manually. Finally, Tseng et al, divide the disparity map into a set of small blocks [6]. Beginning from a block that lies on the frame boundaries and has a number of pixels with negative disparities greater than a threshold, its neighbouring blocks in perpendicular directions are checked for negative disparity values. This way large blocks of negative disparity regions that lie on frame borders are detected. This SWV detection method, although simple, may suffer from a significant false alarm rate as it does not take into account the interpolated disparity values for rivalry regions and the violation duration and width.

3. METHODS DESCRIPTION

In this section we provide a description of the two 3D issues/defects, namely SWV and bent window and present in detail the algorithms we have devised, in order to automatically detect these issues. Each deffect/cinematographic rule violation, the corresponding detection algorithm and some representative examples are presented in separate subsections.

Disparity estimation plays a crucial role in the proposed algorithms, since the disturbing stereo issues are related to visual depth perception. The various disparity estimation algorithms may be classified into two main groups, local and global methods. Local algorithms use a finite neighbourhood window, give less accurate results than global ones but are far less time consuming. Global methods try to minimize the energy of a global cost function and produce quite good results with the disadvantage of high computational complexity. In our experiments two methods for disparity extraction were used, the one described in [7] and [8] and a publicly available one [9], which is part of the OpenCV 2.3 library [10]. The first method gives much more accurate disparity maps by exploiting, among others, temporal information available in a video sequence. The second method generates worse results, thus giving us the opportunity to test the proposed algorithms in cases when a less accurate disparity estimation is available.

We define the image width and height as W and H respectively and thus the pixel indices are in the ranges [0, W-1]x[0, H-1]. The left and right disparity maps (i.e. the disparity maps from the left image to the right one and from the right image to the left one respectively) are defined as $d^l_{u,v}$ and $d^r_{u,v}$, where $u = 0 , .. , W - 1$ and $v = 0 , .. , H - 1$. Disparity takes values in the range $[-128, 128]$. The sign is indicative of whether a pixel is displayed in front or behind the screen plane.

In order to define the thresholds we use in the proposed algorithms, we use the simple percentage rules followed by most 3D cinematographers, which are based on Percival's comfort zone boundaries [11]. More specifically, a negative left disparity up to $2 - 3\%$ of the frame width and a positive left disparity up to $1 - 2\%$ of the frame width is allowed, in order the stereo content to be viewed comfortably.

3.1. Stereoscopic Window Violation

In 3D cinematography and 3DTV, we are observing the 3D world through a window, the Stereoscopic Window (SW) [12], namely the TV or cinema screen. The 3D depth perception is based on the fact that a feature in the left image is horizontally shifted, with respect to the corresponding feature in the right image. An unavoidable consequence of the above principle is retinal rivalry on the left or right frame edges, when object parts that are positioned close to left image's left or right border do not have correspondence (are not displayed) in the right frame and vice versa. For objects with zero disparity no retinal rivalry is observed. When part of an object is cut off by the edge of the display, it results in the so called SWV and is interpreted as occlusion by the viewer.

The SWV does not create any problems when it happens behind the screen (objects with positive left disparity), because both disparity and occlusion cues dictate that the object is behind the screen. However, when SWV involves objects that appear in front of the screen (i.e. they have negative left disparity) the occlusion cue conflicts the disparity one. Generally, occlusion supersedes disparity cue, and, thus, finally, the object is perceived as being behind the screen plane. The above are true for a mild SWV, where only a small part of the object that interferes with left or right frame edge is missing from the other image, while in a severe SWV the missing part is so big that the human brain cannot fuse the images and eventually see 3D. SWV in negative disparities is not only undesirable, but may also be painful. A frequently used approach to fix the SWVs is the so called floating window. A floating window is created by adding black masks on the sides of the left or right image.

The rule about SWV states that a cinematographer has to avoid violating the stereoscopic window while an object has negative left disparity. There is one notable exception, related to object speed [2]. Objects that exit, enter or traverse the video frame in no more than half a second cause no problem. It must be pointed out that all the above apply to cases of mild SWV, when the rivalry region is relatively narrow.

The proposed algorithm for SWV detection firstly selects pixels u, v for which $d^l_{u,v} < -T_1$ in the left disparity map and $d^r_{u,v} > T_1$ in the right disparity map. In order to exclude objects that do not appear significantly in front of the screen, we set the threshold T_1 to a suitable value and perform connected component analysis with an 8-point neighbourhood to extract objects (connected components) that are displayed significantly in front of the screen. A value of $T_1 = 0,0025W$ worked well in our experiments. To reduce noise, objects with small width (less than T_w) or height (less than T_h) are rejected. Threshold values of $T_w = 0.02W$ and $T_h = 0.04H$ work well. The detected objects are then enclosed into rectangular bounding boxes (Regions of Interest, ROIs). Thus, two sets of ROIs $\mathbf{R}^r = \{R^r_1, R^r_2, ..., R^r_n\}$ and $\mathbf{R}^l = \{R^l_1, R^l_2, ..., R^l_k\}$ are created for the left and right images, respectively. These ROIs are represented by their upper left and lower right corner coordinates $[X^j_{i,min}, Y^j_{i,min}]^T$ and $[X^j_{i,max}, Y^j_{i,max}]^T$, where $j = \{r, l\}$ and i is the ROI index.

Two types of disturbing SWVs can be defined. In the first type, namely left SWV, the SWV happens on the left frame border, since there is a region in the left image which is missing from the right one. By detecting this phenomenon one can detect the SWV. The algorithm works as follows. If one or more object ROIs R^r_i with disparity characteristics such as those described in the paragraph above lie on the left border of the right image, that is, if $X^r_{i,min} = 0$, a SWV is present. This is because $X^l_{i,min} = X^r_{i,min} + d^r_{i,j} > 0$ and thus, the region $[0, d^r_{i,j}]$ in the left image is not present in the right one. In order to reduce false alarms because of inaccuracies in disparity maps, another condition is introduced. The number of pixels that belong to the object in the two leftmost ROI columns must be greater than a threshold T_2, expressed as a percentage of the ROI height, to decide that this object signals a SWV. In our experiments T_2 is set to $0.3h_{ROI}$, where h_{ROI} is the ROI height.

A similar procedure is followed for the detection of the second type of SWV, namely right SWV. If one or more object ROIs detected in the left disparity map R^l_i lie on the right border of the left image, i.e., if $X^l_{i,max} = W - 1$, a SWV is present. This is because $X^r_{i,max} = X^l_{i,max} + d^l_{i,j} < W - 1$ and therefore the region $[W + d^l_{i,j}, W - 1]$ in the right image is not present in the left one. The false alarm reduction approach is applied here too.

As already mentioned, a SWV can be tolerated by the human brain, if its duration is short enough. Therefore, when a SWV of duration d_{SWV} frames is detected, either in the left or right image, the following condition is checked:

$$d_{SWV} > fps/2, \quad (1)$$

where fps is the video frame rate. Essentially, this condition checks whether the SWV lasts more than half a second. If the above condition is true, an annoying left or right SWV is signaled.

In the case of an annoying SWV, the floating window width, that can fix the SWV, is estimated as follows. In the case of a left SWV, in order to estimate the appropriate mask width that must be applied on the left image, we first calculate the mean value $\overline{m}^r_i$ of the first three columns of right image object disparities for every object that creates SWV:

$$\overline{m}^r_i = \left(\sum_{u=0}^{2} \sum_{v=Y_{min}}^{Y_{max}} d^r_{u,v} \right) / \, 3\, h_i, \quad \forall R^r_i : X^r_{i,min} = 0, \quad (2)$$

where h_i is the height of object R^r_i and $X^r_{i,min}$ is the object's left vertical boundary. The appropriate width FW_l of the left floating window mask is the mean value of all $\overline{m}^r_i$:

$$FW_l = \left(\sum_{i=1}^{n} R^r_i \right) / \, n, \quad (3)$$

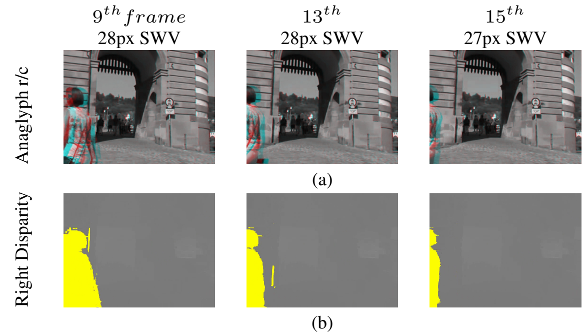

Fig. 1. A left stereoscopic window violation example. The left and right images are combined in an anaglyphic red cyan image. The right disparity maps are also depicted. Yellow marked pixels have disparities greater than 5 pixels.

where n is the number of objects that cause SWV, when detected in the right disparity map. This is done because the disparities of boundary ROI pixels that create SWV point at the boundary line of the region visible to only one eye. The right floating window mask width FW_r is estimated using a similar approach. For every object in the left disparity map that creates a SWV, a mean disparity value $\overline{m}^l_i$ is calculated using the three last disparity map columns and the mean value of all $\overline{m}^l_i$ gives the floating window width FW_r.

Although the floating window is a quick and effective way to correct SWVs, it only works with mild SWVs. To take this into account, we have set a threshold T_{FW} on the floating window width. If this threshold is exceeded, the SWV is characterized as strong SWV and no floating window can be applied to fix it. A value for T_{FW} that performed well in our experiments is equal to $0.03W$.

3.1.1. *Examples of Stereoscopic Window Violation Detection*

In the first example, shown in Figure 1, a left SWV occurs on the left side of the video frame. The right disparity map (Figure 1b) signals the beginning of the violation, when the lady hits the right image border in the second frame ($n = 2$), while being in front of the screen. The algorithm detects a SWV that starts at the second frame and ends when the lady disappears at frame $n = 17$. The detected SWV duration is 16 frames or $16/25 = 0.64$ seconds (the video fps is 25). Thus, the duration threshold $T_d = 25/2 = 13$ frames, is exceeded and the violation is labelled as annoying. Therefore a floating window is needed to fix it. Its width ranges from 28 to 30 pixels, which means that the violation is mild and can indeed be fixed by applying a floating window mask on every left frame where the SWV occurs. Disparity in this and the next case was estimated using the algorithm described in [7], [8].

In the example shown in Figure 2, a right SWV is demonstrated. Here the left disparity map, shown in Figure 2b signals the beginning of the SWV, when the lady enters the frame while being in front of the screen. The algorithm stops sig-

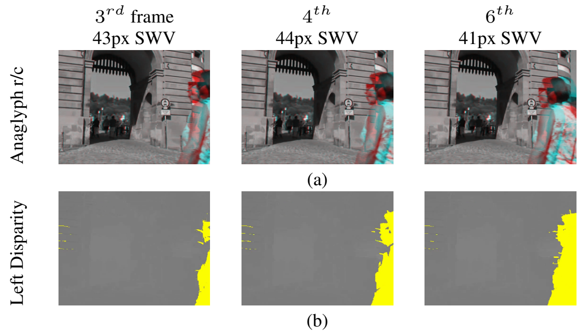

Fig. 2. A right stereoscopic window violation example. Left and right images are combined in an anaglyphic red cyan image. The right disparity maps are also depicted. Yellow marked pixels have disparities lower than -5 pixels.

nalling SWV when the lady disappears from the right image (frame $n = 9$). The SWV lasts for 7 frames which is lower than $fps/2 = 13$ frames and the violation is not annoying. However, the violation width range of 36 to 44 pixels, though mild, is significant. Thus a floating window mask to every right image involved in the SWV can fix the problem.

3.2. The Bent Window Effect

Although the most distracting SWVs are those that occur at the left or right side of the screen, because they cause retinal rivalry, a violation can also happen at the top and bottom borders of a frame. In such a case, the brain perceives contradictory cues. Indeed if an object with strong negative disparity is positioned so that its top and bottom sides are cut off by the frame's top and bottom borders, this indicates that it cannot be in front of the screen. However, due to its disparity, the rest of the object is clearly in front of the screen. In most cases, the brain handles this conflict by deciding that the stereoscopic window is bent towards the viewer [2].

The algorithm that detects a bent window effect operates on the disparity map of a video frame (e.g., the left disparity map). At first, objects that have significantly negative disparity are detected by performing connected component analysis only on pixels with negative disparity lower than a threshold $-T_1$. A value of $T_1 = 0.025W$, helps detecting objects that appear clearly in front of the screen. Every such object is enclosed in a rectangular ROI, whose upper-left and lower-right corner coordinates are denoted by $[X_{min}, Y_{min}]^T$ and $[X_{max}, Y_{max}]^T$ respectively. Thus, a set of ROIs $\overline{R} = \{R_1, R_2, ..., R_N\}$ is generated. Subsequently, the algorithm checks if any of the objects ROIs R_i is in contact with the upper and lower frame boundary. If such a case, the object is marked as the cause of a bent window effect. In more detail, if $Y_{min} = 0$ and $Y_{max} = H - 1$, the object R_i causes a bent window effect. When detected, the bent window can be alleviated by reducing the disparity value of the object that causes it, by translating left image ROIs with respect to right ones.

3.2.1. Bent Window Detection Example

In the video frame depicted in Figure 3 the negative disparity of the tree trunk is close to -30 pixels. The trunk intersects the top and bottom edges of the frame. As a result, the algorithm signals a bent window effect. Disparity was estimated using the algorithm described in [9], showing that the proposed algorithm can operate on less accurate disparity values.

Fig. 3. A bent window effect example. The ROI that encloses the detected object is marked red. Yellow marked regions have disparity values lower than -15 pixels.

4. CONCLUSIONS

Stereoscopic defects in the 3DTV content may negatively affect the 3D viewing experience or even cause eye strain and headaches. In this paper novel algorithms were presented that automatically detect two such stereoscopic defects, namely, stereoscopic window violations and bent window effects, by exploiting disparity information. Examples are provided that showcase their effectiveness in detecting the above mentioned defects. Future work includes performing subjective evaluation tests in order to verify that the detected defects are indeed annoying and for fine-tuning the involved thresholds. Algorithms for the detection of other such issues are also currently under development.

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.

6. REFERENCES

[1] 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.

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

[3] F. Zilly, M. Muller, P. Eisert, and P. Kauff, "The stereoscopic analyzer: An image-based assistance tool for stereo shooting and 3D production," in *Proceedings of IEEE 17th International Conference on Image Processing*, September 2010.

[4] S. Heinzle, P. Greisen, D. Gallup, C. Chen, D. Saner, A. Smolic, A. Burg, W. Matusik, and M. Gross, "Computational stereo camera system with programmable control loop," *ACM Transactions on Graphics*, vol. 30, no. 4, 2011.

[5] S. Koppal, L. Zitnick, M. Cohen, S. Kang, B. Ressler, and A. Colburn, "A viewer-centric editor for stereoscopic cinema," *IEEE Computer Graphics and Applications*, vol. 31, no. 1, 2011.

[6] K.-L. Tseng, W.-J. Huang, A.-C. Luo, W.-H. Huang, Y.-C. Yeh, and W.-C. Chen, "Automatically optimizing stereo camera system based on 3D cinematography principles," in *Proceedings of 3DTV Conference*, October 2012.

[7] 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.

[8] C. Riechert, F. Zilly, and P. Kauff, "Real time depth estimation using line recursive matching," in *Proceedings of European Conference on Visual Media Production*, November 2011.

[9] 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.

[10] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with intel open source computer vision library," *Intel Technology Journal*, vol. 9, no. 2, pp. 119–130, 2005.

[11] A. S. Percival, "The relation of convergence to accommodation and its practical bearing," *Ophthalmological Review*, vol. 11, pp. 131–328, 1892.

[12] B. Mendiburu, Y. Pupulin, and S. Schklair, *3D TV and 3D cinema. Tools and processes for creative stereoscopy*, Focal Press, 2012.