DETECTION OF STUCK-TO-BACKGROUND OBJECTS IN CONVERTED S3D MOVIES

Stanislav Dolganov, Mikhail Erofeev, Dmitriy Vatolin, Yury Gitman

Lomonosov Moscow State University
Moscow, Russian Federation

ABSTRACT

The creation of S3D movies by converting 2D captured footage often introduces depth-map inaccuracies. Such artifacts can significantly degrade the viewing experience even if they occur only in unsalient background objects.

In this paper we propose a method for detecting foreground objects that are stuck to the background. Our method extracts information about motion in the scene and detects conversion-related discrepancies between motion strength and depth. We demonstrate the performance of the method by applying it to 39 full-length converted 3D movies and by providing the results of our analysis as well as examples of detected problem shots.

Index Terms**—** Stereoscopic video, quality assessment, stereo matching, 2D-to-3D conversion, depth estimation.

1. INTRODUCTION

Interest in 3D cinema and 3D television is rising continuously, creating demand for stereoscopic content, but development of content-generation techniques is much slower. Content generation comes in three common forms: capture using a stereoscopic camera system, conversion from 2D footage, and computer rendering. The last approach relates specifically to animated films and postproduction effects. Capturing video using stereoscopic cameras requires alignment of color and geometry characteristics—a difficult and thus extremely error-prone process. Considering the recent progress in 2D-to-3D conversion tools [12, 15], many filmmakers prefer the conversion approach. This preference becomes obvious when comparing the ratio of converted/captured films over the last 60 years [18].

Although existing tools greatly simplify the various stages of the conversion process, the work is still laborious and far from automated. Mike Seymour [13] lists the main problems with stereo generation given a 2D source: inaccurate inpainting of background occlusions, lack of depth texture in foreground objects (the cardboard effect), cropping discrepancies in negative-parallax foreground objects that cross the screen frame (floating window), aggressive stereo-budget distribution between neighboring scenes, and general depth-map quality.

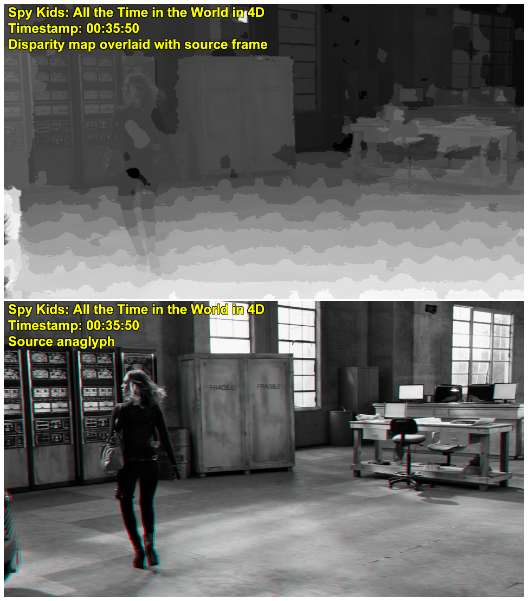

Spy Kids: All the Time in the World in 4D
Timestamp: 00:35:50
Disparity map overlaid with source frame
Spy Kids: All the Time in the World in 4D
Timestamp: 00:35:50
Source anaglyph

Fig. 1. Example of a problem frame detected using proposed method: the foreground object is stuck to the background. The filmmakers failed to draw the actress's body on the depth map, so the scene has unnatural perceptual qualities and may cause visual discomfort.

"Cardboarding" and other depth-map defects that arise during 2D-to-3D conversion often diminish the perceptual experience. The most noticeable problems relate to salient areas having inconsistencies between their depth and motion edges [9, 10], which the viewer perceives as in-depth motion or object deformation.

We aim in this work to provide a tool that automatically examines depth-map quality, allowing filmmakers to produce more-accurate depth maps. Our proposed method requires as input a stereoscopic-3D video sequence and, optionally, the depth map used during its conversion to 3D format. In cases where depth information is unavailable, the method estimates the disparity map using [14, 22]. It detects problems by comparing the depth-map edges of each frame with the

The reported study was funded by RFBR according to the research project 15-01-08632 A.

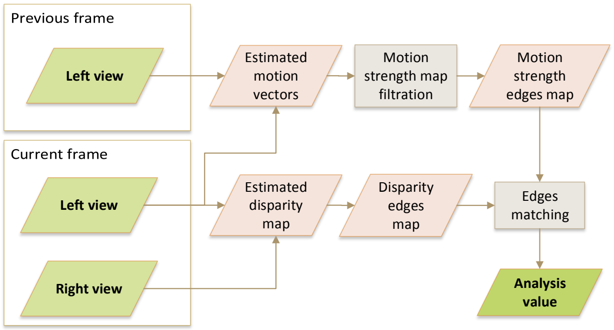


Fig. 2. Workflow of proposed method. The disparity-estimation step is optional and can be skipped if the original depth map is available.

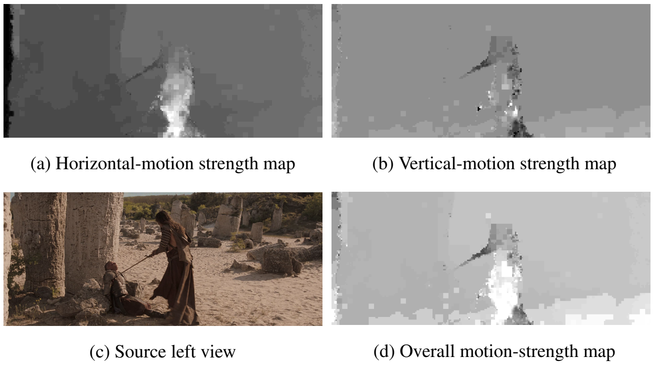
motion-strength-map edges. This approach allows us to detect moving objects that are partially present in or completely absent from the depth map. Since no generally accepted fast and reliable method of motion estimation is available, we apply [14] to the sequence of left views and then enhance an estimated map using spatio-temporal filtering [5, 11, 7]. The final per-frame quality score is the intensity of the depth and motion mismatch. The per-scene score is the average of the corresponding per-frame scores weighted by a motion-estimation confidence value. Section 3 describes each step in more detail. Section 4 discusses the results of applying the proposed method to 39 full-length S3D movies. Fig. 1 shows an example of a detected artifact.

2. RELATED WORK

Over the years researchers have studied 2D-video quality assessment. Wang [17] offers a comprehensive survey of progress in this area so far. When assessing the quality of stereoscopic content, the main particularity relates to multiple inconsistencies between views and also to depth-budget temporal consistency. Voronov et al. [16] recently proposed an approach for detecting inter-view color, sharpness, and geometry mismatches. Akimov et al. [1] proposed a framework for channel-mismatch detection.

Quality artifacts in converted content are more diverse and have undergone relatively little study. Bokov et al. [2] proposed a way to detect the three common problems in converted S3D content: edge-sharpness mismatch, flat objects, and flat scenes. Expectedly, their method can detect problems only for objects that appear in the depth map. In this paper we propose a complementary solution for detecting foreground objects that are absent from or only partially present in the depth map.

Our proposed method is a composition of "depth from motion" extraction and depth- and motion-contour matching. Both problems are well studied.



(a) Horizontal-motion strength map

(b) Vertical-motion strength map

(c) Source left view

(d) Overall motion-strength map

Fig. 3. Example maps of horizontal (a) and vertical (b) motion extracted using a block-based motion-estimation algorithm [14] for the source frame (c). The results are combined into an overall motion-strength map (d).

A common approach to depth-from-motion extraction assumes optimization of the energy function with the data term being based on the motion-strength estimate. Zhang et al. [20] additionally include the temporal-coherence constraint, and they allow every color segment to rotate in the direction that reduces energy. Their approach produces depth maps with extremely high temporal coherence. It takes enormous amount of time, however, and thus is unsuitable for analyzing long sequences. A recently proposed edge-aware filter [7] allows us to employ a less sophisticated approach: we obtain a rough motion-strength map through a block-based motion-estimation algorithm [14] and then enhance the result using edge-aware convolution.

To address the contour-matching problem, several very precise methods of closed-contour matching have recently appeared. For instance, Xu et al. [19] proposed a method based on contour-flexibility descriptors. Extension to open contours is possible but requires extraction of problem edges using a watershed transform. This transform leads to significant growth in second-order errors, however. Felzenszwalb et al. [6] proposed a competing approach that uses a hierarchical shape tree. Unfortunately, this method is not extensible to discontinuous contours, which are critical to our case because motion extraction is imperfect and the resulting motion-edge maps contain numerous discontinuities. Thus, we employ a less precise but more robust approach to input-contour-map inaccuracies that is similar to a distance transform [3].

3. PROPOSED METHOD

Fig. 2 shows the scheme of the proposed method for detecting inconsistencies between object motion and the depth map used in converting 2D to 3D video. The method input is S3D video converted from 2D and, optionally, the depth map used during conversion. The method outputs a per-frame score that can be interpreted as a perimeter of objects that were absent from the

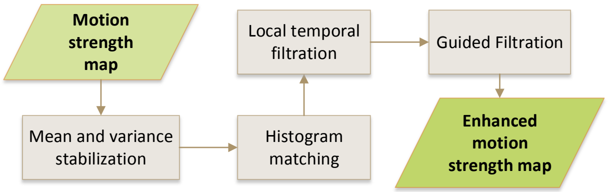
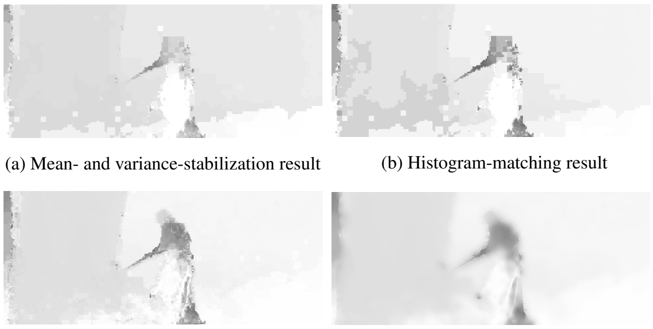

Motion strength map
Local temporal filtration
Guided Filtration
Mean and variance stabilization
Histogram matching
Enhanced motion strength map

Fig. 4. Pipeline of spatio-temporal filtering. This approach enables us to enhance the rough motion-intensity map that we obtain from the motion-estimation algorithm.



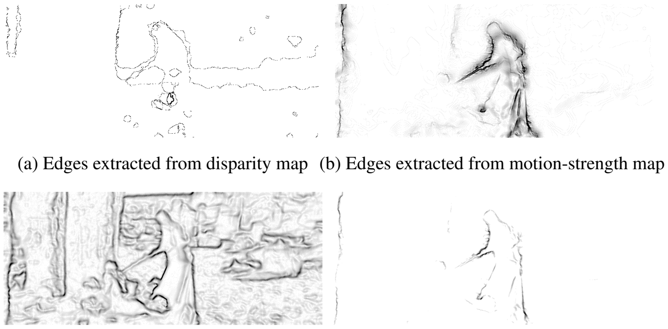
(a) Mean- and variance-stabilization result (b) Histogram-matching result

(c) Temporal-filtration result (d) Guided-filtration result

Fig. 5. Illustrations of each intermediate stage in spatio-temporal filtration pipeline from Fig. 4.

depth map. It comprises the following steps:

1. Estimate left-view motion vectors between previous and current frames.
2. Estimate disparity map between left and right view (if no depth map is available). We can use a disparity map instead of a depth map because our method requires only information about depth-map edges, which match the disparity-map edges.
3. Compute motion-strength maps using a motion-vector field refined by temporal and spatial processing to improve its temporal coherence.
4. Perform edge matching between motion-strength and depth maps.
5. Compute area of motion-strength-map edges that are absent from the depth map. This area is the final frame score.



(a) Edges extracted from disparity map (b) Edges extracted from motion-strength map

(c) Edges extracted from source frame (d) Intersection of source and motion edges

Fig. 6. Edges extracted from the disparity map (a), filtered motion-strength map (b), and source frame filtered using a scale-aware method (c). The intersection of the motion and source-frame edge maps appears in (d).

To extract both the motion and disparity maps, we employ the block-based matching approach described in [14]. To estimate a confidence value for the resulting maps, we use a left/right-consistency constraint (LRC) [4] (which takes the form of a previous/current-consistency constraint in the case of motion estimation).

We compute motion-strength map M as the magnitude of a motion-vector field (see Fig. 3 (a) and (b)). Then, to improve the temporal coherence of the motion-strength map, we apply the pipeline of temporal and spatial filters shown in Fig. 4.

We begin by applying a linear transform to the motion-strength map; this transform sets the minimum value in the map equal to 0 and the maximum value equal to 1 (i.e., auto levels). Fig. 3 (d) shows an example intermediate result for this step. Owing to changes in camera and object velocity as well as errors in the motion-vector field, the resulting motion-strength map contains a good deal of flickering. To reduce this effect, we set the mean and standard deviation of each frame equal to the means of these values computed over the previous n frames (our experiments used $n = 10$):

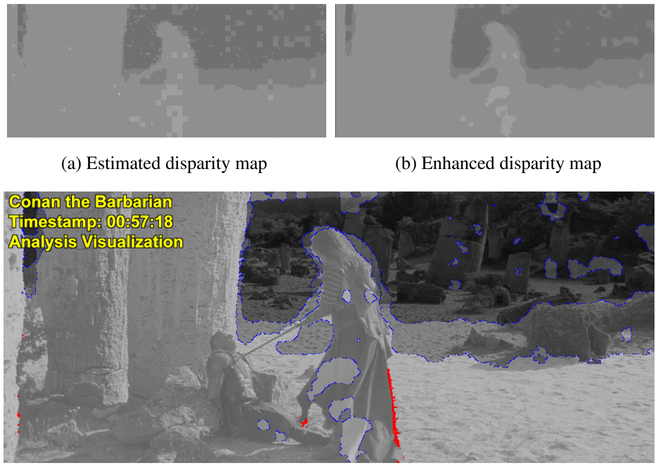
$$M_i^{'k} = \frac{\mathsf{Std}_{k-n}^{k-1}}{\mathsf{Std}\left[M^k\right]}\left(M_i^k - \mathsf{E}\left[M^k\right]\right) + \mathsf{E}_{k-n}^{k-1} \tag{1}$$

Here, E and Std are the mean and standard deviation, respectively; M_i^k is pixel i of the motion-strength map for frame k; and E_{k-n}^{k-1} and Std_{k-n}^{k-1} are the temporal mean and standard deviation, respectively. We calculate these last two values as follows:

$$\mathsf{E}_{k-n}^{k-1} = \frac{1}{\#}\sum_i\left(\frac{1}{n}\sum_{p=k-n}^{k-1} M_i^p\right) \tag{2}$$

$$\mathsf{Std}_{k-n}^{k-1} = \sqrt{\frac{1}{\# - 1}\sum_i\left(\frac{1}{n}\sum_{p=k-n}^{k-1} M_i^p - \mathsf{E}_{k-n}^{k-1}\right)^2} \tag{3}$$

In (2) and (3) # denotes the number of pixels in the frame. Fig. 5 (a) shows an intermediate result from this step. As a



(a) Estimated disparity map

(b) Enhanced disparity map

(c) Visualization of final result

Fig. 7. Examples of a disparity map before (a) and after (b) weighted-median filtering. The resulting visualization (c) that we offer to content creators is a disparity map overlaid on the source frame. We use blue to mark the disparity edges and red to mark any inconsistencies between motion- and disparity-map edges.

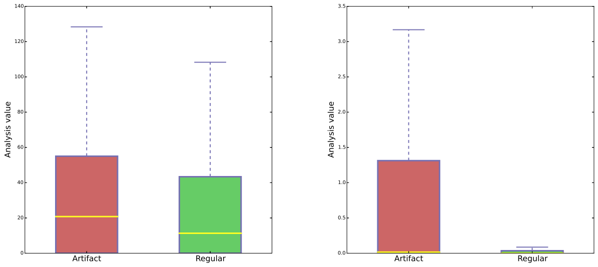
final global transform we apply histogram matching [5] to the sequence of motion-strength maps. This transform makes the histogram of each these maps match the mean histogram of the previous n frames. An example result appears in Fig. 5 (b).

To remove any local flickering that persists after the global transforms, we use the temporal-stabilization method proposed in [11] (see the example result in Fig. 5 (c)).

Finally, to improve the consistency of the motion-strength-map edges with image edges and to fill areas of low confidence, we use a guided filter [7] with the input frame as the guide. Fig. 5 (d) shows an example of the final motion-strength map.

Now consider the case where no depth map is available and the disparity map is estimated using a block-based matching algorithm.To improve the disparity map we apply a fast version of the weighted-median filter described in [22]. This filter refines the object contours and removes extraneous edges from what should be smooth disparity transitions. An example of an estimated disparity map before and after filtration appears in Fig. 7 (a) and (b).

We apply a Scharr operator [8] to both the filtered motion strength-map and the depth map to detect their edges. Fig. 6 (a) and (b) shows example edge maps. Since the motion-strength edge map contains blurred edges caused by guided filtration, we use edge information from the source RGB frame to sharpen them. We cannot, however, use the edge map of the source frame as is, since it contains numerous texture edges that are not actual object edges. To remove most texture edges we apply scale-aware filtration [21] to the source frame and extract the edge map from the filtered frame (see Fig. 6 (d)).



(a) Scores computed using initial algorithm

(b) Scores computed using modified algorithm

Fig. 8. Mean scores of the initial algorithm (a) for true- and false-positive scenes, as well as mean scores computed for the same scenes using an algorithm modified to exclude RGB uniform regions (b).

Finally, we compute the intersection of the edges we extracted from the motion-strength map (M) and the filtered source frame (I) using the equation below:

$$M'^{k}_{i} = M^{k}_{i} \max_{j \in B_r(i)} (I^{k}_{j}) \tag{4}$$

Here, $B_r(i)$ is the neighborhood of radius $r = 0$ around pixel i. We use the same procedure with $r = 5$ to refine the disparity map.

To get a final quality score for the current frame, we intersect the edges of the refined motion-strength map with the edges of the disparity map using a method inspired by the distance transform. For each pixel in the motion-strength edge map we iteratively calculate the distance value:

$$\begin{aligned} M^{k,0}_{i} &= M^{k}_{i}, \\ M^{k,t}_{i} &= M^{k,t-1}_{i} - \max_{j \in B_t(i)} w(i,j) \frac{\left(\vec{M}^{k,t-1}_{i}, \vec{D}_j\right)}{\|\vec{D}_j\|} \end{aligned} \tag{5}$$

In this expression, $w(i,j)$ is a distance weight function:

$$w(i,j) = \exp\left(\frac{\|i-j\|^2}{2\sigma^2}\right) \tag{6}$$

Here, $\vec{M}^{k,t-1}_{i}$ and $\vec{D}_j$ are motion-strength and disparity edge maps extracted using a Scharr operator. The resulting map contains only edges that are present in the motion-strength map but are missing from the depth map. These edges likely belong to objects that do not appear in the depth map. An example of a final result appears with red lines in Fig. 7 (c).

4. EXPERIMENTAL EVALUATION

We used the algorithm described in the previous section to evaluate the false-positive ratio for a set of five movies. To create a

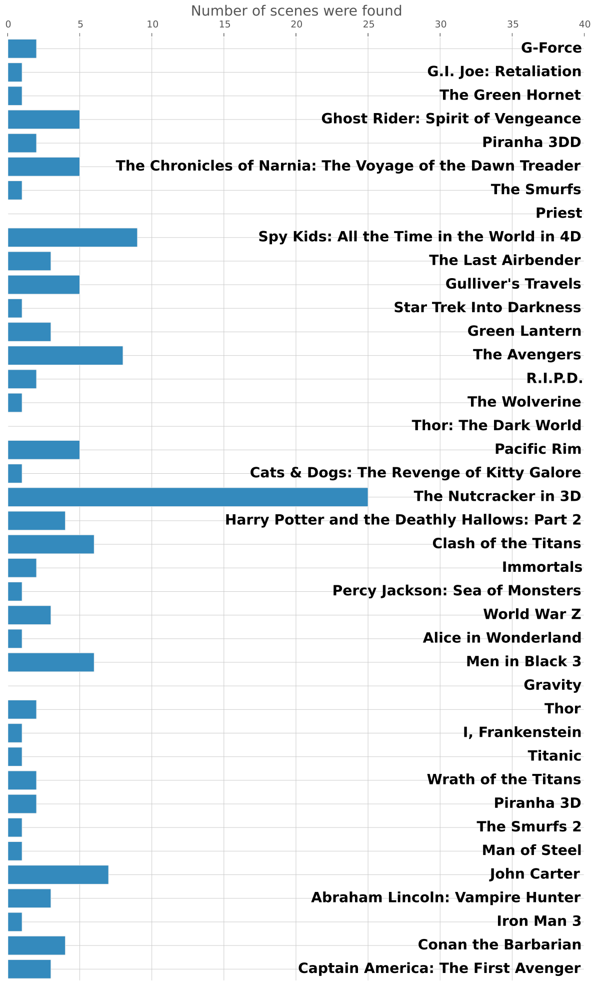


Fig. 9. Number of scenes with major artifacts detected during analysis of the films.

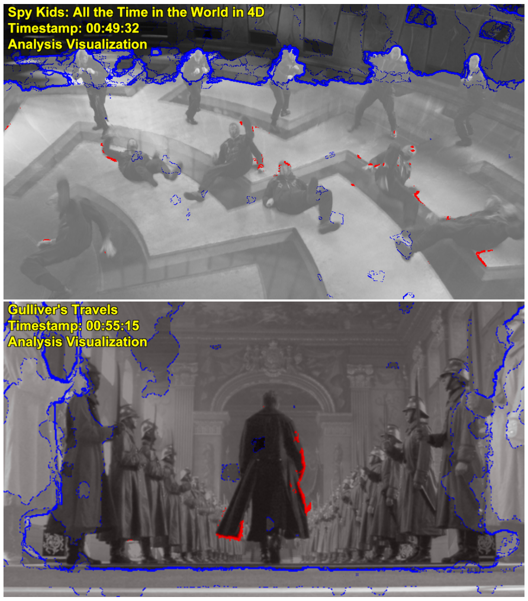


Fig. 10. Example algorithm detection results. In these scenes the foreground objects stick to background. For visibility, the source frame is overlaid with a disparity map, a disparity edge map (blue), and a map of disparity edges that the algorithm found to be absent (red).

validation data set, we manually classified the detected frames as either true positive or false positive. The data set consisted of 21 true-positive scenes and 31 false-positive scenes (see Table 1).

The most common source of false-positive alerts were objects in front of textureless backgrounds (e.g., sky). Since the block-based disparity-estimation method fails to reliably estimate disparity for textureless areas, the proposed pipeline is unable to detect differences in object and background disparity even if the object appears in the depth map. Moreover, the human visual system is quite tolerant of missing depth differences in areas that lack color texture. Thus, the reasonable approach to decreasing the false-positive ratio is to exclude from the algorithm's consideration any objects that appear in front of a uniform background. We did so by weighting the output motion-edge map with the RGB variances of the input frame. Fig. 8 shows a comparison of scores computed using the original and modified algorithms for scenes from the validation data set. The modified algorithm manages to distinguish scenes with artifacts from scenes mistakenly detected by its unmodified version.

After modifying the algorithm we applied it to 39 S3D converted movies. In total, the algorithm detected 125 problem scenes. Fig. 9 depicts the per-film distribution.

The average run time of the proposed pipeline when processing a 960 × 540 frame size is 3.5 seconds on a 2.67 GHz Intel Core i7 processor with 24 GB of RAM.

Film	True positive	False positive
Clash of the Titans	5	5
Conan the Barbarian	4	12
Star Trek Into Darkness	1	7
Harry Potter and the Deathly Hallows: Part 2	4	7
The Avengers	7	0

Table 1. Number of true- and false-positive detections by initial algorithm for the validation data set.

5. CONCLUSION

In this paper we proposed a method for assessing the quality of converted stereoscopic video. The method detects foreground objects that are stuck to the background in the depth map and allows filmmakers to correct the problem.

We demonstrated the performance of our method using a data set containing scenes from five S3D movies. We reduced the false-positive ratio by forcing it to ignore regions of uniform color. Use of fast depth-from-motion extraction enabled us to analyze 39 full-length converted S3D films and detect 125 scenes with significant artifacts that could cause visual discomfort.

6. REFERENCES

[1] D. Akimov, A. Shestov, A. Voronov, and D. Vatolin. Automatic left-right channel swap detection. In *3D Imaging (IC3D), 2012 International Conference on*, pages 1–6. IEEE, 2012.

[2] A. Bokov, D. Vatolin, A. Zachesov, A. Belous, and M. Erofeev. Automatic detection of artifacts in converted s3d video. In *IS&T/SPIE Electronic Imaging*, pages 901112–901112. International Society for Optics and Photonics, 2014.

[3] G. Borgefors. Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344–371, 1986.

[4] G. Egnal, M. Mintz, and R. P. Wildes. A stereo confidence metric using single view imagery with comparison to five alternative approaches. *Image and vision computing*, 22(12):943–957, 2004.

[5] U. Fecker, M. Barkowsky, and A. Kaup. Time-constant histogram matching for colour compensation of multiview video sequences. In *Proc. 26th Picture Coding Symp.(PCS 2007)*, 2007.

[6] P.F. Felzenszwalb and J.D. Schwartz. Hierarchical matching of deformable shapes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[7] K. He, J. Sun, and X. Tang. Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409, 2013.

[8] B. Jähne, H. Scharr, and S. Körkel. Principles of filter design. *Handbook of computer vision and applications*, 2:125–151, 1999.

[9] Y. J. Jung, S. Lee, H. Sohn, H. W. Park, and Y. M. Ro. Visual comfort assessment metric based on salient object motion information in stereoscopic video. *Journal of Electronic Imaging*, 21(1):011008–1, 2012.

[10] J. Li, M. Barkowsky, and P. Le Callet. Visual discomfort of stereoscopic 3d videos: Influence of 3d motion. *Displays*, 35(1):49–57, 2014.

[11] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4. IEEE, 2011.

[12] P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand. Depth image-based rendering with advanced texture synthesis for 3-d video. *Multimedia, IEEE Transactions on*, 13(3):453–465, 2011.

[13] M. Seymour. Art of stereo conversion: 2D to 3D — 2012. `http://www.fxguide.com/featured/art-of-stereo-conversion-2d-to-3d-2012/`, 2012.

[14] K. Simonyan, S. Grishin, D. Vatolin, and D. Popov. Fast video super-resolution via classification. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 349–352. IEEE, 2008.

[15] E. Tolstaya, P. Pohl, and M. Rychagov. Depth propagation for semi-automatic 2d to 3d conversion. In *IS&T/SPIE Electronic Imaging*, pages 939303–939303. International Society for Optics and Photonics, 2015.

[16] A. Voronov, D. Vatolin, D. Sumin, V. Napadovskiy, and A. Borisov. Towards automatic stereo-video quality assessment and detection of color and sharpness mismatch. In *3D Imaging (IC3D), 2012 International Conference on*, pages 1–6. IEEE, 2012.

[17] Y. Wang. Survey of objective video quality measurements. *Computer Science Faculty Publications*, 2006.

[18] Is it Real or Fake 3D? `http://www.http://realorfake3d.com`.

[19] C. Xu, J. Liu, and X. Tang. 2d shape matching by contour flexibility. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):180–186, 2009.

[20] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):974–988, 2009.

[21] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *Computer Vision–ECCV 2014*, pages 815–830. Springer, 2014.

[22] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (wmf). In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2830–2837. IEEE, 2014.