CHANNEL-MISMATCH DETECTION ALGORITHM FOR STEREOSCOPIC VIDEO USING CONVOLUTIONAL NEURAL NETWORK

Sergey Lavrushkin, Dmitriy Vatolin

Lomonosov Moscow State University, Russian Federation

ABSTRACT

Channel mismatch (the result of swapping left and right views) is a 3D-video artifact that can cause major viewer discomfort. This work presents a novel high-accuracy method of channel-mismatch detection. In addition to the features described in our previous work, we introduce a new feature based on a convolutional neural network; it predicts channel-mismatch probability on the basis of the stereoscopic views and corresponding disparity maps. A logistic-regression model trained on the described features makes the final prediction. We tested this model on a set of 900 stereoscopic-video scenes, and it outperformed existing channel-mismatch detection methods that previously served in analyses of full-length stereoscopic movies.

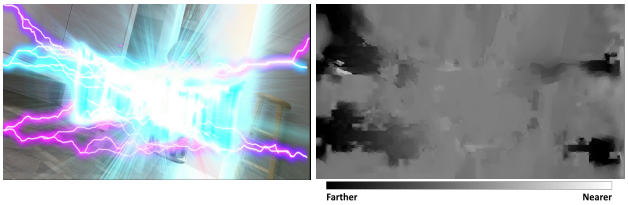
Index Terms — Stereoscopic video, channel mismatch, quality assessment, machine learning, convolutional neural networks

1. INTRODUCTION

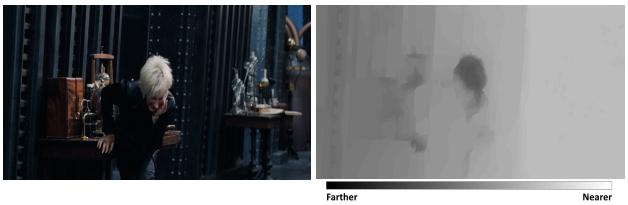
A large number of stereoscopic movies are produced each year. In theaters, however, audiences tend to watch them less often than the 2D versions. Most of these viewers experience some kind of discomfort after watching a stereoscopic movie: fatigue, tension, eye pain, headaches and other symptoms. These are the main reasons why many people lose interest in stereoscopic movies. Discomfort caused by watching 3D movies is primarily due to the quality of the stereoscopic-display equipment, as well as the quality of the stereo content. A number of problems arise during stereoscopic-content creation that do not arise when working with 2D movies. For example, geometric, luminance, brightness, and sharpness distortions often occur when shooting 3D movies [1] because the configurations of the cameras were different and/or some camera components were broken. Although no such problems occur with stereoscopic movies created using CGI or 2D-to-3D conversion, these methods have other flaws – e.g., the cardboard effect, edge-sharpness mismatch, and other artifacts, which can occur during 2D-to-3D conversion [2].

Channel mismatch is one such stereoscopic-video artifact. Although it is less common than the others mentioned above, it can occur in every production method (Figure 1), and if present, it can cause substantial viewer discomfort [3]. This artifact can appear during almost any production stage. In addition to simple view swapping, the channel-mismatch effect also arises when computer graphics and titles are improperly combined with the original stereoscopic-video views and when the conversion from 2D to 3D is incorrect — for example, owing to an inaccurate depth map or low-quality conversion method. Although viewers experience discomfort when watching a stereoscopic scene with channel mismatch, they often fail to understand what caused that discomfort. But this artifact is easier to fix than other artifacts: just swap views, or apply the appropriate operations to the original views if the channel mismatch occurred because of incorrect

(a) Example of completely swapped views (the scene is from *Stalingrad* (2013)).



(b) Example of incorrect computer graphics usage (the scene is from *A Very Harold & Kumar 3D Christmas*).



(c) Example of incorrect 2D-to-3D conversion (the scene is from *Nutcracker in 3D*).

Figure 1: Channel mismatch examples from full-length stereoscopic movies for different stereoscopic video production methods.

2D-to-3D conversion. Automatically detecting it is difficult, however. In this work, we present a novel high-accuracy algorithm for channel-mismatch detection that can be used during stereoscopic-video production to find most occurrences of this artifact.

2. RELATED WORK

Virtually all channel-mismatch detection methods are based on finding discrepancies between certain predefined depth cues and the disparity map obtained by stereo matching. Depending on which depth cues they rely on and which features they employ to numerically assess those cues, we can classify them into the following categories:

1. *Depth-ordering methods.* These algorithms use monocular features to evaluate the disparity map — that is, the disparity map is built using information from only one stereo-

978-1-5386-6125-3/18/$31.00

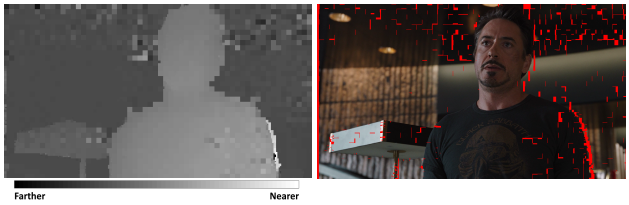
scopic view. Lee et al. [4] employ the simple single-image segmentation method based on saliency maps to find the foreground and background regions. They evaluated this method by applying it to 40 stereo pairs and comparing the results with subjective viewer judgments.

2. *Disparity-distribution analysis methods*. Many stereoscopic-movie scenes have a similar structure (for example, the distance to objects near the bottom of the frame is less than the distance to objects near the top of the frame), so they have similar disparity maps. Disparity-distribution analysis checks whether disparity maps obtained from stereo matching demonstrate this similarity. To identify channel mismatch, Knee [5] calculates the correlation between the analyzed disparity map and a template disparity map, which is a mean of 6,000 disparity maps that display the above-mentioned similarity.
3. *Occlusion-analysis methods*. These methods restore the relative object order using information from occlusions. Bouchard et al. [6] analyze the location of the binocular half-occlusions in the stereo pair relative to the foreground objects. To detect channel mismatch, they calculate the centroids of the binocular half-occlusions for the left and right views and then compare the horizontal coordinates of the centroids. The authors evaluated this method using 52 stereoscopic sequences and compared them to the subjective viewer judgment.
4. *Compositional methods*. This type of approach combines different features to detect channel mismatch and typically yields more-accurate results. Shestov et al. [7] combine two criteria. The first is based on an analysis of edges near binocular half-occlusions and therefore detects the relative order of foreground objects. The second evaluates the disparity distribution by calculating disparity-histogram moments for each view; it is used when the calculated binocular half-occlusions are unsuitable for the first criterion. Bokov et al. [3] also combine several criteria: three are based on simple disparity-distribution heuristics, the fourth uses color near binocular half-occlusions to distinguish foreground and background areas, and the fifth uses motion occlusions to find the corresponding sets of occluded and occluder points and then checks neighboring disparity map regions.

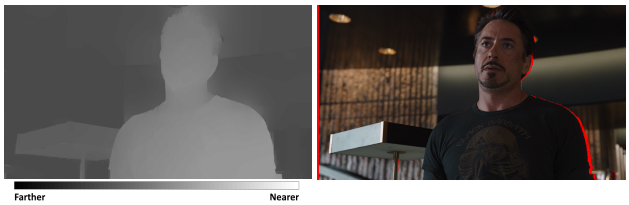
Our proposed method is based on [3], using four of the five criteria as features. It introduces a novel feature computed by a convolutional neural network to considerably improve the resulting classifier performance, as we show.

3. PROPOSED METHOD

Our channel-mismatch detection algorithm is based on five features. Four of them coincide with the perspective, disparity-distribution, binocular half-occlusion, and motion-occlusion criteria from [3]. The fifth feature employs a convolutional neural network to predict channel-mismatch probability in the current frame using the left view and its corresponding disparity map. The algorithm calculates each feature value for each scene frame. The value for a specific scene is the feature mean for all frames in that scene. This step allows us to smooth outliers, which can appear owing to errors in estimating disparity maps, optical-flow fields, and occlusion masks. Such errors are especially common when analyzing highly dynamic portions of scenes. We train a logistic-regression model (without bias) on the computed scene features, then use the result to determine the probability that channel mismatch appears in the scene. Formally, the result of the algorithm for the stereoscopic-video scene is



(a) Unfiltered disparity map and corresponding occlusion map



(b) Filtered disparity map and corresponding occlusion map

Figure 2: Illustration of how disparity-map filtering affects the constructed occlusion maps (highlighted in red).

$$p = \frac{1}{1 + \mathrm{e}^{-z}}, \quad p \in [0, 1],$$

$$z = \sum_{i=1}^{5} \beta_i \overline{x}_i,$$

$$\overline{x}_i = \frac{1}{n_S} \sum_{j \in S} x_{ij},$$

where x_{ij}, $i = \overline{1,5}$, $j = \overline{1,n}$ is the ith feature value for jth frame, n_S is the number of frames in the scene, S is the set of scene frame numbers, and β_i, $i = \overline{1,5}$ are the logistic regression parameters. To train the logistic-regression model, we prepared a training data set consisting of 1,000 stereoscopic scenes containing 30 frames each. The data set included 200 scenes from each of the following stereoscopic movies: *Alice in Wonderland*, *Pirates of the Caribbean: On Stranger Tides*, *Rio 2*, *Thor*, and *Green Lantern*. For half of the scenes (chosen randomly), we artificially swapped the views.

To compute disparity maps and optical flow for these features, we use fast local block matching [8]. Since errors can arise in disparity maps and optical flow, we construct the corresponding confidence maps on the basis of the LRC criterion [9] and block RGB variance (uniform areas are considered to have low confidence). To refine our block-based disparity maps and optical-flow fields, we use a fast global smoothing filter [10] with computed confidence values. This step also improves the quality of our constructed occlusion maps (Figure 2), thereby increasing the accuracy of the occlusion-based features described in [3].

3.1. Scene change detection

We detect the last frame of a scene by comparing the left-view brightness block histograms for the current and subsequent frames. For each frame brightness Y_t^L, we construct the histograms $H_{ij}^{Y_t^L}$ with $n_b = 64$ bins for blocks of size $s = 32$. To compare the current and subsequent frames, we compute the difference between

our obtained brightness block histograms:

$$\mathrm{Dif}_{ij,k}^{H^{Y_t^L},H^{Y_{t+1}^L}} = H_{ij,k}^{Y_t^L} - H_{ij,k}^{Y_{t+1}^L}.$$

In addition, we convolve each row of the computed histogram differences with the kernel $\{0.05, 0.1, 0.2, 0.3, 0.2, 0.1, 0.05\}$ to smooth the result. The algorithm uses mirroring to complete the values needed to filter the left and right boundaries. It accumulates all the computed histogram differences as follows:

$$\mathrm{Dif}^{Y_t^L,Y_{t+1}^L} = \frac{1}{s^2\, h_b\, w_b} \sum_{i=1}^{h_b} \sum_{j=1}^{w_b} \sum_{k=1}^{n_b} |\mathrm{Dif}_{ij,k}^{H^{Y_t^L},H^{Y_{t+1}^L}}|.$$

$\mathrm{Dif}^{Y_t^L,Y_{t+1}^L}$ describes the degree of similarity between frames Y_t^L, Y_{t+1}^L. We assume that when the scene changes, $\mathrm{Dif}^{Y_t^L,Y_{t+1}^L}$ will be large, and that it will greatly differ from the accumulated differences for the previous and subsequent frames. To check this last statement, we calculate the following characteristic:

$$\mathrm{Dis}^{Y_t^L} = 6\,\mathrm{Dif}^{Y_t^L,Y_{t+1}^L} - \sum_{k\in[-3,-1]\cup[1,3]} \mathrm{Dif}^{Y_{t+k}^L,Y_{t+k+1}^L}.$$

If during the analysis of Y_t^L the values of $\mathrm{Dif}^{Y_t^L,Y_{t+1}^L}$ and $\mathrm{Dis}^{Y_t^L}$ exceed specified thresholds, we consider t to be the last scene frame. Figure 3 shows an example of a detected scene change.

3.2. Convolutional-neural-network feature

3.2.1. Neural-network architecture

To predict channel mismatch using a convolutional neural network, we employ a model that takes the left view and its corresponding disparity map as inputs. For the disparity-map inputs we choose refined disparity maps. The neural-network architecture is a stack of convolutional layers with size 3×3 and stride 1 as well as max-pooling layers with size 2×2 and stride 2. After this layer stack we use an average-pooling layer followed by two fully connected layers. For convolutional layers we selected the ReLU activation function; for the first fully connected layer we used the linear function, and for the second we used softmax function. To prevent overfitting we also employ dropout layers [11] with a 0.5 rate before each fully connected layer and batch normalization [12] after each convolutional layer. Figure 4 illustrates this neural-network architecture.

3.2.2. Neural-network training

To train the neural network we prepared a training data set consisting of images and corresponding disparity maps. We took every tenth frame from following stereoscopic movies: *Titanic*, *Alice in Wonderland*, *Pirates of the Caribbean: On Stranger Tides*, *Rio 2*, *The Avengers*, and *The Amazing Spider-Man*. In total, we prepared 113,000 samples.

The input-data size is 224×400. In addition, we normalized the input images and disparity maps. Each disparity map is linearly scaled so the result has a zero mean and unit norm. The target label is randomly selected during training. If the selected label corresponds to a channel mismatch, the sign of the disparity-map values also reverses, thereby modeling the presence of channel mismatch in that frame.

We initialize all neural-network weights with random values from a normal distribution with a mean of 0 and a standard deviation of 0.01 and optimize the neural network using a cross-entropy loss function:

$$L = -\frac{1}{N} \sum_{i=1}^{N} (y_i\, \log(p_{i1}) + (1 - y_i)\, \log(p_{i2})),$$

Table 1: Test results of channel-mismatch detection algorithms.

Method \ Metric	AUC	Accuracy	F-measure
Proposed method	0.9963	0.9784	0.9789
Bokov et al. [3]	0.957	0.8946	0.8928
Shestov et al. [7]	0.901	0.8378	0.8409

where N is the batch size, y_i is the channel mismatch label for sample i, p_{ij} is the neural network output for sample i. To additionally reduce overfitting, we use L_2-regularization with a 0.0005 rate for all network weights along with data augmentation that includes:

- random image scaling by up to 10% of the input size;
- random horizontal image flipping;
- random changes in image brightness and saturation.

We trained our network over 180,000 iterations using stochastic gradient descent with a batch size of 32 and a momentum of 0.9. At the beginning we use a 10^{-2} learning rate, then decrease it to 10^{-3} at 80,000 iterations and to 10^{-4} at 160,000 iterations.

4. EXPERIMENTAL EVALUATION

To test our proposed channel-mismatch detection method and compare it with analogs, we prepared a test set consisting of 900 stereoscopic scenes, each having 30 frames. This test set includes 300 scenes from each of following stereoscopic movies: *Wrath of the Titans*, *Conan the Barbarian*, and *The Three Musketeers*. We randomly chose half of the scenes and artificially swapped the views.

Using this data set we compared our method with [3] and [7], which have been used to analyze full-length stereoscopic movies. Our evaluation used following metrics:

- accuracy;
- area under ROC-curve (AUC);
- F-measure.

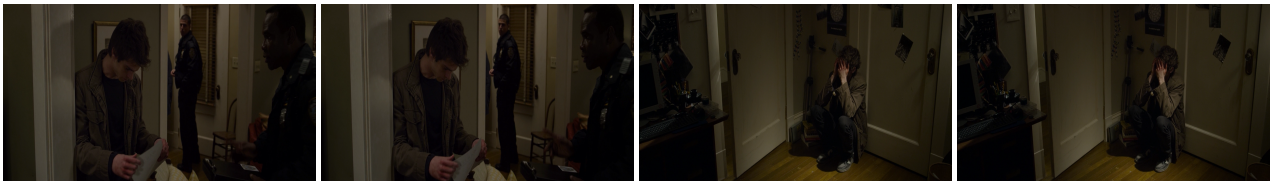
Table 1 shows the resulting metric values, and Figure 5 shows the corresponding ROC curves. Our proposed method outperforms existing channel-mismatch detection methods that were previously used in analyses of full-length stereoscopic movies. With accuracy of 0.9784 it significantly reduces the number of false positive results minimizing amount of manual work required to find all channel mismatches in stereoscopic movie.

5. CONCLUSION

In this paper, we proposed a novel channel-mismatch detection method that uses a convolutional neural network. It achieves an AUC score of 0.9963 when tested on a data set containing 900 S3D video clips, half of which have swapped views. It also achieves an accuracy of 0.9784, which is 8% higher than that of the best alternative channel-mismatch detection methods [3].

6. REFERENCES

[1] Darya Khaustova, Jérôme Fournier, Emmanuel Wyckens, and Olivier Le Meur, "An objective method for 3D quality prediction using visual annoyance and acceptability level," in *SPIE/IS&T Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93910P–93910P.



$\mathrm{Dif}^{Y^L_{t-1},Y^L_t} = 0.0676$ $\quad$ $\mathbf{Dif}^{\mathbf{Y^L_t},\mathbf{Y^L_{t+1}}} = \mathbf{1.3105}$ $\quad$ $\mathrm{Dif}^{Y^L_{t+1},Y^L_{t+2}} = 0.0178$ $\quad$ $\mathrm{Dif}^{Y^L_{t+2},Y^L_{t+3}} = 0.0181$

$\mathrm{Dis}^{Y^L_{t-1}} = -1.123$ $\quad$ $\mathbf{Dis}^{\mathbf{Y^L_t}} = \mathbf{7.6222}$ $\quad$ $\mathrm{Dis}^{Y^L_{t+1}} = -1.3888$ $\quad$ $\mathrm{Dis}^{Y^L_{t+2}} = -1.3436$

Figure 3: Scene-change example along with calculated $\mathrm{Dif}^{Y^L_t,Y^L_{t+1}}$ and $\mathrm{Dis}^{Y^L_t}$ values for each frame. The frames are from *The Amazing Spider-Man*.

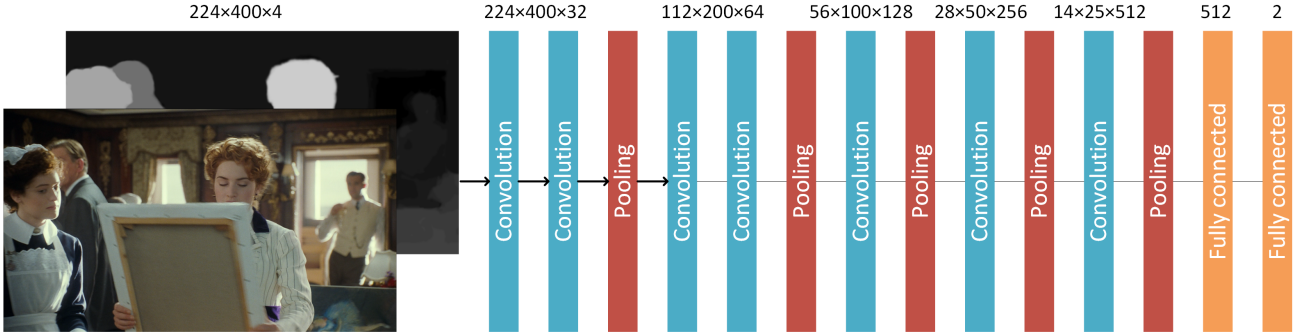


Figure 4: Convolutional-neural-network architecture for channel-mismatch detection based on view and corresponding disparity map. The frame is from *Titanic*.

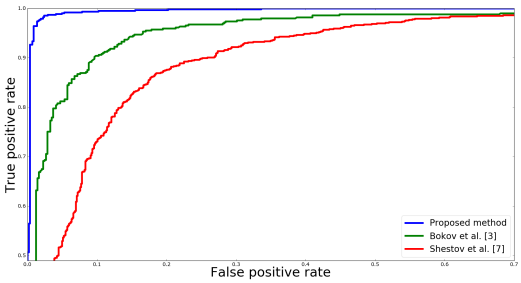


Figure 5: ROC curves of the proposed channel-mismatch detection method as well as the methods described in [3] and [7].

[2] Alexander Bokov, Dmitriy Vatolin, Anton Zachesov, Alexander Belous, and Mikhail Erofeev, "Automatic detection of artifacts in converted S3D video," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2014, pp. 901112–901112.

[3] Alexander Bokov, Sergey Lavrushkin, Mikhail Erofeev, Dmitriy Vatolin, and Alexey Fedorov, "Toward fully automatic channel-mismatch detection and discomfort prediction for S3D video," in *3D Imaging (IC3D), 2016 International Conference on*. IEEE, 2016, pp. 1–7.

[4] Jaeho Lee, Chanho Jung, Changick Kim, and Amir Said, "Content-based pseudoscopic view detection," *Journal of Signal Processing Systems*, vol. 68, no. 2, pp. 261–271, 2012.

[5] Mike Knee, "Getting machines to watch 3D for you," *SMPTE Motion Imaging Journal*, vol. 121, no. 3, pp. 52–58, 2012.

[6] Jonathan Bouchard, Yasin Nazzar, and James J Clark, "Half-occluded regions and detection of pseudoscopy," in *3D Vision (3DV), 2015 International Conference on*. IEEE, 2015, pp. 215–223.

[7] Alexey Shestov, Alexander Voronov, and Dmitriy Vatolin, "Detection of swapped views in stereo image," in *22nd GraphiCon International Conference on Computer Graphics and Vision*, 2012, pp. 23–27.

[8] Karen Simonyan, Sergey Grishin, Dmitriy Vatolin, and Dmitriy Popov, "Fast video super-resolution via classification," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 349–352.

[9] Geoffrey Egnal and Richard P Wildes, "Detecting binocular half-occlusions: Empirical comparisons of five approaches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1127–1133, 2002.

[10] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[12] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 448–456.