NEURAL-NETWORK-BASED DETECTION METHODS FOR COLOR, SHARPNESS, AND GEOMETRY ARTIFACTS IN STEREOSCOPIC AND VR180 VIDEOS

Sergey Lavrushkin, Konstantin Kozhemyakov, Dmitriy Vatolin

Lomonosov Moscow State University, Russian Federation

ABSTRACT

Shooting video in 3D format can introduce stereoscopic artifacts, potentially causing viewers visual discomfort. In this work, we consider three common stereoscopic artifacts: color mismatch, sharpness mismatch, and geometric distortion. This paper introduces two neural-network-based methods for simultaneous color- and sharpness-mismatch estimation, as well as for estimating geometric distortions. To train these networks we prepared large datasets based on frames from full-length stereoscopic movies and compared the results with methods that previously served in analyses of full-length stereoscopic movies. We used our proposed methods to analyze 100 videos in VR180 format—a new format for stereoscopic videos in virtual reality (VR). This work presents overall results for these videos along with several examples of detected problems.

Index Terms— objective quality assessment, color mismatch, sharpness mismatch, geometric distortions, stereoscopic video, vr180, deep learning

1. INTRODUCTION

Stereoscopic videos are now widespread and familiar to almost everyone. When watching them, viewers experience the illusion of a three-dimensional image, achieved by showing two video sequences, the so-called left and right views, one to each eye. Another format that is even more immersive than plain 3D video is 360-degree or spherical video; the greatest immersion comes through the use of dedicated head-mounted displays or VR headsets. When watching 360-degree video, the viewer sees only part of the sphere corresponding to the current viewing direction. Additionally, spherical videos can also be stereoscopic by showing a separate video sequence to each eye: left and right views, similar to the usual stereoscopic video format. VR headsets enable viewers to watch these videos.

Creation of spherical videos usually involves a special rig of multiple cameras that simultaneously film different but overlapping views around the viewing point. Stitching algorithms can later merge these views into one 360-degree video. This approach, however, introduces quality problems in the final video, depending on the quality of the stitching and calibration of the cameras. Moreover, the main action in spherical videos usually occurs on one side of the sphere, but the device receives the entire stream, leading to transmission and storage of redundant information. To address these issues, Google in 2018 introduced a new VR video format, VR180 [1], that achieves stereoscopy by projecting one view on a hemisphere and the other view on the remaining hemisphere. Instead of using a rig of multiple cameras to shoot a video, VR180 only requires two cameras with fisheye lenses, similar to conventional 3D. This approach reduces the cost of the final device considerably. At the same time, it greatly simplifies the shooting technique, since all conventional-camera methods remain applicable (except the result is potentially more spectacular and immersive). All in all, VR180 provides an even more immersive experience than 360-degree video, is cheaper to produce, is easier to shoot, and has no stitching problems.

But as with conventional stereoscopic format, VR180 suffers from stereoscopy-related problems specific to 3D shooting, also known as stereoscopic artifacts. These artifacts can cause viewer discomfort, from fatigue and eyestrain to headaches [2], that if unfixed can decrease the popularity of and demand for stereoscopic VR. In this work, we consider the most common stereoscopic artifacts for 3D shooting: color, sharpness, and geometry mismatches between stereoscopic views. They appear when the cameras are configured differently, a component of one camera is damaged, or both. Stereoscopic videos produced using conversion or computer graphics avoid such problems, so more and more stereoscopic-movie makers are moving away from 3D shooting in favor of 2D-to-3D conversion.

The novel aspects of this work include neural-network-based methods for simultaneous color- and sharpness-mismatch estimation, as well as geometry-mismatch estimation in stereoscopic videos, including VR180; and an objective quality assessment of 100 VR180 videos from YouTube using our proposed methods, revealing their overall low quality.

The rest of the paper is organized as follows. Section 2 presents the state of stereoscopic-video-quality assessment. Section 3 describes our proposed neural-network-based methods for simultaneous color- and sharpness-mismatch detection, and Section 4 describes geometry-mismatch detection. In Section 5 we present experimental results for objective quality assessment of 100 VR180 videos. Finally, Section 6 summarizes the paper.

2. RELATED WORK

In general, evaluating distortions usually requires view matching followed by analysis of the corresponding pixels. Approaches without stereo matching are also possible, however. For color-mismatch detection, Winkler [3] calculated the Pearson correlation of histograms for two views in the HSV color space, but this approach is incapable of localizing color distortions. A similar approach described by Dong et al. [4] introduces a simple global color-mismatch measure. Those authors also evaluated geometric distortions using sparse SIFT [5] matching. To evaluate the vertical shift, this technique analyzes the histogram of vertical vector components; to estimate the zoom, it uses the scaling of matched points from SIFT; and to estimate the rotation, it chooses the rotation angle that minimizes the difference between the rotated left view and the original right view.

The problem of evaluating sharpness mismatch caused by focal-distance variation between cameras was previously considered by Devernay et. al. [6]. The authors assumed the input stereopair is rectified, and they only used horizontal pixel matching to compute a dense disparity map. Their technique employs sum of modified Laplacian to obtain per-pixel estimates of sharpness mismatch and fits them to a model that relates sharpness differences between views to disparity values. Liu et al. [7] proposed a different approach that analyzes width deviations of corresponding edge pairs between stereoscopic views. These approaches, however, avoid directly measuring the sharpness-mismatch magnitude and produce only five possible values in the case of Devernay and only an overall sharpness-mismatch probability in the case of Liu.

Any model-fitting method can enable geometric-distortion estimates on the basis of matching results——for example, RANSAC [8] and its modifications [9, 10]. A slightly different proposal appears in [11, 12], using a neural network to compute correspondence weights before evaluating the model. It can serve as an additional step to better filter the acquired correspondences between stereoscopic views. Rocco et al. [13] propose replacing the stereo-matching step by calculating the full correlation of two feature maps that neural networks have extracted from the views. After completely matching the feature maps, the authors add a regression neural network that predicts an affine-transformation matrix to match the left and right views. A variation of this algorithm involves training a regression network to predict a matrix of completely arbitrary geometric transformations. In further work, Rocco et al. [14] additionally proposed estimating the matrix of projective transformations.

This effort starts from the ideas of the VQMT3D project [15] for objective quality assessment of stereoscopic video. Here, we build on our previous methods from this project, which are based on the standard pipeline of stereoscopic-view matching and artifact evaluation. The color-mismatch-detection method evaluates color differences between corresponding pixels, the sharpness-mismatch-detection method evaluates blur differences in the frequency domain, and the geometry-mismatch-detection method evaluates an affine transformation using RANSAC. We employed these techniques to evaluate more than 100 stereoscopic movies. Unlike previous approaches from the VQMT3D project, the main idea of our proposal in this work is to use neural networks to directly predict the distortion between stereoscopic views.

Fig. 1: A left view with generated color and sharpness distortions and an interpolated right view. The scene is from *Captain America: The First Avenger*.

3. NEURAL-NETWORK-BASED METHOD FOR COLOR- AND SHARPNESS-MISMATCH ESTIMATION

Color and sharpness distortions are extremely common in stereoscopic videos made by shooting native 3D, both conventional stereoscopic and VR180 format since they use the same methods. These distortions are typical for systems consisting of two cameras. The slightest inconsistency in the camera settings, malfunction of one camera, or both can produce these artifacts.

In this work, we consider the problem of simultaneously detecting color and sharpness mismatch between stereoscopic-video views. Both of these artifacts lead to brightness and/or color differences between the views, so when using separate algorithms to detect them, numerous false positives can occur.

3.1. Color- and sharpness-mismatch model and dataset generation

Let L^{gt} and R^{gt} respectively denote the left and right views of a stereopair that lacks color and sharpness mismatches. We consider images in the YUV color space. To model color and sharpness artifacts, we modify undistorted frames as follows:

$$L_c(x,y) = a_c(x,y) \times (G(\sigma^{pos}(x,y)) * L_c^{gt})(x,y) + b_c(x,y),$$

$$R_c(x,y) = (G(\sigma^{neg}(x,y)) * R_c^{gt})(x,y),$$

where L and R are the resulting distorted left and right views; c is a color channel in the YUV color space; $a_c(x,y)$ and $b_c(x,y)$ are the linear and constant coefficients, respectively, for modeling color distortions (generated using Perlin noise for each pixel with coordinates (x,y)); $G(\sigma(x,y))$ is an 11×11 Gaussian kernel; $\sigma(x,y)$ is the standard deviation of the

Gaussian distribution (generated using Perlin noise for each pixel with coordinates (x, y)) that sets the blur strength for $G(\sigma(x, y))$; σ^{pos} and σ^{neg} are the generated standard-deviation matrices containing positive numbers and the absolute value of negative numbers, respectively, from the standard-deviation matrix $\sigma(x, y)$, with zeros in the remaining entries; and $*$ is the convolution operator. We use a linear model of color distortions, which we only added to the left view (enough to obtain a color difference between the stereoscopic views). We also use a Gaussian blur with a strength that varies from pixel to pixel in order to model sharpness distortions, applying it either to the left or right view. To generate the coefficients for the linear model and the matrix of standard deviations for the Gaussian blur, our approach employs Perlin gradient noise, allowing us to set a continuous distortion-strength change that depends on the pixel coordinates. This technique enables us to produce complex artifacts corresponding to uneven heating of the camera sensors in the case of color distortions, as well as different objects being in focus in different views in the case of sharpness distortions.

Additionally, we use a simple constant distortion model that changes the original stereopair similarly for each pixel:

$$L_c = a_c \times (G(\sigma^{pos}) * L_c^{gt}) + b_c,$$

$$R_c = G(\sigma^{neg}) * R_c^{gt},$$

where the parameters a_c, b_c, σ^{pos}, and σ^{neg} are constant for a stereopair and are independent of pixel coordinates. Either σ^{pos} or σ^{neg} is equal to zero, so only one of the two views is blurred. This model corresponds to simpler variations of color and sharpness distortions.

To generate datasets based on the models described above, we gathered 9,488 stereopairs of size 960×540 without color or sharpness distortions (this group included only stereopairs with near-zero distortion estimates from the corresponding VQMT3D methods [15]). The frames were from 16 stereoscopic movies, comprising films produced by 3D shooting as well as by 2D-to-3D conversion and computer graphics. When preparing the dataset, we considered cases with and without color distortions as well as with and without sharpness distortions. Generation of the resulting stereopairs randomly employed one of the two models presented above. Figure 1 shows an example with distortions added.

3.2. Neural-network-based method

To estimate color and sharpness differences between stereoscopic views, we propose a neural-network-based method. The first step is to compute a disparity map using fast local block matching [16]. Since the result can contain errors, we construct a corresponding confidence map based on the LRC criterion [17] and block RGB variance.

Furthermore, the neural network takes as input the original left view; the right view, interpolated according to the computed disparity map; and the corresponding confidence map. The left and right views are both in the YUV color space. These inputs are concatenated before being fed to the network, which simultaneously predicts color-difference maps between stereoscopic views and a blur map corresponding to the standard-deviation matrix by which the Gaussian kernel specified the distortion strength during dataset generation. Figure 2 shows the general scheme of the proposed method.

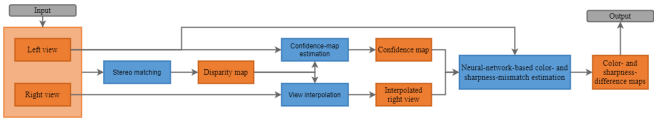


Fig. 2: General scheme of the proposed method for detecting color and sharpness mismatch between stereoscopic views.

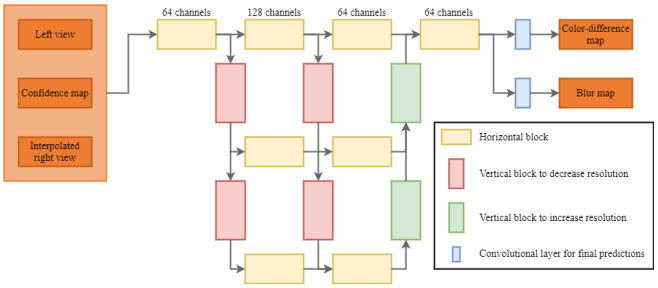


Fig. 3: Architecture of the GridNet network.

The final score for a stereopair's color mismatch, m_c, and sharpness mismatch, m_d, is based on the predicted difference maps:

$$m_c = \frac{\sum_{i=1}^{n} \mathrm{conf}_i \times (\hat{c}_i^Y + \hat{c}_i^U + \hat{c}_i^V)}{3 \sum_{i=1}^{n} \mathrm{conf}_i},$$

$$m_d = \frac{\sum_{i=1}^{n} \mathrm{conf}_i \times \hat{d}_i}{\sum_{i=1}^{n} \mathrm{conf}_i},$$

where $\hat{c}$ is the predicted color-difference map for each YUV color channel, $\hat{d}$ is the predicted blur map, conf is the disparity confidence map that serves as the input confidence map for the neural network, and n is the number of pixels in the image.

We employed the modified GridNet convolutional neural network [18], a variation of the encoder-decoder architecture, to predict color- and sharpness-difference maps. This architecture substantially reduces the network size relative to a standard encoder-decoder, and it also increases the prediction accuracy thanks to a feature-map stream with full spatial resolution. In total, GridNet has three block types: one horizontal for sequentially processing feature maps of one resolution, and two vertical for decreasing and increasing the feature-map resolution as well as for transmitting them to downstream and upstream flows, respectively. We used the same block configuration as in [18]. After the last horizontal block, two parallel convolutional layers predict color- and sharpness-difference maps. Figure 3 illustrates the overall network architecture.

Table 1: Test results for estimating color and sharpness mismatches between stereoscopic views on the prepared Sintel dataset.

Method	PCC	SROCC
Color mismatch		
MAE	0.1254	0.1626
MAE with right view interpolation	0.1338	0.2039
Winkler [3]	-0.4430	-0.4093
VQMT3D [15]	0.8136	0.8760
Proposed method	**0.9696**	**0.9602**
Sharpness mismatch		
MAE	0.1482	0.2635
MAE with right view interpolation	0.2683	0.3505
VQMT3D [15]	0.7686	0.6815
Proposed method	**0.9762**	**0.9078**

3.3. Neural-network training

To train neural networks we used a dataset based on the previously described distortion model. Our basic loss function for predicting both color- and sharpness-difference maps was the sum of squared differences between the predicted and ground-truth values, weighted by the disparity-map confidence:

$$L_c(\hat{c}, c) = \frac{\sum_{i=1}^{n} \mathrm{conf}_i \times ((\hat{c}_i^Y - c_i^Y)^2 + (\hat{c}_i^U - c_i^U)^2 + (\hat{c}_i^V - c_i^V)^2)}{3\sum_{i=1}^{n} \mathrm{conf}_i},$$

$$L_d(\hat{d}, d) = \frac{\sum_{i=1}^{n} \mathrm{conf}_i \times (\hat{d}_i - d_i)^2}{\sum_{i=1}^{n} \mathrm{conf}_i},$$

where $\hat{c}$ and c are the predicted and ground-truth color-difference maps, respectively, for each YUV color channel; $\hat{d}$ and d are the predicted and ground-truth blur maps, respectively; conf is the input disparity confidence map for the neural network; and n is the number of pixels in the image. Additionally, we used L_2-regularization with the regularization parameter 10^{-2}. The final loss function is the following:

$$L(\hat{c}, c, \hat{d}, d, \theta) = L_c(\hat{c}, c) + L_d(\hat{d}, d) + L_2(\theta),$$

where θ is the vector of neural-network weights.

We used the Xavier initialization method [19] to initialize the convolutional-layer weights before training and chose Adam [20] as an optimization method. The neural-network training took place over 100 epochs. We set the learning rate to 10^{-4}, decreasing it by a factor of 10 every 40 epochs. The batch size was 8, and the resolution of the training examples was 256×256. Our approach randomly cut out image sections of this size from the full images during training. Also, to further augment the data, we used random horizontal or vertical image reflection as well as normally distributed noise with a standard deviation of 0.02 and zero mean.

3.4. Model evaluation

To test the proposed method, we prepared a dataset based on Sintel [21], which contains 23 stereoscopic-video sequences with a resolution of 1024×436, as well as ground-truth optical-flow and disparity values for each frame. The original video sequences lack any color or sharpness distortions, as they were obtained using computer graphics. To prepare a test dataset, we added to each sequence artificial distortions based on the aforementioned general distortion model. Every sequence appeared three times in the test set, each with added distortions of different types and/or strengths. Using this prepared dataset, we evaluated our proposed method, several simple methods, as well as corresponding methods from the VQMT3D project [15]. Table 1 presents the results. Our proposed method is superior to previous methods from the VQMT3D project both in Pearson correlation and Spearman correlation.

4. NEURAL-NETWORK-BASED METHOD FOR GEOMETRY-MISMATCH ESTIMATION

Geometric distortions between stereoscopic views often occur in 3D shooting. The most common types include vertical shift, rotation, and scaling. Geometry mismatch can occur because of incorrect camera calibration as well as slightly inconsistent tilting or vertical positioning of the cameras. Stereoscopic-video production often overlooks quality control, causing such artifacts to appear in movies and in videos on popular sharing platforms. When watching them, viewers may experience discomfort. We therefore propose a neural-network-based algorithm for detecting the abovementioned geometry distortions.

4.1. Geometry-mismatch model and dataset preparation

Application of an affine transformation to one view is sufficient to model the geometric distortions described above. Let $p = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ and $p' = \begin{bmatrix} x' & y' & 1 \end{bmatrix}^T$ denote the homogeneous coordinates of two points before and after the transformation, respectively. We then model the geometry mismatch between stereoscopic views through the following affine transformation:

$$p' = A \times p,$$

$$A = \begin{bmatrix} (1+k)\cos(\alpha) & -(1+k)\sin(\alpha) & 0 \\ (1+k)\sin(\alpha) & (1+k)\cos(\alpha) & t \\ 0 & 0 & 1 \end{bmatrix},$$

where α is the rotation angle, k is the scaling coefficient, and t is the vertical shift. We therefore consider the problem of estimating the parameters $\theta = \begin{bmatrix} \alpha & k & t \end{bmatrix}$.

On the basis of this model, we prepared a dataset using frames from 39 3D movies. Among them there were films produced by 3D shooting and films produced by 2D-to-3D conversion. Since geometric distortions between views in a stereoscopic movie have no ground-truth values, we evaluated them using the VQMT3D project's geometric-distortion-detection

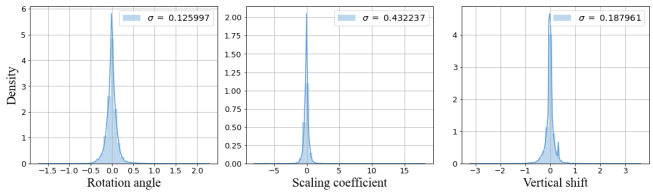


Fig. 4: Distributions of and computed standard deviations for the geometric distortions for thirty-nine 3D movies.

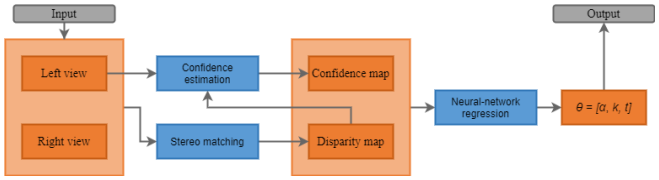


Fig. 5: General scheme of proposed method for detecting geometry mismatch between stereoscopic views.

algorithm [15]. This method also evaluates geometric distortions in terms of rotation, scaling, and vertical shift of one view relative to the other. Figure 4 shows distributions of the corresponding distortions for all analyzed films. We computed the standard deviation for each distortion type and gathered stereopairs for which all three of these parameters had absolute values less than $\frac{\sigma}{10}$. We selected frames with a certain step to avoid duplicating scenes in the dataset. In total, we extracted 22800 stereo pairs. We distorted each stereopair by applying the affine transformation to one view, using random parameters generated from a normal distribution with zero mean and a standard deviation five times larger than that of the estimated distribution (Figure 4). This choice for the standard deviation expanded the coverage of possible distortion values in the films.

4.2. Neural-network-based method

Our proposed method for estimating geometry mismatch consists of two main steps: stereo matching with confidence estimation and neural-network prediction of geometric distortions (Figure 5). We use the same disparity- and confidence-map-estimation approach as we did in our method for simultaneously detecting color and sharpness mismatch.

To estimate the geometry-mismatch parameters we employ a neural-network architecture, similar to ResNet-18 [22]. We put four residual blocks between every increase in channel size and do not use batch normalization [23]. Besides, the final layer produces a vector of length three, containing the parameters of the predicted geometric distortions. The neural network's inputs are the disparity map and corresponding confidence map. The spatial dimensions of the input tensor are arbitrary—the last block produces a fixed-size vector using a global-average-pooling layer at the end.

4.3. Neural-network training

We used the following loss function to train the neural network:

$$L(\theta, \theta_{gt}, \theta_b) = L_{SE}(\theta, \theta_{gt}) + L_{Grid}(\theta, \theta_{gt}) + L_{Siam}(\theta, \theta_b),$$

where θ is the neural network's prediciton based on the disparity and confidence maps for the left view, θ_{gt} is the vector of ground-truth distortion parameters, and θ_b is the neural network's prediction based on the disparity and confidence maps for the right view. The proposed loss function includes three main terms.

The first term, L_{SE}, is the squared difference between the predicted and ground-truth distortion parameters, with empirically chosen weights for each distortion type:

$$L_{SE}(\theta, \theta_{gt}) = w^{\alpha}_{SE}\,(\alpha - \alpha_{gt})^2 + w^{k}_{SE}\,(k - k_{gt})^2 + w^{t}_{SE}\,(t - t_{gt})^2\,,$$

where $w^{\alpha}_{SE} = 1$, $w^{k}_{SE} = 10^4$, and $w^{t}_{SE} = 10^4$.

The second term, L_{Grid}, is the loss between two grids transformed using the predicted and ground-truth affine transformations. Let $G \in R^{H \times W \times 3}$ denote homogeneous coordinates of points on the plane. We chose $H = W = 21$ and selected points from the square $[-1; 1] \times [-1; 1]$ with a step $h = 0.1$. To calculate this term, we decomposed the parameter vector into three separate vectors: $\theta^{\alpha} = \begin{bmatrix}\alpha & 0 & 0\end{bmatrix}$, $\theta^{k} = \begin{bmatrix}0 & k & 0\end{bmatrix}$, and $\theta^{t} = \begin{bmatrix}0 & 0 & t\end{bmatrix}$. Next, we sequentially applied each affine transformation T to the original grid G using the predicted parameters, as well as the ground-truth parameters, to generate new grids corresponding to each geometric distortion:

$$\begin{aligned} G^{\alpha} &= T(G, \theta^{\alpha}),\ G^{\alpha}_{gt} = T(G, \theta^{\alpha}_{gt}),\\ G^{k} &= T(G^{\alpha}, \theta^{k}),\ G^{k}_{gt} = T(G^{\alpha}_{gt}, \theta^{k}_{gt}),\\ G^{t} &= T(G^{k}, \theta^{t}),\ G^{t}_{gt} = T(G^{k}_{gt}, \theta^{t}_{gt}). \end{aligned}$$

The mean squared error between corresponding grids and the weighted sum of these errors form the second loss term:

$$\begin{aligned} L_{Grid} = w^{\alpha}_{Grid} MSE(G^{\alpha}, G^{\alpha}_{gt}) + w^{k}_{Grid} MSE(G^{k}, G^{k}_{gt})\\ + w^{t}_{Grid} MSE(G^{t}, G^{t}_{gt}), \end{aligned}$$

where $w^{\alpha}_{Grid} = 5,000$, $w^{k}_{Grid} = 3,000$, and $w^{t}_{Grid} = 3,000$.

Finally, the last term measures the consistency between the neural network's predictions of the disparity and confidence maps for the left and right views. If the predictions are correct, the network should yield the same distortion-parameter values for the left view if we feed it disparity and confidence maps for the right view, except with opposite sign. In other words, $\theta = -1 \cdot \theta_b$. So the third loss term penalizes the difference between the predicted vectors:

$$L_{Siam}(\theta, \theta_b) = L_{SE}(\theta, -1 \cdot \theta_b).$$

To calculate this term, we additionally predict the distortion parameters θ_b using right-view data during training. But during

Table 2: Absolute error for each geometric distortion.

Method	Rotation angle	Scaling coefficient	Vertical shift
No model	0.63406	0.6507	0.57497
Yi et al. [11]	0.05115	0.10810	0.19109
Rocco et al. [13]	0.43735	1.23582	0.82534
VQMT3D [15]	0.01158	0.02622	0.02004
Proposed method	**0.01029**	**0.02071**	**0.00947**

inference, disparity and confidence maps for the left view are sufficient to evaluate geometric distortions in the stereopair.

We used the He initialization method [24] to initialize weights and trained the neural network using the Adam optimizer [20]. Our approach employed standard parameters, except for the learning-rate coefficient, which was 10^{-4}. We trained the model over 120 epochs.

4.4. Model evaluation

We compared our method with the neural-network-based algorithms Rocco et al. [13] and Yi et al. [11], as well as with the previous version from the VQMT3D project [15]. We trained both neural-network-based analogs on the training part of our dataset. Testing for all methods employed the testing part of our dataset. Table 2 presents the results. It contains mean values of the absolute error for each of the three geometric-transformation parameters. "No model" predicts zero for each geometric distortion.

5. ANALYSIS OF VR180 VIDEOS

To create the VR180 dataset, we simulated search queries on YouTube with the filter set to VR180 video. To make the selection unbiased, we retrieved video IDs from the first one or two pages using a total of 36 nonempty search queries: 26 English letters plus 10 digits, one by one, leading to about 200 links to YouTube VR180-video pages. We excluded some of the videos (for being unavailable, poor in resolution, or nonstereoscopic) and selected older and more-popular videos for the dataset. The result was 100 VR180 videos in 4K resolution.

All VR180 video frames initially appear in an equirectangular projection. The preprocessing step remaps them into a cubemap projection. The algorithm chooses the front edge of the resulting cubemap projection because it contains more information than the other blocks. Besides, after this transformation, the views will not contain artifacts that appeared on the equirectangular projection closer to the image poles. Furthermore, we processed the views in the same way we processed the stereoscopic frames.

We processed all 100 samples and found that VR180 videos contain many artifacts. Figure 6 presents several example artifacts. The overall charts show average metric values relative to VR180-video release date (Figure 7) and view count (Figure 8) on YouTube. According to the results, VR180's technical quality is terrible. Distortions even increase with time, and they are almost uncorrelated with video popularity, meaning people continue to watch poor-quality VR180 content and no one corrects the problems. This situation indicates a need to develop correction tools to improve VR180 stereoscopic quality.

6. CONCLUSION

In this paper we proposed two novel neural-network-based methods for simultaneous color- and sharpness-mismatch estimation, as well as for geometric-distortion estimation. Our methods exhibited a significant quality increase relative to previous versions from the VQMT3D project. We used these methods to analyze 100 VR180 videos gathered from YouTube, revealing the overall bad quality of current VR180 videos, which tend to exhibit at least one of the problems we described.

We plan to continue improving the objective quality metrics from the VQMT3D project using neural-network-based approaches in addition to developing correction algorithms for the artifacts we describe herein. Improvements in this area can simplify the detection and correction of all stereoscopic artifacts, increasing the final product's quality and potentially leading to another wave of interest in S3D.

We plan to soon publish a report describing a more thorough analysis of VR180 videos. This effort will expand our VR180 dataset to 1,000 videos and will include other stereoscopic artifacts specific to 3D shooting. The report will be available on the main VQMT3D project page: `http://videoprocessing.ml/stereo_quality/`.

6.1. Acknowledgments

This work was supported by the START program of the State Fund for Support of Small Enterprises in the Scientific-Technical Fields under the project "Development of a system for automatic objective quality assessment and correction of stereoscopic video and video in VR180 format."

This work was partially supported by the Russian Foundation for Basic Research under Grant 19-01-00785 a.

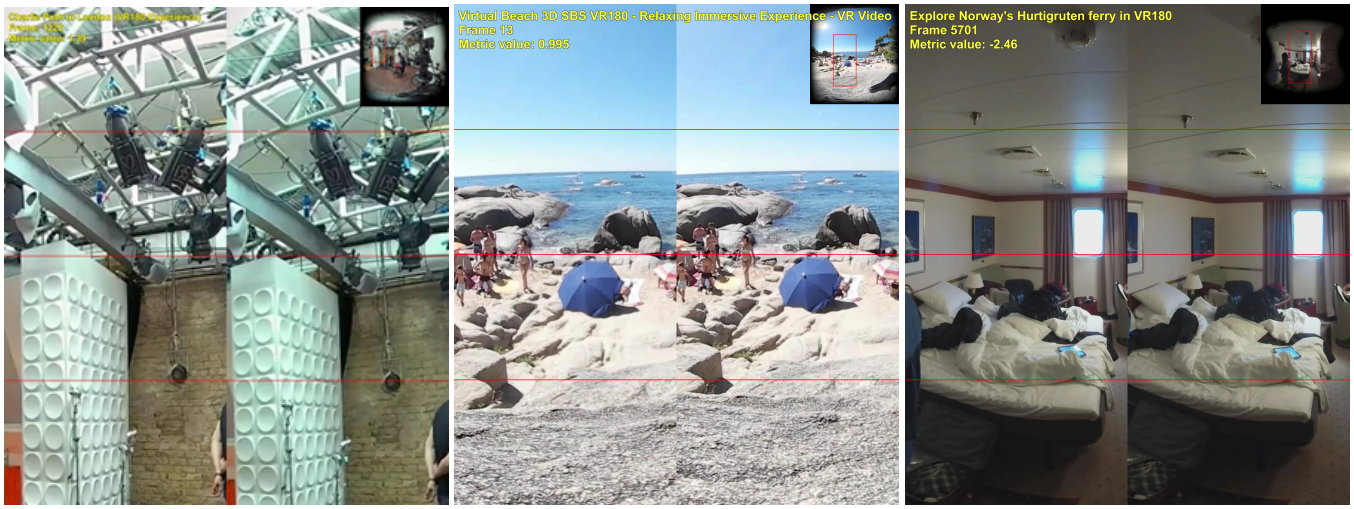
We performed all neural-network training using the IBM Polus high-performance cluster of the Faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University.

7. REFERENCES

[1] "VR180." available online: `https://arvr.google.com/vr180/`.

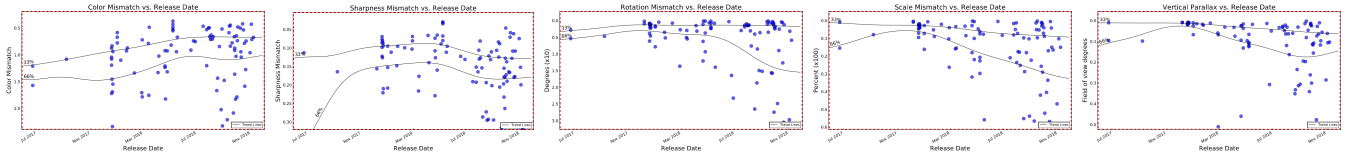
[2] A. Antsiferova and D. Vatolin, "The influence of 3D video artifacts on discomfort of 302 viewers," in *2017 International Conference on 3D Immersion (IC3D)*, pp. 1–8, IEEE, 2017.

(a) Color mismatch (b) Sharpness mismatch



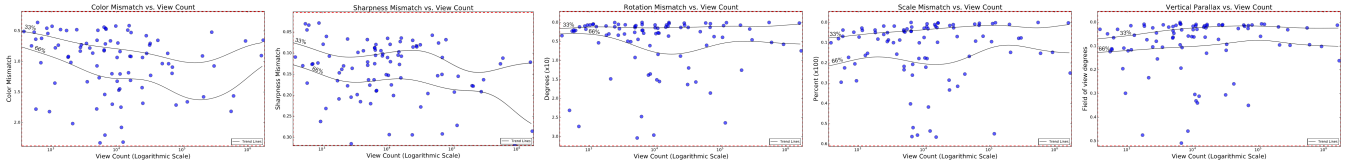
(c) Rotation mismatch (d) Scale mismatch (e) Vertical shift

Fig. 6: Examples of artifacts in analyzed videos.



(a) Color mismatch (b) Sharpness mismatch (c) Rotation mismatch (d) Scale mismatch (e) Vertical shift

Fig. 7: Average metric values relative to video release date. Videos that are higher on a chart have lower estimated distortion values.



(a) Color mismatch (b) Sharpness mismatch (c) Rotation mismatch (d) Scale mismatch (e) Vertical shift

Fig. 8: Average metric values relative to video view count on YouTube. Videos that are higher on a chart have lower estimated distortion values.

[3] S. Winkler, "Efficient measurement of stereoscopic 3D video content issues," in *Image Quality and System Performance XI*, vol. 9016, p. 90160Q, International Society for Optics and Photonics, 2014.

[4] Q. Dong, T. Zhou, Z. Guo, and J. Xiao, "A stereo camera distortion detecting method for 3DTV video quality assessment," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–4, IEEE, 2013.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[6] F. Devernay and S. Pujades, "Focus mismatch detection in stereoscopic content," in *Stereoscopic Displays and Applications XXIII*, vol. 8288, p. 82880E, International Society for Optics and Photonics, 2012.

[7] M. Liu and K. Müller, "Automatic analysis of sharpness mismatch between stereoscopic views for stereo 3D videos," in *2014 International Conference on 3D Imaging (IC3D)*, pp. 1–6, IEEE, 2014.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC-differentiable RANSAC for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692, 2017.

[10] E. Brachmann and C. Rother, "Neural-guided RANSAC: learning where to sample model hypotheses," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4322–4331, 2019.

[11] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2666–2674, 2018.

[12] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "Attentive context normalization for robust permutation-equivariant learning," *arXiv preprint arXiv:1907.02545*, 2019.

[13] I. Rocco, R. Arandjelović, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6148–6157, 2017.

[14] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6917–6925, 2018.

[15] D. Vatolin, A. Bokov, M. Erofeev, and V. Napadovsky, "Trends in S3D-movie quality evaluated on 105 films using 10 metrics," *Electronic Imaging*, vol. 2016, no. 5, pp. 1–10, 2016.

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

[17] G. Egnal and R. 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.

[18] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," in *28th British Machine Vision Conference*, 2017.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*, pp. 611–625, Springer, 2012.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[23] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.