STEREOSCOPIC DATASET FROM A VIDEO GAME: DETECTING CONVERGED AXES AND PERSPECTIVE DISTORTIONS IN S3D VIDEOS

Kirill Malyshev, Sergey Lavrushkin, Dmitriy Vatolin

Lomonosov Moscow State University, Russian Federation

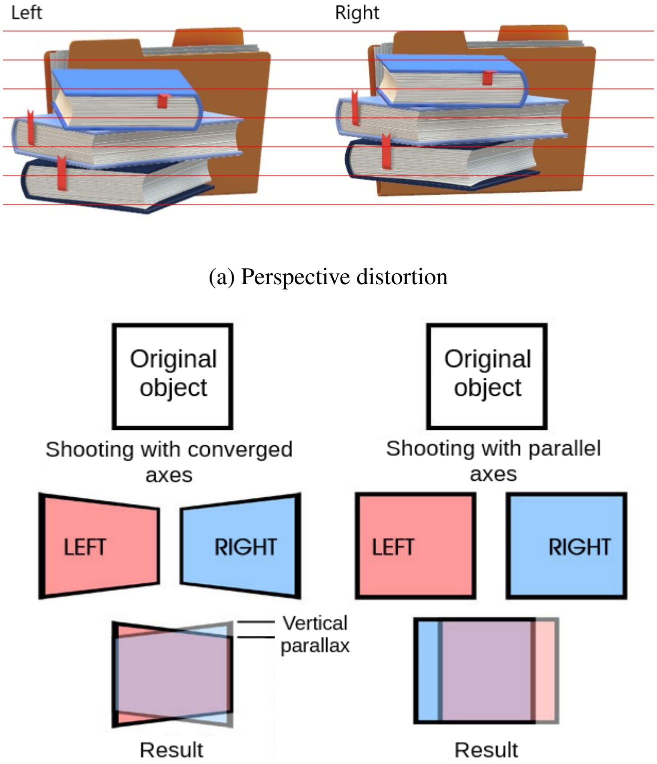
ABSTRACT

This paper presents a method for generating stereoscopic or multi-angle video frames using a computer game (Grand Theft Auto V). We developed a mod that captures synthetic frames allows us to create geometric distortions like those that occur in a real video. These distortions are the main cause of viewer discomfort when watching 3D movies. Datasets generated in this way can aid in solving problems related to machine-learning-based assessment of stereoscopic- or multi-angle-video quality. We trained a convolutional neural network to evaluate perspective distortions and converged camera axes in stereoscopic video, then tested it on real 3D movies. The neural network discovered multiple examples of these distortions.

Index Terms— Perspective distortion, converged axes, geometric distortions, stereoscopic video, deep learning.

1. INTRODUCTION

Recent years have seen a tremendous leap in the use of deep learning to analyze and process stereoscopic video. For example, Zbontar and LeCun [1] used this approach to solve the problem of stereo matching. Li et al. [2] trained a convolutional neural network that predicts the right-image segmentation on the basis of the left-image segmentation. Also, neural networks can assess the quality of stereoscopic images [3]. This technique, however, has disadvantages. For instance, training the model requires a set of data labeled for a specific task. Researchers must either create such a dataset on their own or use an existing one, if available. Until recently, there were no large sets of stereoscopic sequences. Kits such as KITTI stereo 2012 (194 training-image pairs and 195 test-image pairs) [4], KITTI stereo 2015 (800 training scenes and 800 test scenes) [5] and Middlebury Stereo (33 short sequences) [6] are too small for many tasks. But in recent years, datasets containing over 100,000 stereoscopic pairs have emerged, including IRS (103,316 synthetic samples) [7] and DrivingStereo (182,188 real samples) [8]. Unfortunately, they are unsuitable for some tasks: stereoscopic-video analysis and processing, for example, require special frames that such datasets lack. For instance, when training a model to detect converged camera axes, stereoscopic pairs must have an angle between the views. These tasks are important because stereoscopic-frame distortion leads to viewer discomfort, and many types of distortions can afflict stereoscopic video [9]. In such cases, creating even small datasets with real sequences can be too inefficient.



(a) Perspective distortion

(b) Shooting with converged camera axes

Fig. 1: Schematic representation of distortions.

One solution to this problem is to automatically generate synthetic stereoscopic-video sequences. A convenient environment is the Grand Theft Auto V (GTA V) computer game, which allows the use of mods to capture frames of the surrounding space and the associated data. Because of the game's realism, these frames are high definition and lifelike. The single-angle frame sequences Playing for Benchmarks [10], PreSIL [11] and GTA-3D [12] have already employed this

2020 International Conference on 3D Immersion (IC3D) | 978-1-6654-0336-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/IC3D51119.2020.9376375

978-0-7381-1220-6/20/$31.00 ©2020 IEEE

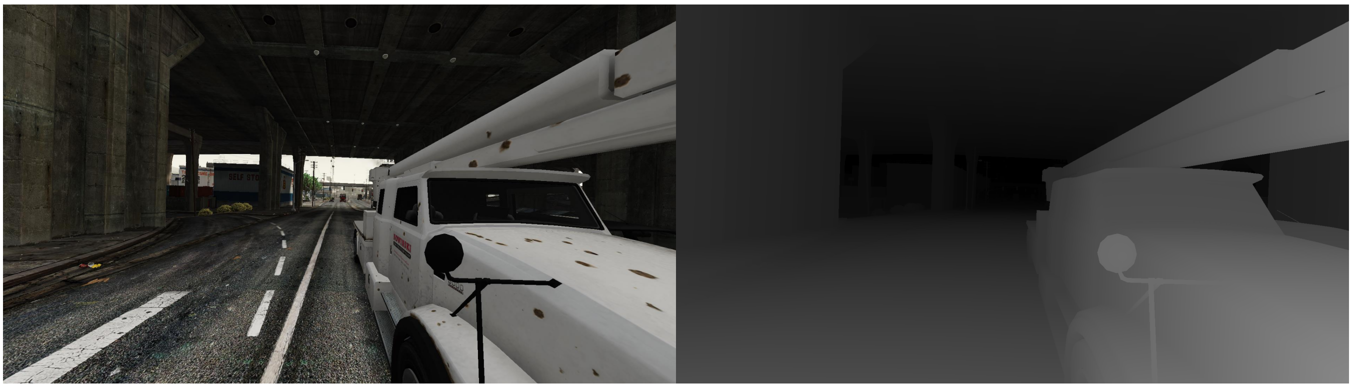


Fig. 2: A captured image and its depth map.

approach.

In this article, we propose a method that uses GTA V to automatically generate a synthetic set of frame sequences with the following variable characteristics:

- Time step between frames
- Number of camera angles
- Camera direction (can be random for each frame)
- Horizontal/vertical camera rotation for some angles (can be random for each frame)
- Horizontal/vertical/depth camera shift for some angles (can be random for each frame)
- Weather (can be random for each frame)
- Time of day (can be random for each frame)

This variability is achievable through a convenient mod-creation interface that controls the weather, time of day, camera position and speed of object movement. The interface can also freeze the scene completely enable shooting from multiple angles. Employing a small time step with a fixed camera direction, fixed weather conditions and a fixed time of day yields a video sequence, whereas a large time step with random camera directions, random weather conditions and a random time of day yields a set of dissimilar frames. This method can produce frames with 1,920×1,080 resolution or lower. Each image is annotated with the following:

- Time of day and weather conditions
- Depth map (Figure 2)
- Horizontal and vertical camera-rotation angle (in degrees)
- Horizontal/vertical/depth camera shift (in centimeters)

As an experiment, we used this method to generate data for solving problems related to evaluating perspective distortion (Figure 1a) and converged camera axes (Figure 1b). Perspective distortion occurs when the left and right cameras have a vertical offset, creating incorrect occlusions. When shooting close-ups with converged axes, vertical parallax and unpleasant distortions of object shapes can occur. These effects can cause viewers to suffer headaches or nausea.

We modified the resulting sequences by adding noise and blur to make the frames more realistic, then divided them into training and validation samples. Next, we trained a convolutional neural network on the synthetic sequences with appropriate distortions. It showed high accuracy when tested on a validation set. We also tested the model on real stereoscopic films, manually selecting frames from among those in which the model exhibited a high distortion score.

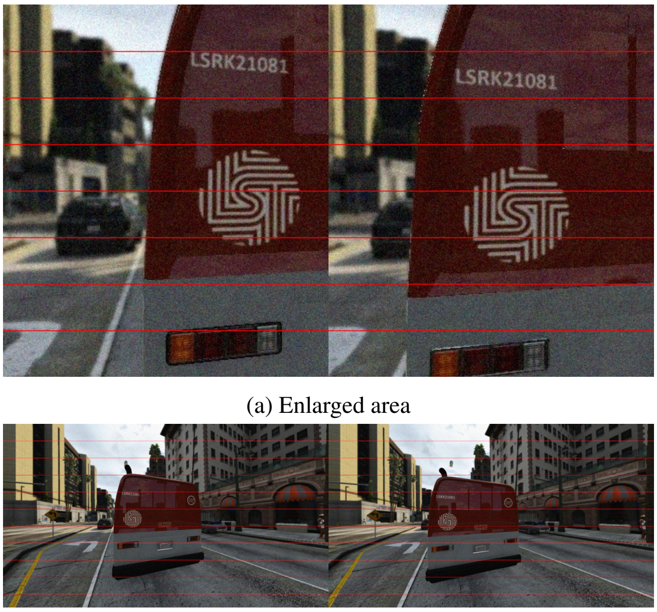
2. RELATED WORK

2.1. Synthetic datasets

Because of the need to obtain large datasets, synthetic data generation is often the method of choice. It greatly reduces ground-truth annotation. Naturally, the problem is how to generate sufficiently realistic annotated ground-truth frames. Synthetic-data-set creators employ different techniques to make the generated data realistic.

A particularly famous artificial-video set is MPI-Sintel [13]. It contains multiple sequences and serves as a benchmark for evaluating algorithms that construct optical-flow and disparity maps. To obtain this dataset, its creators used the Sintel animation engine, which is open source. Wrenninge et al. [14] went the other way: they designed a method for procedurally generating street landscapes, forming the basis of the Synscapes dataset. The advantage of this method is its numerous independently adjustable parameters, but it can only produce static images, not video sequences.

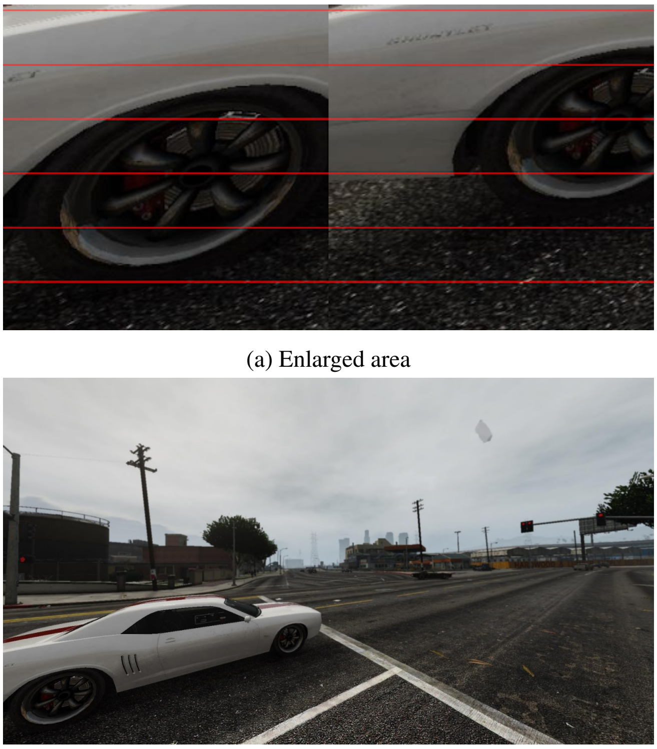
The engines that drive modern video games can create highly realistic frames, making them of great interest. For



(a) Enlarged area

(b) Left and right images.

Fig. 3: Processed frame with perspective distortion. The right camera is 8 cm higher than the left.



(a) Enlarged area

(b) Left image

Fig. 4: Processed frame from cameras with axes converging at a 5-degree angle. Vertical parallax is noticeable in the frame's lower-left corner.

example, Qiu and Yuille [15] developed the UnrealCV plugin to produce annotated datasets using Unreal Engine 4. Shafaei et al. [16] generated a VG dataset containing over 60,000 frames from the game Half-Life 2. Particularly relevant to researchers is GTA V, which has a large, open world and realistic graphics. Unfortunately the source code is closed, but modifications are possible. Johnson-Roberson et al. [17] developed the GTAVisionExport plugin, which can create from the game a depth map and pixel-wise object stencil buffer of frames. Of special note, this plugin aided in creating the Precise Synthetic Image and LiDAR (PreSIL) [11] dataset for autonomous-vehicle perception, as well as a set of omnidirectional images with semantic segmentation and a depth map (OmniScape) [18]. Richter et al. produced two sets of video sequences: Playing for Data [19], which contains semantic maps, and Playing for Benchmarks [10], which is annotated with ground-truth data for both low- and high-level vision tasks, including optical flow, semantic-instance segmentation, object detection and tracking, object-level 3D-scene layout, and visual odometry. GTA V was also the source for the GTA-3D [12] dataset containing 2D frames, 3D point-cloud data, and 3D vehicle-bounding-box labels.

2.2. Distortion estimation

Because geometric distortions of stereoscopic-video angles cause viewer discomfort, detecting them and evaluating their magnitude is important. Until now, machine-learning methods have seldom been a tool for solving this problem. Stereo rectification is a common approach to correcting geometric distortion: it involves projection of the left and right images onto a common plane parallel to the line connecting the optical centers. It then searches for the corresponding point pairs between the two views. Most such methods focus on estimating the fundamental matrix that easily corrects a stereo pair. Evaluating this matrix requires numerous correspondences between the two angles. For example, Zilli et al. [20] developed a stereo-rectification method by evaluating the fundamental matrix. The method shows relatively high accuracy, but it only works for small angles. Georgiev et al. [21] proposed an approach that imposes restrictions on the camera positions while still remaining practical. It therefore avoided the need to evaluate the fundamental matrix and thus accelerated the algorithm. Napieralski and Kowalczyk [22] suggested the sliding-window method to estimate the vertical shift. This technique avoids difficulties that arise when the image has many repeating patterns, which can cause incorrect matching of the fragments.

3. PROPOSED METHOD

3.1. Mod description

We developed our own mod to extract data from GTA V. As with the PreSIL and OmniScape datasets, our approach used

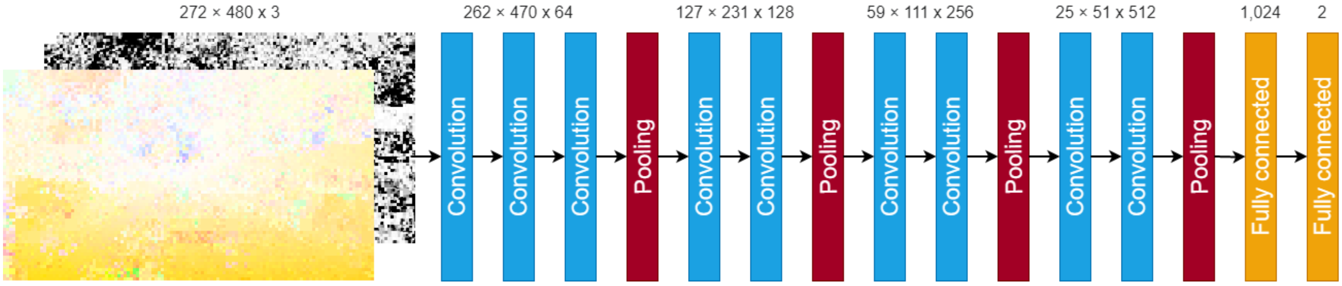


Fig. 5: Schematic of trained convolutional neural network.

the GTAVisionExport [17] plugin to generate depth maps. The virtual camera is fixed on an invisible machine, and it takes pictures at regular intervals from a given number of angles. To automate the process, we employed the VAutodrive mod, which can control a car via autopilot, choosing a random direction at each turn.

To create datasets for different tasks, we included in the mod a set of parameters for adjusting the camera, listed in the Section 1. As mentioned, our method allows generation of stereoscopic sequences with various geometric distortions. It therefore simplifies creation of training datasets and use of machine learning to evaluate stereoscopic-video distortion. The proposed method for creating synthetic datasets can generate various distortions, including perspective distortion, scale mismatch, rotation mismatch and converged camera axes. If one camera is closer than the other, the result will be a stereo pair with scale mismatch. Tilting one camera slightly to the left or right generates frames with rotation mismatch. Or the camera axes can converge. Combining various distortion types can yield a dataset for a more general method. By adjusting the number of angles, we can create datasets of not only stereoscopic frames but also polygonal frames.

Since the mod API can set the weather conditions and time of day, we included this feature in our mod to generate more- varied frames. The following are the possible weather types: extra sunny, clear, clouds, smog, foggy, overcast, rain, thunderstorm, clear, snow, light snow. We excluded some of the available weather types to avoid excessive artificiality.

The time-step setting allows us to adjust the video's frame rate. A large step yields frames with little relation to each other. A small step is necessary to generate video; stopping movement in the game to allow shooting from multiple angles, however, makes video recording slower than real time. Setting a random shooting direction also increases the dissimilarity of the captured frames. In this case, the camera turns a randomly selected angle of 0 to 360 degrees for each frame. In addition, randomly changing weather conditions and time of day generates highly disparate frames, allowing us to train the neural network without shifting the domain toward frames of the same type.

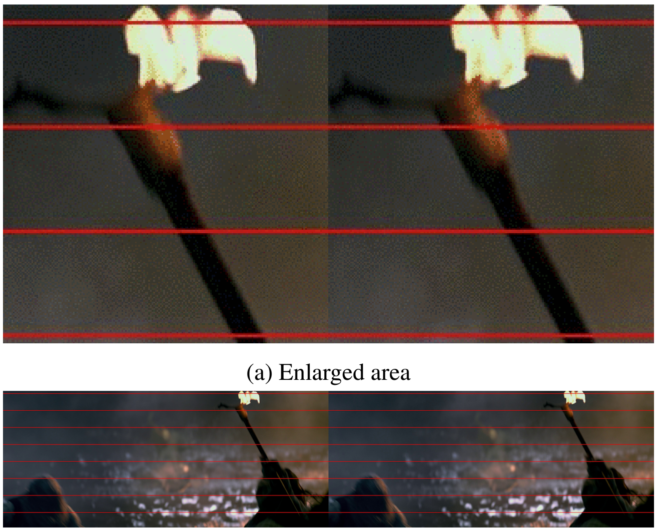
3.2. Dataset generation

To test our data-set-generation method, we trained a neural network to estimate perspective distortion and converged camera axes. To this end, we created a set of stereoscopic frames containing these distortions. The amount of perspective distortion and the angle of the converged axes were random. The convergence angle of the camera axes was between 0 and 10 degrees (Figure 4); the perspective distortion of one chamber, up or down, was no more than 20 centimeters (Figure 3). Most frames exhibited both distortions, but some exhibited just one or neither. Since we developed a model that predicts the distortion amount on the basis of one frame, we tuned the mod to prevent adjacent frames from being similar: they were captured with a time step of 1,000 ms, each with a random camera direction, random weather conditions and a random time of day. The result was a total of 4,500 frames with a resolution of 1,920×1,080: 4,000 were for the training set and 500 for the test set. We included images of both city streets and countryside landscapes. The generation process took just over three hours.

For more-realistic views, we modified most frames with noise and/or blur. In particular, we added Gaussian noise to 80% of the frames. To fluctuate the noise power, the variance was random in the range 0 to 0.01 with an average value of 0. Gaussian blur was in 80% of the frames as well. To imitate the camera's operation, the image blur was only in the foreground or background rather than both. Information about object distance came from the depth maps we generated with the frames. The Gaussian-kernel size was (11, 11) with a randomly chosen standard deviation of 1 to 4 in the horizontal direction. After these transformations, the synthetic images were more similar to the real ones.

3.3. Model architecture

The model's architecture (Figure 5) is a convolutional neural network. The input data is a calculated disparity map [23] for the left view, scaled down to 480×272, and a corresponding confidence map. The model returns two numbers normalized from -1 to 1: the angle between the converging optical axes (in degrees) and the perspective distortion (in centimeters).



(a) Enlarged area

(b) Left and right images.

Fig. 6: A scene from *Pirates of the Caribbean: On Stranger Tides* with perspective distortion.

In the first experiments, the model estimated only one of two distortions. But we settled on such an option to simultaneously detect perspective distortion and converged axes, since the resulting model showed good results on the validation set. The neural network has 2,886,754 trainable parameters. For the loss function, we chose mean squared error (MSE):

$$L_{MSE} = \frac{1}{2}((persp_{est} - persp_{gt})^2 + (conv_{est} - conv_{gt})^2).$$

4. RESULTS

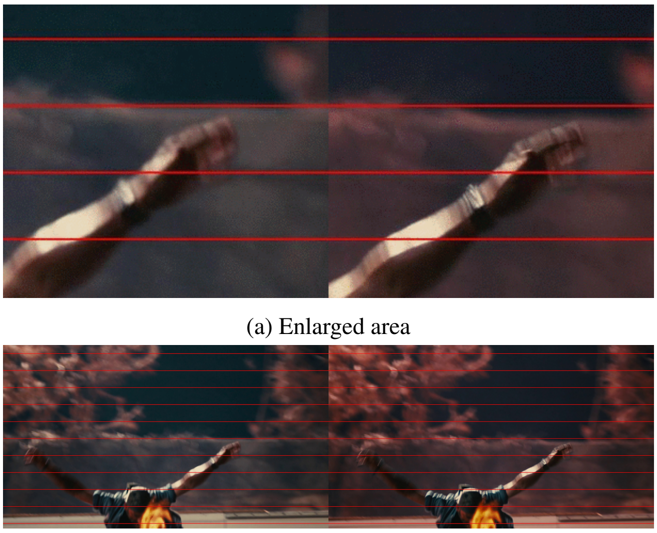
4.1. Limitations

The speed of the mod depends on the time step between frames — the period during which there is no image capture or depth measurement but the car is moving. Also, the greater the number of shooting angles, the longer it takes to produce the frames. The Table 1 shows the number of frames the mod generates per minute as these parameters change. We took the measurements using a system with a 2.80GHz Intel Core i7-7700HQ processor, an Nvidia GeForce GTX 1050 Ti graphics card and 16 GB of RAM.

Because a car can move autonomously, data generation can proceed without operator participation. We can therefore efficiently create synthetic datasets.

4.2. Training results

We trained the model on 4000 prepared samples over 70 epochs using the Adam optimizer with an initial learning rate of 10^{-4}. Every 10 epochs without a fall in the validation error, we reduced the learning rate by 10 times. Testing of the trained model employed a validation sample of 500 frames. For accuracy metric we use the Pearson correlation coefficients (PCCs) between ground-truth distortion and model-estimated values. For converged axes, the PCC was 0.956, and for perspective distortion, it was 0.859. This indicates a high correlation of the predicted values with ground-truth values.



(a) Enlarged area

(b) Left and right images.

Fig. 7: A scene from *Drive Angry* with perspective distortion.

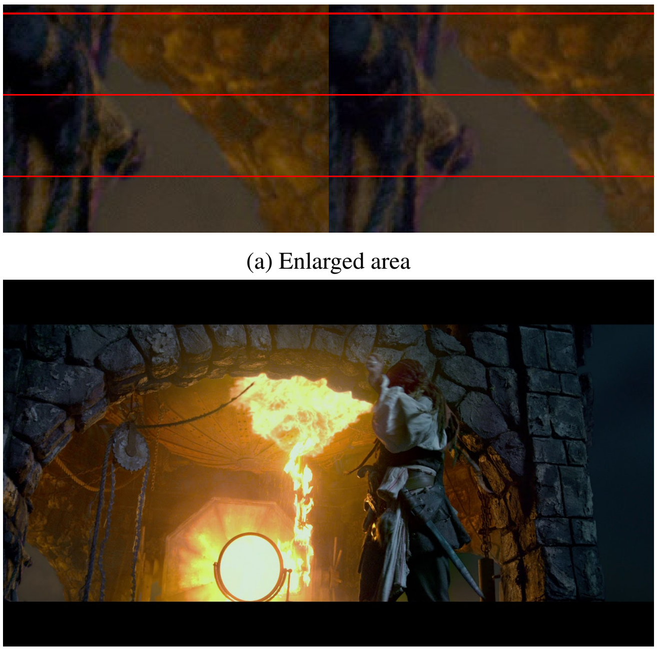
Time step between frames (ms)	Number of views			
	1	2	3	4
40	58	30	22	18
1,000	28	22	16	13

Table 1: Number of frames the mod generates per minute.

4.3. Checking movies

We also used the trained model to find distortion examples in stereoscopic films: *Drive Angry* and *Pirates of the Caribbean: On Stranger Tides*. For each frame in these movies, we estimated the magnitude of the perspective distortion and the convergence angle of the cameras. To select the frames of interest to us, we estimated threshold values — any frames below those thresholds received no further considered. Moreover, among the rest of the frames, we manually selected those that contain the distortions of interest.

Examples of perspective distortion can be seen in Figures 6 and 7. Magnified objects show that the camera has moved up or down. Figures 8 and 9 show frames with converged camera axes. Vertical parallax occurs in the corners of the



(a) Enlarged area

(b) Left image

Fig. 8: A scene from *Pirates of the Caribbean: On Stranger Tides* with converged axes.

frame.

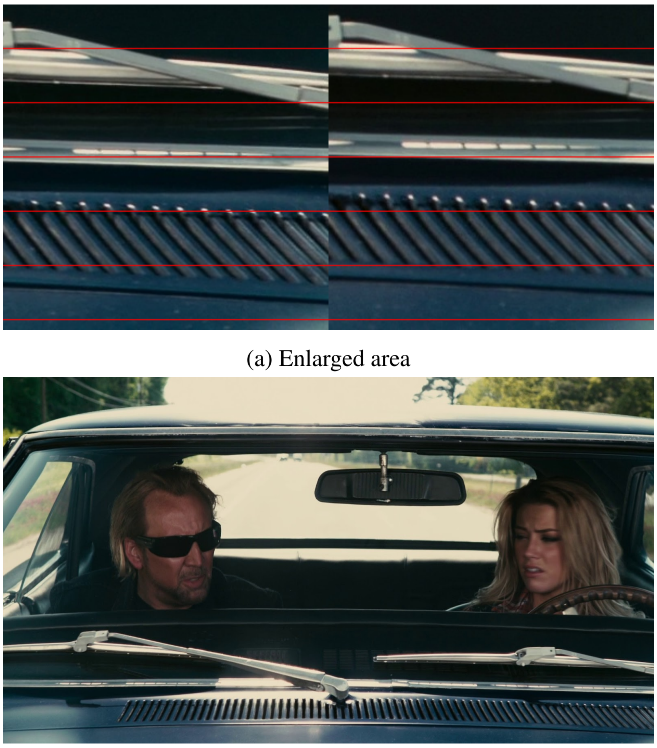
5. CONCLUSION

Our proposed method facilitates creation of datasets to aid in processing and analyzing stereoscopic and multi-angle video. Programmatically setting camera parameters is much more convenient than working with real cameras. The GTA V computer game enables automatic generation of large synthetic datasets with specified characteristics at a minimal cost. This capability opens new avenues for applying machine learning to various problems. In particular, we showed this technique can help search for perspective distortion and converged camera axes in stereoscopic video, and we used the trained models to find these artifacts in films. We plan to improve our approach to assessing stereoscopic-video quality by, for example, considering more distortion types.

6. ACKNOWLEDGMENTS

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.



(a) Enlarged area

(b) Left image

Fig. 9: A scene from *Drive Angry* with converged axes.

7. REFERENCES

[1] J. Zbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” *J. Mach. Learn. Res.*, vol. 17, pp. 65:1–65:32, 2016.

[2] X. Li, H. Huang, H. Zhao, Y. Wang, and M. Hu, “Learning a convolutional neural network for propagation-based stereo image segmentation,” *The Visual Computer*, vol. 36, pp. 39–52, 2018.

[3] S. Li, X. Han, M. Zubair, and S. Ma, “Stereo image quality assessment based on sparse binocular fusion convolution neural network,” *2019 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, 2019.

[4] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[5] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3061–3070, 2015.

[6] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *GCPR*, 2014.

[7] Q. Wang, S. Zheng, Q. Yan, F. Deng, K. Zhao, and X. Chu, "Irs: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation," *ArXiv*, vol. abs/1912.09678, 2019.

[8] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 899–908, 2019.

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

[10] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2232–2241, 2017.

[11] B. Hurl, K. Czarnecki, and S. Waslander, "Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception," *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2522–2529, 2019.

[12] O. McNulty, "3d object localisation with convolutional neural networks," bachelor's thesis, The University of Sydney, Sydney, Australia, 2017.

[13] D. Butler, J. Wulff, G. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*, 2012.

[14] A. Tsirikoglou, J. Kronander, M. Wrenninge, and J. Unger, "Procedural modeling and physically based rendering for synthetic data generation in automotive applications," *ArXiv*, vol. abs/1710.06270, 2017.

[15] W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," *ArXiv*, vol. abs/1609.01326, 2016.

[16] A. Shafaei, J. Little, and M. Schmidt, "Play and learn: Using video games to train computer vision models," *ArXiv*, vol. abs/1608.01745, 2016.

[17] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 746–753, 2017.

[18] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, "The omniscape dataset," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1603–1608, 2020.

[19] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016.

[20] F. Zilly, M. Müller, P. Eisert, and P. Kauff, "Joint estimation of epipolar geometry and rectification parameters using point correspondences for stereoscopic tv sequences," *Proceedings of 3DPVT*, 01 2010.

[21] M. Georgiev, A. Gotchev, and M. Hannuksela, "A fast and accurate re-calibration technique for misaligned stereo cameras," *2013 IEEE International Conference on Image Processing*, pp. 24–28, 2013.

[22] P. Napieralski and M. Kowalczyk, "Detection of vertical disparity in three-dimensional visualizations," *Open Physics*, vol. 15, pp. 1028 – 1033, 2017.

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