A Study of Conventional and Learning-Based Depth Estimators for Immersive Video Transmission

Smitha Lingadahalli Ravi
Orange Labs
35510 Cesson Sévigné, France
smitha.lingadahalliravi@orange.com

Marta Milovanović
Orange Labs
78280 Guyancourt, France
marta.milovanovic@orange.com

Luce Morin
INSA Rennes - IETR
35000 Rennes, France
luce.morin@insa-rennes.fr

Félix Henry
Orange Labs
35510 Cesson Sévigné, France
felix.henry@orange.com

***Abstract*—Obtaining an accurate depth map of a scene is very important for major applications like immersive video, robotics, autonomous driving, and many more. The different methods to estimate depths can be classified as conventional and learning-based methods. While these methods have been studied for their depth accuracy, less attention has been paid to studying their performance in the use case of depth image-based rendering (DIBR). Here we study and evaluate two conventional methods and five learning-based methods for a real-world use case of immersive video transmission in the context of MPEG-I. The user-requested views are synthesized using Test Model for Immersive Video (TMIV) from the depth maps obtained by all methods and original texture views. The synthesized images are compared with their original counterparts using various quality metrics.**

***Index Terms*—depth estimation, deep learning, view synthesis, immersive video transmission, MPEG-I**

I. Introduction

In conventional video processing, a three-dimensional real-world scene is represented as a two-dimensional image or a video. An essential component is lost during this process, corresponding to the third dimension: the depth component. Of course, two-dimensional displays are usually sufficient for many applications. However, applications like autonomous driving, immersive video, robotics, 3D reconstruction, augmented reality, and biometrics require the third dimension. In robotics, depth is a principal component to enable performing tasks like perception, navigation, and planning. The depth estimation techniques can be broadly classified into conventional and learning-based depth estimators. In conventional methods, they typically rely on computing the disparity of each pixel across rectified images by matching corresponding pixels along the epipolar lines, thus allowing depth estimation through triangulation [1]. Humans can quickly identify the approximate size of an object, its location, its disparity and can infer how far it is from our eyes. This is because our brain has enabled us to make use of prior knowledge, *i.e.*, previously seen scenes, and develop mental models of the three-dimensional world. In learning-based methods, this prior knowledge is used to solve the problem as a learning task [2]. In recent years, depth estimation methods have been compared against each other by their intrinsic capacity to estimate depth values that are as close as possible to the given ground truth. The performance measure is a variant of a signal-to-noise ratio between the estimated and original depth maps. While this is certainly a very relevant purpose when developing high-performance methods, depth maps are generally not the final goal but only a stage in a processing pipeline of a given use case. The ISO-MPEG-I MIV standard [14, 15], which intends to facilitate the storage and transmission of immersive video content (for AR/VR applications), requires a non-normative depth estimation component to perform view synthesis. While we participated in the MIV standard, we noticed that researchers tend to assume that learning-based depth estimation would soon outperform the conventional depth estimators that are currently used. Therefore, it is important to inform the community about the real performance of these approaches for this essential use case. Hence, in this paper, we study the performances of two state-of-the-art conventional depth estimation methods and five state-of-the-art learning-based depth estimation methods for a specific use case of immersive video synthesis. In this use case, videos are captured from several viewpoints of a scene (typically 10 to 20). These videos are used to construct the depth map associated with each view. Using DIBR techniques [3], any additional viewpoint can be synthesized. Our goal is to compare the impact of the different depth estimation methods on the quality of the synthesized views. The rest of the paper is structured as follows: Section 2 summarizes the studied depth estimation methods, Section 3 introduces MPEG-I test sequences used for evaluation, as well as the experimental setup and analysis. Finally, the obtained results, discussion, and conclusion are in Section 4 and Section 5, respectively.

II. Depth Estimators

A. Conventional depth estimators

These first-generation depth estimation methods solely rely on matching pixels across multiple rectified images. For our analysis, we have selected two state-of-the-art depth estimators

978-1-6654-7189-3/22/$31.00 © 2022 IEEE

that have been used in the MPEG standardization group, Depth Estimation Reference Software (DERS) [4] and Immersive Video Depth Estimation (IVDE) Software [5, 6]. In DERS, the error cost for each pixel in the center view and possible depth are computed by combining errors of all neighboring views (left, right, top, bottom). Finally, it is subjected to optimization through graph cuts to get the optimal depth estimation per pixel. In IVDE, the depth estimation is carried out simultaneously on all the input views. Instead of estimating depth for each pixel, it is estimated for segments, where the segment is a small homogeneous patch of the image. Hence, processing time and the quality of the depth maps are dependent on the size of the segments. Finer segments attain high-quality depth maps, whereas larger segments provide faster computation. Currently, in MPEG immersive video, the depths of five out of six natural sequences are generated by IVDE.

B. Learning-based depth estimators

These are the second generation of depth estimation methods based on neural networks. For our analysis, we have selected two stereo based depth estimators, Guided Aggregation Network (GA-Net) [7] and Group-Wise Correlation Network (GWC-Net) [8], and two multi-view stereo (MVS) based depth estimators, Depth Inference for Unstructured Multi-view Stereo (RMVSNet) [9] and Adaptive Aggregation Recurrent Multi-view Stereo Network (AA-RMVSNet) [10]. We have also included a state-of-the-art learning-based synthesizer, IBRNet [11], in our experimentation, as it can be used as a depth estimator.

The stereo-based depth estimators can be broadly classified as encoder-decoder with 2D convolution (ED-Conv2D) [12] and cost volume matching with 3D convolution (CVM-Conv3D) networks [13]. ED-Conv2D methods use an encoder-decoder structure for the neural network. The encoder produces a cost volume with Weight x Height x Disparity dimension, with a cost associated with each disparity per pixel, and the decoder part predicts the disparity based on the cost volume. The ED-Conv2D methods are computationally efficient but are limited in performance efficiency. To overcome the accuracy problem in disparity estimation, researchers proposed CVM-Conv3D networks. Firstly, the left and right feature maps of stereo pairs of images are computed. Then, a 4D cost volume is constructed, which has the feature map dimension in addition to the other dimensions. This forces the neural network to extract the features that are more relevant to solve the disparity estimation problem. For our analysis, we have selected two CVM-Conv3D models, GA-Net and GWC-Net.

A typical learning-based MVS network consists of three parts, a feature extraction network, a cost volume constructor, and a cost volume regularization network. Firstly, using a shared convolutional neural network (CNN), it extracts a feature representation of each input image. The information obtained from the feature maps is aggregated to form a 3D cost volume for depth regularly sampled between the depth range (it is usually provided with the dataset). The cost volume only encodes local information. A further step of 3D convolutions is then performed to propagate this local information. The depth map is finally extracted from the refined cost volume using a differentiable *argmin* operation. The main disadvantage of MVS networks is that they cannot handle high-resolution images since the memory demand of learning cost volume regularization rises with the model's resolution. Also, it should be noted that these models do not use any specific data augmentation and produce depth maps that are four times smaller than the original input. For our analysis, we have selected two learning-based MVS depth estimators: RMVSNet and AA-RMVSNet.

Sequence	Type	Resolution	Frames	Views
Frog	Natural	1920x1080	300	13x1
Kitchen	Synthetic	1920x1080	97	5x5
Shaman	Synthetic	1920x1080	150	5x5
Painter	Natural	2048x1088	350	4x4
Street	Natural	1920x1088	250	9x1
Carpark	Natural	1920x1088	250	9x1
Fan	Synthetic	1920x1080	97	5x3
Mirror	Synthetic	1920x1080	97	5x3

Table 1. The MPEG-I visual test sequences used for evaluation along with their nature, resolution, and the number of frames utilized for tests.

Recently, texture-based synthesizers which render the requested view directly from the captured views have emerged as a very promising area of research. They are capable of rendering high-resolution views with good generalization properties. We have therefore adopted IBRNet as the final method in our tests. It shares principles with the neural rendering method NeRF [25], where it directly aggregates information from nearby images to synthesize views and involves volume rendering along a camera ray. Currently, it is a state-of-the-art learning-based view synthesizer and can be used to generate depth maps.

III. EXPERIMENTAL SETTINGS AND ANALYSIS

For our study, we used the test sequences of the MPEG Common Test Conditions [16]. Table 1 shows the list and characteristics of the sequences. We only consider perspective and rectified content for the analysis, as most pre-trained models are trained on such datasets. The MPEG-I dataset is comprised of multi-view sequences captured by sparse camera setup. Our experiment aims at comparing the depth estimation methods described in Sections 2 and 3 for the specific use case of view synthesis using depth-based image rendering (DIBR). To this end, we will use the setup of the TMIV software [14, 15]. TMIV is the reference software implementing the non-normative encoding, normative decoding, and non-normative rendering techniques according to the MPEG Immersive Video coding standard (MIV). For our study, a subset of the tools of the TMIV processing chain is used. Only the decoder pipeline is used by replacing each view's decoded texture and depth component with the original texture and our tested depth component, respectively. The camera parameters from the bitstream are used to perform view synthesis using a

Sequence	1	2	3	4	5	6	7
Frog	805.9	486.0	16.9	1.8	5.6	47.3	165.7
Kitchen	796.1	410.8	16.7	1.6	5.3	47.1	164.8
Shaman	823.3	504.3	18.1	1.7	5.4	47.7	165.9
Painter	801.4	435.7	16.8	1.7	5.3	47.2	165.3
Street	704.3	320.4	16.2	1.4	5.0	46.2	163.5
Carpark	789.5	358.2	16.4	1.5	5.1	46.9	164.1
Fan	892.3	572.7	22.4	2.5	6.4	47.3	167.3
Mirror	852.5	530.6	16.9	1.9	6.2	47.1	165.4

Table 2. Comparison of runtime in seconds per frame for depth estimation. 1:DERS, 2:IVDE, 3:GA-Net, 4:GWC-Net, 5:RMVSNet, 6:AA-RMVSNet, 7:IBRNet.

view weighting synthesizer. The inputs to the renderer are 10-bit texture and 10-bit depth (normalized disparities), camera parameters list, and finally, the target camera parameters for a perspective viewport. The output of the renderer is a perspective view. The synthesized output view is provided in luma and chroma 4:2:0 format with 10-bit support for texture. In order to evaluate the performance of the synthesis, and for each of the tested sequences, we synthesize the viewport at the location and angle of an existing view. Of course, this existing view is excluded from the list of input views available for synthesis. This is performed for all views at all time instants of the video. Both DERS and IVDE use the configurations recommended by the Common Test Conditions of MPEG. We used DERS 8.0, IVDE 1.0, and TMIV 4.0 software versions for our evaluation. The GA-Net and GWC-Net produce disparity maps, which are subsequently converted into depth maps [17]. Both GA-Net and GWC-Net used the model pre-trained on the KITTI dataset [18, 19], and RMVSNet and AA-RMVSNet used the model pre-trained on the DTU dataset [20] for testing. Evaluation of IBRNet was conducted in two ways: 1) directly testing with default model, which is trained using both synthetic [21] and real data [22, 23, 24], on MPEG sequences without any fine-tuning, 2) fine-tuning the default pre-trained model on MPEG content before testing. Due to the modest size of the MPEG dataset, we have fine-tuned one instance of IBRNet for each sequence by using all other sequences as training set. Finally, in order to present a broad comparison that takes into consideration different types of distortions, the performance of the synthesis was evaluated using various quality metrics like Immersive Video PSNR (IV-PSNR) [26] which is designed to reflect virtual view synthesis artifacts, Learned Perceptual Image Patch Similarity (LPIPS) [27], which evaluates the distance between image patches, and the Structural Similarity Index (SSIM) [28] which is a perceptual metric that quantifies image quality degradation.

IV. Results and Discussion

The quality of view synthesis from depth maps produced by different depth estimators evaluated with various quality metrics is shown in Table 3. The "original" column corresponds to a synthesis done with the original depth maps provided with the MPEG sequences. These depth maps are not perfect and are typically estimated offline from the input

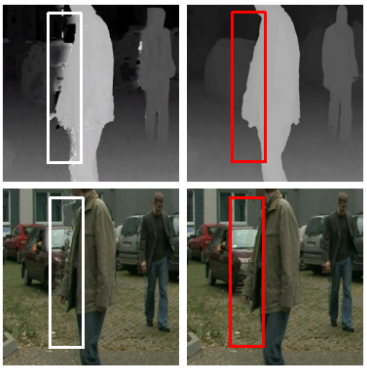

Fig. 1. Qualitative comparison of depth maps and synthesized views of Carpark sequence

views and then further refined "by hand". As shown in Table 3, results vary from sequence to sequence, and more than half of the sequences have the best quality for learning-based depth estimators (GA-Net, fine-tuned IBRNet) in the objective comparison using IV-PSNR and SSIM. For the LPIPS metric, IVDE has the best average quality of the synthesized views. Overall, IVDE produces the best average results in all the three quality metrics. Both conventional and learning-based depth estimators have their advantages and disadvantages. The conventional methods produce noisy depth maps, which can cause degraded synthesized views (Fig. 1, white box), but they recover sharp edges of objects. The learning-based stereo networks produce smooth and "cloudy" depth maps and generate temporally coherent synthesized views (Fig. 1, red box), but they have difficulty in recovering accurate boundaries. Indeed, as the output depth maps are four times smaller than the original input, learning-based MVS networks fail to recover thin structures. Also, the learning-based MVS networks are temporally inconsistent and have difficulty reconstructing accurate boundaries around the moving objects when compared to conventional methods (Fig. 2, yellow, green and orange box).

The stereo-based networks are not optimized to produce good quality depth maps for wide baseline data, and when tested with such stereo pairs, they fail to produce meaningful depth maps. It should be noted that the depth maps from IBRNet produce better synthesized image quality and recover finer details for synthetic sequences like Kitchen and Shaman when the network is fine-tuned with MPEG content (Fig. 2, blue box). Due to the limitations of our simulation platform and the large size of MPEG sequences, the other methods could not be fine-tuned, but a similar behaviour would be expected. Also, the learning-based methods like GANet, GWCNet require huge amount of computational resources (eight, 24GB GPUs) and this is impractical for generating the depth maps needed for immersive video transmission. Table 2 shows the time taken to produce one depth map on a GeForce RTX 2080 Ti GPU except for conventional methods, which only use CPU. It

	IV-PSNR ↑								
Sequences	Original	DERS	IVDE	GANet	GWCNet	RMVSNet	AARMVSNet	IBRNet	IBRNet(FT)
Frog	36.92	34.96	35.13	**35.87**	35.29	30.28	34.48	31.52	32.18
Kitchen	40.81	38.46	38.92	39.27	38.63	35.74	37.61	38.64	**39.42**
Shaman	45.74	41.18	42.06	41.38	41.55	36.79	38.25	40.34	**42.28**
Painter	45.35	39.37	**39.82**	38.91	39.67	37.02	39.29	38.14	38.47
Street	41.59	39.61	38.81	**40.56**	40.08	35.09	39.92	38.94	37.96
Carpark	42.17	39.45	38.29	**40.39**	39.78	34.78	37.16	37.68	37.41
Fan	37.88	35.13	**36.71**	34.84	34.29	33.68	34.48	34.92	35.21
Mirror	41.52	38.56	**40.46**	35.67	35.12	33.46	34.18	36.92	38.18
Average	41.37	38.34	**38.77**	38.36	38.05	34.66	36.92	37.13	37.65
	LPIPS ↓								
Sequences	Original	DERS	IVDE	GANet	GWCNet	RMVSNet	AARMVSNet	IBRNet	IBRNet(FT)
Frog	0.118	0.137	0.149	**0.134**	0.141	0.212	0.178	0.198	0.193
Kitchen	0.112	0.131	**0.126**	0.129	0.129	0.151	0.140	0.129	0.132
Shaman	0.087	0.102	0.112	0.109	0.106	0.184	0.168	0.119	**0.101**
Painter	0.091	0.121	**0.114**	0.132	0.118	0.149	0.125	0.141	0.138
Street	0.109	0.120	0.126	0.125	0.119	0.165	**0.112**	0.153	0.151
Carpark	0.104	0.124	0.129	**0.118**	0.128	0.193	0.152	0.148	0.145
Fan	0.105	0.130	**0.121**	0.135	0.138	0.152	0.141	0.138	0.132
Mirror	0.101	0.118	**0.109**	0.117	0.130	0.148	0.131	0.128	0.121
Average	0.103	0.124	**0.123**	0.125	0.126	0.169	0.143	0.144	0.139
	SSIM ↑								
Sequences	Original	DERS	IVDE	GANet	GWCNet	RMVSNet	AARMVSNet	IBRNet	IBRNet(FT)
Frog	0.864	0.852	0.849	**0.856**	0.852	0.784	0.817	0.795	0.798
Kitchen	0.912	0.881	0.885	0.892	0.882	0.861	0.872	0.879	**0.902**
Shaman	0.926	0.916	0.901	0.906	0.912	0.882	0.892	0.915	**0.919**
Painter	0.920	0.884	**0.896**	0.882	0.887	0.859	0.879	0.872	0.875
Street	0.915	0.887	0.880	**0.894**	0.889	0.865	0.881	0.871	0.874
Carpark	0.919	0.879	0.871	**0.883**	0.880	0.856	0.861	0.868	0.869
Fan	0.904	0.875	**0.889**	0.872	0.868	0.852	0.856	0.864	0.870
Mirror	0.924	0.889	**0.908**	0.892	0.889	0.875	0.882	0.885	0.891
Average	0.910	0.882	**0.884**	**0.884**	0.881	0.854	0.867	0.868	0.874

Table 3. The comparison of average quality of synthesized views using IV-PSNR (higher means better), LPIPS (lower means better) and SSIM (higher meansbetter) metrics with respect to the various depth estimators.

Fig. 2. Qualitative comparison of synthesized views of Frog, Street, Shaman and Fan (top to bottom) obtained by using various depth estimators.

can be observed that the GWCNet is the fastest depth estimator compared to other depth estimators.

Overall, the conventional depth estimators produce better quality depth maps compared to learning-based depth estimators. The main reason is probably that the learning-based methods are trained on ground-truth depth maps which are imperfect (especially for natural content) and this causes degradation in the quality of depth maps, which in turn produces artifacts in the synthesized views. Also, the learning-based methods only try to improve the quality of the depth maps, but they should rather be trained end-to-end with the synthesized view quality as loss function, something that is not feasible due to the synthesis being non-differentiable.

V. Conclusion

It is the first time to our knowledge that a comparative study on conventional and learning-based depth estimation methods following MPEG Common Test Conditions has been conducted. Our study is instrumental in determining which depth estimator to use in immersive video transmission depending on the requirements regarding synthesized image quality, computational power, and run-time. Our work also benefits the broader research community in understanding the behavior of networks to input images captured by sparse camera setup. We have measured the impact of the different depth estimation methods on the specific use case of view synthesis for immersive video. Surprisingly, learning-based depth estimation is not substantially better than a conventional approach for this use case. Besides, learning-based methods do not exhibit the graceful degradation of conventional methods in difficult areas, such as object boundaries. We believe that further exploration is needed in terms of optimizing the networks for wider-baseline and for high-resolution data. However, given the relatively short amount of time since learning-based approaches are used for depth estimation and their already state-of-the-art performance in terms of depth accuracy, there is little doubt that they will improve substantially in the near future for the use case of view synthesis.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," IJCV, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015.

[3] Christoph Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," San Jose, CA, May 2004, pp. 93–104.

[4] Eduardo Juarez et. al., Manual of Depth Estimation Reference Software, (DERS 8.0), ISO/IEC JTC 1/SC 29/WG 11 N18450, Geneva, Switzerland, March 2019.

[5] Dawid Mieloch, Adrian Dziembowski, Jakub Stankowski, Olgierd Stankiewicz, Marek Domanski, Gwangsoon Lee, and Yun Young Jeong, "Immersive video depth estimation," ISO/IEC JTC 1/SC 29/WG 11 m53407, Apr. 2020.

[6] Manual of Immersive Video Depth Estimation, ISO/IEC JTC1/SC29/WG11 MPEG2020/N19224, Online, April 2020.

[7] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in CVPR, 2019.

[8] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in CVPR, 2019.

[9] Yao, Yao, et al. "MVSNet: Depth inference for unstructured multi-view stereo." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[10] Wei, Zizhuang, et al. "AA-RMVSNet: Adaptive aggregation recurrent multi-view stereo network." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[11] Wang, Qianqian, et al. "IBRNet: Learning multi-view image-based rendering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[12] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," PAMI, vol. 30, no. 2, pp. 328–341, 2008.

[13] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," PAMI, vol. 35, no. 2, pp. 504–511, 2012.

[14] "Working Draft 4 of Immersive Video," ISO/IEC JTC 1/SC 29/WG 11 N19001, Feb. 2020

[15] Test Model 4 for Immersive Video, ISO/IEC JTC1/SC29/WG11 MPEG/N18795, October 2019, Geneva, Switzerland.

[16] Joel Jung, Bart Kroon, and Jill Boyce, "Common Test Conditions for Immersive Video," ISO/IEC JTC 1/SC 29/WG 11 N18997, Feb. 2020.

[17] "Depth map formats used within MPEG 3D technologies", ISO/IEC JTC1/SC29/WG11 MPEG2017/N16730, January 2017, Geneva, CH

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," IJRR, vol. 32, no. 11, pp. 1231–1237, 2013.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in CVPR, 2012.

[20] Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. International Journal of Computer Vision (IJCV) (2016)

[21] Google Research, Google scanned objects. (https://tinyurl.com/2dbynjct)

[22] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. CVPR, 2019.

[23] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima K. Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM TOG, 2019

[24] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. SIGGRAPH, 2018

[25] Mildenhall, Ben, et al. "NeRF: Representing scenes as neural radiance fields for view synthesis." European conference on computer vision. Springer, Cham, 2020.

[26] J. Stankowski, A. Dziembowski, "Even faster implementation of IV-PSNR software", ISO/IEC JTC1/SC29/WG04 MPEG/M54896, October 2020, Online.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. CVPR, 2018.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," TIP, vol. 13, no. 4, pp. 600–612, 2004.