The Visual Computer manuscript No.
(will be inserted by the editor)

Comparison of Light Field Compression Methods

David Barina · Marek Solony · Tomas Chlubna · Drahomir Dlabaja · Ondrej Klima · Pavel Zemcik

Received: date / Accepted: date

Abstract In this article, we compare the impact of state-of-the-art light field compression methods. It addresses quality of (a) refocused images and (b) point clouds reconstructed from 4D light field data. The methods include recent video compression formats, specifically H.265, AV1, XVC, and H.266/VVC (finalized in 2020). In addition, we have extended a standard image compression method into four dimensions and compared it with the video compression formats. It turned out that the new VVC format demonstrated superior performance, closely followed by the underrated XVC. Apart from the comparison, we show that the four-dimensional light field data can be compressed with a higher ratio than independent still images while maintaining the same visual quality of a perceived picture.

Keywords 4D Light Fields · Plenoptic Imaging · Compression · Image Refocusing · 3D Reconstruction

1 Introduction

A scene can be rendered from every position of the virtual camera when information about light is available for every point in a 3D space and every direction relative to this point coming from this direction to the given point. In other words, a field of light in the scene can be described by a function, and with this knowledge, any scene can be rendered just using this visual information. An ideal light field (LF) representation can be defined in terms of geometric optics using a 5D plenoptic function. 5D plenoptic function returns color radiance for arguments representing a three-dimensional position of a point in space and a direction defined by two spherical angles [1]. The domain of this function is defined over the whole space of the scene. This function can also be extended to 7D by adding time and wavelength dimensions. Using these functions directly, 7D and 5D light field is defined for a given space. The 5D plenoptic function is, however, usually being replaced by a 4D representation while the scene is closed in a convex hull. Intersection points of the virtual camera rays with two planes that are enclosing the scene are then used as the parameters for the radiance function. The virtual camera is then restricted to "look at" the scene from outside of this hull. Also, instead of a physical quantity of radiance, a color value (usually RGB values) is used when sampling the scene with rays. The time dimension is ignored in this representation, reducing the dimensionality to four. The advantage of this approach is that for a visual reconstruction of the scene, a set of photos

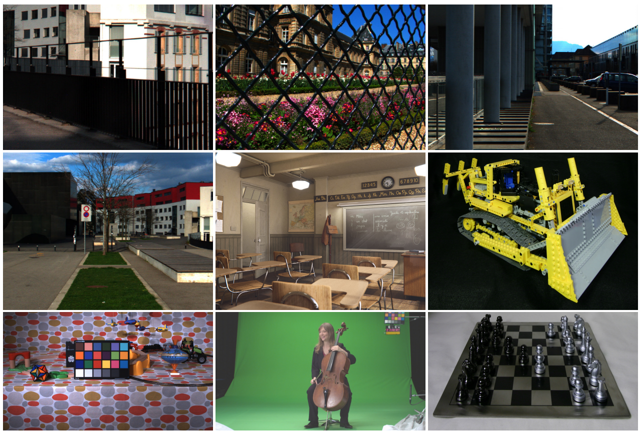


Fig. 1: Dataset of 4D light fields used in this paper.

Centre of Excellence IT4Innovations
Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno, Czech Republic
E-mail: {ibarina,isolony,ichlubna,iklima,zemcik}@fit.vutbr.cz

description	source	resolution	disparity
Black Fence	École polytechnique fédérale de Lausanne (EPFL)	$15 \times 15 \times 624 \times 432$	-0.1 to 0.5
Palais du Luxembourg	École polytechnique fédérale de Lausanne	$15 \times 15 \times 624 \times 432$	-0.4 to 1
Pillars	École polytechnique fédérale de Lausanne	$15 \times 15 \times 624 \times 432$	0.1 to 0.55
Red & White Building	École polytechnique fédérale de Lausanne	$15 \times 15 \times 624 \times 432$	0.05 to 0.5
Classroom	Saarland University (synthetic light field)	$8 \times 8 \times 1936 \times 1216$	-3 to 35
Lego Bulldozer	Stanford Computer Graphics Laboratory	$17 \times 17 \times 1536 \times 1152$	-1 to 8
HaToy	Saarland University	$8 \times 8 \times 1280 \times 720$	51 to 77
Take2_1	Saarland University	$8 \times 8 \times 1280 \times 720$	40 to 48
Chess	Stanford Computer Graphics Laboratory	$17 \times 17 \times 1400 \times 800$	-1 to 3

Table 1: Dataset used in this paper. The adjacent image disparity range (last column) is given in pixels.

of the scene can be used. These photos are then mapped on one of the planes or surfaces of the 4D parametrizations [2, 3]. The result is a discrete representation of the light field, also known as the 4D light field [4] or lumigraph [5]. Each photo is a view coming from one of the cameras in the virtual camera grid, capturing the light field of the scene.

A light field can be viewed as an extension of classic photography or video. Additional edits can be performed in post-processing. Users can, for example, change the focusing distance or the position of the virtual camera of the scene without the need to physically recapture the scene. Light fields can be used in modern computer games as photorealistic and computationally inexpensive assets [6] or in film industry for interactive playback or extended editing [7].

Since the storage and transmission requirements for 4D light field data are tremendous, compression techniques for these data are gaining momentum in recent years [8]. In this paper, we evaluate the impact of state-of-the-art video compression methods on light field data. Decoded light field data are not meant to be viewed as raw images but rendered using specific interpolation methods. Therefore, the compression artifacts might affect the resulting quality in a different way than in classic 2D images that are the usual targets for already published benchmarks. The description of the compression methods is discussed in Section 2. The comparison methodology is described in Section 3. The results of the comparison and additional discussion can be found in Section 3 as well. Section 4 concludes this article on the basis of the results.

2 Related Work

This article follows our previous work [9]. This section presents the compression formats selected for our comparison in this paper.

Light field views are usually captured as images from various positions in the scene taken by a multi camera-array, single moving camera, or plenoptic camera capturing multiple views on a single sensor. Since these views resemble video frames, state-of-the-art video formats were chosen for the experiments described in this paper. In accordance to our previous experience [9], we have chosen H.265, AV1, XVC, and upcomming VVC video formats.

H.265 (also called HEVC) is a successor of widely used H.264. Compared to its predecessor, H.265 offers almost 50% better compression in certain cases [10] while maintaining the same visual quality. The main difference is that while H.264 uses the discrete cosine transform (DCT) on fixed-sized blocks, H.265 uses similar transforms on coding tree units (CTUs) having variable sizes up to 64×64 pixels. Improved motion compensation and spatial prediction methods in H.265 come with the cost of higher computation requirement than H.264. The main advantage of H.265 is its support in various areas including GPUs. Note that we use the x265 encoder to compress light field data in this article.

AV1 is the competitor of H.265 and achieves generally the same compression performance [11, 12]. The main purpose of AV1 is to offer a royalty-free alternative to H.265. It has been developed by the Alliance for Open Media as a successor of VP9 adopting concepts from VP10 development. While AV1 aims to be an Internet video standard, the hardware requirements are higher than of H.265. We use the reference libaom library to compress light field data.

Divideon released the new xvc (referred to herein as XVC) codec in 2017 aiming to offer better compression quality than both H.265 and AV1 and to be less computationally complex than AV1. The xvc format uses the same block-based compression scheme as the previously mentioned formats. One of the main differences is that xvc uses non-square coding units in transform and prediction phase. Another features providing better results are adaptive motion vector prediction, affine motion prediction, cross-component prediction, transform selection and local illumination compensation. As a result, xvc can reduce bitrate up to 25% [13] while maintaining the same visual quality compared to AV1. Note that we use the official xvc codec.

VVC was finalized at the end of 2020. The main motivation for its development is the expectation of 4K and 16K video resolution becoming a standard video format along with increased popularity of 360-degree and HDR videos. Preliminary tests show that at least 30% quality improvement can be reached just by improving methods used in H.265 [14] using newer algorithms. The number of intra prediction directions is raised from 33 (H.265) to 65, rectangular and larger blocks are allowed and new chroma prediction is included. Four separable discrete cosine/sine transforms are used instead of one, and a new dependent scalar quantization method is applied. The adaptive loop filtering method which was proposed but not included in H.265 has been included in the VVC standard. The block partitioning scheme has been extended using two stages of tree-based splitting. Unfinished VVC implementation already outperforms AV1 in certain cases [15]. In this article, we use the VTM reference software for VVC (still under development).

According to our experience from the previous work, we also decided to include a standard image method extended to four dimensions. This method is referred to as the LF4 format. The method begins by finding the optimal disparity for offsetting image views, minimizing the average error. Views are further interpreted as a four-dimensional body, divided into an array of arbitrarily sized hyperblocks and compressed by a method that extends the JPEG into four dimensions. Optionally, each hyperblock is predicted by an optimal direction vector from previously encoded hyperblocks. A four-dimensional discrete cosine transform is applied to these hyperblocks, and the resulting coefficients are quantized to the desired quality. Unlike the original JPEGs Huffman encoder, quantized coefficients are encoded using a context-adaptive arithmetic encoder. A discrete cosine transform can make good use of sample similarity within a single block. Therefore, the method is expected to be efficient for light fields with a strong similarity between adjacent views. This is also the reason why disparity search and mutual shift are carried out in the first step. Note that here we use our own software implementation available under the terms of the BSD license.[1] More technical information can be found in [9].

Because our work deals with the comparison of compression methods for light field data, we consider it appropriate to briefly summarize the competitive work here. In recent years, several papers compared and evaluated the compression performance of video codecs on light field imagery. The authors of [16] evaluated the performance of the main image coding standards, H.264, and H.265 format. They however compressed individual views independently (using the intra profile). The H.265 proved to be the most efficient compression method. In [17], the authors compared the compression performance of three strategies using the H.265 (the lenslet image, light field views as a pseudo-temporal sequence, subset of lenslet images). Their results show that coding the 4D light field leads to better performance when compared to coding lenslet images. The method in [18] decomposes the 4D light field into homography parameters and residual matrix. The matrix is then factored as the product of a matrix containing several basis vectors and a smaller matrix of coefficients. The basis vectors are then encoded using the H.265. In [19, 20], the authors propose a hierarchical coding structure for light fields. The 4D light field is decomposed into multiple views, organized into a coding structure according to the spatial coordinates, and the views are then encoded hierarchically. The scheme is implemented in the reference H.265 software. The authors in [21] propose a scheme that splits the 4D light field into several central views and remaining adjacent views. The adjacent views are subtracted from the central views, and both groups are then encoded using H.265 coder. Finally, the authors of [22, 23] feed the 4D light field into the H.265 exploiting the inter prediction mode for individual views.

Also, a lot of work dealing with the comparison of video compression methods can be found in the literature. For example, in [12], the authors compared the coding efficiency of the AV1, H.265, and VP9 formats. They found that both AV1 and H.265 significantly overcome the VP9. The compression performance of AV1 was slightly below H.265 on average. The authors, however, judged this difference quite insignificant and highly dependent on the contents used in tests. The authors of [24] compared the compression performance of VP9, AV1, H.265, and an early version of VVC (JEM software). They observed that compression efficiency has been improved significantly for AV1 and VVC over their respective predecessors. They also observed significant bitrate overhead of AV1 relative to VVC. Finally, the paper [25] compares the performance of three major contemporary video codecs: H.265, AV1, and VVC, based on both objective and subjective assessments. The authors found that H.265 and AV1 are not significantly different in terms of perceived quality at the same bit rates. The VVC was, however, performing significantly better than H.265 and AV1.

De Carvalho et al. [26] proposed a coding scheme based on exploiting the 4D redundancy of light fields by using a 4D transform and hexadeca-trees. It divides

[1] `https://github.com/xdlaba02/light-field-image-format`

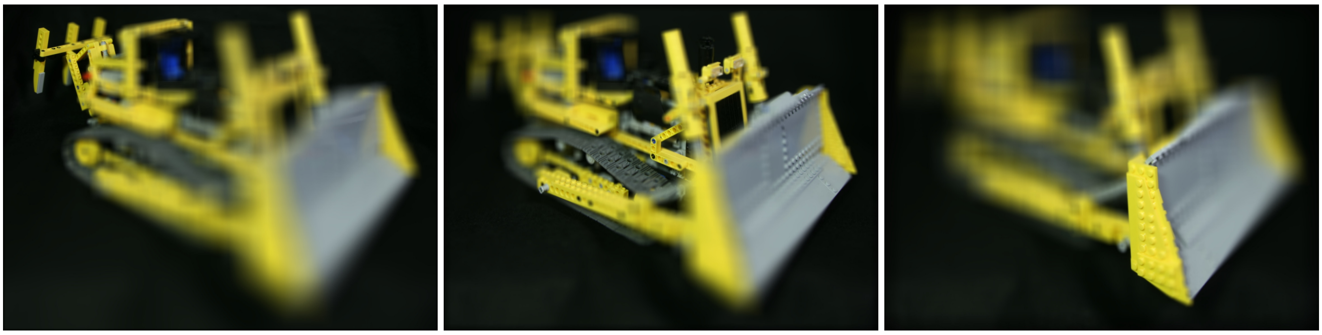


Fig. 2: Lego Bulldozer. Views rendered from the 4D light field for three focal planes.

the light field into 4D blocks and computes a 4D DCT of each one. Then the transform coefficients of the 4D block are grouped using hexadeca-trees and encoded using an adaptive arithmetic coder. This procedure was also adopted by JPEG PLENO [27]. The JPEG Committee also provides a publicly available EPFL Light-field database, which subset was also used for comparison in this paper.

3 Evaluation

Here we introduce the dataset and methodology used for the comparison, followed by the actual comparison and accompanied by a brief discussion.

Our dataset consists of nine 4D light fields based on all three types of capturing devices. The first four light fields were captured using Lytro Illum B01 plenoptic camera, another two using conventional moving camera (using simple motorized gantry and Canon Digital Rebel XTi camera). The other two were captured using 8×8 multi-camera array (grid), and the last one is 8×8 synthetic light field rendered on a computer. Corresponding resolutions and adjacent image disparity ranges are listed in Table 1. For convenience, the central view for each light field is also shown in Figure 1. The Classroom light field has been rendered in Blender.

To make our research reproducible, we provide more information about datasets here. The 4D light fields coming from the EPFL Light-field data set are provided directly by the JPEG committee.[2] The images are provided for research purposes. Light fields provided by the Stanford Computer Graphics Laboratory are freely available on their website in the form of original as well as rectified and cropped images.[3] We used the rectified form in our comparison since it seems more suitable for our 4D compression method. The university does not state any license terms. Light fields from the Saarland University are available on the SAUCE project website.[4] This dataset is intended for research and education purposes only. All images were further cropped to multiples of 8×8 pixels and converted to the color depth of 8 bits per pixel. This is necessary due to the limitations of some video codecs.

We evaluate the impact of compression methods on the quality of refocused images and point clouds reconstructed from 4D light field data.

The refocus of the 4D light fields at the virtual focal plane is achieved using the shift-sum algorithm [28]. Figure 2 helps to better understand the situation. This algorithm shifts the views according to the camera baseline with respect to the reference view and accumulates the corresponding pixel values. The refocused image is thus an average of transformed views. The computation of each pixel value of the refocused image is given by the distance of the synthetic plane from the main lens. We perform interpolation in the last two of the four dimensions to convert the sampled light field function into a continuous one. In order to reduce the block artifacts in the refocused image caused by a large camera baseline, we employ frame-interpolation algorithm to compute intermediate frames. The intermediate frames, computed from interpolation of the dense optical flow [29], are subsequently added to the resulting refocused image, increasing the spatial resolution of the LF and smoothing the artifacts.

The 3D pointclouds are computed using modified incremental Structure from Motion (SfM) pipeline [30], constrained by the known configuration of LF camera array or gantry such as the inherent grid structure, camera calibration parameters and baseline. The first step of SfM algorithm extracts visual features and descriptors in images from all cameras, and subsequently, the matches between images are estimated exploiting epipolar geometry of the scene. We opted for SIFT [31] features and

[2] `https://jpeg.org/jpegpleno/`
[3] `http://lightfield.stanford.edu/`
[4] `https://www.sauceproject.eu/`

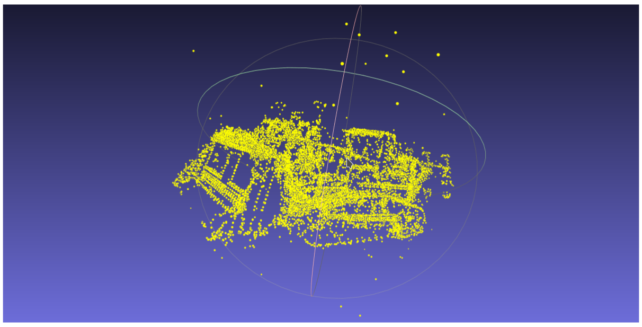


Fig. 3: Lego Bulldozer. Point cloud reconstructed from the original 4D light field.

descriptors because they produced consistent matches even in scenes with little structure such as Take2_1.

The incremental 3D reconstruction starts with a pair of cameras with the most valid corresponding 2D points and estimates the relative poses of the cameras. Additional poses of cameras are computed sequentially, in the order given by the number of corresponding 2D points between new camera and camera with an already known pose. After each added camera input, non-linear optimization is applied to the system consisting of cameras, 3D points and 2D measurements to minimize the reprojection error, refine 3D structure and camera parameters and to detect outlying measurements and matches. We utilize a non-linear graph optimization library SLAM++ [32] to perform the optimization task. SLAM++ is a very efficient implementation of several non-linear least squares solvers, based on sparse block matrix manipulation for solving linear problems. This library allows the implementation of custom edges, which is used to constrain the camera array positions to a grid, and robust edge implementation applied to detect outliers and improve the accuracy of the 3D structure.

To investigate the performance of the evaluated methods, we measure a distortion of reconstructed point clouds with respect to the bitrate of their source images. The distortion is evaluated as a difference from the point cloud reconstructed from a ground-truth data. In each data set, the ground truth point cloud is registered to distorted ones using the iterative closes point (ICP) [33] algorithm. To avoid the influence of outlier vertices on the registration, one percent of points with the highest distances are omitted. The distortion is evaluated as a root mean square error (RMSE) between the ground-truth and their closest vertices, measured in meters.

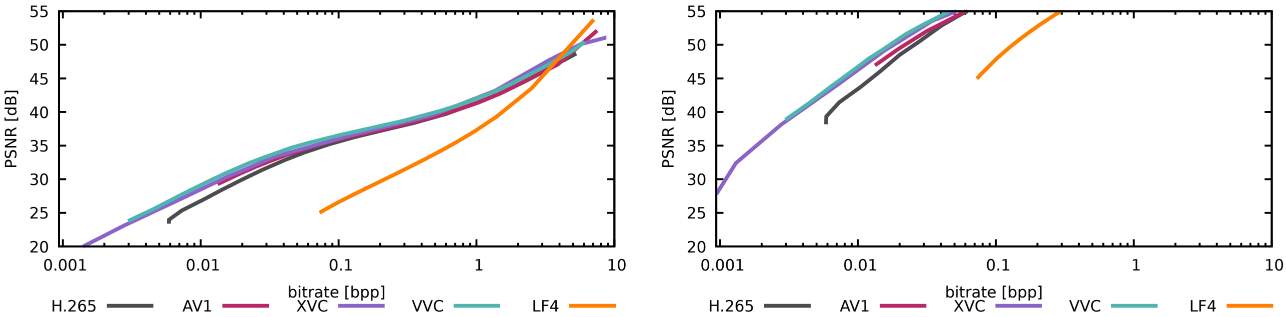
In the beginning, we wondered whether it was really necessary to compare the image quality on views rendered for multiple focal planes rather than the original 4D light field. The experiment in Figure 4 reveals that a huge difference can be observed between the former and the latter. This difference is about 10 to 20 dB in the PSNR scale. This can be explained by the fact that any pixel in the rendered view is a sum of pixels from the original 4D LF. The sum all together suppresses compression artifacts. This leads us to the conclusion that we can afford to compress the 4D light fields much more than independent images while maintaining the same visual quality of a screened picture. Furthermore, considering the rendered views, we can also notice a failure of all formats except the XVC. The issue is that the formats are unable to cover lower bitrates, ca. below 0.01 bpp, and consequently, they cannot handle quality below ca. 40 dB.

Note that meaningful values for the PSNR are between 30 and 50 dB, provided the bit depth is 8 bits. The mean squared error (MSE) value 1 leads to ca. 48.1 dB. So anything above this limit only improves the fractions of the least significant bit plane on average. For this reason, we have limited the y-axis in our graphs to the 20–55 dB interval.

The rest of the paper deals with the question, "What is the best compression method for 4D light field data?" As a side problem, we also deal with the question of whether it is better to compress the 4D light fields as a sequence of 2D frames, or as a four-dimensional body. To answer these questions, we compressed the original light field using different compression formats, and then assessed compression performance on both (1) refocused images rendered from the distorted light fields and (2) point clouds reconstructed from the same data. The results of these two measurements are summarized in Figures 5 and 6.

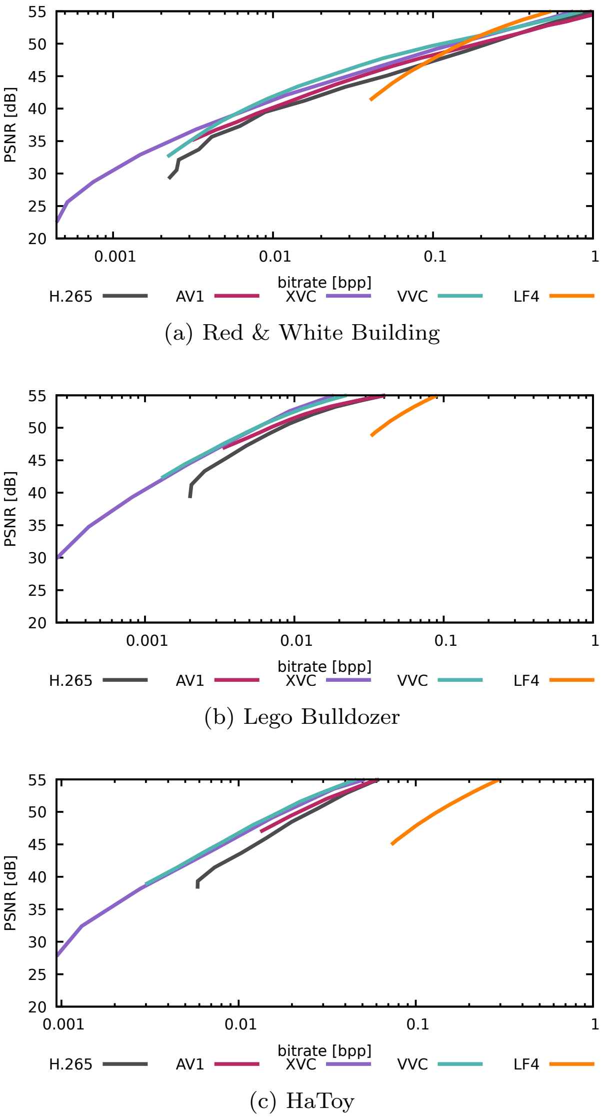
First, we will focus on the first figure. It visualizes dependencies of the PSNR on the bitrate. Due to a lack of space, only three representative measurements are shown. However, we have obtained similar results also for other measurements. The first thing we can notice is that only the XVC was able to cover the lowest bitrates and qualities. For most bitrates, the XVC is dominated by the VVC. However, this difference is very small. It should be noted that the VVC format has not yet been finalized and improvements can be expected with the advent of high-quality encoders. Maybe a little surprising, H.265 exhibits consistently the worst performance. Finally, except for the highest bitrates on Lytro light fields, the four-dimensional compression method (LF4) failed on all data and all bitrates.

Now, we will focus on Figure 6. The distortion rises with lower bitrate, which is caused by higher noise and reduced number of vertices present in the point clouds. Although the trend of increasing distortion with lower bitrate is clear, slight oscillations can be observed. These



(a) HaToy, 4D (b) HaToy, rendered 2D

Fig. 4: Comparison of the image quality (PSNR/bitrate) computed between (a) original 4D bodies and (b) 2D views rendered from 4D LF for multiple focal planes (average for 10 focal planes).

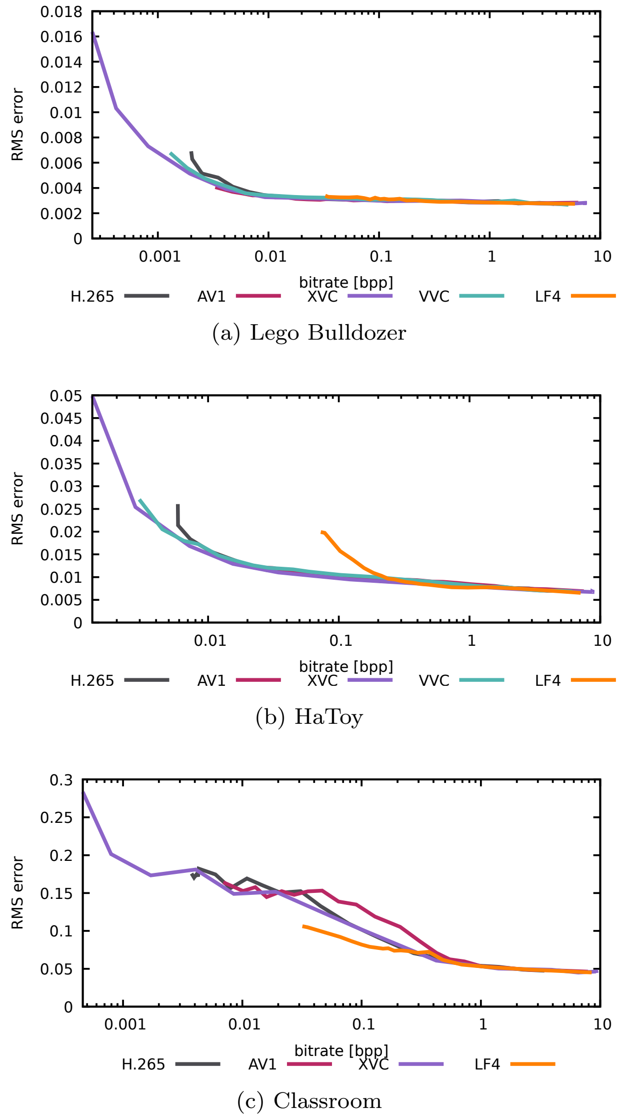


(a) Red & White Building

(b) Lego Bulldozer

(c) HaToy

Fig. 5: Comparison of different compression formats on views rendered for multiple focal planes.



(a) Lego Bulldozer

(b) HaToy

(c) Classroom

Fig. 6: Comparison of different compression formats on 3D point clouds reconstructed from 4D light fields.

occurs due to a rather complex process of the point cloud reconstruction in combination with the closest point registration. Apart from these oscillations, the findings described in the previous paragraph are still valid.

Finally, let us note that the findings presented in this article are consistent with the findings of our previous research in [9], and also agree with the results of comparisons of video codecs published elsewhere [12, 24, 25]. Although these results never evaluated the XVC codec.

4 Conclusions

Our paper compared several methods for lossy compression of four-dimensional light fields. It turned out that H.265 has already been overcome in any case. Currently, the best results are achieved by VVC and XVC. The VVC was finalized in 2020. Making this happen will be the best option for compression of the 4D light field imagery. Unfortunately, the VVC and XVC still lack broad software support. In future work, it would be worthwhile to make a comparison with more advanced VVC encoders. Although the AV1 is a state-of-the-art compression format, it was always dominated by XVC and VVC in our experiments. But expanding software support speaks in its favor.

Compressing light field data as a four-dimensional body did not prove viable, mainly due to too small similarity between adjacent views. Eventually, it turned out that light fields can be compressed much more than independent images while maintaining the same perceived visual quality.

Acknowledgements This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (LQ1602), the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737475, and the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 780470.

References

1. Adelson EH, Bergen JR (1991) The plenoptic function and the elements of early vision. In: Landy M, Movshon JA (eds) Computational models of visual processing, MIT Press, pp 3–20
2. Camahort E, Lerios A, Fussell D (1998) Uniformly sampled light fields. In: Rendering Techniques' 98, Springer, pp 117–130
3. Camahort E, Fussell D (1999) A geometric study of light field representations. Tech. rep., The University of Texas at Austin
4. Levoy M, Hanrahan P (1996) Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, SIGGRAPH '96, pp 31–42, DOI 10.1145/237170.237199
5. Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF (1996) The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, Association for Computing Machinery, SIGGRAPH '96, pp 43–54, DOI 10.1145/237170.237200
6. McGuire M, Mara M, Nowrouzezahrai D, Luebke D (2017) Real-time global illumination using precomputed light field probes. In: Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Association for Computing Machinery, New York, NY, USA, I3D '17, DOI 10.1145/3023368.3023378
7. Trottnow J, Spielmann S, Lange T, Chelli K, Solony M, Smrz P, Zemcik P, Aenchbacher W, Grogan M, Alain M, et al (2019) The potential of light fields in media productions. In: SIGGRAPH Asia 2019 Technical Briefs, pp 71–74
8. Wu G, Masia B, Jarabo A, Zhang Y, Wang L, Dai Q, Chai T, Liu Y (2017) Light field image processing: An overview. IEEE Journal of Selected Topics in Signal Processing 11(7):926–954
9. Barina D, Chlubna T, Solony M, Dlabaja D, Zemcik P (2019) Evaluation of 4D light field compression methods. In: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Part I, Computer Science Research Notes (CSRN), vol 2901, pp 55–61, DOI 10.24132/CSRN.2019.2901.1.7
10. Grois D, Marpe D, Mulayoff A, Itzhaky B, Hadar O (2013) Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders. In: Picture Coding Symposium (PCS), pp 394–397, DOI 10.1109/PCS.2013.6737766
11. Grois D, Nguyen T, Marpe D (2016) Coding efficiency comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC encoders. In: Picture Coding Symposium (PCS), pp 1–5, DOI 10.1109/PCS.2016.7906321
12. Akyazi P, Ebrahimi T (2018) Comparison of compression efficiency between HEVC/H.265, VP9 and AV1 based on subjective quality assessments. In: Tenth International Conference on Quality of Multimedia Experience (QoMEX), pp 1–6, DOI 10.1109/QoMEX.2018.8463294

13. Samuelsson J (2019) The XVC video codec – a revolutionary software-defined video compression format. SMPTE Motion Imaging Journal 128(10):1–8, DOI 10.5594/JMI.2019.2937737
14. Chen J, Karczewicz M, Huang YW, Choi K, Ohm JR, Sullivan GJ (2019) The joint exploration model (JEM) for video compression with capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology DOI 10.1109/TCSVT.2019.2945830
15. Panayides AS, Pattichis MS, Pantziaris M, Constantinides AG, Pattichis CS (2020) The battle of the video codecs in the healthcare domain – a comparative performance evaluation study leveraging VVC and AV1. IEEE Access 8:11,469–11,481, DOI 10.1109/ACCESS.2020.2965325
16. Alves G, Pereira F, da Silva EA (2016) Light field imaging coding: Performance assessment methodology and standards benchmarking. In: IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp 1–6, DOI 10.1109/ICMEW.2016.7574774
17. Viola I, Rerabek M, Ebrahimi T (2017) Comparison and evaluation of light field image coding approaches. IEEE Journal of Selected Topics in Signal Processing 11(7):1092–1106, DOI 10.1109/JSTSP.2017.2740167
18. Jiang X, Pendu ML, Farrugia RA, Guillemot C (2017) Light field compression with homography-based low-rank approximation. IEEE Journal of Selected Topics in Signal Processing 11(7):1132–1145, DOI 10.1109/JSTSP.2017.2747078
19. Li L, Li Z, Li B, Liu D, Li H (2017) Pseudo sequence based 2-D hierarchical coding structure for light-field image compression. In: Data Compression Conference (DCC), pp 131–140, DOI 10.1109/DCC.2017.10
20. Li L, Li Z, Li B, Liu D, Li H (2017) Pseudo-sequence-based 2-D hierarchical coding structure for light-field image compression. IEEE Journal of Selected Topics in Signal Processing 11(7):1107–1119, DOI 10.1109/JSTSP.2017.2725198
21. Han H, Xin J, Dai Q (2018) Plenoptic image compression via simplified subaperture projection. In: Advances in Multimedia Information Processing (PCM), pp 274–284
22. Li Y, Sjostrom M, Olsson R, Jennehag U (2014) Efficient intra prediction scheme for light field image compression. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 539–543, DOI 10.1109/ICASSP.2014.6853654
23. Li Y, Sjostrom M, Olsson R, Jennehag U (2016) Coding of focused plenoptic contents by displacement intra prediction. IEEE Transactions on Circuits and Systems for Video Technology 26(7):1308–1319, DOI 10.1109/TCSVT.2015.2450333
24. Nguyen T, Marpe D (2018) Future video coding technologies: A performance evaluation of av1, jem, vp9, and hm. In: 2018 Picture Coding Symposium (PCS), pp 31–35
25. Zhang F, Katsenou AV, Afonso M, Dimitrov G, Bull DR (2020) Comparing vvc, hevc and av1 using objective and subjective assessments. ArXiv:2003.10282 (eess.IV)
26. de Carvalho MB, Pereira MP, Alves G, B da Silva EA, Pagliari CL, Pereira F, Testoni V (2018) A 4d dct-based lenslet light field codec. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp 435–439, DOI 10.1109/ICIP.2018.8451684
27. Astola P, da Silva Cruz LA, da Silva EAB, Ebrahimi T, Freitas PG, Gilles A, Oh KJ, Pagliari C, Pereira F, Perra C, Perry S, Pinheiro AMG, Schelkens P, Seidel I, Tabus I (2020) Jpeg pleno: Standardizing a coding framework and tools for plenoptic imaging modalities. ITU Journal: ICT Discoveries, Special issue on The Future of Video and Immersive Media 3(1)
28. Ng R, Levoy M, Bredif M, Duval G, Horowitz M, Hanrahan P (2005) Light field photography with a hand-held plenoptic camera. Tech. rep., Stanford, stanford Tech Report CTSR 2005-02
29. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: Pajdla T, Matas J (eds) Computer Vision – ECCV 2004, Springer Berlin Heidelberg, pp 25–36
30. Snavely N, Seitz SM, Szeliski R (2006) Photo tourism: Exploring photo collections in 3D. ACM Transactions on Graphics 25(3):835–846, DOI 10.1145/1141911.1141964
31. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110, DOI 10.1023/B:VISI.0000029664.99615.94
32. Ila SV, Polok L, Solony M, Svoboda P (2017) SLAM++ – a highly efficient and temporally scalable incremental SLAM framework. The International Journal of Robotics Research 2017(1):210–230, DOI 10.1177/0278364917691110
33. Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2):239–256, DOI 10.1109/34.121791