applied sciences

MDPI

Article

On Alpha-Expansion-Based Graph-Cut Optimization for Decoder-Side Depth Estimation

Dawid Mieloch *, Dominika Klóska and Olgierd Stankiewicz

Institute of Multimedia Telecommunications, Poznan University of Technology, 60-965 Poznań, Poland; dominika.kloska@put.poznan.pl (D.K.); olgierd.stankiewicz@put.poznan.pl (O.S.)
* Correspondence: dawid.mieloch@put.poznan.pl

Abstract: In order to achieve high realism an acceptable level of user experience in immersive videos, it is crucial to provide both the best possible quality of depth maps and minimize computational time. In this paper, we propose a new approach to the decoder-side depth estimation (DSDE) problem, which uses the hierarchical alpha-expansion algorithm with additional improvements for the estimation designed to be more robust to compressed video artifacts and limited computational resources. As shown by the experimental results, the proposal simultaneously results in reduction of computation time of the estimation process (by almost 40%) and an improvement of quality of estimated depth maps. The increased quality is demonstrated by more than 6% Bjøntegaard delta gain compared to the Moving Picture Experts Group (MPEG) immersive video state-of-the-art DSDE method.

Keywords: immersive video; decoder-side depth estimation; graph-cut

Citation: Mieloch, D.; Klóska, D.; Stankiewicz, O. On Alpha-Expansion-Based Graph-Cut Optimization for Decoder-Side Depth Estimation. *Appl. Sci.* **2024**, *14*, 5768. https://doi.org/10.3390/app14135768

Academic Editor: Vladimir A. Golovko

Received: 13 June 2024
Revised: 27 June 2024
Accepted: 28 June 2024
Published: 1 July 2024

1. Introduction

The evolution of immersive video has dramatically enhanced how users interact with digital content, as new immersive experiences characterized by high levels of realism and interactivity are presented [1]. In complex immersive video systems, it is necessary to provide 3D representation of a scene. This is a critical component in achieving realism, as it is required to render virtual views requested by the viewer [2]. It can be acquired, for instance, using the depth estimation process or depth cameras [3].

When a larger number of depth maps is sent, the bitrate required to preserve proper geometry becomes too high, especially for low bitrates (as it can constitute more than 50% of the whole bitstream [4]). The use of a dedicated codec for depth maps provides satisfactory results, but such codecs are not standardized and are not sufficiently widespread to be used in practical cases. To address this, the current state-of-the-art ISO/IEC 23090-12 Moving Picture Experts Group (MPEG) Immersive Video (MIV) coding standard [2] is codec-agnostic; i.e., it assumes the use of the same internal video compression to encode texture and geometry.

Compressing depth maps with traditional video encoding methods is also challenging due to the unique statistical properties of depth maps, which differ significantly from natural video content [4]. Such traditional video encoding methods struggle to preserve sharp edges and precise depth information critical for rendering accurate 3D scenes. It can lead to potential inaccuracies and a diminished user experience in immersive applications [5].

To overcome these challenges, decoder-side depth estimation has been proposed [6] as one of the types of immersive video coding in which depth information must be reconstructed from compressed video data. In such an approach, the primary challenge lies in dealing with compression artifacts and ensuring that depth estimation algorithms operate effectively within the constraints of minimizing the complexity of the decoding process. Compression-induced artifacts can impact the quality of the video and, as a result, depth

estimation accuracy. These errors include the displacement of object edges (which can lead to inaccuracies in spatial relationships between views) and minimized levels of detail (which reduces the texture and nuances within the scene, making it challenging to achieve precise depth estimation) [7].

In the incoming second edition of MIV [8], the depth estimation step is assumed to simultaneously enhance the quality of a subset of transmitted depth maps as well as the quality of depth maps estimated for remaining views [9]. This further increases the importance of providing faster and more reliable depth estimation methods for immersive video. Unfortunately, conventional depth estimation methods often struggle to provide high quality in real time, while learning-based methods can be used only for content captured with a linear arrangement of cameras [4]. Even for such content, studies show that the quality of depth estimated with learning-based methods for immersive video is lower than for conventional depth estimators [10].

Graph-cut optimization techniques, particularly the alpha-expansion algorithm, have shown promise in various computer vision tasks because of their ability to solve pixel labeling problems efficiently [11]. Graph-cut-based techniques constitute the basis for the two last reference depth estimation methods of MPEG video coding (i.e., Depth Estimation Reference Software—DERS [12] and Immersive Video Depth Estimation—IVDE [4]). Unfortunately, this step has not been fully optimized for decoder-side depth estimation, where specific challenges such as compressed video artifacts and limited computational resources must be addressed. Adapting these techniques to handle the distortions introduced by compression is crucial for improving depth estimation accuracy in immersive video.

The abovementioned challenges motivated our research presented in this paper. To provide an initial speed-up of the depth estimation for the immersive video, we have improved IVDE with the use of hierarchical alpha-expansion [13] for faster optimization using graph-cuts. Unfortunately, as presented in the results of experiments shown in Section 4.2, estimation is faster at the expense of a slight decrease of depth map quality. Therefore, using the findings of the previous research on main problems with depth estimation for immersive video [14,15] and evaluations performed by MPEG Video Coding experts [4], we introduce two further novel improvements to the hierarchical alpha-expansion method:

- The second cycle label offset introduces a change in the starting point for the second iteration cycle, effectively narrowing the search space for depth values. This scheme is designed to leverage the insights gained from the first cycle of graph-cut, effectively optimizing the selection of depth values for subsequent analysis.
- By including depth values of adjacent segments into optimization, neighboring segments label examination counteracts the boundary-blurring effects of video compression.

These novel proposals are detailed in Section 3.

2. State of the Art in Alpha-Expansion-Based Graph-Cut Optimization

Depth estimation is often based on minimizing a cost function [11]. Typically, the cost function can be simplified to

$$E\left(\overline{d}_p\right) = \sum_{p\in \mathrm{P}} D_p\left(\overline{d}_p\right) + \sum_{p\in \mathrm{P}} \sum_{q\in \mathrm{Q}} V_{p,q}\left(\overline{d}_p, \overline{d}_q\right), \quad (1)$$

where P is the set of all points of the input view, p is a point of the input view, $\overline{d}_p$ is the currently considered depth of a point p, D_p is the data term that represents the cost of assigning the depth $\overline{d}_p$ to the point p, Q is the set of points in the neighborhood of the point p, q is a point in the neighborhood of the point p, $\overline{d}_q$ is the currently considered depth of a point q, and $V_{p,q}$ is the smoothness term that represents the intra-view discontinuity cost of assignment of the depth d_p to the point p and depth d_q to the point q.

The data term D_p establishes a correspondence between neighboring views, typically by calculating the sum of absolute differences between color values of point p and its

corresponding point in a neighboring view for the considered depth $\overline{d}_p$. This metric can be replaced or computed within a small window to minimize noise influence [16].

The smoothness term $V_{p,q}$ deals with surfaces lacking texture. Its discontinuity model is based on depth similarity:

$$V_{p,q}\left(\overline{d}_p, \overline{d}_q\right) = \beta_0 \cdot \left|\overline{d}_p - \overline{d}_q\right|, \quad (2)$$

where β_0 is the smoothing coefficient provided by the user or calculated from camera parameters [4], $\overline{d}_p$ is the currently considered depth of a point p, and $\overline{d}_q$ is the currently considered depth of a point q. This formulation aims to estimate smooth depth on scene objects. The β_0 coefficient sometimes is also dependent on the similarity between neighboring points p and q [17]. Nevertheless, this way, $V_{p,q}$ becomes prone to the effects of compression of input data, as blurred edges in processed video can negatively influence the accuracy of the performed optimization.

Additional terms can be included in (1) to ensure inter-view [11,15] and temporal [18] consistencies. A final solution for $E\left(\overline{d}_p\right)$, i.e., the assignment of one of available depth values to all points of input views, can be estimated using a relevant optimization method. One of the most common among these are graph cut-based methods [19], commonly used for binary problem optimization in image processing. Each image point becomes a node in a graph, with edges representing the cost function. The algorithm finds the optimal cut, which assigns nodes to labels in a way that minimizes the cost function.

For multi-label segmentation, solutions are obtained through a series of consecutive optimizations. As can be seen above, the depth estimation process requires multi-label optimization, where each depth level is represented as a label.

The two primary methods for minimizing multi-label problems are the alpha–beta swap and alpha-expansion [19]. In the alpha–beta swap, the graph cut algorithm iterates through all pairs of labels α and β, resulting in a large number of optimizations $(size(d)^2)$ required. In the alpha-expansion method, during each iteration, the graph cut algorithm assigns a label α or retains the current label (referred to as a non-α label). For instance, if points are initially labeled with label '0' in the first iteration, each point in the first optimization is assigned either label '0' or another label '1'. Subsequently, points are assigned to further labels or retain the previous labeling (i.e., '0' or '1'). This method requires only $size(d)$ iterations.

An example of alpha-expansion-based optimization performed for 8 depth levels is presented in Figure 1. At the top, the results of each optimization can be seen. The result of the last optimization (for 7th depth level), enlarged at the bottom, is the final depth map. After all the iterations are performed, the process of optimization can be performed again, in the latter cycle of iterations [14], so the 'final depth map' (Figure 1) becomes the input to another optimization.

Not all nodes have to be a part of each iteration. They can be turned off in chosen iterations in order to provide temporal consistency [14], or if some input depth was provided into the optimization and it should be unchanged during the estimation. Still, these nodes can influence the depth assigned to other nodes. Therefore, the cost function is

$$E\left(\overline{d}_p\right) = \sum_{p \in P_{on}(\overline{d}_p)} D_p\left(\overline{d}_p\right) + \sum_{p \in P_{on}(\overline{d}_p)} \sum_{q \in Q \cap P_{on}(\overline{d}_p)} V_{p,q}\left(\overline{d}_p, \overline{d}_q\right), \quad (3)$$

where $P_{\text{on}}\left(\overline{d}_p\right)$ is the set of active nodes.

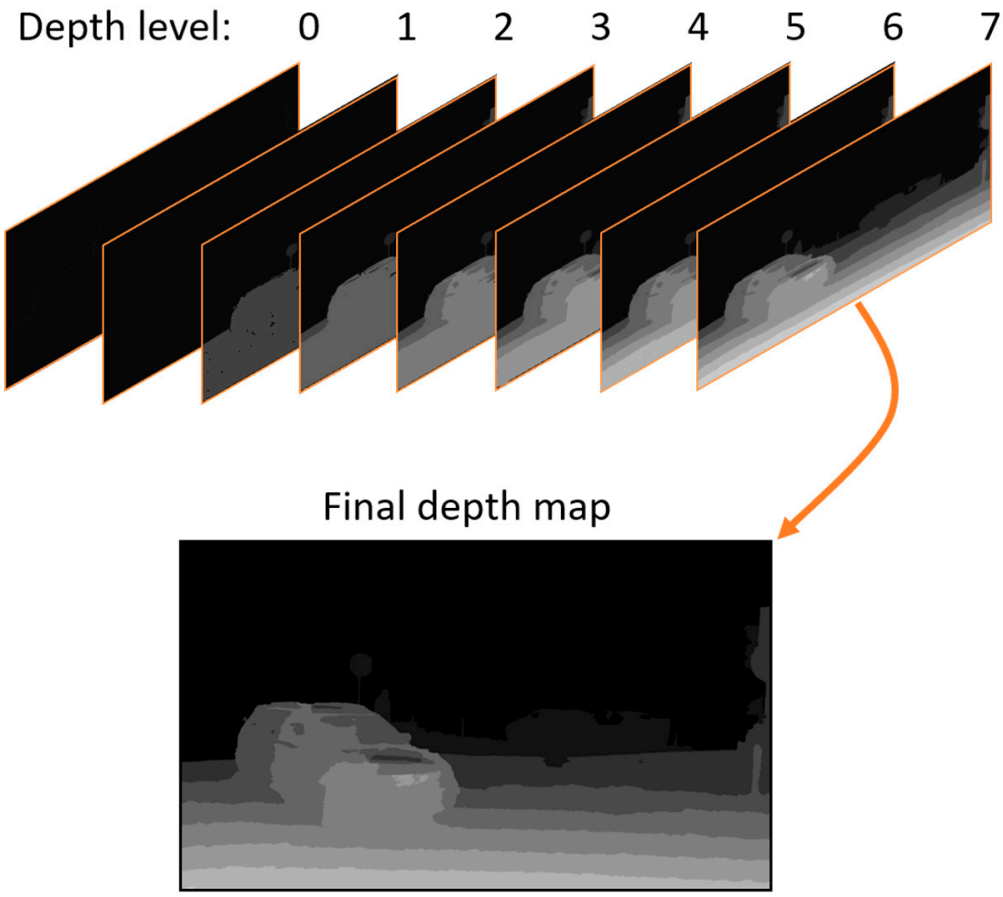

Figure 1. An example of alpha-expansion-based depth estimation.

Performing the first, coarse optimization leads to an initial, low-detail depth map if used for geometry estimation (Figure 2A). This rough estimation is the start of the next cycle of iterations in which the number of labels (depth levels) required to be examined for each segment is reduced from $size(d)$ to only $\pm n/2$ labels (Figure 2B). Therefore, the number of pixels active in these iterations is significantly decreased, and their set is

$$P_{on}\left(\overline{d}_p\right) = \left\{r : \left|\overline{d}_p - r\right| < n\right\}. \quad (4)$$

By leveraging the result of coarse optimization, the algorithm focuses only on the refinement of previous labeling, reducing unnecessary computations and potentially reaching better solutions. Therefore, this algorithm decreases the complexity of optimization twofold. Firstly, it limits the number of labels to be analyzed in the first cycle, and secondly, it optimizes smaller graphs in the second cycle.

As one of the reviews has shown [20], this scheme provides a satisfactory trade-off between the computational complexity and the resulting quality of depth maps when compared to other global optimization-based methods. Unfortunately, as our initial experiments have shown (Section 4.2), this scheme results in slight reduction accuracy of estimated depth maps.

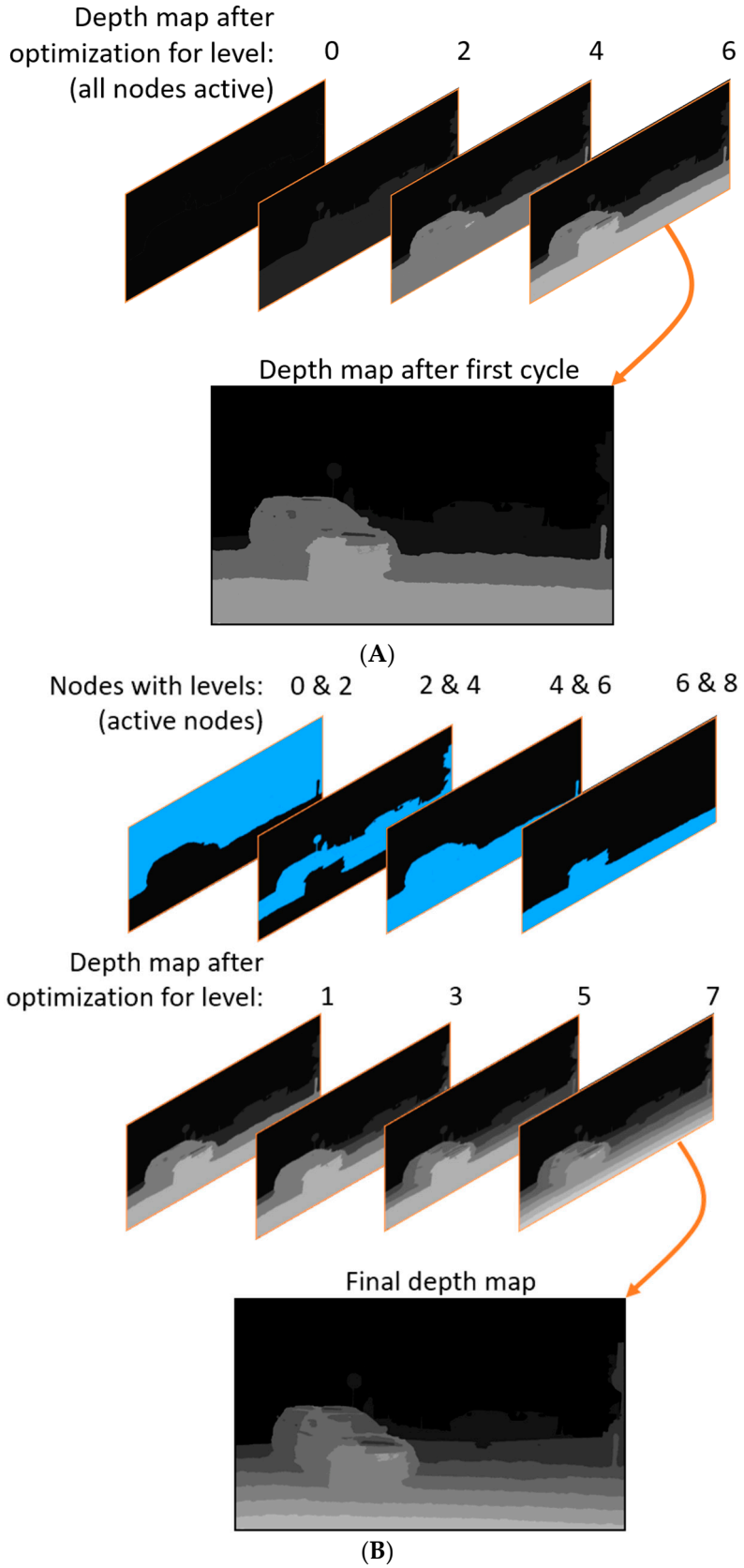

Figure 2. An example of hierarchical alpha-expansion-based depth estimation. The currently active nodes are shown as the blue area. (**A**) First cycle of graph cut (coarse optimization). (**B**) Second cycle of graph cut (refinement optimization).

3. The Proposed Method

This paper presents the results of the first implementation of hierarchical alpha-expansion [13] for depth estimation performed for immersive video. Moreover, two novel further improvements of this scheme are proposed, to provide even faster estimation and improve the quality of depth maps in the demanding task of decoder-side depth estimation.

3.1. Second-Cycle Label Offset

As shown in other works [14], at least two global cycles of graph cut optimization are required to provide satisfactory quality of estimated depth maps. The second cycle removes a noticeable number of erroneous values, especially in areas which were occluded in some views. Fortunately, the increase of quality when the third cycle is used is negligible [14], so the further increase of optimization time is not required. The necessity of performing a second cycle was also proven with subjective quality measurements performed by MPEG Video Coding experts. It led to using two cycles in the evaluations of the DSDE compression scheme in coding performed by MIV. Therefore, in a hierarchical alpha-expansion approach used in depth estimation for immersive video, it is also necessary to perform two coarse global cycles for every n-th of available depth values (so the cycle shown in Figure 2A is performed twice). Then, refinement iterations examine $\pm n/2$ of adjacent depth values to the values acquired from the first two cycles. Such a hierarchical scheme is still less complex than the basic alpha-expansion. Nevertheless, the additional cycle requires a significant amount of additional time for the depth estimation, which is already the most time-consuming part of immersive video decoding.

In our proposal, assuming that coarse cycles are done for every n-th of available depth values, we offset by $n/2$ the initial depth value from which the second coarse global cycle is being performed. For instance, if n would be equal to 4, then the second cycle would start from depth level '2', instead of '0'. Therefore, $\bar{d}_p$ in the cycle c is a set defined as

$$\bar{d}_p \in \begin{cases} \{nk : k \in \mathbb{N} \bigwedge nk < N\} & if \mathrm{c} = 0 \\ \{nk + n/2 : k \in \mathbb{N} \bigwedge (nk + n/2) < N\} & if \mathrm{c} = 1 \end{cases} \quad (5)$$

With the proposed second-cycle label offset, the time required to estimate depth is noticeably decreased. If n is equal to 4, then the number of depth levels to be examined is the same as it would be for n equal to 2 in traditional approach (as every second depth label was already examined in two first cycles). Moreover, for $n = 2$ there is no need to perform any refinement iterations, as all depth values were already examined for all segments after two coarse iterations.

Using the offset which is a result of dividing n by odd values (e.g., offset by $n/3$) would result in non-optimal depth values to be checked in the second coarse cycle, as these values would not be evenly distributed between values for the first cycle. It would negatively influence the refinement cycle, so the values to be checked would not be $\pm n/2$. For example, for $n = 4$, $\{-1, 3\}$, the largest distance from the initial depth would increase from 2 to 3. The larger this distance is, the worse the results of the optimization.

In addition, as our results have shown, increasing n negatively influences the quality of estimated depth maps (Section 4.2). Therefore, this proposal is beneficial both in terms of the quality and time of estimation, as only a small number of depth values is examined in refinement cycles. Naturally, these considerations are true only if we assume that the first two cycles are performed for all depth levels, as was explained at the beginning of this section.

3.2. Neighboring Segments Label Examination

One of the main issues observed in depth maps, which heavily influences the perceived quality of final virtual view synthesis, is the displacement of the object boundaries [5]. This issue is addressed in IVDE by using superpixels [15] in graph-cut optimization instead of pixels. Superpixels tend to follow the border of objects in encoder-side depth estimation [21].

Nevertheless, video compression can decrease their sharpness in decoder-side schemes, making it harder to estimate the proper shape of objects. Depth errors produced at this stage can propagate throughout the entire sequence, e.g., due to temporal consistency improvements based on utilizing depth values from previous frames [14]. It significantly reduces the final quality of virtual views presented to a viewer.

Let us assume that as a consequence of a too-large value of n in the first two coarse graph-cut cycles, a segment on the edge of the object was incorrectly labeled as having the depth of the foreground. If the unmodified hierarchical alpha-expansion is used, the error will be hard to correct in the optimization cycles, as for this segment, only the closest values of depth will be examined, so the values that represent background will not be proposed. On the other hand, there is a high probability that, among the neighboring segments, we can find depth values closer to correct one. Therefore, we propose to also examine depth values from the neighboring segments in graph-cut refinement cycles thusly:

$$P_{on}\left(\overline{d}_p\right) = \left\{r : \left|\overline{d}_p - r\right| < n \bigwedge r = \overline{d}_q\right\}. \quad (6)$$

At the expense of slight increase in the number of depth values to be examined, the proposal can be used to remove some of incorrectly estimated depth values resulting from, among other things, object boundary displacements. The basic idea of using the depth of neighboring segments has already been researched and found to be useful in depth refinement [22], as it was proven to highly increase the final quality of depth maps. In our proposal, we incorporate this idea directly into the optimization itself. Such an approach removes the unnecessary computational overhead required to run additional refinement after the depth estimation. Reducing this overhead aligns with the requirements of immersive video coding based on decoder-side depth estimation.

4. Experimental Results

To evaluate the proposal, we compare it with state-of-the-art decoder-side depth estimation encoding. In total, six different configurations were tested:

(1) basic alpha-expansion,
(2) hierarchical alpha-expansion with optimization step $n = 2$ (current state of the art and reference method in MIV experiments)
(3) as in (3), but with $n = 4$,
(4) proposed hierarchical alpha-expansion with second-cycle label offset with optimization step $n = 2$,
(5) as in (4), but with $n = 4$,
(6) proposed hierarchical alpha-expansion with second-cycle label offset and neighboring segments label examination (with $n = 4$).

4.1. Methodology

We conducted experiments based on the Common Test Conditions (CTC) defined by ISO/IEC MPEG Video Coding [23]. This helps us to ensure a fair comparison between different methods for immersive video coding. To carry out the experiments, we used the TMIV—Test Model for MPEG Immersive Video 16.0 [23]—which is designed to implement the MPEG Immersive Video coding standard. Used test sequences are listed in Table 1.

For depth estimation performed at the decoder side, we used Immersive Video Depth Estimation (IVDE [4]), which is used in MIV experiments conducted by ISO/IEC MPEG Video Coding. We use our modification of publicly available IVDE 8.0. Once depth maps are estimated, the TMIV renderer is utilized to produce virtual views that are positioned in the same position as all views of the test sequences used. To measure the quality of the virtual views in comparison to the real views of the sequences, PSNR and IV-PSNR metrics [16] are utilized. Eventually, BD-rate (Bjøntegaard delta [24]) is computed to assess the percentage change in the bitrate required to attain the same quality for the tested techniques relative to the reference method.

Table 1. List of test sequences used.

Sequence	Source	Type		Resolution	Views
Classroom Video	[25]	Equirectangular projection	Computer generated	4096 × 2048	15
Museum	[26]	Equirectangular projection	Computer generated	2048 × 2048	24
Chess	[27]	Equirectangular projection	Computer generated	2048 × 2048	10
Guitarist	[28]	Equirectangular projection	Computer generated	2048 × 2048	23
Hijack	[29]	Equirectangular projection	Mixed	4096 × 2048	10
Cyberpunk	[29]	Equirectangular projection	Mixed	2048 × 2048	10
Kitchen	[30]	Perspective, planar	Computer generated	1920 × 1080	25
Cadillac	[31]	Perspective, planar	Computer generated	1920 × 1080	15
Mirror	[32]	Perspective, planar	Computer generated	1920 × 1080	15
Fan	[33]	Perspective, planar	Computer generated	1920 × 1080	15
Group	[34]	Perspective, convergent	Computer generated	1920 × 1080	21
Dancing	[35]	Perspective, convergent	Computer generated	1920 × 1080	24
Painter	[36]	Perspective, planar	Natural content	2048 × 1088	16
Breakfast	[37]	Perspective, planar	Natural content	1920 × 1080	15
Barn	[38]	Perspective, planar	Natural content	1920 × 1080	15
Frog	[39]	Perspective, planar	Natural content	1920 × 1080	13
Carpark	[40]	Perspective, planar	Natural content	1920 × 1088	9
Street	[40]	Perspective, planar	Natural content	1920 × 1088	9
Fencing	[41]	Perspective, convergent	Natural content	1920 × 1088	9
CBABasketball	[42]	Perspective, convergent	Natural content	1920 × 1080	34
MartialArts	[43]	Perspective, convergent	Natural content	1920 × 1080	15

The tested methods were also evaluated based on their computational complexity. These evaluations are presented as a runtime ratio when compared to DSDE, which was performed without hierarchical alpha-expansion.

4.2. Results

First of all, we establish the performance of the hierarchical alpha-expansion (reference state-of-the-art method) by comparing it with the basic alpha-expansion. The results shown in Table 2 indicate that, on average, using hierarchical scheme with the optimization step equal to 2 decreases the time of decoding (which includes parsing the bitstream, depth estimation, and virtual view rendering) by almost 20%. On the other hand, the loss in the quality of estimated depth maps results in 5% of BD-rate loss in both reported quality metrices.

When the optimization step is equal to 4, further decrease of runtime can be seen (by another 18%) as well as progressing BD-rate loss (Table 2). This indicates that in immersive video applications, the state-of-the-art hierarchical alpha-expansion used for depth estimation cannot be used to further decrease the complexity of a decoder—the quality losses become too significant to accept.

The proposed second-cycle label offset, as presented in Table 3, overcomes the limitations of the state-of-the-art hierarchical alpha-expansion. If the optimization step is equal to 2, then the runtime is decreased by more than 40% and the quality is only negligibly decreased. For the step equal to 4 (Table 3), the quality loss becomes noticeable and the runtime is longer than in the previous configuration (as the refinement iterations have to be performed in this scheme).

Table 2. State-of-the-art hierarchical alpha-expansion with $n = 2$ (anchor) and $n = 4$, compared to basic alpha-expansion.

Sequence	$n = 2$			$n = 4$		
	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime
ClassroomVideo	−7.1%	−2.8%	75.5%	2.8%	−0.6%	61.8%
Museum	−2.8%	−0.9%	108.5%	−4.8%	−1.3%	93.2%
Chess	−99.2%	−17.8%	99.8%	19.9%	−12.0%	81.3%
Guitarist	13.6%	2.4%	109.8%	76.1%	34.3%	87.6%
Hijack	50.9%	110.6%	95.8%	−11.4%	2.0%	86.2%
Cyberpunk	−2.4%	−0.7%	90.7%	1.6%	−3.3%	77.5%
Kitchen	2.5%	2.7%	75.0%	3.8%	2.6%	52.6%
Cadillac	−1.2%	−0.1%	76.3%	0.1%	1.5%	57.8%
Mirror	1.9%	2.7%	77.1%	2.4%	2.3%	57.4%
Fan	−1.5%	−0.8%	77.8%	−2.0%	−1.3%	59.9%
Group	39.7%	31.2%	98.5%	44.3%	21.6%	84.6%
Dancing	3.8%	2.4%	75.9%	7.4%	5.0%	58.6%
Painter	0.6%	1.0%	70.4%	1.7%	1.5%	51.0%
Breakfast	−20.4%	−12.9%	83.4%	−11.2%	−6.1%	66.5%
Barn	1.4%	−0.8%	77.4%	2.8%	−1.4%	54.9%
Frog	0.1%	−0.0%	57.6%	−0.2%	−0.3%	37.7%
Carpark	13.9%	7.9%	78.0%	8.3%	5.7%	55.6%
Street	2.3%	0.4%	68.6%	17.0%	11.3%	53.0%
Fencing	49.9%	20.8%	66.9%	147.7%	71.3%	43.8%
CBA Basketball	−5.3%	−60.1%	79.9%	2.9%	−0.2%	62.7%
MartialArts	67.1%	19.7%	89.8%	56.7%	81.8%	68.4%
Average	5.1%	5.0%	82.5%	17.4%	10.2%	64.4%

BD-Rate change higher than 3% was highlighted in red (for efficiency loss) or green (for efficiency gain). Decoding and Rendering Runtime was highlighted in green if it was smaller than 90%.

Nevertheless, turning on neighboring segment label examination (Table 4) significantly improves the quality, even over the configuration without hierarchical alpha-expansion (gain of 6.6% for IV-PSNR-based BD-rate).

The results show that it is not optimal to examine all depth levels, as is done in hierarchical alpha-expansion with second-cycle label offset with step 2. It is better to first provide a coarse proposal of the depth map and then refine it only with depth levels that are more probable (i.e., the neighboring depth levels and depth levels from neighboring segments). This scheme is the only one that provides quality gain, and, moreover, the runtime is still one of the shortest of all (only 5% longer than the fastest scheme)

The comparison of estimated depth maps is visualized in Figures 3 and 4. As can be seen, most of the differences between depth maps are present on the edges of objects.

Table 3. Hierarchical alpha-expansion with second-cycle label offset with $n = 2$ and $n = 4$, compared to basic alpha-expansion.

Sequence	$n = 2$			$n = 4$		
	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime
ClassroomVideo	−7.3%	−4.6%	66.5%	−5.7%	−3.5%	60.1%
Museum	−6.5%	−2.1%	57.6%	−3.5%	−1.1%	91.5%

Table 3. *Cont.*

Sequence	$n = 2$			$n = 4$		
	BD-Rate Y-PSNR	**BD-Rate IV-PSNR**	**Decoding and Rendering Runtime**	**BD-Rate Y-PSNR**	**BD-Rate IV-PSNR**	**Decoding and Rendering Runtime**
Chess	−9.3%	−13.0%	59.0%	−7.9%	−4.2%	77.8%
Guitarist	−0.4%	1.3%	53.2%	26.5%	34.0%	86.3%
Hijack	45.3%	40.2%	65.3%	26.4%	−43.0%	78.7%
Cyberpunk	−7.3%	−5.0%	57.7%	−6.1%	−2.8%	71.2%
Kitchen	1.1%	0.3%	56.5%	1.9%	1.4%	51.8%
Cadillac	0.3%	1.1%	54.8%	−0.4%	1.1%	54.4%
Mirror	1.2%	1.8%	56.7%	2.6%	2.6%	55.8%
Fan	0.1%	0.7%	53.8%	−2.2%	−0.9%	54.9%
Group	40.4%	31.0%	58.2%	31.1%	124.6%	77.6%
Dancing	3.7%	2.4%	58.1%	5.5%	3.5%	55.1%
Painter	0.5%	0.6%	57.9%	0.7%	0.0%	45.4%
Breakfast	−8.2%	−5.2%	57.9%	−12.9%	−7.1%	60.7%
Barn	3.3%	0.3%	59.6%	1.4%	−1.1%	55.5%
Frog	0.2%	0.2%	50.3%	0.2%	0.1%	33.7%
Carpark	12.0%	7.8%	56.7%	15.7%	10.4%	53.8%
Street	−1.3%	−1.3%	57.2%	3.3%	0.7%	49.9%
Fencing	−1.1%	−3.3%	56.9%	45.9%	18.3%	41.0%
CBA Basketball	−25.4%	−27.0%	57.2%	−4.3%	−9.3%	59.7%
MartialArts	−7.6%	−10.8%	54.3%	32.9%	4.4%	64.2%
Average	1.6%	0.7%	57.4%	7.2%	6.1%	60.9%

BD-Rate change higher than 3% was highlighted in red (for efficiency loss) or green (for efficiency gain). Decoding and Rendering Runtime was highlighted in green if it was smaller than 90%.

Table 4. Hierarchical alpha-expansion with second-cycle label offset and neighboring segments label examination with $n = 4$ compared to basic hierarchical alpha-expansion.

Sequence	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime
ClassroomVideo	−5.6%	−2.9%	60.2%
Museum	−3.7%	−1.0%	89.9%
Chess	−14.5%	−5.2%	80.1%
Guitarist	−8.9%	−89.1%	88.8%
Hijack	6.6%	25.4%	78.8%
Cyberpunk	−9.7%	−6.6%	69.8%
Kitchen	2.3%	1.3%	51.4%
Cadillac	0.8%	1.8%	57.8%
Mirror	2.3%	2.3%	57.1%
Fan	−2.9%	−1.3%	57.3%
Group	−1.3%	−27.4%	78.6%
Dancing	1.3%	1.5%	56.8%
Painter	2.4%	2.3%	45.6%
Breakfast	−16.3%	−8.5%	63.5%
Barn	2.0%	0.2%	58.0%
Frog	0.2%	0.3%	36.6%

Table 4. *Cont.*

Sequence	BD-Rate Y-PSNR	BD-Rate IV-PSNR	Decoding and Rendering Runtime
Carpark	10.8%	6.5%	55.1%
Street	3.4%	0.5%	51.4%
Fencing	15.8%	8.2%	41.3%
CBA Basketball	−19.4%	−21.9%	61.3%
MartialArts	−28.7%	−25.0%	65.9%
Average	−3.0%	−6.6%	62.2%

BD-Rate change higher than 3% was highlighted in red (for efficiency loss) or green (for efficiency gain). Decoding and Rendering Runtime was highlighted in green if it was smaller than 90%.

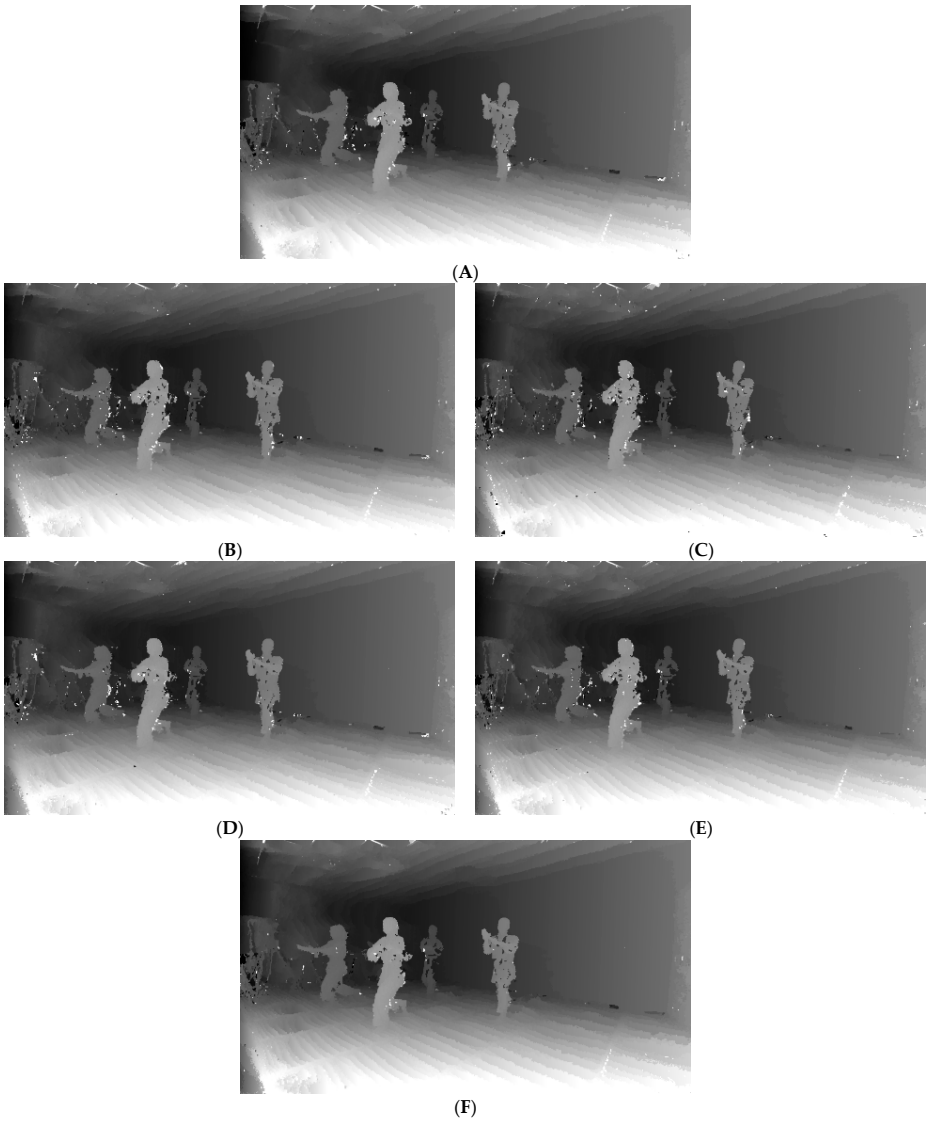

Figure 3. The comparison of depth maps estimated for the MartialArts sequence. Depth maps estimated using graph-cut optimization with: (**A**) alpha-expansion, (**B**) hierarchical alpha-expansion ($n = 2$), (**C**) hierarchical alpha-expansion ($n = 4$), (**D**) hierarchical alpha-expansion with second-cycle label shift ($n = 2$), (**E**) hierarchical alpha-expansion with second-cycle label shift ($n = 4$), (**F**) hierarchical alpha-expansion with second-cycle label shift and neighboring segments label examination.

Figure 4. The visualization of differences in depth maps estimated using alpha-expansion and the proposal. Areas estimated differently (shown as white areas) are overlaid on input view.

5. Conclusions

This paper presents a novel hierarchical alpha-expansion-based graph-cut optimization technique for decoder-side depth estimation in immersive video applications. Our approach significantly enhances the computational efficiency and accuracy of depth map estimation, addressing the critical challenges of rendering high-quality immersive videos. Through comprehensive experimentation, we have demonstrated that our method outperforms the state-of-the-art technique, in terms of both depth accuracy and computational load, making it a viable solution for more practical immersive video applications. The proposal is particularly relevant for streaming immersive content to resource-limited receivers, where optimizing decoder-side computations is crucial for a user experience.

The experimental results have shown that examining selected, more plausible depth levels provides more accurate estimation than examining all available labels. Therefore, future directions for research can include exploring adaptive hierarchical structures that can dynamically adjust based on content complexity and computational complexity constraints (such as the number of CPU cores or the size of RAM available in the decoder). In conclusion, the hierarchical alpha-expansion-based graph-cut optimization presents a significant step forward in the research on using decoder-side depth estimation for immersive video compression.

Author Contributions: Conceptualization, D.M. and D.K.; methodology, D.M.; software, D.M. and D.K.; validation, D.M.; formal analysis, D.M. and O.S.; investigation, D.M.; writing—original draft preparation, D.M.; writing—review and editing, D.K. and O.S.; visualization, D.M.; supervision, O.S.; project administration, D.M. and O.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by Ministry of Science and Higher Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The research was conducted under ISO/IEC JTC1/SC29 WG04 MPEG Video Coding Common Test Conditions for MPEG Immersive Video: https://mpeg-miv.org/, accessed on 13 June 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Debarba, H.G.; Montagud, M.; Chagué, S.; Herrero, J.G.-L.; Lacosta, I.; Langa, S.F.; Charbonnier, C. Content format and quality of experience in virtual reality. *Multimed. Tools Appl.* **2022**, *81*, 14269–14303. [CrossRef]
2. Boyce, J.M.; Dore, R.; Dziembowski, A.; Fleureau, J.; Jung, J.; Kroon, B.; Salahieh, B.; Vadakital, V.K.M.; Yu, L. MPEG Immersive Video Coding Standard. *Proc. IEEE* **2021**, *109*, 1654–1676. [CrossRef]
3. Zhang, Y.; Yang, J.; Liu, Z.; Wang, R.; Chen, G.; Tong, X.; Guo, B. VirtualCube: An Immersive 3D Video Communication System. *IEEE Trans. Vis. Comput. Graph.* **2022**, *28*, 1681–1690. [CrossRef] [PubMed]
4. Mieloch, D.; Garus, P.; Milovanovic, M.; Jung, J.; Jeong, J.Y.; Ravi, S.L.; Salahieh, B. Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 4282–4296. [CrossRef]
5. Chan, Y.L.; Fu, C.H.; Chen, H.; Tsang, S.H. Overview of current development in depth map coding of 3D video and its future. *IET Signal Process.* **2020**, *14*, 1–14. [CrossRef]
6. Garus, P.; Jung, J.; Maugey, T.; Guillemot, C. Bypassing Depth Maps Transmission for Immersive Video Coding. In Proceedings of the 2019 Picture Coding Symposium (PCS), Ningbo, China, 8–10 May 2019.
7. Dziembowski, A.; Domanski, M.; Grzelka, A.; Mieloch, D.; Stankowski, J.; Wegner, K. The influence of a lossy compression on the quality of estimated depth maps. In Proceedings of the 2016 International Workshop on Systems, Signals and Image Processing (IWSSIP), Bratislava, Slovakia, 18–20 May 2016.
8. Vadakital, V.K.M.; Dziembowski, A.; Lafruit, G.; Thudor, F.; Lee, G.; Alface, P.R. The MPEG Immersive Video Standard—Current Status and Future Outlook. *IEEE MultiMedia* **2022**, *29*, 101–111. [CrossRef]
9. Mieloch, D.; Dziembowski, A.; Klóska, D.; Szydełko, B.; Jeong, J.Y.; Lee, G. A New Approach to Decoder-Side Depth Estimation in Immersive Video Transmission. *IEEE Trans. Broadcast.* **2023**, *69*, 611–624. [CrossRef]
10. Ravi, S.L.; Milovanovic, M.; Morin, L.; Henry, F. A Study of Conventional and Learning-Based Depth Estimators for Immersive Video Transmission. In Proceedings of the 2022 IEEE 29th International Conference on Image, Video and Signal Processing (MMSP), Shanghai, China, 25–27 August 2022.
11. Kolmogorov, V.; Zabih, R. Multi-camera Scene Reconstruction via Graph Cuts. In Proceedings of the 7th European Conference on Computer Vision (ECCV), London, UK, 28 May–2 June 2002.
12. Rogge, S.; Bonatto, D.; Sancho, J.; Salvador, R.; Juarez, E.; Munteanu, A.; Lafruit, G. MPEG-I Depth Estimation Reference Software. In Proceedings of the 2019 International Conference on 3D Immersion (IC3D), Brussels, Belgium, 18–20 September 2019.
13. Papadakis, N.; Caselles, V. Multi-label Depth Estimation for Graph Cuts Stereo Problems. *J. Math. Imaging Vis.* **2010**, *38*, 70–82. [CrossRef]
14. Mieloch, D.; Grzelka, A. Segmentation-based Method of Increasing the Depth Maps Temporal Consistency. *Int. J. Electron. Telecommun.* **2018**, *64*, 283–289. [CrossRef]
15. Mieloch, D.; Stankiewicz, O.; Domanski, M. Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation. *IEEE Access* **2020**, *8*, 5760–5776. [CrossRef]
16. Dziembowski, A.; Mieloch, D.; Stankowski, J.; Grzelka, A. IV-PSNR—The Objective Quality Metric for Immersive Video Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7575–7591. [CrossRef]
17. Mieloch, D.; Dziembowski, A.; Grzelka, A.; Stankiewicz, O.; Domański, M. Graph-based multiview depth estimation using segmentation. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017.
18. Lei, J.; Liu, J.; Zhang, H.; Gu, Z.; Ling, N.; Hou, C. Motion and Structure Information Based Adaptive Weighted Depth Video Estimation. *IEEE Trans. Broadcast.* **2015**, *61*, 351–362. [CrossRef]
19. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [CrossRef]
20. Tippetts, B.; Lee, D.J.; Lillywhite, K.; Archibald, J. Review of stereo vision algorithms and their suitability for resource-limited systems. *J. Real-Time Image Process.* **2016**, *11*, 5–27. [CrossRef]
21. Xue, T.; Owens, A.; Scharstein, D.; Goesele, M.; Szeliski, R. Multiframe stereo matching with edges, planes, and superpixels. *Image Vis. Comput.* **2019**, *91*, 103806. [CrossRef]
22. Adrian, D.; Adam, G.; Dawid, M.; Olgierd, S. Depth map upsampling and refinement for FTV systems. In Proceedings of the 2016 International Conference on Signals and Electronic Systems (ICSES), Kraków, Poland, 11–13 September 2016.
23. *ISO/IEC JTC1/SC29/WG4 MPEG2023/N0406*; Common Test Conditions for MPEG Immersive Video. ISO: Geneva, Switzerland, 2023.
24. Bjøntegaard, G. Calculation of average PSNR differences between RD-Curves. In Proceedings of the ITU-T VCEG Meeting, Austin, TX, USA, 2–4 April 2001.
25. *ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415*; 3DoF+ Test Sequence ClassroomVideo. ISO: Geneva, Switzerland, 2018.
26. *ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349*; Technicolor 3DoF+ Test Materials. ISO: Geneva, Switzerland, 2018.
27. *ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787*; New Test Content for Immersive Video—Nokia Chess. ISO: Geneva, Switzerland, 2020.
28. *ISO/IEC JTC1/SC29/WG04 MPEG2021/M58080*; A New Computer Graphics Scene, Guitarist, Suitable for MIV Edition-2. ISO: Geneva, Switzerland, 2021.

29. *ISO/IEC JTC1/SC29/WG11, MPEG2021/M58433*; [MIV] ERP Content Proposal for MIV ver.1 Verification Test. ISO: Geneva, Switzerland, 2021.
30. *ISO/IEC JTC1/SC29/WG11 MPEG2018/M43318*; Proposition of New Sequences for Windowed-6DoF Experiments on Compression, Synthesis and Depth Estimation. ISO: Geneva, Switzerland, 2018.
31. *ISO/IEC JTC1/SC29/WG4, MPEG2021/M57186*; [MIV] New Cadillac Content Proposal for Advanced MIV v2 Investigations. ISO: Geneva, Switzerland, 2021.
32. *ISO/IEC JTC1/SC29/WG11 MPEG2020/M55710*; Interdigital Mirror Content Proposal for Advanced MIV Investigations on Reflection. ISO: Geneva, Switzerland, 2021.
33. *ISO/IEC JTC1/SC29/WG11 MPEG/M54732*; InterdigitalFan Content Proposal for MIV. ISO: Geneva, Switzerland, 2020.
34. *ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731*; InterdigitalGroup Content Proposal. ISO: Geneva, Switzerland, 2020.
35. *ISO/IEC JTC1/SC29/WG4 MPEG2021/M57751*; [MIV] Dancing Sequence for Verification Tests. ISO: Geneva, Switzerland, 2021.
36. *ISO/IEC JTC1/SC29/WG11 MPEG2017/M40010*; Light Field Content from 16-Camera Rig. ISO: Geneva, Switzerland, 2017.
37. *ISO/IEC JTC1/SC29/WG4 MPEG2021/M56730*; MIV] Breakfast New Natural Content Proposal for MIV. ISO: Geneva, Switzerland, 2021.
38. *ISO/IEC JTC1/SC29/WG4 MPEG2021/M56632*; Barn New Natural Content Proposal for MIV. ISO: Geneva, Switzerland, 2021.
39. *ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748*; Kermit Test Sequence for Windowed 6DoF Activities. ISO: Geneva, Switzerland, 2018.
40. *ISO/IEC JTC1/SC29/WG11 MPEG2019/M51598*; Natural Outdoor Test Sequences. ISO: Geneva, Switzerland, 2020.
41. *ISO/IEC JTC1/SC29/WG11 MPEG2018/M38247*; Multiview Test Video Sequences for Free Navigation Exploration Obtained Using Paris of Cameras. ISO: Geneva, Switzerland, 2016.
42. *ISO/IEC JTC1/SC29/WG11 MPEG2021/M58500*; [MIV] Undistorted CBA Basketball Test Sequence for MPEG-I Visual. ISO: Geneva, Switzerland, 2021.
43. *ISO/IEC JTC1/SC29/WG04 MPEG2023/M61949*; [MIV] New Natural Content—MartialArts. ISO: Geneva, Switzerland, 2023.