



POZNAN UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTING AND TELECOMMUNICATIONS
INSTITUTE OF MULTIMEDIA TELECOMMUNICATIONS



Doctoral Dissertation

Immersive Video Coding Using
Decoder-Side Depth Estimation

Dominika Klóska

Supervisors: dr hab. inż. Dawid Mieloch, prof. PP
 dr hab. inż. Adrian Dziembowski, prof. PP

I would like to express my sincere gratitude to Professor Dawid Mieloch and Professor Adrian Dziembowski. Their generous support, expertise, and boundless patience have been a constant source of guidance throughout this journey.

I would also like to thank my colleagues at the Institute of Multimedia Telecommunications for creating such an inspiring and collaborative environment over the years. Furthermore, I owe my sincere thanks to Dr Tomasz Grajek and Jakub Stankowski for their time and careful review of the manuscript.

This work is dedicated to my beloved parents, Krystyna and Stanisław, for their unwavering and unconditional love and support.

ABSTRACT

This dissertation explores and advances the technology of Decoder-Side Depth Estimation (DSDE) in immersive video compression. At the initial stage of this research, DSDE was a newly proposed concept within the MPEG Immersive Video (MIV) standard, presenting significant opportunities for improvement. By leveraging the MIV standard and the Immersive Video Depth Estimation (IVDE) reference software, this work systematically addresses these limitations, establishing a rigorous framework to validate its foundational research theses.

To achieve this, the impact of the views selected for transmission on the quality of immersive video at the decoder is investigated first. Following a comprehensive review of existing literature, a progression of solutions is introduced, evolving from a basic transmission scenario to a versatile view selection method tailored specifically for DSDE.

Subsequently, the focus shifts to the refinement of the depth estimation algorithm itself, optimising it to yield high-quality results when operating on the decoder side with decoded views. Furthermore, a key modification to the graph cut algorithm, used in the IVDE depth estimation software, is proposed to optimise a goal function. The proposed algorithmic adjustment significantly accelerates the estimation process while simultaneously improving the quality of the estimated depth maps.

Building upon these developments, a comprehensive immersive video coding scheme based on the MIV standard is established. This proposed framework entails transmitting selected depth maps within one of the atlases to expedite DSDE further. The system is evaluated across two distinct scenarios: the first involves the transmission of full depth maps, while the second additionally transmits depth map patches corresponding to occluded scene elements. Notably, the innovative coding framework developed and presented in this chapter has been officially included in the MPEG MIV standard and is protected by a patent.

STRESZCZENIE

Niniejsza rozprawa doktorska poświęcona jest badaniu i rozwojowi technologii estymacji głębi po stronie dekodera. W początkowej fazie prowadzonych badań technologia ta znajdowała się we wczesnym stadium rozwoju, co otwierało szerokie pole do optymalizacji i wdrożenia udoskonaleń. W ramach badań wykorzystano normę MPEG Immersive Video (MIV) oraz oprogramowanie referencyjne Immersive Video Depth Estimation (IVDE) w celu wprowadzenia modyfikacji usprawniających technologię wizji immersyjnej, co pozwoliło na precyzyjne zarysowanie zakresu pracy oraz zweryfikowanie sformułowanych tez badawczych.

W pierwszej kolejności skupiono się na analizie wpływu wyboru konkretnych widoków przeznaczonych do transmisji na jakość wizji immersyjnej po stronie dekodera. Po szczegółowym przeglądzie literatury, zaproponowano szereg nowych rozwiązań: od prostego wariantu transmisji aż po kompleksową metodę selekcji widoków, zaprojektowaną ściśle dla estymacji głębi po stronie dekodera.

Następnie, istotną część badań poświęcono udoskonaleniu samego oprogramowania referencyjnego do estymacji głębi, optymalizując je pod kątem zapewnienia wysokiej jakości map głębi w sytuacjach, gdy proces ten zachodzi na podstawie zdekodowanych widoków. Ponadto, zaproponowano kluczową modyfikację algorytmu cięcia grafu (ang. *graph cut*), stosowanego w oprogramowaniu IVDE. Modyfikacja ta znacząco przyspiesza proces obliczeniowy, poprawiając jednocześnie jakość estymowanych map głębi.

Opierając się na rozwiązaniach opracowanych we wcześniejszych etapach pracy, w kolejnym kroku zaprezentowano kompleksowy schemat kodowania wizji immersyjnej, bazujący na normie MIV. Zaproponowany schemat zakłada transmisję wybranych map głębi w ramach jednego z atlasów, co pozwala na dodatkowe podniesienie ich jakości oraz przyspieszenie estymacji po stronie dekodera. Rozwiązanie to przetestowano w dwóch odrębnych scenariuszach: pierwszy z nich zakłada transmisję pełnych map głębi, natomiast w drugim przesyłane są dodatkowo ich fragmenty zawierające przysłonięte elementy sceny. Przedstawiony nowatorski schemat kodowania, stanowiący zwieńczenie pracy, został oficjalnie włączony do normy MPEG MIV oraz objęty ochroną patentową.

PUBLICATIONS INCLUDED IN THIS DISSERTATION

This dissertation consists of five research publications, which collectively document the systematic advancement of immersive video coding using Decoder-Side Depth Estimation. For each publication the Ministry points (M) and the Impact Factor (IF) are provided, along with the contribution statement of the author. The list concludes with a granted USPTO patent, which is included as a supporting publication rather than forming part of the core publication cycle.

- [P1] D. Mieloch, A. Dziembowski, **D. Klóska**, B. Szydełko, J. Jeong, G. Lee, “A New Approach to Decoder-Side Depth Estimation in Immersive Video Transmission,” *IEEE Transactions on Broadcasting*, Vol. 69, No. 4, 2023, pp. 951-965.

[M = 140, IF = 4.8]

Own contribution: ~35% – co-designing of IDMA and eIDMA methods, significant role in the implementation of proposals, conducting experiments, collection and processing of results, analysing the results, writing the text of the manuscript.

- [P2] **D. Klóska**, A. Dziembowski, A. Grzelka, D. Mieloch, “On the Selection of Transmitted Views for Decoder-Side Depth Estimation,” *Applied Sciences*, Vol. 16, No. 1, 2026, pp. 72-1 - 72-18.

[M = 100, IF = 2.5]

Own contribution: ~80% – main contribution in designing the method and the experimental methodology, conducting experiments, collecting and processing the results, analysing the results, writing the text of the manuscript.

- [P3] D. Mieloch, **D. Klóska**, O. Stankiewicz, “On Alpha-Expansion-Based Graph-Cut Optimization for Decoder-Side Depth Estimation,” *Applied Sciences*, Vol. 14, No. 13, 2024, pp. 5768.

[M = 100, IF = 2.5]

Own contribution: ~45% – co-designing of the method, partial implementation of solutions, conducting a subset of experiments, collecting the results, analysing the results, partial preparation of the manuscript.

- [P4] **D. Klóska**, A. Dziembowski, J. Samelak, “Versatile input view selection for efficient immersive video transmission,” 31st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2023, Pilsen, Czech Republic, 2023-05-15.

[M = 70]

Own contribution: ~80% – main contribution in designing the method and the experimental methodology, conducting experiments, collecting and processing the results, analysis of the results, preparation of the manuscript.

- [P5] D. Mieloch, **D. Klóska**, M. Woźniak, "Point-to-Block Matching in Depth Estimation," 29th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: WSCG 2021, Prague, Czech Republic, 2021-05-17.

[M = 70]

Own contribution: ~30% – co-designing of the method, implementing the solution, conducting state-of-the-art review, performing experiments, collecting and processing data, analysing the results, writing the manuscript.

SUPPORTING PUBLICATION:

- [P6] J. Y. Jeong, G. Lee, D. Mieloch, M. Domański, A. Dziembowski, B. Szydełko, **D. Klóska**, „Method for decoding immersive video and method for encoding immersive video,” Patent office: USPTO, Patent number: US12530806, Publication date: 2026-01-20

Own contribution: ~35% – co-designing of IDMA and eIDMA methods, significant role in the implementation of proposals, preparation of figures for patent application.

TABLE OF CONTENTS

1.	INTRODUCTION.....	8
1.1.	SCOPE OF THE DISSERTATION	8
1.2.	GOALS AND THESES	14
1.3.	OVERVIEW OF THE DISSERTATION	15
2.	VIEW SELECTION STRATEGIES FOR EFFICIENT IMMERSIVE VIDEO CODING.....	17
2.1.	BASELINE TRANSMISSION SCENARIO.....	19
2.1.1.	EXPERIMENTAL METHODOLOGY	20
2.1.2.	RESULTS	20
2.2.	OMNIDIRECTIONAL CONTENT TRANSMISSION SCENARIO.....	21
2.2.1.	EXPERIMENTAL METHODOLOGY	23
2.2.2.	RESULTS	24
2.3.	DECODER-SIDE DEPTH ESTIMATION SCENARIO.....	26
2.3.1.	EXPERIMENTAL METHODOLOGY	29
2.3.2.	RESULTS	30
3.	REDUCING THE IMPACT OF LOSSY COMPRESSION ON DECODER-SIDE DEPTH ESTIMATION	35
3.1.	THE PROPOSAL OF POINT-TO-BLOCK MATCHING.....	37
3.1.1.	RESEARCH DESIGN AND METHODOLOGY	38
3.1.2.	ASSESSMENT OF THE QUALITY OF VIRTUAL VIEWS	40
3.2.	ADAPTING GRAPH CUT OPTIMISATION FOR DECODER-SIDE DEPTH ESTIMATION	43
3.2.1.	PROPOSED ALPHA-EXPANSION BASED OPTIMISATION.....	46
3.2.2.	EXPERIMENTAL METHODOLOGY	48
3.2.3.	RESULTS AND ANALYSIS.....	49
4.	INPUT DEPTH MAP ASSISTANCE	53
4.1.	PROPOSED DEPTH ASSISTANCE APPROACHES.....	54
4.2.	EXPERIMENTAL METHODOLOGY	61
4.3.	RESULTS AND COMPARATIVE ANALYSIS.....	62
5.	SUMMARY OF THE DISSERTATION.....	73
	REFERENCES	75
	PUBLICATIONS OF THE AUTHOR.....	83
	INVOLVEMENT IN RESEARCH PROJECTS.....	86

1. INTRODUCTION

1.1. SCOPE OF THE DISSERTATION

The research presented in this dissertation focuses on immersive video technologies, which enable virtual navigation within three-dimensional (3D) scenes. Such interaction is typically facilitated by head-mounted display (HMD) [1] devices designed to provide the user a sense of immersion within the virtual environment. However, immersive content is also frequently rendered on conventional displays. In these instances, viewpoint modification is performed through peripheral input devices or tactile interfaces [2, 3].

In conventional television and video streaming services, viewers are presented with only one, non-changeable perspective of the captured scene (Fig. 1A). A further evolution of this medium gave us stereoscopic television (Fig. 1B), which more closely mirrors human vision by providing two viewpoints of a scene, enabling depth perception [2]. However, these viewpoints remain static, as the perspective is predetermined during the acquisition stage and cannot be altered by the viewer during playback.

To facilitate interactive content that can be modified by the user and viewed from different directions, spherical (360-degree) cameras were employed, as illustrated in Fig. 1C. In this instance, interaction is restricted to three degrees of freedom (3DoF), replicating the rotational movements of the head of the viewer [4].

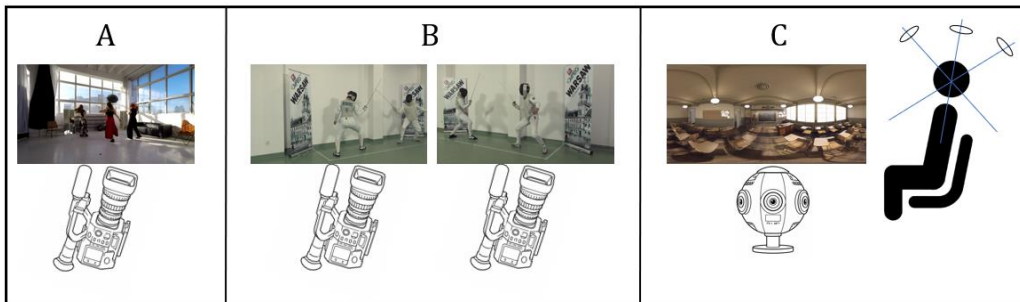


Figure 1. Conventional video (A), stereoscopic video (B), 3DoF video (C).

Immersive video systems, one of the latest generations of visual multimedia, offer full interactive movement with six degrees of freedom (6DoF), enabling users to autonomously alter both their spatial position and viewing orientation within the captured 3D scene (Fig. 2).

In the 6DoF scenario, the transmission of a single perspective is insufficient. Instead, it is necessary to convey the three-dimensional (3D) geometry of the scene. One established approach to represent 3D scenes for immersive video involves using multiple views along with their corresponding depth maps. This format is referred to as Multiview Video plus Depth (MVD) [5] and its example is illustrated in Fig. 3.

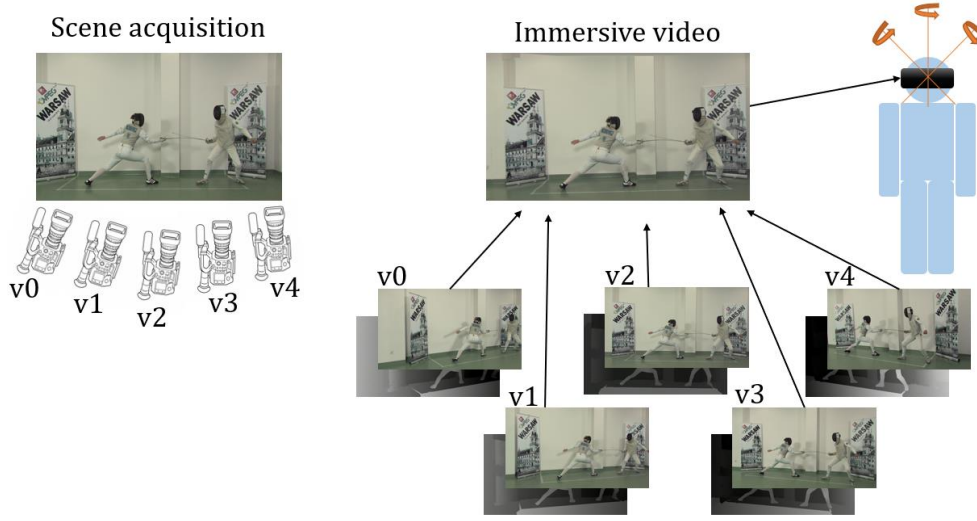


Figure 2. Immersive video (6DoF).

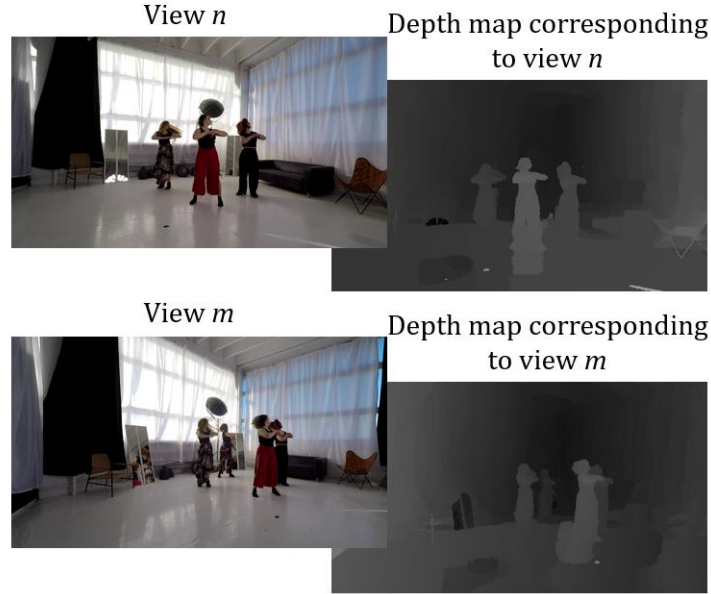


Figure 3. Multiview Video plus Depth representation.

The camera configuration and the selection of sensor types are primarily dictated by the scale of the captured scene and the requirements of the intended application [6]. Depth maps are typically represented as greyscale images (Fig. 3B) where pixel values encode the distance from the observed scene to the camera plane.

The most prevalent method for representing depth information is normalised disparity [7], which correlates with the non-linear human perception of depth. Typically, depth maps are estimated algorithmically from the captured views of the scene [8]. In this context, the fundamental principle for determining the depth value of a specific point is identifying its corresponding point in an adjacent view.

Real-time depth acquisition is achievable with the use of dedicated depth cameras which appear to be potentially ideal candidates for immersive video systems [9]. However, the practical deployment of such hardware is often hindered by several significant drawbacks, including low spatial resolution of these cameras and a restricted measurement range. Furthermore, in multi-camera configurations, there is a substantial risk of mutual interference when multiple cameras are operated simultaneously [10].

Given the acquired views, the camera parameters, and the 3D geometry of the scene, it is possible to synthesise virtual views. These views are rendered at arbitrary positions selected by the user within the immersive video system [11] (Fig. 4).

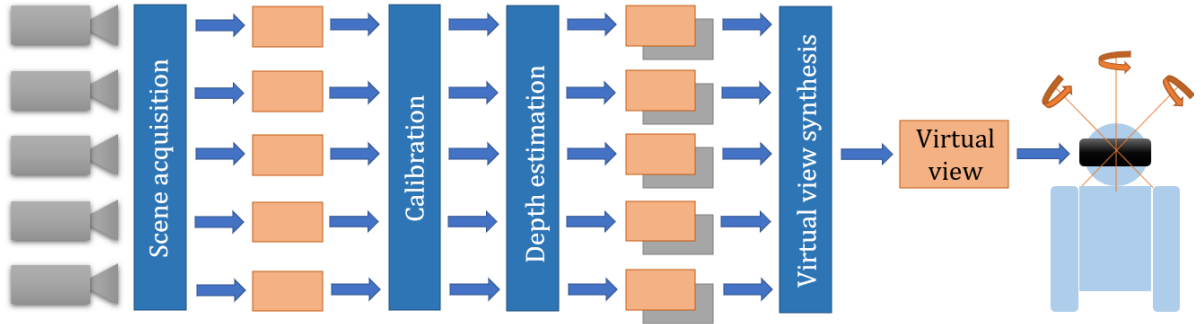


Figure 4. Virtual view synthesis in the position requested by the viewer.

The estimation of camera parameters (involving intrinsic properties, such as focal length, and extrinsic attributes, such as spatial positioning) is relatively straightforward and can be achieved with sufficient precision using widely accessible methods [8]. On the other hand the quality of the synthesised virtual views is dependent on the accuracy of the depth maps [7, 12]. This dependency exists because the virtual view synthesis process integrates both the input texture views and their corresponding depth information. Consequently, the fidelity with which a depth map represents the scene geometry directly influences the visual quality of the resulting synthesis.

As in typical streaming or broadcasting, the transmission of immersive video requires data compression. However, data volumes in the MVD representation are substantially larger than those in conventional video sequences. Consequently, techniques for immersive video compression must be characterised by very high coding efficiency. A straightforward approach to addressing this challenge involves multi-view simulcast transmission (Fig. 5).

Unfortunately, the independent encoding of views and their corresponding depth maps is not particularly efficient and results in substantial bitrate [13]. Furthermore, this approach necessitates simultaneous operation of multiple decoder instances, which poses significant challenges for hardware implementations in consumer devices [14]. When transmitting full-scene representations, an additional constraint is the pixel rate, defined as the total number of pixels that must be decoded per second to facilitate the synthesis of virtual views [14]. Even if these requirements are satisfied, conventional video encoders are primarily optimised for the compression of video depicting natural

scenes [15]. Characteristics of depth maps substantially differ from traditional videos and therefore, their coding is significantly less efficient, so specialised tools have to be employed [16]. However, utilising different codecs for texture and depth data increases computational complexity of encoding process and compromises the interoperability of the system.

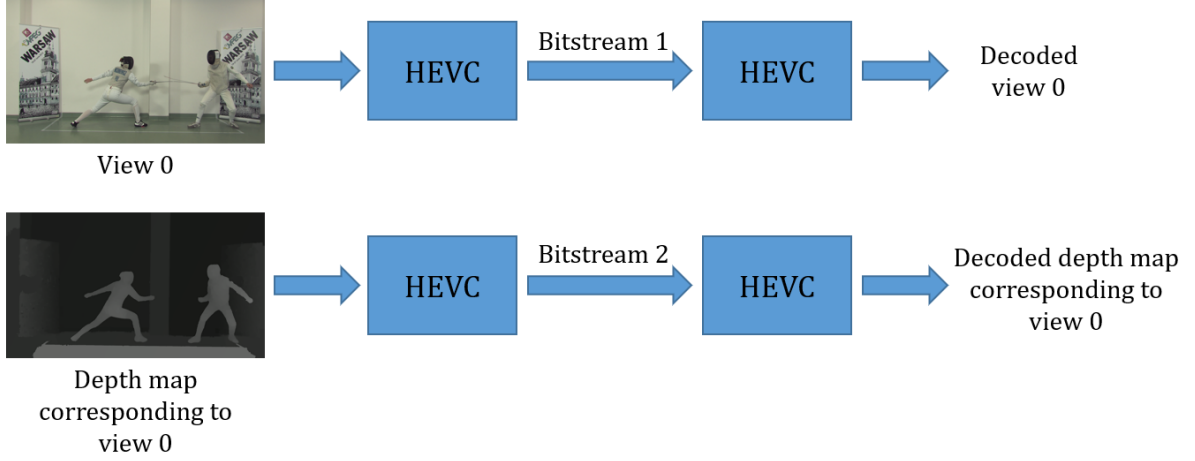


Figure 5. Multiview simulcast.

Further advancement in immersive video coding is the MPEG Immersive Video (MIV) standard [13]. The MIV standard exclusively defines the bitstream format and the decoding processes, alongside its supported profiles. Its associated reference software, the Test Model for MPEG Immersive Video (TMIV) [17], further encompasses the non-normative components, including the encoding and rendering processes.

The MIV standard incorporates several profiles, each defining distinct methodologies for handling immersive video pre- and post-processing. The baseline profile is MIV Main. By employing mechanisms such as view selection, pruning, and packing, this profile generates atlases which significantly reduce the volume of data transmitted to the decoder. The block diagram of the MIV Main profile is illustrated in Fig. 6.

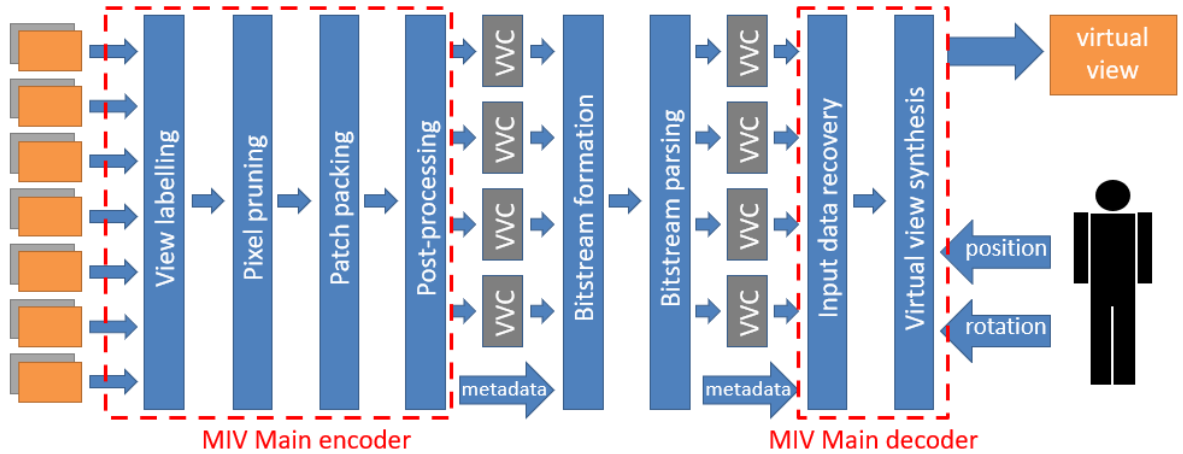


Figure 6. MIV Main block scheme.

The atlases within the MIV Main profile comprise full views (texture component), their associated depth maps (depth component), and patches derived from both components (Fig. 7). These patches represent the residual information remaining after redundancy removal (pruning) has been performed on views that were not labelled as base views. Typically, the MIV Main profile generates four distinct atlases: the first contains the full texture views (and if there is any empty place left, it is being filled with the mosaic of patches), while the second provides the corresponding depth maps. The third and fourth atlases are reserved for texture patches and their associated depth patches, respectively.

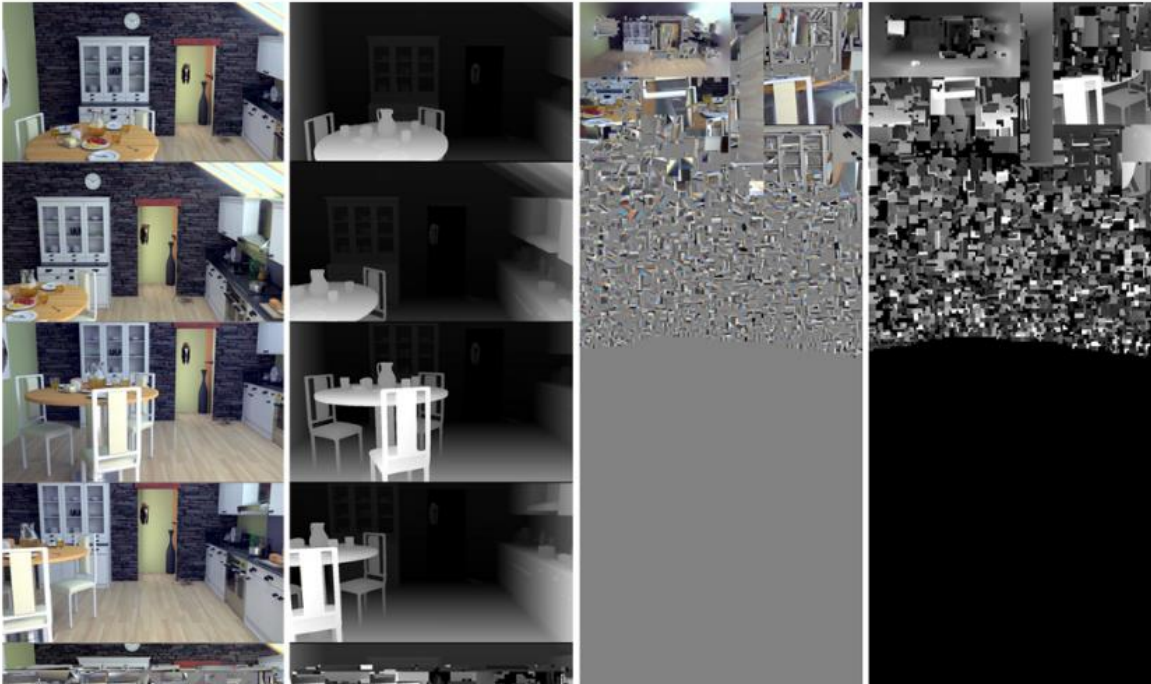


Figure 7. *MIV Main atlases.*

Following this, each atlas is individually encoded using a conventional video codec, such as Versatile Video Coding (VVC) [18], and transmitted to the decoder. At the decoder side, the inversed set of operations is performed. It includes the reconstruction of views from the atlases and the final rendering process, resulting in immersive video sequence.

Nevertheless, the MIV Main profile does not address the fundamental challenge to which this doctoral dissertation is dedicated. Within the MIV Main framework, conventional video encoders (such as VVC) are still utilised for depth map compression, despite the fact that these codecs were not inherently designed for such data. This has a detrimental impact on the quality of synthesised virtual views, as the use of conventional codecs for geometry data often results in sub-optimal rate-distortion performance, as these tools are not tuned to preserve sharp discontinuities typically present in depth maps.

To address this challenge, MPEG introduced the Geometry Absent (MIV GA) profile [19] which leverages a mechanism known as Decoder-Side Depth Estimation (DSDE). Unlike MIV Main, this profile does not transmit depth maps; instead, it delegates the

reconstruction of scene geometry to the decoder side. The block diagram of the MIV Geometry Absent profile is illustrated in Fig. 8.

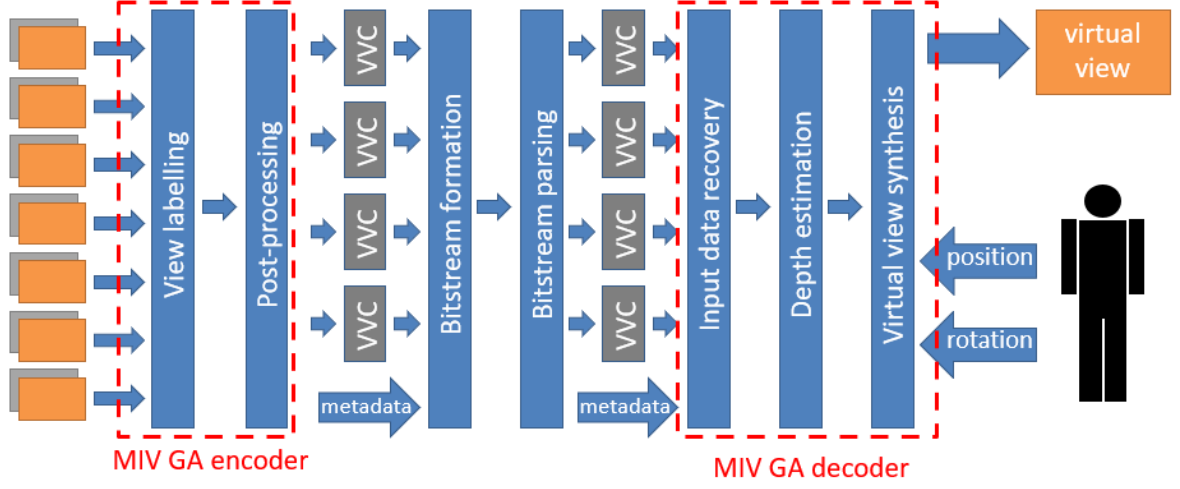


Figure 8. MIV GA block scheme.

Consequently, this approach obviates the need to include a depth component within the MIV bitstreams transmitted to the decoder (Fig. 9). Instead, the bitstream consists exclusively of full views.



Figure 9. MIV Geometry Absent atlases.

Within this profile, the atlases contain exclusively texture component, with depth being estimated at the decoder side before the rendering stage. Notably, the specific depth estimation algorithm is not defined within the MIV GA specification, categorising it as a non-normative component.

The application of the MIV GA profile is particularly advantageous in scenarios where depth information is either unavailable, or computationally expensive to estimate at the encoder side. Furthermore, it facilitates significant bandwidth savings by eliminating the

need to transmit geometry bitstreams. For certain applications, such as dense light fields or live event streaming, generating high-quality depth maps for every view at the encoding stage is computationally expensive. The MIV GA profile addresses this by allowing the decoder to selectively estimate depth only for the views necessary to render the target viewport, thereby reducing the total computational overhead compared to full-scene depth estimation. Additionally, this profile supports distributed architectures where MIV GA is utilised for captured multi-view sequence, while depth estimation is performed at an intermediate location (such as a cloud server), to then generate MIV Main bitstreams for end-user distribution.

Research presented in [19, 20] indicates that for specific content types, notably natural scenes with low-fidelity initial depth maps, estimating geometry from decoded texture data at the receiver side can yield superior visual fidelity compared to using compressed depth maps. This holds true even when a restricted number of views are available for the estimation process. Consequently, the MIV GA profile not only bypasses the limitations of conventional depth encoders, but also may enhance the final quality for natural content, which is a critical requirement for practical immersive video systems.

Accordingly, this doctoral dissertation focuses on the Decoder-Side Depth Estimation approach, specifically aiming to further optimise its performance and enhance individual stages of its processing pipeline.

1.2. GOALS AND THESES

The primary objective of this doctoral dissertation is to enhance the efficiency and visual fidelity of immersive video systems based on Decoder-Side Depth Estimation (DSDE). By conducting a comprehensive analysis of the MPEG Immersive Video (MIV) Geometry Absent profile, this research aims to optimise the processing pipeline from initial view selection to final synthesis, ensuring robust performance even in the presence of lossy compression. To achieve this, the following research theses have been proposed and validated.

Thesis 1

The selection of views transmitted for Decoder-Side Depth Estimation should depend on scene characteristics, specifically the proportion of occlusions.

The initial stage of the DSDE-based compression pipeline involves selecting a subset of views for transmission. While conventional methods focus on texture redundancy, DSDE-based systems are uniquely sensitive to occlusions, where a lack of multi-view correspondence prevents accurate depth estimation. The original research presented in this dissertation demonstrates that adapting the selection criteria to account for scene

geometry significantly reduces estimation errors in regions that would otherwise remain occluded.

Thesis 2

It is possible to modify the depth estimation method to improve the quality of depth maps estimated from encoded views.

Standard depth estimation algorithms are typically designed for uncompressed, high-fidelity input. However, in practical transmission scenarios, the decoded texture components exhibit specific artefacts, such as ringing and edge-shifting, which degrade the precision of spatial matching. The original methods described in this thesis proposes a modification to the inter-view matching and the optimization performed within the depth estimation algorithm which enhances the robustness against compression-related distortions and, thereby, improves the quality of the estimated depth maps and resulting virtual views presented to a viewer.

Thesis 3

It is possible to enhance the quality of virtual view synthesis while maintaining the same bitrate as the state-of-the-art Decoder-Side Depth Estimation methods by leveraging additional input depth information at the decoder side.

Building upon the limitations of purely texture-based estimation on the decoder side, this work proposes a hybrid architecture. By leveraging information available at the encoder side, the original method proposed in this dissertation simplifies or bypasses the most computationally expensive and error-prone stages of DSDE. This approach yields superior synthesis results and faster processing times without exceeding the bandwidth constraints of the state-of-the-art compression method based on MIV GA profile.

1.3. OVERVIEW OF THE DISSERTATION

This dissertation is structured into five chapters, tracing the progression from theoretical foundations and data pre-processing to the development of novel architectures for immersive video coding. The content of each chapter is outlined below.

Chapter 1:

Introduction provides the motivation for this research and delineates the scope of the study. It introduces the state-of-the-art MPEG Immersive Video (MIV) Geometry Absent profile and establishes the research theses that guide the subsequent investigations.

Chapter 2:

View selection strategies for efficient immersive video coding addresses the first research thesis. It examines the principles of view selection in a general transmission scenario and in transmission specifically for DSDE. This chapter details the experimental methodology and presents results demonstrating how occlusion-aware selection enhances the quality of the reconstructed scene. The research presented in this chapter was described in [P4] and [P2].

Chapter 3:

Reducing the impact of lossy compression on Decoder-Side Depth Estimation focuses on the second research thesis. It introduces a novel correspondence search method designed to mitigate the impact of compression-related artefacts. By departing from conventional point-to-point comparisons in favour of a localised window-based approach, this method improves depth map accuracy when operating on encoded texture views. This research chapter was described in [P5]. The chapter also presents a technical contribution aimed at enhancing the depth estimation software. While not tied to a specific thesis, this chapter additionally describes the implementation of graph-cut-based optimisation techniques to accelerate the estimation process and improve the spatial consistency of the resulting depth maps. This research was described in [P3].

Chapter 4:

Input depth map assistance constitutes the core contribution of this dissertation and validates the third research thesis. It introduces two novel methodologies that leverage the MIV framework to integrate encoder-side information into the decoder-side estimation pipeline. Detailed experimental analysis confirms that these hybrid approaches yield superior synthesis quality while remaining within strict bitrate constraints. The research presented in this chapter was described in [P1] and [P6].

Chapter 5:

Summary of the dissertation provides a final synthesis of the research findings. It reviews the validation of the proposed theses, discusses the practical implications of the work, and suggests potential avenues for future research in the field of immersive media delivery.

2. VIEW SELECTION STRATEGIES FOR EFFICIENT IMMERSIVE VIDEO CODING

This chapter examines view selection as a key mechanism for effective data reduction in immersive video systems. Specifically, it investigates whether a selection strategy exists that consistently guarantees optimal virtual view quality. Sections 2.1 and 2.2 outline the transmission scenario analysed in the context of the MIV Main profile (illustrated in Fig. 2.1), while Section 2.3 presents an analysis of view selection for the DSDE scenario.

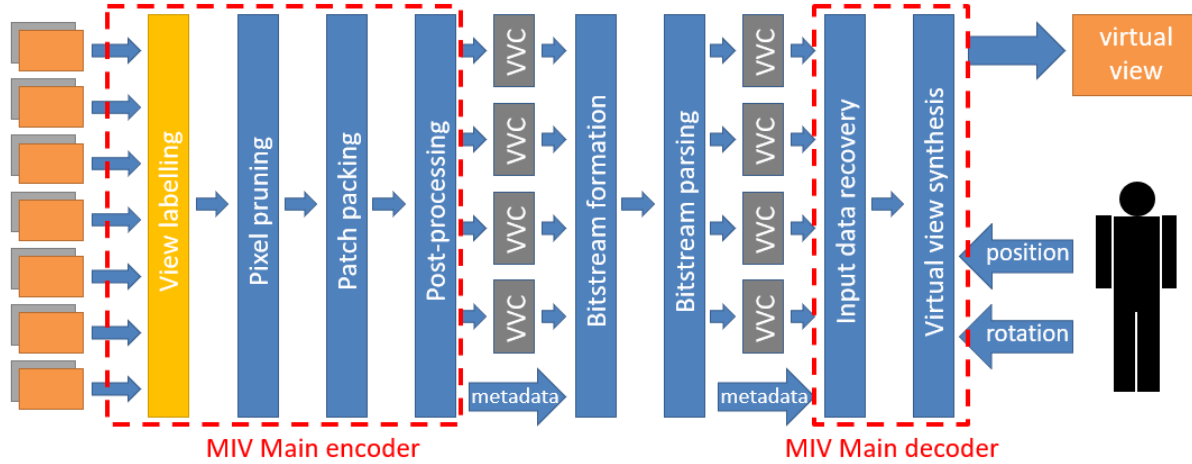


Figure 2.1. MIV Main block scheme. The component highlighted in yellow represents the stage subject to the modifications proposed in Sections 2.1 and 2.2.

Practical multicamera systems employ 10–20 cameras [7], generating vast amounts of data for processing and transmission. The possibility to immerse into the scene is achieved by rendering user-requested viewports [21, 22], which requires comprehensive 3D scene information.

As previously noted in Chapter 1, the MIV standard represents the state-of-the-art approach to the challenge of reducing immersive video data. Within this chapter, it is essential to examine its operational principles in detail, particularly how it employs a view selection mechanism designed to reduce data redundancy.

Before atlas creation, the MIV encoder undergoes a view selection process to determine which views should be integrated into the atlases (Fig. 2.1). During this process, a subset of the available input data is selected. The chosen views are those considered most critical from the perspective of a viewer and are designated as “base views”. Base views are incorporated into the atlases in their entirety (i.e., without any cropping or modification). A separate atlas subsequently contains the depth maps corresponding to these base views.

The remaining views that are not selected as base views are labelled as “additional views” and undergo a separate processing stage. As these additional views contain redundant data, they are pruned to remove it and therefore reduce the required bitrate. Following the pruning operation, the redundant information is discarded, leaving only the “patches”

which are integrated into the atlases as a patch mosaic. Texture patches are first packed into the available empty space within the full-view texture atlas. Any remaining patches are then packed into an empty texture atlas. The corresponding depth information (full views and patches) is packed into the remaining atlases [23].

Effective input view selection is crucial for immersive video quality and coding efficiency, regardless of the architecture employed. While previous studies addressed view selection, the focus of these studies was set on scenarios distinct from those explored in this dissertation.

In [24], the authors investigated how view selection affects virtual view synthesis in scenarios where the renderer has access to all available input views. In addressing the input view selection problem, the authors identified three primary constraints: first, the presence of occlusions, which vary depending on the positioning of cameras; second, the finite resolution of the video, which dictates, that in order to maintain the continuity of objects within the scene, the objects should be projected from views in which they exhibit their maximum width; and finally, the presence of non-Lambertian surfaces, which reflect light differently depending on the capture angle. Based on these limitations, it was proven that the highest quality of rendered views is achieved when the nearest left and right views are selected for synthesis. Therefore, in this scenario, it would be optimal to transmit only these two views. However, such a selection is only feasible if the position requested by the viewer is known prior to transmission.

In a practical immersive video system, where multiple viewers receive the same bitstream and independently select their own viewpoints [3], it cannot be assumed that the position of a viewer is known *a priori*. Consequently, there is a requirement to select input views in a manner that guarantees the highest average quality of the reconstructed content, regardless of the user’s specific perspective. Furthermore, an effective view selection method must be compatible with various input formats, as immersive video sequences are captured by a diverse range of multicamera systems. Therefore, the selection process should yield optimal results whether the sequence was captured by perspective cameras arranged in a planar matrix or by omnidirectional cameras positioned along an arc.

Section 2.1 describes and evaluates the impact of view selection on virtual view quality within a basic scenario featuring a linear camera arrangement. Following this, Section 2.2 focuses on a more complex scenario involving omnidirectional content, where the selection of views must align with how the user observes the immersive scene. Finally, drawing upon the results obtained in Sections 2.1 and 2.2, Section 2.3 addresses view selection for a Decoder-Side Depth Estimation scenario.

2.1. BASELINE TRANSMISSION SCENARIO

To accommodate arbitrary user positions in immersive video transmission, the view selection method in [24] must be extended. Given that the quality of a synthesised view is the highest when rendering is performed using the nearest left and right views, a simulation was conducted to evaluate this relationship.

The simulation assumed a practical linear multicamera system with a sustainable pixel rate [13] and 13 evenly distributed cameras [25]. Of the four views selected for transmission, views 1 and 13 were fixed to ensure reference availability for 100 virtual positions. The remaining two views were varied iteratively to determine their optimal positions. For each of the 100 virtual viewer positions, the total distance to the nearest left and right reference views was calculated. The resulting data is presented in Fig. 2.1.1, where the horizontal axis represents the index of the third real view utilised for synthesis, and the vertical axis indicates the index of the second real view.

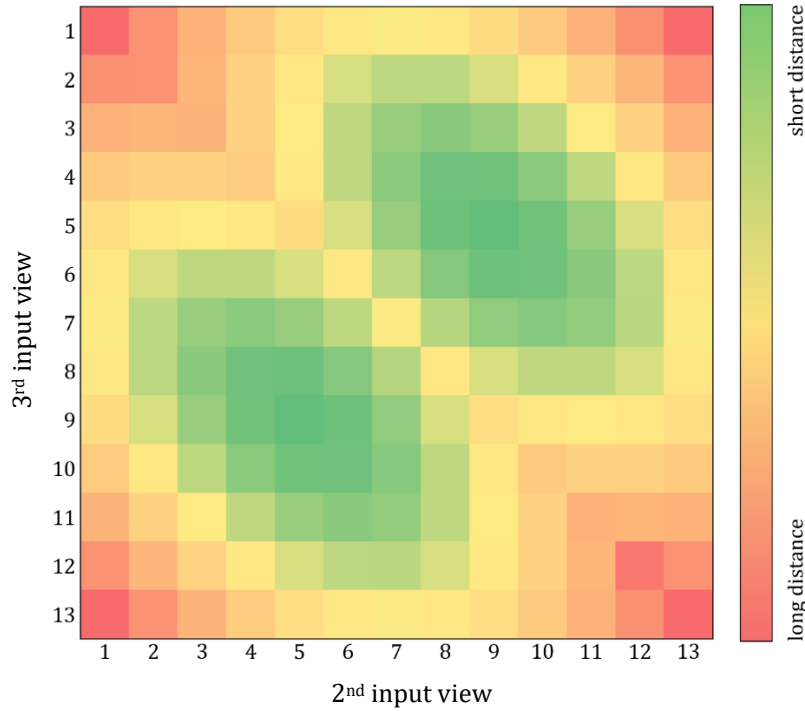


Figure 2.1.1. Total distance between the nearest left and the nearest right view calculated for all real views used for the experiment.

The green regions in Fig. 2.1.1 indicate arrangements in which the total distance between the virtual view and its neighbouring transmitted views is minimal. Contrarily, the red areas indicate camera arrangements with the largest baselines. These results clearly demonstrate that the distance is minimised when the input views selected for transmission are uniformly distributed.

While based on a linear arrangement and interpolation distances, these findings extend to more complex systems. To address the lack of direct quality analysis in the simulation, subsequent experiments used real-world data for empirical validation.

2.1.1. EXPERIMENTAL METHODOLOGY

To validate the aforementioned considerations, an experiment was conducted to assess the quality of virtual views synthesised using various combinations of input views. The synthesis was performed using the MPEG reference software – VVS [26].

The experiment was conducted using the *BBB Butterfly* and *BBB Flowers* sequences [27], each consisting of 79 views along an arc (v6–v84). This high camera density enabled comprehensive validation of the simulation results. Seven views (v6, v19, v32, v45, v58, v71, and v84) were designated as potential inputs, with the remainder serving as ground truth. Following MIV Common Test Conditions (CTC) [28], the quality of synthesised views was evaluated using two full-reference metrics: WS-PSNR and IV-PSNR [29, 30]. Mirroring the initial simulation, four views were selected for transmission. The first and last selected views, v6 and v84, were fixed, while the positions of the remaining two were varied to determine the optimal configuration.

2.1.2. RESULTS

Table 2.1.2.1 presents the mean IV-PSNR, the lowest IV-PSNR, and Δ IV-PSNR for the synthesised views of the *BBB Flowers* sequence. The results were averaged across 75 views (v6–v84), excluding the four input views which exhibit perfect fidelity. In the table, dark green and red shading highlight the highest and lowest quality results, respectively, for both mean and lowest IV-PSNR. Regarding the Δ IV-PSNR, green shading indicates the smallest variance between the minimum and maximum quality levels across all synthesised views, thereby denoting a consistent user quality of experience (QoE). Results for the *BBB Butterfly* sequence have been omitted from this dissertation due to their comparable trends; however, they remain available in [P4].

Mean IV-PSNR results show that the highest quality of synthesised view was achieved when views were distributed evenly. Lowest IV-PSNR results indicate that the even distribution of input views provides the highest quality even in the most unfavourable scenario.

Table 2.1.2.1. Objective quality results for synthesised views (averaged across 75 views) calculated for different combinations of transmitted input views for sequence *BBB Flowers*.

		Mean IV-PSNR [dB]					Lowest IV-PSNR [dB]					Δ IV-PSNR				
3 rd input view	v19		31.32	32.06	32.32	31.34		22.95	26.94	23.50	23.40		18.99	14.99	18.74	21.14
	v32	31.32		32.10	33.16	32.57	22.95		26.94	29.72	25.72	18.99		14.99	12.52	18.82
	v45	32.06	32.10		31.78	31.59	26.94	26.94		24.41	24.17	14.99	14.99		17.83	20.37
	v58	32.32	33.16	31.78		30.74	23.50	29.72	24.41		21.41	18.74	12.52	17.83		23.15
	v71	31.34	32.57	31.59	30.74		23.40	25.72	24.17	21.41		21.14	18.82	20.37	23.15	
		v19	v32	v45	v58	v71	v19	v32	v45	v58	v71	v19	v32	v45	v58	v71
2 nd input view																

Δ IV-PSNR results demonstrate that a uniform distribution of transmitted views minimises the variance between the minimum and maximum quality levels across all synthesised views. This effectively enhances the stability of the user experience, as the fluctuations in perceived quality are significantly reduced during virtual navigation within the scene.

In the preceding experiment, the conclusions drawn from the simulation were validated using the *BBB Flowers* and *BBB Butterfly* sequences, both of which were captured by perspective cameras in a linear arrangement.

However, a fundamental premise of this research is that the view selection process should be versatile. Specifically, it must remain effective regardless of camera type or the physical geometry of the multicamera system. To demonstrate this adaptability, the following section extends these considerations to the transmission of omnidirectional content.

2.2. OMNIDIRECTIONAL CONTENT TRANSMISSION SCENARIO

While Section 2.1 assumed a uniform viewpoint probability, real-world HMD usage shows that the visual attention of users is non-uniform. Saliency maps reveal that users focus primarily on the equatorial line, rarely gazing towards the poles [31, 32]. This concentration of attention causes the assumption of equal observation probability across all viewpoints to be invalid.

Consequently, the previously proposed view selection algorithm, which prioritises input views based on maximum distances between them, may result in a sub-optimal selection for omnidirectional content. For instance, the views captured by cameras oriented towards the poles might be selected at the expense of the views containing essential information regarding the equatorial regions, which are of greater importance to the viewer (Fig. 2.2.1).

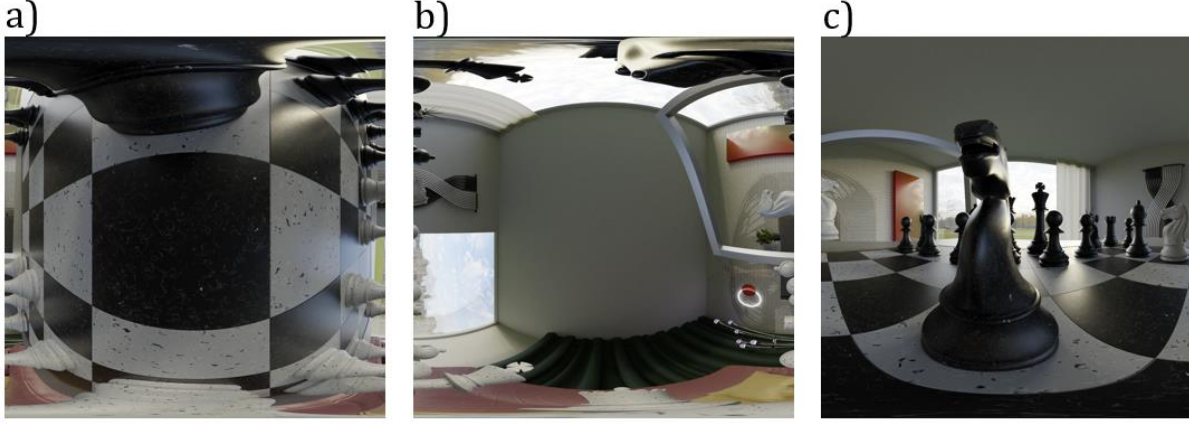


Figure 2.2.1. The three views of Chess sequence selected for transmission by the reference software.

To account for non-uniform visual significance, a modified view selection algorithm is proposed. It down-weights vertical camera distance to prioritise horizontal views, where attention is concentrated. This contrasts with the previously used TMIV [33] method, which selects views by maximising the total distance between them:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (r_{i,j}^z)^2}, \quad (2.2.1)$$

where $r_{i,j}^x$, $r_{i,j}^y$ and $r_{i,j}^z$ are the distances between views i and j along three axes of the global coordinate system

The proposed modification to (2.2.1) accounts for the non-uniform probability of the viewing direction by reducing the significance of the vertical distance. To achieve this, the camera distances calculated during the view selection process are rendered non-homogeneous:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (w \cdot r_{i,j}^z)^2}, \quad (2.2.2)$$

where w is the inhomogeneity coefficient.

The proposed modification was put forward in the standardisation document [34] and was appreciated by the experts of the ISO/IEC JTC1/SC29/WG4 MPEG VC group by including it to the further versions of Test Model for MIV [35].

A series of experiments was conducted to evaluate the influence of this modified view selection algorithm on both the objective and subjective quality of the resulting immersive video. The experimental methodology is detailed in Section 2.2.1, while the corresponding results and their analysis are presented in Section 2.2.2.

2.2.1. EXPERIMENTAL METHODOLOGY

The experiments were conducted in accordance with the MIV CTC [28]. However, the test set was restricted exclusively to omnidirectional sequences. The characteristics of the sequences included in the experiment are detailed in Table 2.2.1.1 below.

Table 2.2.1.1. Test sequences used in experiments.

Sequence name	Camera type	Content type	Resolution	Source
Chess	Semi-spherical	Omnidirectional	2048×2048	[36]
ChessPieces	Semi-spherical	Omnidirectional	2048×2048	[37]
ClassroomVideo	Spherical	Omnidirectional	4096×2048	[38]
Hijack	Semi-spherical	Semi-omnidirectional	4096×2048	[39]
Museum	Semi-spherical	Omnidirectional	2048×2048	[39]

Using TMIV 14 as a baseline, the proposed method replaced (2.2.1) with (2.2.2) using an inhomogeneity coefficient of 0.4 which was determined through preliminary testing. Sequences were compressed with the VVC encoder across five rate points (RP1–RP5, 2.5–20 Mbps) and evaluated across three configurations (MIV, MIV View, and DSDE) per MIV CTC. Objective quality was measured using WS-PSNR and IV-PSNR.

Subjective quality was assessed by 45 naïve viewers using MIV CTC pose-traces [13] and the Pair Comparison (PC) method (Rec. ITU-T P.910 [40], ITU-R BT.500 [41]). Participants used a seven-point CCR scale (-3 to +3). To prevent fatigue, only RP1, RP3, and RP5 were tested, focusing exclusively on cases where the proposed algorithm altered the camera configuration from the baseline (Table 2.2.1.2).

Table 2.2.1.2. Overview of the sequences and different MIV configurations. “✖” indicates the scenario in which view selection result was the same as with the unmodified TMIV 14 software.

Sequence name	MIV Configuration		
	MIV Main	MIV View	MIV DSDE
Classroom	✖	✖	✖
Museum	✓	✓	✓
Hijack	✖	✖	✓
Chess	✖	✖	✓
ChessPieces	✖	✖	✓

The results for subjective quality evaluation are presented as an average quality change caused by the proposal and the 95% confidence interval, calculated according to ITU-R recommendations [41]:

$$CI = 1.96 \cdot \frac{SD}{\sqrt{N}}, \quad (2.2.1.1)$$

where CI is a confidence interval, SD stands for standard deviation and N indicates the number of viewers.

2.2.2. RESULTS

SUBJECTIVE RESULTS

Fig. 2.2.2.1 presents the results of the subjective quality evaluation of MIV DSDE across different test sequences. In three out of the four test sequences, a quality gain induced by the proposed view selection modification is observed. Furthermore, this improvement in quality is statistically significant in seven out of nine cases. The only sequence for which the quality remained unchanged is Hijack. However, it should be emphasised that the proposed method did not cause any degradation in quality.

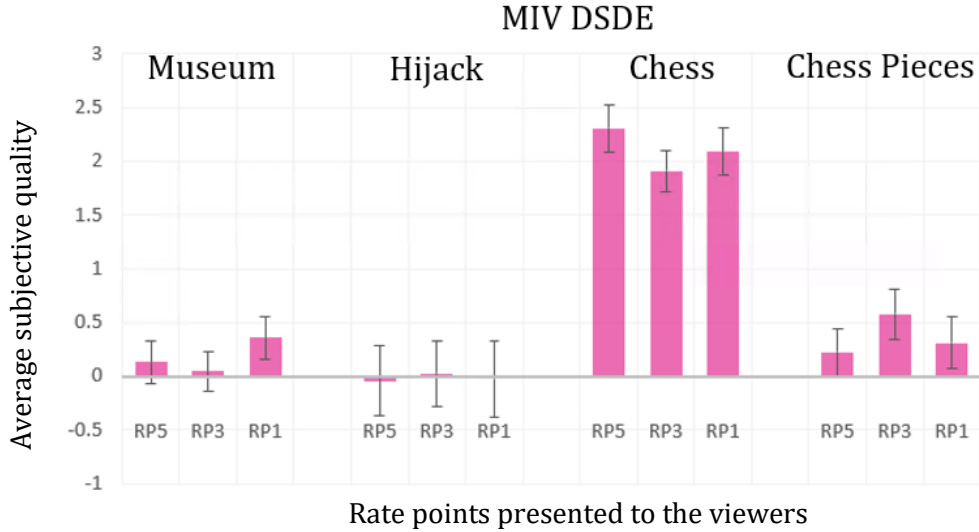


Figure 2.2.2.1. Subjective results for four sequences in MIV DSDE configuration.

OBJECTIVE RESULTS

The objective quality results were evaluated and presented as Bjøntegaard Delta (BD) metrics [42]; specifically, the BD-rate (Table 2.2.2.1) and BD-PSNR (Table 2.2.2.2). These metrics facilitate a direct comparison of the coding efficiency between the reference and the proposed view selection methods across the entire range of tested bitrates.

Table 2.2.2.1. Objective metrics BD-rates obtained for four lowest rate points (Low-BR) and for the four highest rate points (High-BR); “---” denotes that the BD-rate calculation was not possible because of non-overlapping curves.

Sequence name	MIV configuration	WS-PSNR		IV-PSNR	
		High-BR BD-rate	Low-BR BD-rate	High-BR BD-rate	Low-BR BD-rate
Museum	MIV Main	-3.4%	0.2%	0.9%	2.3%
	MIV View	-4.6%	-5.5%	-7.7%	-3.2%
	MIV DSDE	57.2%	32.5%	13.1%	10.2%
Hijack	MIV DSDE	---	---	---	---
Chess	MIV DSDE	---	---	---	---
ChessPieces	MIV DSDE	---	---	---	---

Table 2.2.2.2. Objective metrics BD-PSNRs obtained for four lowest rate points (Low-BR) and for four highest rate points (High-BR).

Sequence name	MIV configuration	WS-PSNR		IV-PSNR	
		High-BR BD-PSNR	Low-BR BD-PSNR	High-BR BD-PSNR	Low-BR BD-PSNR
Museum	MIV Main	0.4%	0.1%	-0.0%	-0.2%
	MIV View	0.2%	0.3%	0.8%	0.3%
	MIV DSDE	-1.6%	-1.3%	-0.8%	-0.6%
Hijack	MIV DSDE	-13.2%	-13.2%	-11.1%	-11.1%
Chess	MIV DSDE	-6.4%	-6.2%	-5.9%	-5.5%
ChessPieces	MIV DSDE	-11.3%	-11.0%	-8.3%	-7.9%

Counter-intuitively, the objective results indicate that, on average, the proposed method performs less effectively than the reference view selection algorithm implemented in TMIV 14. However, it must be emphasised that the data presented in Tables 2.2.2.1 and 2.2.2.2 were derived by averaging the IV-PSNR and WS-PSNR values across all synthesised views, including the “base views”. This global averaging often obscures localised quality improvements. A representative example of this phenomenon is provided in Table 2.2.2.3, which details the specific IV-PSNR values for each of the ten individual views within the Chess sequence.

Table 2.2.2.3. IV-PSNR of synthesised views, RP1, similar bitrate for both approaches (21.3 Mbps for TMIV 14 and 20.8 Mbps for proposed); Chess sequence, MIV DSDE configuration.

View	IV-PSNR [dB]		
	TMIV 14	Proposed	Δ
v0	57.00	36.17	-20.84
v1	37.84	37.99	0.15
v2	56.78	56.76	-0.02
v3	33.20	56.28	23.09
v4	56.73	56.44	-0.29
v5	56.01	55.93	-0.08
v6	38.85	56.11	17.26
v7	56.02	56.04	0.03
v8	57.08	57.18	0.10
v9	56.34	30.08	-26.26
Average (all views)			-0.69
Average (only non-base views)			-0.02

As demonstrated in Table 2.2.2.3, the average IV-PSNR for the "additional views" maintained parity with the quality levels achieved by the unmodified TMIV 14. Significant quality degradation showed exclusively in views v0 and v9 which correspond to cameras capturing the ceiling and the floor. As established previously, these views are of marginal importance to the viewer, and thus their degradation does not negatively impact the overall user experience.

2.3.DECODER-SIDE DEPTH ESTIMATION SCENARIO

In this Section, the view selection for the DSDE scenario is examined (Fig. 2.3.1).

In immersive video systems employing the Decoder-Side Depth Estimation (DSDE) approach, it is often necessary to transmit only a subset of the input views due to bitrate and pixel-rate constraints [13]. Although it was demonstrated in previous sections that a uniform selection of views was optimal for the previously analysed scenarios, this strategy does not inherently guarantee the highest reconstruction quality at the decoder side. Within the DSDE framework, view selection must account for the fact that depth estimation is performed using decoded input views. Consequently, the input views should be selected in a manner that ensures the maximum possible quality of the estimated depth maps. Therefore, the primary objective of the view selection method in the DSDE

scenario is to maximise the fidelity of the synthesised virtual views while adhering to the constraints regarding the number of transmitted views, as well as the overall bitrate and pixelrate.

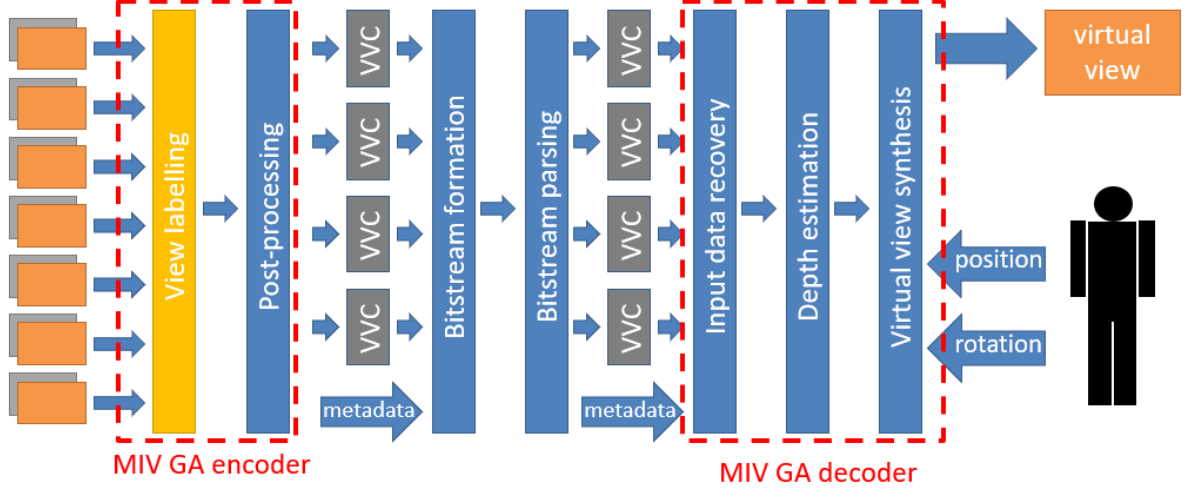


Figure 2.3.1. MIV GA block diagram. The component highlighted in yellow represents the focus of the proposed modifications.

The study in [7] explores the impact of view selection on virtual view quality in simple free-navigation systems. The authors observed that reducing the number of views increases the summarised inter-view distance, which adversely affects depth estimation accuracy. Theoretically, the depth of a pixel can only be computed if it is visible to at least two cameras. Increasing the baseline distance typically leads to a higher proportion of occlusions within the scene. For such occluded regions, depth cannot be explicitly estimated and must be interpolated or extrapolated. Furthermore, greater spatial disparity between cameras often results in divergent lighting conditions and reflections, further degrading the depth estimation process. Consequently, it was hypothesised that a short baseline ensures that only a small fraction of the scene is occluded while maintaining similar photometric conditions between views. Conversely, a short baseline inherently limits the triangulation accuracy of the estimated depth maps. The study evaluated the impact of both camera distance and occlusion levels on the final virtual view quality. The results differed between gains derived from baseline adjustment and those from occlusion reduction. While long baselines can improve depth resolution, uniform layouts in complex scenes often suffer from a high number of occlusions [3, 43], leading to horizontal displacements in the synthesis. In typical two-step synthesis pipelines, such regions require inpainting, which results in lower visual quality [44, 45]. For sequences with insignificant occlusions (<25%), the optimal results were achieved using a uniform camera distribution. However, for sequences where occlusions exceeded 25%, a camera pairing strategy yielded superior quality. Notably, the experiments in [7] utilised the DERS algorithm [43] which is currently considered legacy software, and were conducted prior to the introduction of the MIV DSDE profile.

Building upon the findings in [7], this work proposes a view selection method specifically tailored for DSDE. This method relies on the analysis of occlusions across input views.

The occluded regions are detected through the reprojection of 3D points between views. This process identifies pixels for which depth estimation is not possible because they are visible in only a single view. In the proposed approach, occlusion detection is provided using the highly optimised reprojection module from the TMIV encoder [33]. This choice ensures a negligible overhead in the encoder-side computation. Initially, a selected number of evenly spaced input views is selected. Each of these views is afterwards reprojected onto the remaining views. For every pixel, the reprojected depth is compared with the original depth in the target view. A pixel is deemed non-occluded if the reprojected depth does not exceed the original depth. This procedure yields a direct estimation of the occlusion ratio, defined as the proportion of regions where reliable depth estimation cannot be performed.

It was decided to perform the analysis of the occlusion once per group of pictures (GOP), specifically on the initial frame. The resulting view selection remains fixed for the entire GOP. The proposed approach prevents fluctuations in view selection within the GOP, which would otherwise significantly degrade video compression efficiency.

Following the occlusion ratio analysis, the algorithm adopts one of the two view selection strategies. If the analysed sequence exhibits a high occlusion ratio, the algorithm selects multiple camera stereopairs rather than uniformly arranged single cameras, as illustrated in Fig. 2.3.2.

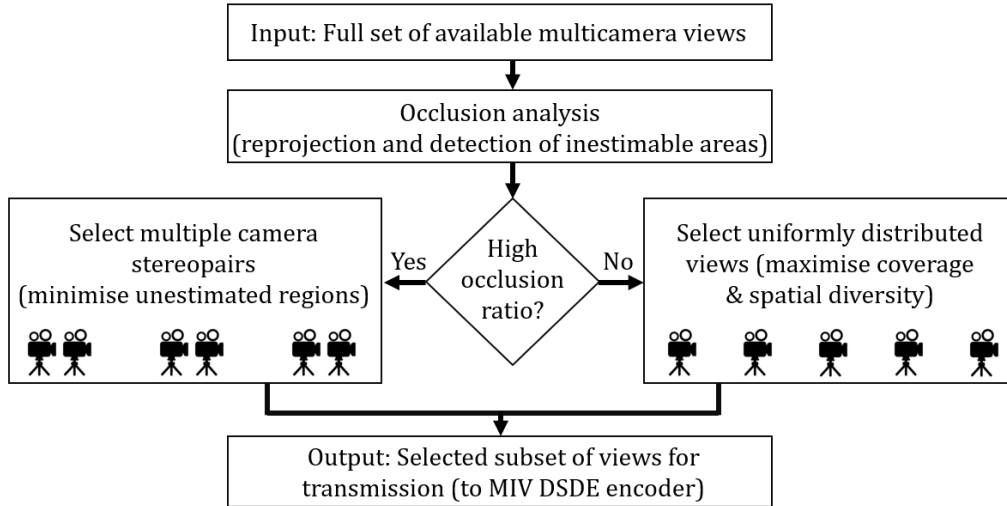


Figure 2.3.2. Scheme of the proposal.

As demonstrated in [P4], this approach reduces the number of non-estimable regions, thereby enhancing the overall quality of the synthesised views. Contrarily, in cases where the sequence contains a low proportion of occlusions, uniformly distributed views are selected to maximise scene coverage and spatial diversity.

The proposed approach is inherently compatible with existing MIV encoders and requires no modifications to the decoder side. Furthermore, by selecting views captured as stereopairs rather than evenly distributed cameras, the method can potentially reduce the total bitrate of the immersive video. Similar neighbouring views integrated into a single atlas can be compressed more efficiently, particularly when screen-content coding tools are employed by the video encoder [46].

2.3.1. EXPERIMENTAL METHODOLOGY

To evaluate the proposed method, two experiments were conducted: the MIV Experiment, using MIV CTC sequences, and a Supplementary Experiment utilising legacy FTV sequences. Both followed the MIV CTC pipeline, employing TMIV for atlas construction and synthesis, VVenC/VVdeC for VVC coding [47], and IVDE [48] for Decoder-Side Depth Estimation. The Supplementary Experiment has been omitted from this dissertation, given that its outcomes align with the findings of the MIV Experiment. For detailed results, the reader is referred to [P2].

To isolate the impact of horizontal baselines on coding and synthesis quality, the study was restricted to a one-dimensional camera layout (line or arc) with four views packed into a single atlas. This controlled, four-view configuration represents the simplest non-trivial case for assessing camera pairing strategies without confounding factors from multidimensional rigs or complex atlas setups. By establishing these fundamental conclusions in a controlled environment, the findings can subsequently be generalised to more complicated scenarios.

MIV Experiment

Six sequences from MIV CTC set satisfied the specific requirements of this study, as they were captured using multi-camera systems with linear or arc-shaped arrangements.

Table 2.3.1.1. Test sequences used in the MIV experiment.

Sequence Name	Number of views	Resolution	Source
Carpark	9	1920×1088	[49]
Street	9	1920×1088	[49]
CBABasketball	34	1920×1080	[51]
Frog	13	1920×1080	[25]
Fencing	9	1920×1088	[50]
MartialArts	15	1920×1080	[52]

To evaluate the influence of camera spacing, a uniformity coefficient U was introduced:

$$U = \frac{b}{d}, \quad (2.3.1.1)$$

where U is the uniformity coefficient, b stands for the baseline of each stereopair and d is the distance between two stereopairs.

To illustrate the components of the uniformity coefficient defined in (2.3.1.1), Fig. 2.3.1.1 depicts various camera baseline configurations.

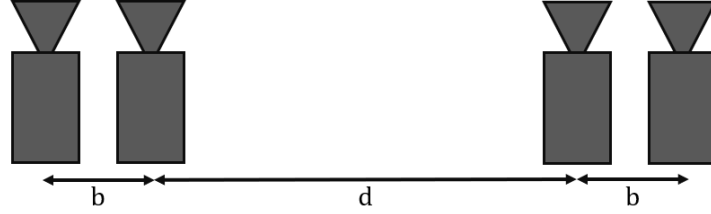


Figure 2.3.1.1. Analysed camera arrangement; b – stereopair baseline (the same for both stereopairs), d – distance between two stereopairs; uniformity coefficient describes the ratio between b and d .

The uniformity coefficient quantifies the regularity of camera placement within the setup. A coefficient of 1 denotes a perfectly uniform distribution with equal distances between all neighbouring cameras, while lower values indicate non-uniform layouts, such as those featuring clustered stereo pairs. On the other hand, a coefficient greater than 1 represents configurations where the distance between two camera pairs is smaller than the baseline of each individual stereo pair, for instance, an arrangement with a central stereo pair flanked by single camera on each side.

Quality evaluation was conducted using the IV-PSNR metric [53]. Rate-distortion performance was assessed by calculating BD-IVPSNR values. This metric was selected over BD-rate because the rate-distortion (RD) curves did not always overlap across different configurations. Thus, BD-IVPSNR ensured a more consistent and robust comparison of the results.

Furthermore, an occlusion analysis for each test sequence was done to quantify the proportion of scene regions not visible across multiple input views. This data facilitated an analysis of how camera arrangement impacts depth estimation and view synthesis quality across varying levels of scene complexity.

2.3.2. RESULTS

The rate-distortion curves presented in Fig. 2.3.2.1 confirm the correlation between the arrangement of selected real views and the occlusion levels within the scene. For sequences with a higher percentage of occlusions (Table 2.3.2.1), the proposed view selection strategy consistently achieves a higher IV-PSNR for a given bitrate.

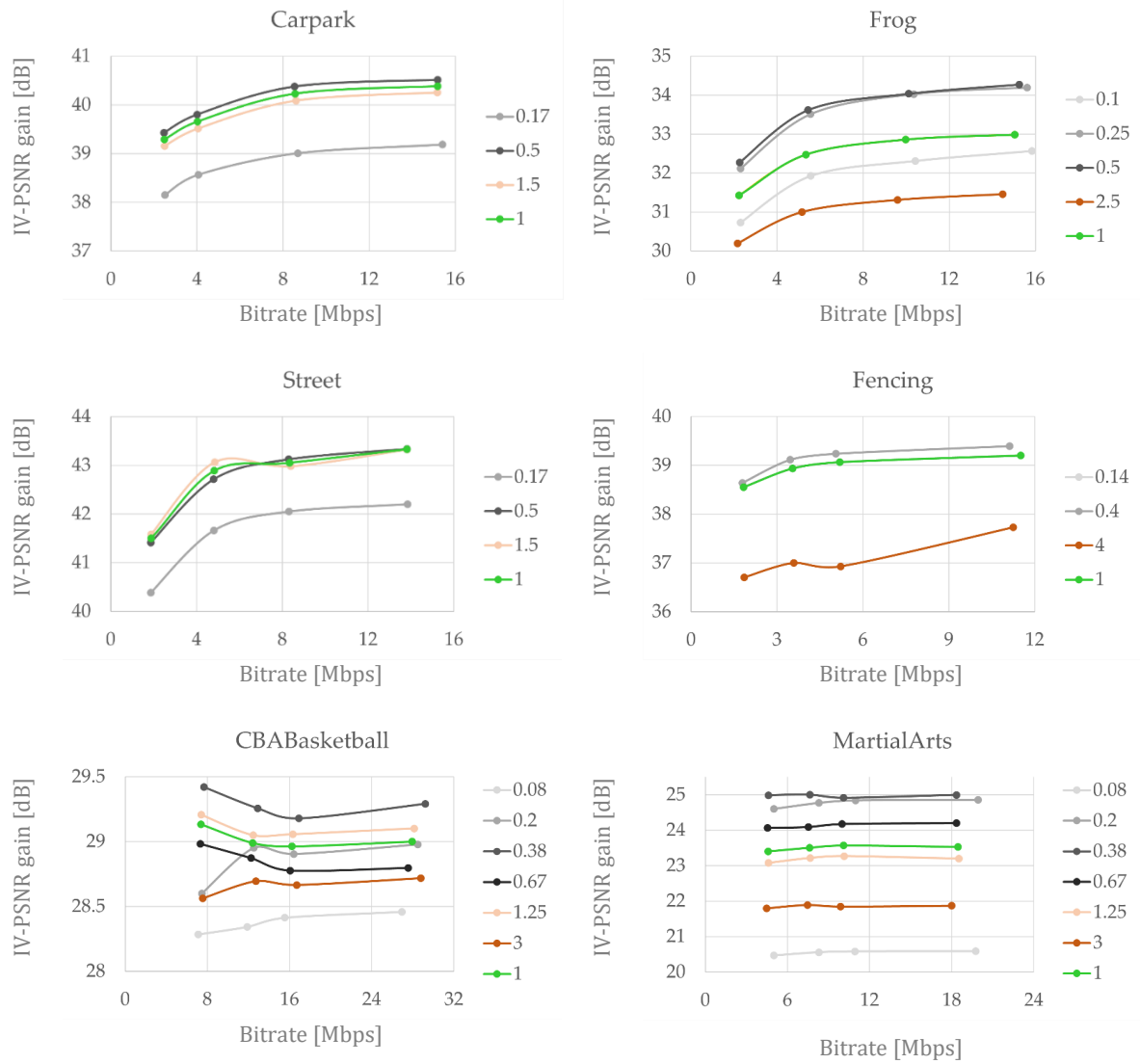


Figure 2.3.2.1. RD-curves of the immersive video system compared to uniform view distribution (uniformity coefficient = 1, **green line**); **grey lines**: two pairs of cameras (uniformity coefficient < 1), **red lines**: a pair of cameras in the middle and two single cameras at each side (uniformity coefficient > 1).

Table 2.3.2.1. Percentage of occluded areas in test sequences.

Sequence name	Occluded areas [%]
Frog	13.47
Carpark	6.13
Street	2.51
Fencing	11.54
CBABasketball	11.29
MartialArts	21.47

Figure 2.3.2.2 summarises these BD-IVPSNR gains as a function of the uniformity coefficient. The results confirm that pairing cameras with a reasonable baseline (specifically when the uniformity coefficient ranges from 0.2 to 0.7) leads to a measurable quality gain for the majority of the tested sequences, particularly those with a higher percentage of occlusions.

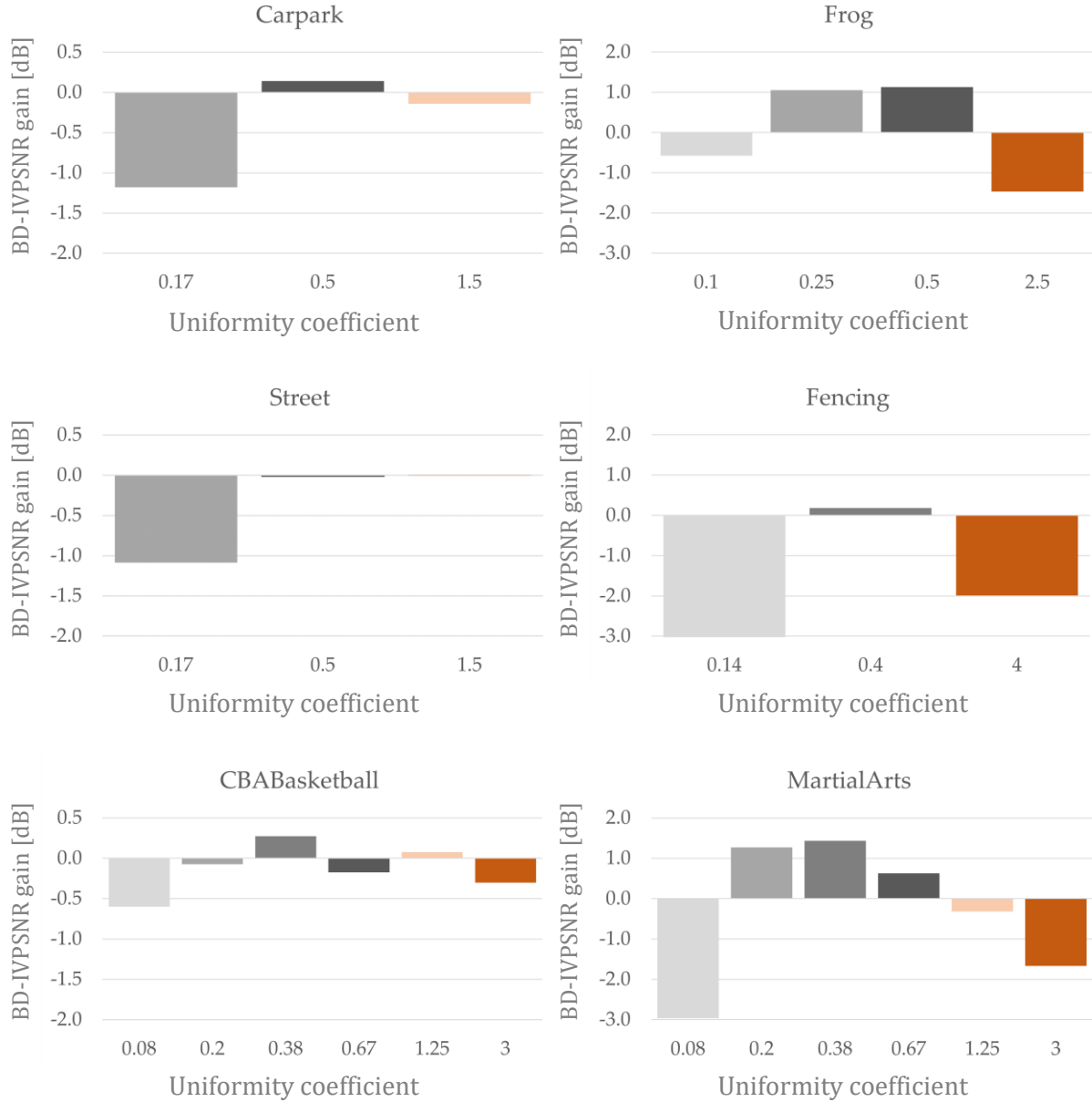


Figure 2.3.2.2. BD-IVPSNR of the immersive video system compared to uniform view distribution (uniformity coefficient = 1); **grey bars**: two pairs of cameras (uniformity coefficient < 1), **red bars**: a pair of cameras in the middle and two single cameras at each side (uniformity coefficient > 1).

For scenes with a minimal proportion of occlusions (*Carpark* and *Street*), the BD-IVPSNR gains resulting from camera pairing remain negligible. In sequences with limited occlusions, all cameras observing the scene capture nearly identical content, hence, regions with non-estimable depth are not significant across all camera arrangements. As

a result, camera pairing does not introduce additional geometric cues that would significantly impact the improvement of depth estimation or view synthesis. Contrarily, uniform camera distribution (as described earlier and in [13]) minimises issues related to the finite resolution of depth maps and the presence of non-Lambertian surfaces within the scene. Therefore, for scenes with limited occlusions, the proposed view selection method intentionally selects uniformly distributed views.

The results demonstrate that the proposed view selection method yields significant improvements when the proportion of occluded regions exceeds approximately 10% (Table 2.3.2.1). In such instances, pairwise camera grouping enhances the reconstruction of occluded areas and reduces geometric distortions in the synthesised views. On the other hand, for sequences with minimal occlusion (*Carpark* and *Street*), a uniform camera distribution remains sufficient. In these cases, camera pairing provides no significant quality gains, as demonstrated in Table 2.3.2.2 and Fig. 2.3.2.3.

Table 2.3.2.2. Gain of camera pairing (compared with the uniform camera arrangement).

Sequence name	BD-IVPSNR Gain [dB]
Frog	1.14
Carpark	0.11
Street	-0.03
Fencing	0.20
CBABasketball	0.27
MartialArts	1.44

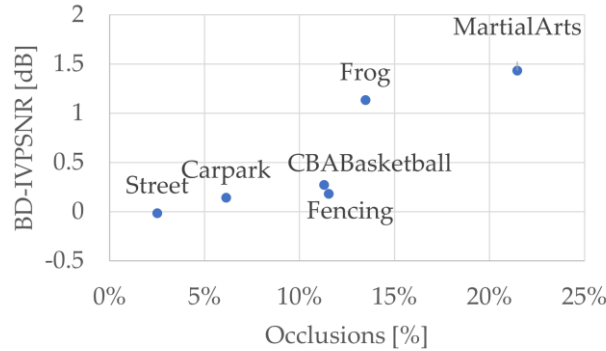


Figure 2.3.2.3. BD-IVPSNR gain from camera pairing (compared with the uniform camera arrangement) as a function of occlusions.

Results show that camera arrangement highly influences immersive video quality. The proposed pair-wise strategy balances occlusion handling and depth estimation precision, particularly in complex scenes where it mitigates occlusion artefacts.

It should be highlighted that the 10% threshold is consistent across the presented MIV Experiment and further confirmed by the supplementary experiment that can be found here [P2]. This threshold is lower than the corresponding threshold reported in [54], where it exceeded 20%. This discrepancy originates from the evolution of the processing

tools employed for depth estimation and view synthesis. The authors of [23] utilised the DERS algorithm [43] for depth estimation and VSRS [55] for virtual view synthesis. In contrast, the current dissertation employs IVDE 7.0 [48] and the TMIV view weighting synthesiser (VWS) [33]. Both IVDE and VWS were developed as successors to DERS and VSRS, introducing critical enhancements such as inter-view, and temporal consistency (IVDE) and the capability to synthesise views using more than two input views (VWS). Therefore, these contemporary tools provide more robust depth estimation, with improved handling of challenging or weakly textured regions, and significantly higher-quality virtual view synthesis. Such advancements reduce the sensitivity to missing information, thereby shifting the effective threshold at which stereopair grouping becomes advantageous.

COMPUTATIONAL OVERHEAD

The proposed method follows the standard TMIV 16.0 workflow [33], introducing only a high-speed occlusion analysis step using the already existing reprojection module. The occlusion ratio is determined by reprojecting four evenly spaced views to identify pixels visible to fewer than two cameras (regions where depth estimation is unfeasible). To minimise overhead, this analysis is performed only once per GOP (initial frame), with the selection fixed for the remainder of the period.

As shown in Table 2.3.2.3, the encoding time increase is negligible ($<0.5\%$). Moreover, the decoding time remains unaffected, which is a critical factor for practical viability.

Table 2.3.2.3. Computational time of the proposed approach compared to the unmodified TMIV encoder in DSDE configuration; results averaged over all sequences.

	MIV Experiment
Original encoding time (per GOP)	58.25 s
Occlusion analysis time (per GOP)	0.255 s
Computational time increase	0.44%

The presented results demonstrate that the proposed method is computationally efficient and viable for real-world immersive video encoding, offering a reliable view selection strategy with a negligible impact on overall processing time.

Overall, the results presented confirm that the spatial arrangement of transmitted views has a measurable impact on the visual quality experienced by a viewer within a DSDE immersive video system. Furthermore, the findings demonstrate that the optimal spatial arrangement of these views is dependent on scene characteristics, specifically the proportion of occlusions. Therefore, results of research presented in this section provide empirical validation for the first thesis presented in Chapter 1: *"The selection of views transmitted for Decoder-Side Depth Estimation should depend on scene characteristics, specifically the proportion of occlusions."*

3. REDUCING THE IMPACT OF LOSSY COMPRESSION ON DECODER-SIDE DEPTH ESTIMATION

This chapter focuses on the depth estimation process on the decoder side (Fig. 3.1). Firstly it presents a method developed to mitigate the adverse impact of image coding artefacts on inter-view matching errors during Decoder-Side Depth Estimation. Additionally, the chapter outlines a modification to the depth estimation reference software aimed at accelerating Decoder-Side Depth Estimation process.

To establish a context for the first proposed solution, a preliminary analysis of the fundamental sources of these inter-view matching errors is required. Understanding how compression-induced artefacts affect the correspondence between different views is essential for evaluating the effectiveness of the subsequent depth estimation enhancement.

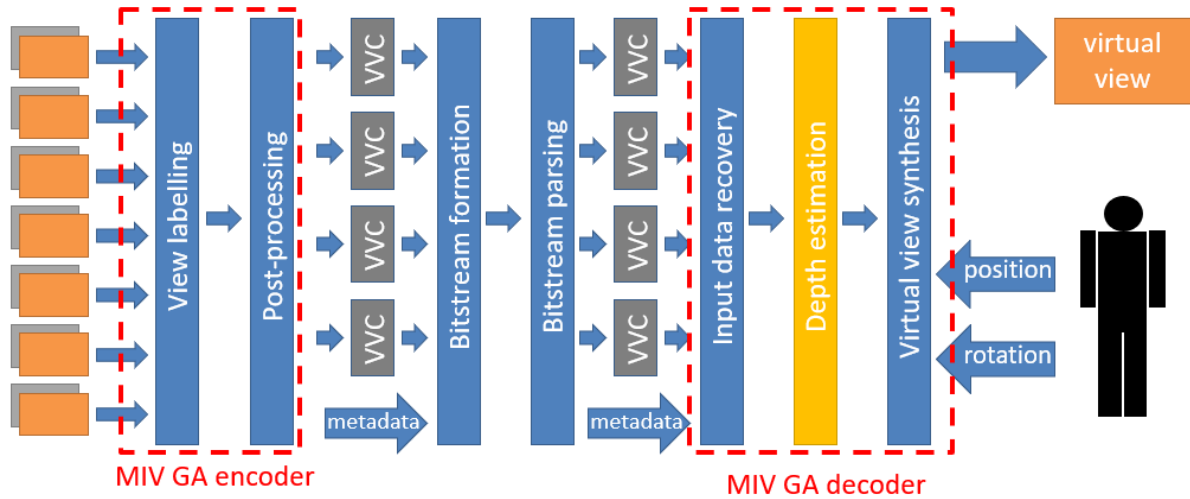


Figure 3.1. MIV GA block diagram. The component highlighted in yellow represents the focus of the proposed modifications.

As defined in Chapter 1, the MIV Geometry Absent profile relies on depth estimation performed at the decoder side, utilising views transmitted from the encoder. Consequently, these input views are inherently affected by compression artefacts (Fig. 3.2). Regions characterised by sharp edges and complex textures are particularly vulnerable to such coding errors, which can severely compromise the precision of the depth estimation process.



Figure 3.2. Visual comparison of a representative video frame across two extreme compression levels: low compression (left) and high compression (right).

Accurate edge representation necessitates the preservation of high-frequency harmonics. However, during the coding process, particularly when transform coefficients undergo quantisation, these high-frequency components are discarded. This loss results in blurring or, in more severe instances, the spatial displacement of edges within the image. Similar spatial shifts may also be found in highly compressed videos, primarily due to an extensive reliance on temporal prediction mechanisms [56].

Beyond compression artefacts, the quality of the estimated depth maps is further compromised by the limited precision of camera parameters, specifically regarding the intrinsic and extrinsic camera parameters [57, 58], as well as the complexity of camera distortion models [59]. Furthermore, computational constraints, such as the limitations of floating-point representation and cumulative rounding errors [60], introduce additional numerical inaccuracies. Finally, the reliability of the inter-view correspondence search is significantly degraded by noise inherent in the input sequences of natural scenes [61], which influences the original colour of pixels.

Following the identification of primary error sources in Decoder-Side Depth Estimation, various strategies have been proposed to mitigate these challenges. Hybrid similarity measures, integrating census or rank transforms with SAD/SSD metrics, have proven effective for noisy sequences [62], while SSIM-based approaches exhibit superior robustness against radiometric distortions [63]. Accuracy in problematic regions such as occlusions is further enhanced by dynamically adapting similarity weights to local features [64] or employing matching windows with variable geometries [65].

Temporal consistency, essential for virtual view quality, is addressed either through the integration of temporal information directly into the matching metric [66] or via sophisticated post-processing filters [67]. Alternative methodologies explore the transform domain to achieve exposure invariance [68]. However, their efficacy may be compromised by independent view compression in simulcast scenarios. Furthermore, while MST-based methods [69] and feature detectors like SURF [70] or KAZE [71] offer robustness in complex scenes, their performance on compressed or noisy data remains insufficiently documented.

To the best of the author’s knowledge, a significant research gap remains: current state-of-the-art matching methods do not specifically account for the unique characteristics of compression-induced artefacts. While existing techniques address noise, they lack dedicated mechanisms tailored to the spatial distortions and frequency-domain alterations inherent in modern video coding standards.

3.1. THE PROPOSAL OF POINT-TO-BLOCK MATCHING

The proposed method, hereafter referred to as "Point-to-block Matching", was developed with a primary focus on the optimisation of depth map quality. To establish the theoretical foundations of this approach, it is essential to examine the nature of the depth estimation problem. In most contemporary frameworks, depth estimation is formulated as a problem of cost-function minimisation [72]. In its most prevalent form, this objective function is defined as follows:

$$E(\bar{d}_p) = \sum_{p \in P} D_p(\bar{d}_p) + \sum_{p \in P} \sum_{q \in Q} V_{p,q}(\bar{d}_p, \bar{d}_q), \quad (3.1.1)$$

where:

- P – set of pixels of the input view,
- p – pixel of the input view,
- $\bar{d}_p$ – currently considered depth of the pixel p ,
- D_p – data term that represents the cost of assigning the depth d_p to the pixel p ,
- Q – currently considered depth of the pixel q ,
- q – pixel in the neighbourhood of the pixel p ,
- $\bar{d}_q$ – currently considered depth of the pixel q ,
- $V_{p,q}$ – smoothness term that represents the intra-view discontinuity cost of assignment of the depth d_p to the pixel p and depth d_q to the pixel q .

The Point-to-block Matching method modifies the data term D_p , which accounts for the correspondence between pixels across different views. Conventionally, D_p utilises a similarity metric to evaluate the relationship between pixel p and its corresponding pixel p' in a neighbouring view, where the correspondence is determined for the considered depth value $\bar{d}_p$.

In the proposed method, the data term does not rely on a direct point-to-point comparison. Instead, it evaluates pixel p against a set of pixels within a small block of pixels R that surrounds the initial pixel p' . The algorithm then selects the specific pixel within this region that yields the minimum matching error (or maximum similarity) relative to the pixel p . Consequently, if the Sum of Absolute Differences (SAD) metric is used, the expression is formulated as follows:

$$D_p(\bar{d}_p) = \min(SAD(p, r)), \quad r \in R \quad (3.1.2)$$

To illustrate this mechanism with a practical example, consider a scenario in which a 3×3 pixel window is selected to represent the region R . In this instance, the data term D_p is determined by identifying the minimum value among nine separate SAD calculations. These computations correspond to the central projection p' and its eight immediate neighbours within the defined local neighbourhood. By evaluating this local search space, the algorithm effectively compensates for minor spatial displacements caused by compression-induced edge shifting.

The Point-to-block Matching algorithm was integrated into the Immersive Video Depth Estimation (IVDE) reference software. IVDE framework operates on the mathematical principles established in (3.1.1.) The implementation was carried out in C++, and required a strategic modification of the matching error calculation to incorporate the local spatial search mechanism.

While the transition from point-to-point correspondence to a Point-to-block Matching method increases the computational overhead of the depth estimation process, requiring nine evaluations instead of one, this added complexity is mitigated by the fact that the similarity calculations are inherently data-parallel. Consequently, the algorithm can be efficiently accelerated through parallelisation.

It must be emphasised that the proposed framework is entirely agnostic to the choice of similarity measure. Any conventional or advanced metric can be seamlessly integrated into the Point-to-block method. Due to its modular nature, this method has the potential to enhance any existing depth estimation architecture by significantly augmenting its robustness to the localised spatial shifts inherent in encoded input sequences.

3.1.1. RESEARCH DESIGN AND METHODOLOGY

To evaluate the proposed solution, a series of experiments was conducted across diverse configurations of the depth estimation software. The IVDE framework operates using data and parameters specified in configuration files, with baseline set of values defined by the MIV Common Test Conditions (CTC). To assess the software’s performance under controlled yet challenging scenarios, these parameters were systematically varied. The standard parameter set typically employed in the IVDE environment is detailed in Table 3.1.1.1.

Table 3.1.1.1. IVDE standard parameter set.

Parameter value	Parameter meaning
256 levels of depth	Depth levels are defined as a series of planes oriented parallel to the coordinate plane of the multiview camera system. Each plane represents a discrete depth value within the estimable range. A reduction in this number diminishes the depth resolution, forcing the algorithm to assign identical depth values to objects situated at slightly different distances from the camera plane. Consequently, such downsampling of the depth search space can lead to the loss of fine geometric detail and the introduction of depth-layering artefacts.
50 000 superpixels per view	The number of superpixels serves as a primary control mechanism for managing the trade-off between the visual fidelity of the estimated depth maps and the computational complexity of the estimation process. By adjusting the superpixel density, one can effectively balance the spatial granularity of the depth representation against the overall processing time.
Uncompressed input views	It should be noted that the IVDE software was originally designed to perform depth estimation using uncompressed source content. Consequently, raw, high-fidelity data is treated as the default input. The software’s underlying algorithms were not initially intended to account for the non-linear distortions and coding artefacts introduced by video compression. This creates a significant disparity between the design assumptions of the reference software and the practical requirements of end-to-end immersive video systems, where bandwidth constraints necessitate the use of compressed bitstreams.
All views used for estimation	All available input data was utilised.

The following changes were introduced to the standard parameters:

1. Decreased number of depth levels:
 - 16 levels of depth,
 - 128 levels of depth.
2. Changed number of superpixels [73]:
 - 5000 superpixels per view,
 - 100 000 superpixels per view.
3. Compression of input views:
 - Low compression,
 - Medium compression,
 - High compression.

4. Decreased number of used cameras:

- 70% of cameras,
- 20% of cameras.

Each configuration was evaluated across three experimental runs: the reference point-to-point method and the Point-to-block Matching using 3x3 and 5x5 windows. Following the CTC, a "one-variable-at-a-time" approach was adopted in order to isolate the impact of specific parameters on estimation robustness. Since depth maps serve as intermediate representations, the quality of synthesised virtual views was analysed. To facilitate direct pixel-to-pixel comparison, virtual views were synthesised at original camera coordinates using uncompressed input views as high-fidelity references. Objective quality was evaluated via IV-PSNR across five different sequences (Table 3.1.1.2), with results averaged across all views to provide a comprehensive measure of reconstruction performance.

Table 3.1.1.2. *Test sequences used in experiments.*

Sequence name	Number of views	Resolution	Source
Carpark	9	1920×1088	[49]
Fencing	10	1920×1080	[50]
Frog	13	1920×1080	[25]
Painter	16	2048×1088	[74]
Street	9	1920×1088	[49]

Virtual view synthesis was performed using VVS 2.0 [26] in its default configuration, reconstructing each viewpoint from the four nearest reference views and their corresponding depth maps. This multi-view blending approach established a rigorous benchmark for evaluating the impact of Point-to-block Matching on image fidelity. Input views were compressed via HEVC (x265 [75]) at CRF levels 0 (near lossless), 26, and 51, facilitating a robustness assessment across a broad spectrum of bitrates and coding artefact intensities.

3.1.2. ASSESSMENT OF THE QUALITY OF VIRTUAL VIEWS

This section presents the results of the conducted experiments. Tables 3.1.2.1 through 3.1.2.3 present the aggregated results averaged across all experimental sequences. Although performance varied between individual sequences, as demonstrated in the preceding sections, these consolidated data indicate that for the utilised test set, the proposed Point-to-block Matching method with a 3x3 window configuration consistently outperformed the conventional point-to-point method.

Table 3.1.2.1. Average IV-PSNR [dB] of views synthesised utilising depth maps estimated from uncompressed input views presented across five distinct IVDE configurations, averaged for all sequences.

Depth estimation configuration	Point-to-point Matching	Point-to-block Matching (3x3)	Point-to-block Matching (5x5)
Standard parameters	38.83	39.35	39.04
16 levels of depth	35.86	36.57	36.61
128 levels of depth	38.90	39.41	39.12
5000 superpixels	35.78	35.80	36.26
100000 superpixels	39.44	39.81	39.36

Table 3.1.2.2. Average IV-PSNR [dB] of views synthesised utilising depth maps estimated from compressed input views presented across three different levels of compression, averaged for all sequences.

Depth estimation configuration	Point-to-point Matching	Point-to-block Matching (3x3)	Point-to-block Matching (5x5)
Low compression	38.94	39.37	39.10
Medium compression	38.93	38.99	38.84
High compression	34.18	35.01	34.87

Table 3.1.2.3. Average IV-PSNR [dB] of views synthesised utilising depth maps estimated from uncompressed input views presented across two scenarios of fewer cameras available, averaged for all sequences.

Depth estimation configuration	Point-to-point Matching	Point-to-block Matching (3x3)	Point-to-block Matching (5x5)
70% cameras	36.72	38.32	38.06
20% cameras	37.24	37.06	37.30

The superior performance was observed across nearly all parameter modifications and experimental scenarios, with the sole exception being the configuration that utilised only 20% of the cameras for depth estimation. The most significant quality gains were recorded in scenarios involving a reduced number of cameras, high compression, and a reduction in depth resolution from 256 to 16 levels. These results suggest that the proposed method is particularly robust in more challenging environments where input data is either limited or significantly degraded.

Figure 3.1.2.1 illustrates the differences between the virtual views synthesised using depth maps derived from point-to-point matching and those obtained using the Point-to-block Matching with both 3x3 and 5x5 windows used.

The experiments detailed above demonstrate that, on average, the proposed method enhanced the quality of the estimated depth maps across nearly all tested configurations. The results for individual sequences verify that the proposed approach yields the most significant improvements when applied to highly compressed input views or in instances where the multi-view sequence contains substantial levels of noise. Furthermore, quality gains were consistently observed across all sequences in scenarios involving a reduced depth resolution during the estimation process.

Such estimation is significantly less computationally intensive, suggesting that the proposed method could be highly effective for systems with limited computational resources, such as consumer-grade hardware equipped with dual webcams. Given that these systems typically operate using compressed input video streams, the demonstrated efficiency of the proposal under such conditions enhances its practical utility and scalability in real-world applications.

Furthermore, the performance gains observed in scenarios involving a reduced number of input views suggest that the proposed method could be effectively integrated with the approach detailed in Chapter 2, specifically in configurations where the number of views available at the decoder side is constrained.

The experimental findings presented in this section provide empirical confirmation of the second thesis statement proposed in Chapter 1, which stated that:

It is possible to modify the depth estimation method to improve the quality of depth maps estimated from encoded views.

Through the implementation and rigorous testing of the proposed Point-to-block Matching method, it has been demonstrated that even a fundamental modification to the inter-view matching process can yield significant quality enhancements. This is particularly evident in the case of encoded input data, where the proposed method consistently outperformed the conventional point-to-point approach by mitigating the negative impact of compression artefacts and noise on depth estimation. Consequently, the measurable increases in IV-PSNR across various test sequences and scenarios confirm that the research objective has been achieved and the thesis has been successfully validated.

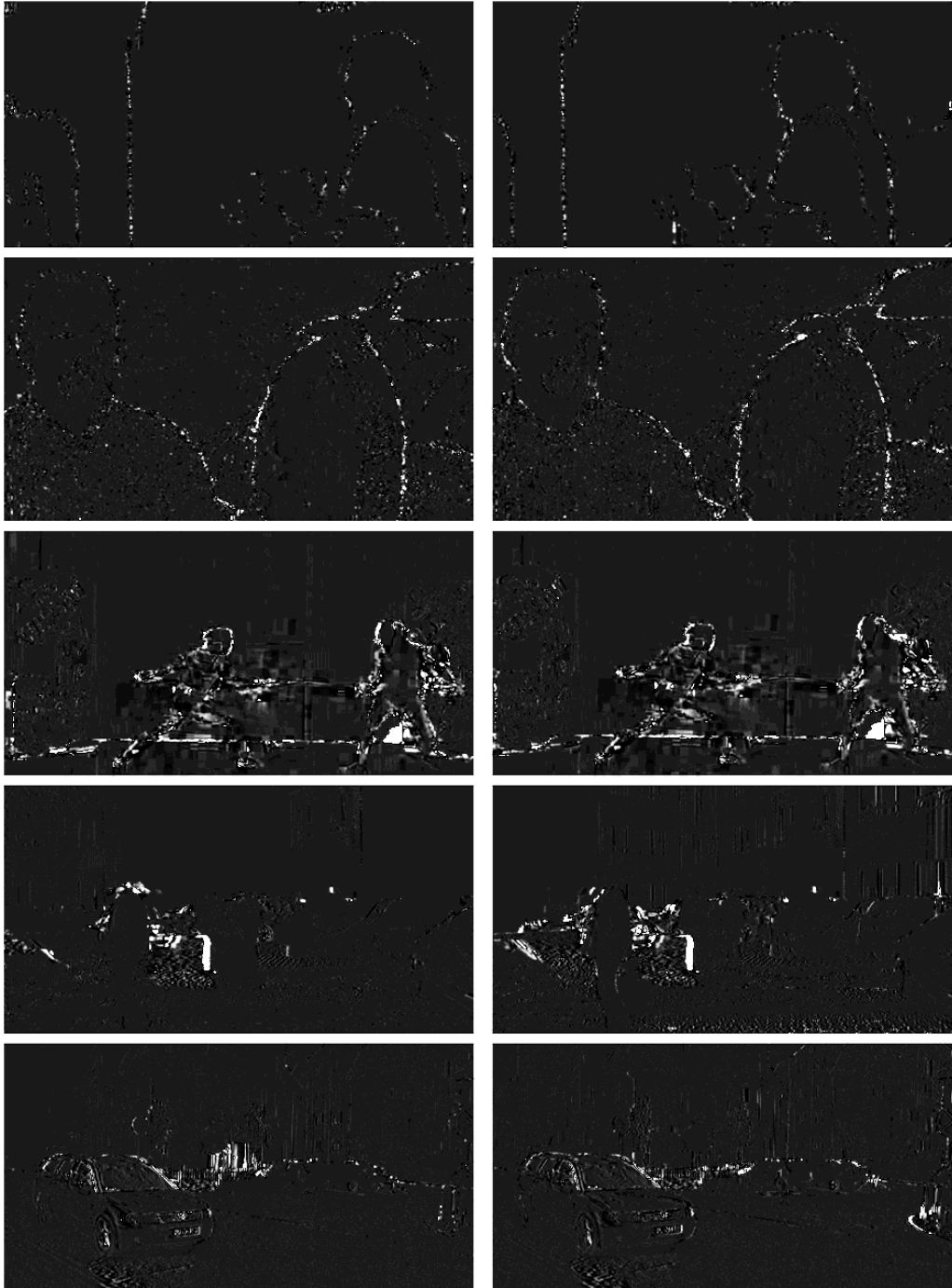


Figure 3.1.2.1. Differences between virtual view synthesised using depth maps estimated using point-to-point matching and Point-to-block Matching with 3x3 window (left)/5x5 window (right). Test sequences (top to bottom): Painter, Frog, Fencing, Carpark, Street.

3.2. ADAPTING GRAPH CUT OPTIMISATION FOR DECODER-SIDE DEPTH ESTIMATION

This section presents another modification to the depth estimation algorithm, specifically addressing challenges inherent in Decoder-Side Depth Estimation. Beyond mitigating the compression artefacts, an estimation method designed for DSDE should operate effectively within the limitations of minimising the complexity of the decoding process.

The second edition of MIV [25] assumes that the depth estimation step simultaneously enhances the quality of both transmitted and estimated depth maps [P1]. This increases the need for faster, more reliable estimation methods for immersive video. However, conventional methods often fail to deliver high-quality results in real-time. Meanwhile, learning-based approaches are frequently restricted to linear camera arrangements [15], lacking the versatility required for immersive systems, and often underperform when compared to conventional techniques [76].

Graph-cut optimisation techniques, specifically the alpha-expansion algorithm, are effective for solving pixel-labelling problems [72]. These techniques are the basis of MPEG reference depth estimation software such as DERS and IVDE. However, graph-cut techniques have not been fully optimised for Decoder-Side Depth Estimation. Adapting them to mitigate compression-induced artefacts is essential for improving accuracy in immersive video systems.

The formulation of depth estimation as a cost function minimisation was established in Chapter 3 in (3.1.1). The smoothness term, which addresses surfaces lacking texture, is defined as:

$$V_{p,q}(\bar{d}_p, \bar{d}_q) = \beta_0 \cdot |\bar{d}_p - \bar{d}_q|, \quad (3.2.1)$$

where:

- β_0 is the smoothing coefficient provided by the user or calculated from camera parameters [15],
- $\bar{d}_p$ is the currently considered depth of a pixel p ,
- $\bar{d}_q$ is the currently considered depth of a pixel q .

Equation (3.2.1) enforces depth smoothness across scene objects. However, the smoothness coefficient is susceptible to compression-induced artefacts, as blurred edges in the processed video can degrade the accuracy of the optimisation process.

To ensure inter-view [72, 77] and temporal [78] consistency, additional terms may be incorporated into (3.1.1). The final solution assigning discrete depth values to all pixels in the input views is determined through a suitable optimisation method. In this framework, pixels are represented as graph nodes, while edges encode the cost function components. The algorithm then identifies the optimal cut to minimise the total cost.

Multi-label segmentation problems are typically solved through successive optimisations, primarily using alpha-beta swap or alpha-expansion [79]. While an alpha-beta swap requires numerous iterations by cycling through all label pairs, an alpha-expansion either assigns a new label, or retains the existing one, reducing the complexity to $\text{size}(d)$ iterations.

Figure 3.2.1 illustrates alpha-expansion-based optimisation for eight depth levels. Upon completion, the process can be repeated in subsequent cycles [80], where the final depth map serves as the input for further refinement.

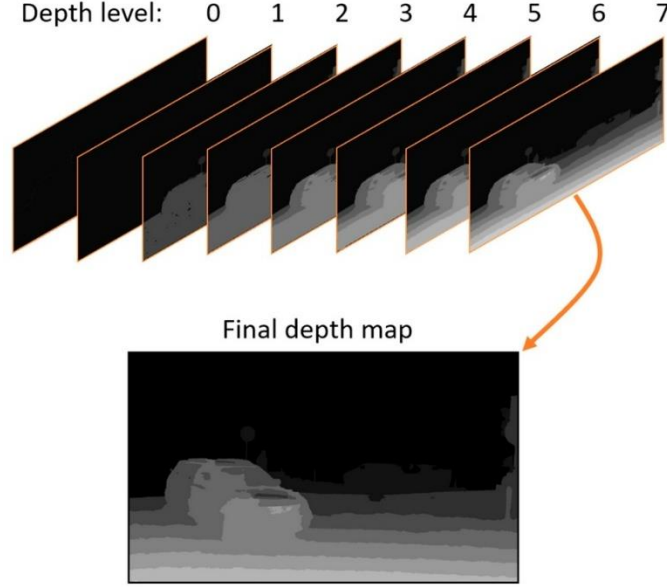


Figure 3.2.1. An example of alpha-expansion-based depth estimation.

Node participation in each iteration can be restricted, by deactivating some of them in chosen iterations, to ensure temporal consistency [80] or to preserve pre-existing depth values. Despite their deactivation, these nodes continue to influence the labels assigned to active neighbours. An initial coarse optimisation yields a low-detail depth map (Fig. 3.2.2A), which serves as the basis for the subsequent cycle. In this phase, the number of labels evaluated for each segment is reduced from $size(d)$ to $\pm \frac{n}{2}$ (Fig. 3.2.2B). Consequently, the number of active pixels is significantly decreased, and their set is defined as:

$$P_{on}(\bar{d}_p) = \{r : |\bar{d}_p - r| < n\}, \quad (3.2.2)$$

Leveraging coarse optimisation results enables the algorithm to focus on refining previous labelling, thereby eliminating redundant computations and potentially improving the final depth map. The proposed approach reduces optimisation complexity twofold: first, by restricting the label space in the initial phase, and second, by processing reduced graph sizes in the subsequent cycle.

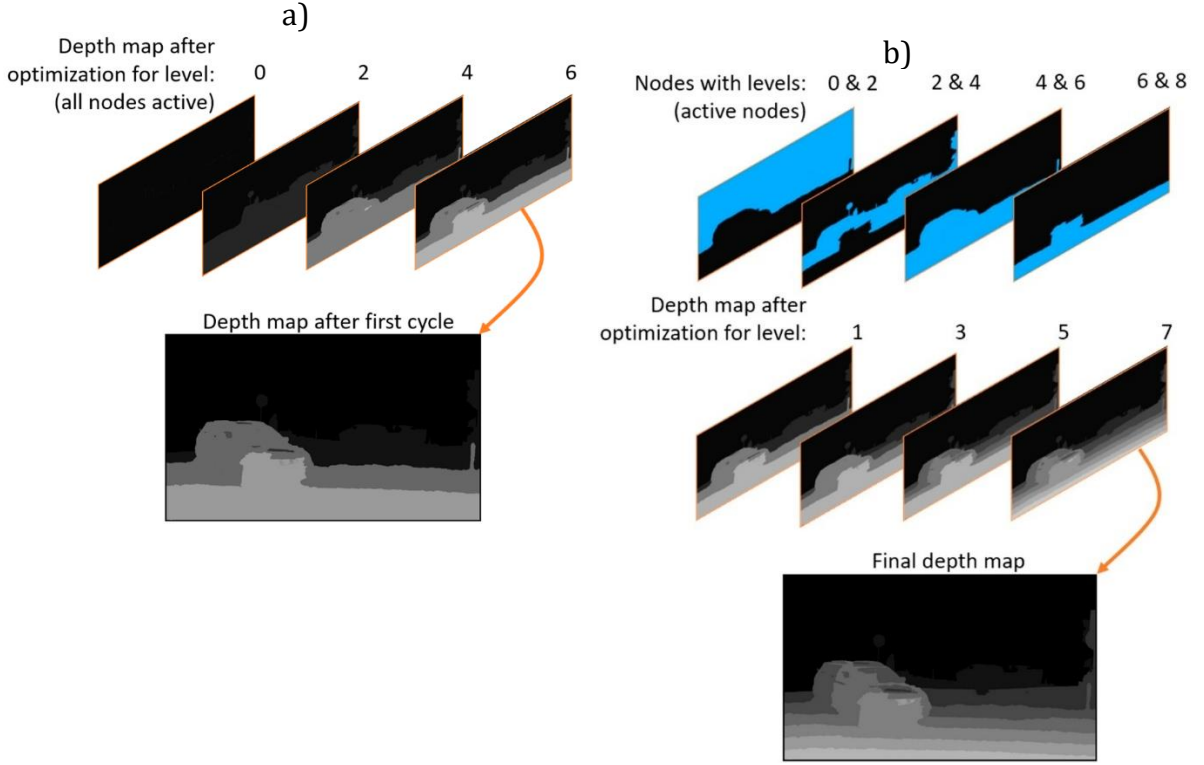


Figure 3.2.2. An example of hierarchical alpha-expansion-based depth estimation. The currently active nodes are shown as the blue area. a) First cycle of graph cut (coarse optimization). b) Second cycle of graph cut (refinement optimization).

3.2.1. PROPOSED ALPHA-EXPANSION BASED OPTIMISATION

SECOND-CYCLE LABEL OFFSET

Prior research [80] indicates that at least two global cycles of graph-cut optimisation are necessary to achieve sufficient depth map quality, as the second cycle effectively mitigates significant errors, particularly in the occluded regions. Given that the third cycle yields negligible improvements, the additional computational overhead is unjustified. This requirement for a dual-cycle approach was further validated through subjective quality assessments by MPEG Video Coding experts, leading to its adaptation in MIV-based evaluations of the DSDE compression scheme. As a result, in the hierarchical-alpha expansion approach utilised 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 Fig. 3.2.2A is performed twice). The additional cycle of refinement iterations requires a significant amount of additional time for the depth estimation, which is already the most time-consuming part of immersive video coding pipeline.

The proposed method assumes that coarse cycles are performed for every n -th available depth value. To improve the estimation, an offset of $n/2$ is applied to the initial depth level of the second coarse global cycle. For example, if $n = 4$, the second cycle

commences at depth level 2 instead of 0. Consequently, the label set $\bar{d}_p$ in a cycle c is defined as:

$$\bar{d}_p \in \begin{cases} \{nk : k \in \mathbb{N} \bigwedge nk < N\} & \text{if } c = 0 \\ \{nk + \frac{n}{2} : k \in \mathbb{N} \bigwedge (nk + \frac{n}{2}) < N\} & \text{if } c = 1 \end{cases} \quad (3.2.1.1)$$

The proposed second-cycle label offset yields a significant reduction in depth estimation time.

NEIGHBOURING SEGMENTS LABEL EXAMINATION

A significant challenge in depth estimation is the displacement of object boundaries [16], which severely degrades the quality of virtual view synthesis. While IVDE addresses this by employing superpixels instead of pixels in graph-cut optimisation [77, 81], the compression done on input views can reduce the sharpness of these segments, hindering precise shape estimation. Such errors may then propagate through the entire sequence, aggravated by temporal consistency mechanisms that rely on depth values from preceding frames [80].

If an excessively large value of n in the initial coarse cycles causes an edge segment to be incorrectly labelled as foreground, standard hierarchical alpha-expansion often fails to rectify the error. This is because the refinement cycles only examine the nearest depth values, effectively excluding background labels from the search space. Conversely, correct depth values are likely present within neighbouring segments. As a result, the proposed method incorporates depth labels from adjacent segments into the refinement cycles:

$$P_{on}(\bar{d}_p) = \{r : |\bar{d}_p - r| < n \bigwedge r = \bar{d}_q\}. \quad (3.2.1.2)$$

While slightly increasing the number of depth labels to be evaluated, this approach effectively rectifies estimation errors arising from object boundary displacements. Although the utility of neighbouring segment depth values for refinement is well-documented [82], the proposed method integrates this concept directly into the optimisation framework. This integration eliminates the computational overhead associated with independent post-processing stages, therefore aligning with the efficiency requirements of DSDE-based immersive video coding.

3.2.2. EXPERIMENTAL METHODOLOGY

To evaluate the performance of the proposed methods, they were compared against state-of-the-art Decoder-Side Depth Estimation.

Experiments were conducted in accordance with the CTC defined by ISO/IEC MPEG Video Coding [83]. The test sequences used are listed in Table 3.2.2.1. The depth estimation process was performed using Immersive Video Depth Estimation (IVDE [15]).

The computational complexity of the tested methods was also evaluated. Results are reported as a runtime ratio relative to the baseline DSDE, which was performed without hierarchical alpha-expansion.

Table 3.2.2.1. List of sequences used in the experiments.

Sequence	Source	Type		Resolution	Views
ClassroomVideo	[38]	Equirectangular projection	Computer generated	4096 × 2048	15
Museum	[39]	Equirectangular projection	Computer generated	2048 × 2048	24
Chess	[36]	Equirectangular projection	Computer generated	2048 × 2048	10
Guitarist	[84]	Equirectangular projection	Computer generated	2048 × 2048	23
Hijack	[39]	Equirectangular projection	Mixed	4096 × 2048	10
Cyberpunk	[85]	Equirectangular projection	Mixed	2048 × 2048	10
Kitchen	[86]	Perspective, planar	Computer generated	1920 × 1080	25
Cadillac	[87]	Perspective, planar	Computer generated	1920 × 1080	15
Mirror	[88]	Perspective, planar	Computer generated	1920 × 1080	15
Fan	[89]	Perspective, planar	Computer generated	1920 × 1080	15
Group	[90]	Perspective, convergent	Computer generated	1920 × 1080	21
Dancing	[91]	Perspective, convergent	Computer generated	1920 × 1080	24
Painter	[74]	Perspective, planar	Natural content	2048 × 1088	16
Breakfast	[92]	Perspective, planar	Natural content	1920 × 1080	15
Barn	[93]	Perspective, planar	Natural content	1920 × 1080	15
Frog	[25]	Perspective, planar	Natural content	1920 × 1080	13
Carpark	[49]	Perspective, planar	Natural content	1920 × 1088	9
Street	[49]	Perspective, planar	Natural content	1920 × 1088	9
Fencing	[50]	Perspective, convergent	Natural content	1920 × 1088	9
CBABasketball	[51]	Perspective, convergent	Natural content	1920 × 1080	34
MartialArts	[52]	Perspective, convergent	Natural content	1920 × 1080	15

3.2.3. RESULTS AND ANALYSIS

Initially, the performance of hierarchical alpha-expansion (the state-of-the-art reference) was established through a comparison with basic alpha-expansion. Results in Table 3.2.3.1 indicate that employing the hierarchical scheme with the optimisation step of $n = 2$ reduces total decoding time (encompassing bitstream parsing, depth estimation, and virtual view rendering) by approximately 20%. Contrarily, the reduced quality of the estimated depth maps results in a 5% BD-rate penalty across both reported metrics.

Table 3.2.3.1. State-of-the-art hierarchical alpha-expansion with $n = 2$ (anchor) and $n = 4$, compared to basic alpha-expansion. 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%.

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
CBABasketball	-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

As demonstrated in Table 3.2.3.2, the proposed second-cycle label offset addresses the constraints of the state-of-the-art hierarchical method. With an optimisation step of $n = 2$, the runtime is reduced by over 40% with only a negligible impact on quality. However, at $n = 4$, quality degradation becomes more pronounced, and the runtime exceeds that of the $n = 2$ configuration, primarily due to the increased overhead of the mandatory refinement iterations required by this scheme.

Furthermore, upon activating the examination of neighbouring segments, the results presented in Table 3.2.3.3 reveal a substantial improvement in quality. The performance

exceeds even that of the non-hierarchical baseline, yielding a 6.6% reduction in the IV-PSNR-based BD-rate.

Table 3.2.3.2. Hierarchical alpha-expansion with second-cycle label offset with $n = 2$ and $n = 4$, compared to basic alpha-expansion. 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%.

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
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
CBABasketball	-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

In conclusion, as performed in the hierarchical alpha-expansion with a second-cycle offset ($n=2$), evaluating all depth levels is suboptimal. Instead, generating an initial coarse depth map followed by refinement using only high-probability candidates, specifically neighbouring depth levels and labels from adjacent segments, yields superior results. This configuration is the only one to provide a quality gain while maintaining a highly competitive runtime, only 5% longer than the fastest tested scheme. Visual comparisons in Fig. 3.2.3.1 and 3.2.3.2 reveal that the most significant improvements occur at object boundaries.

Experimental results indicate that prioritising plausible depth levels over exhaustive label searches significantly enhances estimation accuracy. Consequently, future research should investigate adaptive hierarchical frameworks capable of dynamic scaling based on content complexity and hardware-specific constraints, such as CPU and memory availability. Ultimately, the hierarchical alpha-expansion graph-cut optimisation constitutes a major advancement in leveraging Decoder-Side Depth Estimation for immersive video compression.

Table 3.2.3.3. Hierarchical alpha-expansion with second-cycle label offset and neighbouring 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
Carpark	10.8	6.5	55.1
Street	3.4	0.5	51.4
Fencing	15.8	8.2	41.3
CBABasketball	-19.4	-21.9	61.3
MartialArts	-28.7	-25.0	65.9
Average	-3.0	-6.6	62.2



Figure 3.2.3.1. 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.

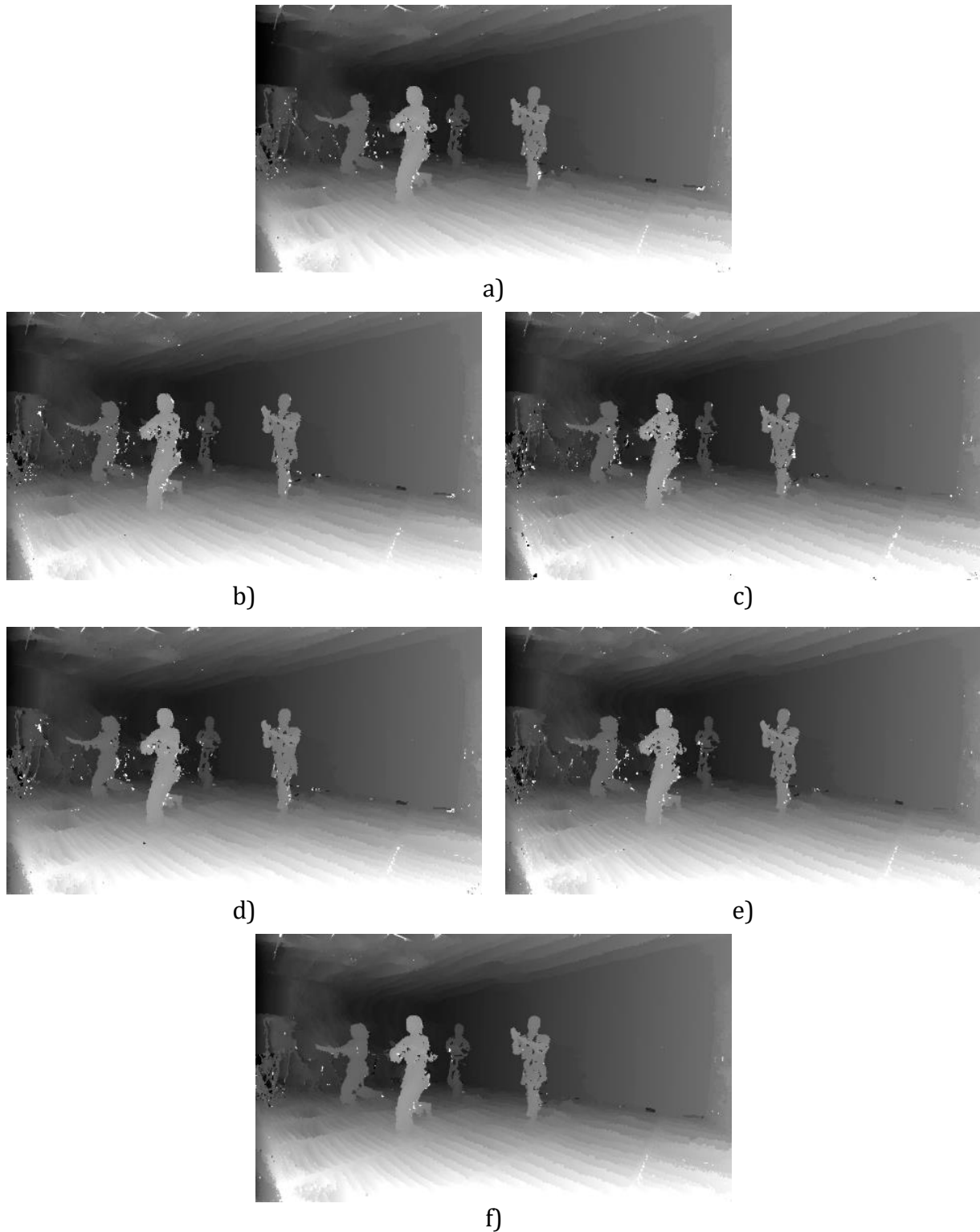


Figure 3.2.3.2. The comparison of depth maps estimated for the MartialArts sequence. Depth maps estimated using graph-cut optimization with: a) alpha-expansion, b) hierarchical alpha-expansion ($n = 2$), c) hierarchical alpha-expansion ($n = 4$), d) hierarchical alpha-expansion with second-cycle label shift ($n = 2$), e) hierarchical alpha-expansion with second-cycle label shift ($n = 4$), f) hierarchical alpha-expansion with second-cycle label shift and neighbouring segments label examination.

4. INPUT DEPTH MAP ASSISTANCE

Chapter 1 detailed the challenges posed by data redundancy in immersive video and its implications for compression. Building on that foundation, this chapter focuses on contemporary coding frameworks and introduces the proposed modifications to these approaches, with the main focus on the Decoder-Side Depth Estimation.

As described in Chapter 1, due to the absence of codecs dedicated to depth map compression, traditional or adapted video encoders are typically employed [5, 16]. The DSDE profile was developed specifically to address the inherent limitations of depth map coding. While this approach relies on high-quality depth estimation at the decoder, such processes are typically computationally intensive [15, 76]. Although depth refinement could be implemented at the decoder side to enhance fidelity, it would further increase computational complexity. Consequently, the quality of the depth maps estimated at the decoder is generally inferior to the quality of those available at the encoder. This disparity stems primarily from pixel-rate constraints which restrict the number of views available at the decoder [77], and compression-induced artefacts that further compromise estimation accuracy [94]. A fundamental limitation of the DSDE approach is its inability to leverage the superior depth information available at the encoder side.

To address this, previous solutions [15, 95, 96] have employed encoder-derived features to guide the Decoder-Side Depth Estimation, for instance, by narrowing the search range for depth levels or identifying blocks suitable for temporal copy-and-apply operations from previous frames. However, this approach necessitates specialised depth estimation algorithms, severely restricting the choice of suitable methods. Furthermore, these features can consume between 1 and 2 Mbit/s [96, 97]. At low bitrates (e.g., below 5 Mbit/s), it disproportionately reduces the bandwidth available for texture data, compromising the information necessary for accurate depth estimation and virtual view synthesis.

An alternative approach [98] distributes the estimation process between the encoder and decoder, using the motion vectors to decide whether a block should be estimated or reconstructed via motion compensation. While this shifts the computational complexity to the encoder, it relies on motion vector extraction from the AVC-encoded streams. Consequently, it violates the codec-agnostic principle of the MIV framework, limiting its general applicability.

Since each of the aforementioned DSDE approaches possesses inherent drawbacks, a novel set of solutions is proposed in Section 4.1.

4.1. PROPOSED DEPTH ASSISTANCE APPROACHES

This section details two proposed immersive video compression methods: Input Depth Map Assistance (IDMA) and Extended Input Depth Map Assistance (eIDMA).

PROPOSED MODIFICATIONS TO THE MIV ENCODER

The core concept of these proposals is illustrated in Fig. 4.1.1. Both methods rely on encoding a subset of the depth maps available at the encoder to enhance the Decoder-Side Depth Estimation process. While conventional DSDE typically performs depth estimation based solely on available texture views (Fig. 4.1.1a), the IDMA approach transmits a selection of original depth maps. This geometry data is simultaneously refined and utilised to improve estimation accuracy for the remaining views (Fig. 4.1.1b). It is achieved through a modified depth estimation based on multi-view optimisation.

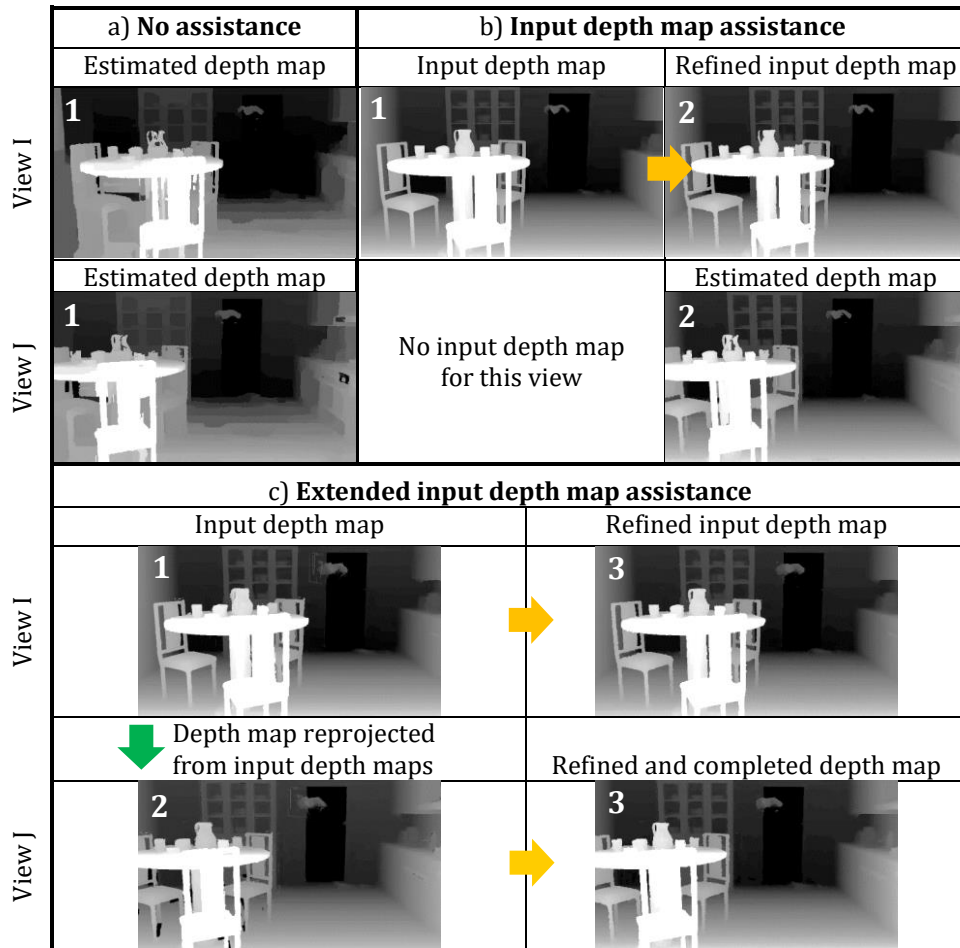


Figure 4.1.1. Simplified overview of DSDE for two views, performed without assistance and for two proposed methods: a) depth maps for views i and j are estimated in the decoder (1); b) input depth map of view i decoded from the bitstream (1) is used in the joint process of estimation and refinement of depth maps for views i and j (2); c) decoded input depth map of view i (1) is reprojected to view j and becomes input depth map for view j (2), both input depth maps are used in the joint estimation and refinement process, resulting in depth maps for views i and j (3).

To further enhance the depth map quality and reduce computational complexity, the eIDMA approach is introduced (Fig. 4.1.1c). In this method, available input depth maps are reprojected onto views with incomplete depth data. These reprojected maps are then processed by the modified depth estimator which refines them (similarly to the IDMA approach) while simultaneously estimating depth for the empty regions. This framework is built upon the MIV compression standard [13].

Implementing these proposals required several modifications to the reference encoder, beginning with the view labelling stage. This process categorises input views as either basic or additional. While basic views are packed into atlases in their entirety, additional views are either pruned into patches (MIV Main) or discarded (MIV GA). Table 4.1.1 compares the view labelling strategies across the MIV Main and MIV GA profiles.

Table 4.1.1. Types of transmitted data for MIV Main and MIV GA profiles.

View type	Input views	
	Basic	Additional
Transmitted data	MIV Main: texture, depth MIV GA: texture	MIV Main: texture, depth MIV GA: none

Both IDMA and eIDMA transmit only a subset of the depth maps, which results in a configuration where texture atlases outnumber depth atlases. Figure 4.1.2 illustrates an example configuration of four atlases in total: three containing texture information and one dedicated to depth.



Figure 4.1.2. Example of **four** atlases produced by MIV encoder in the proposed methods; IDMA – three texture atlases and only the yellow framed part of the depth atlas, eIDMA – three texture atlases and both yellow and green bordered parts of the depth atlas.

In the proposed solutions, depth map selection aims to maximise the scene coverage while minimising overlap between selected views. The IDMA architecture which

incorporates this depth selection module is illustrated in Fig. 4.1.3. Since the standard MIV view labelling algorithm already performs this selection, the IDMA approach executes it in two passes: the first pass identifies the basic views, while the second pass selects 'essential views' (those whose depth maps are transmitted in their entirety) (Table 4.1.2).

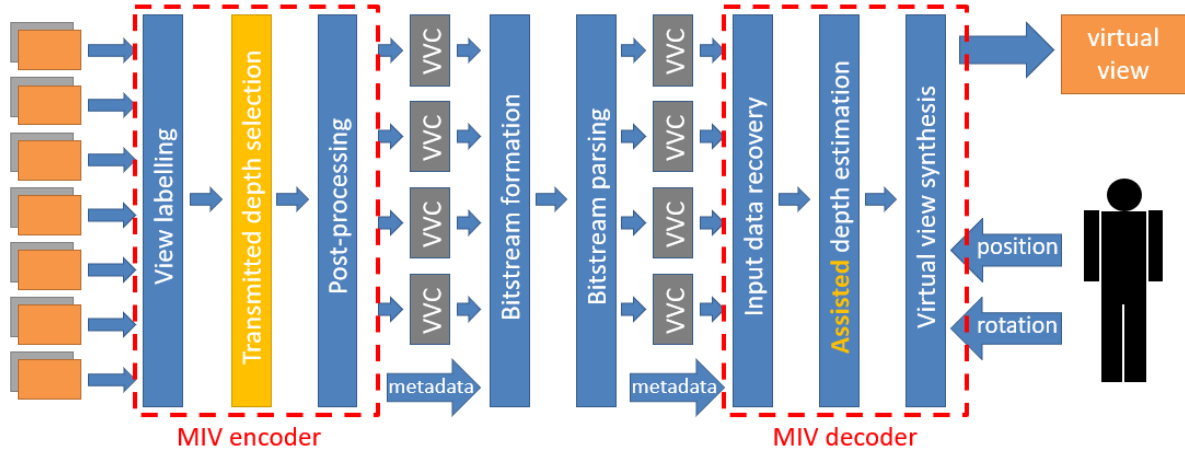


Figure 4.1.3. The scheme of MIV encoder and decoder modified to handle the IDMA proposed method. Changes made to the MIV GA are coloured in yellow.

Table 4.1.2. Types of transmitted data for proposed methods.

View type	Input views		
	Basic		Additional
	Essential	Non-essential	
Transmitted data	IDMA: texture, depth eIDMA: texture, depth	IDMA: texture eIDMA: texture, depth	None

Performance analysis revealed that IDMA is more effective for natural sequences with imperfect depth maps, whereas eIDMA yields superior results for computer-generated content with high-quality depth maps. Consequently, an automatic selection mechanism was implemented to dynamically determine the most suitable approach based on content characteristics. This functionality leverages the existing MIV bitstream "depth quality flag" [13]. The selection is based on a geometry assessment performed on the initial frame, during which the input views are reprojected to the positions of the other views. A pixel is classified as inconsistent if its reprojected depth value exceeds that of the co-located pixel or its neighbours. If the proportion of inconsistent pixels exceeds a threshold of 0.1%, the geometry quality is classified as low.

In eIDMA, a significant portion of the depth map is generated through the reprojection of essential views, reducing the need for full estimation (Fig. 4.1.1c). However, relying exclusively on the depth maps of essential views often proves insufficient for complete

scene reconstruction. To address this, the eIDMA framework incorporates additional stages, such as depth pixel pruning (Fig. 4.1.4).

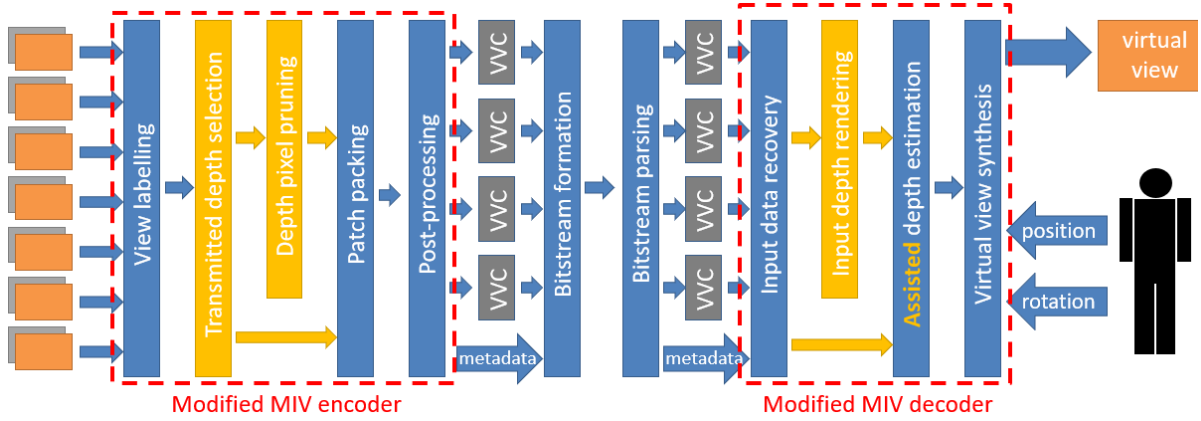


Figure 4.1.4. The scheme of MIV encoder and decoder modified to handle the eIDMA proposed method. Changes made to the MIV GA are coloured in yellow.

An illustrative example of this issue is provided in Fig. 4.1.5, where depth map (b) was rendered exclusively using the information from the essential view atlases (a). The pixels in this depth map can be categorised into three distinct sets:

Set 1: Accurately restored regions which constitute most of of the data derived directly from the essential view atlases.

Set 2: Regions devoid of information, appearing as white areas. These include a substantial portion of the floor and disocclusions behind the chess pieces in the background.

Set 3: Regions containing erroneous data (indicated by the red frame). For example, the head of the knight is incorrectly represented by background information.

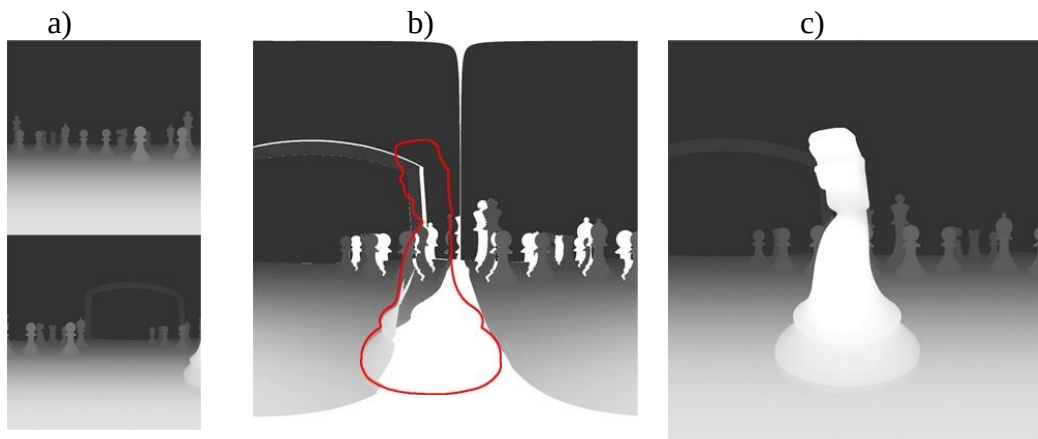


Figure 4.1.5. Motivation for sending additional depth information. a) first depth atlas, b) depth for view rendered using information from the first atlas (silhouette of the knight is highlighted), c) reference depth for the view from the second atlas (not transmitted).

Depth values in Set 1 are correctly reprojected and align with the reference depth map (Fig. 4.1.6c). For pixels in Set 2, depth information is absent and must be estimated at the

decoder side. However, Set 3 poses a significant challenge: while these pixels contain reprojected data, the information is erroneous. Crucially, the decoder cannot autonomously distinguish these incorrect fragments from the valid data in Set 1.

The classification of basic (non-essential) view pixels into these three sets is performed at the encoder side via reprojection from essential views. A pixel is assigned to Set 1 if its depth value is less than or equal to (i.e., closer to the camera than) the value of the co-located reprojected depth. If no information is reprojected to the pixel's coordinates from any essential view, that pixel is assigned to Set 2. Otherwise, the pixel is categorised as Set 3.

As illustrated in Fig. 4.1.4, the eIDMA approach introduces a dedicated depth pixel pruning stage for non-essential views. This stage employs a modified MIV pruning algorithm - a pixel is removed if it is inter-view redundant, or if its depth can be reliably estimated at the decoder using decoded textures. Therefore, only pixels representing inestimable foreground objects are preserved, specifically those that would otherwise be erroneously replaced by reprojected background from other views. The resulting depth map rendered using these additional depth patches is shown in Fig. 4.1.6.b.

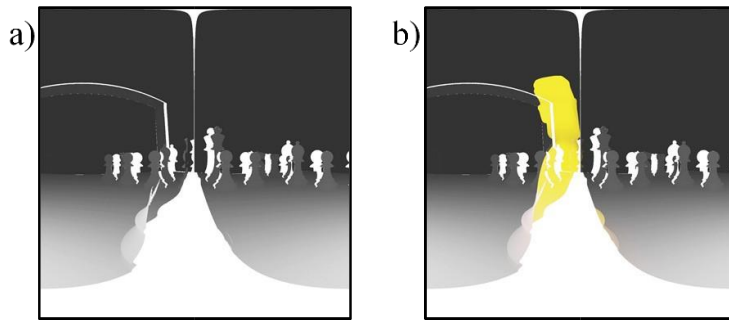


Figure 4.1.6. a) A depth map rendered using depth maps from the first atlas and b) using depth maps from the first atlas together with additional depth patches; depth transmitted in additional patches was highlighted in yellow.

Notably, the height of the depth atlas in eIDMA is twice that of the IDMA approach. However, as MIV by default downsamples depth resolution by a factor of two in both dimensions the total footprint of the depth atlas remains significantly smaller than that of the texture atlases [13, 99].

PROPOSED MODIFICATIONS TO THE MIV DECODER

Initially, the modified MIV decoder extracts all views and depth maps from the received bitstream (Fig. 4.1.3, 4.1.5). This process, based on the standard MIV decoding algorithm [100], recovers all basic views and transmitted depth maps. In the IDMA approach, this includes the depth for essential views, while in eIDMA, the depth for both basic and essential views is recovered.

All the recovered basic views are subsequently used as an input to the modified depth estimation algorithm. The handling of the recovered depth maps depends on the chosen

configuration. In IDMA, the depth estimator receives B recovered input views and E recovered input depth maps, where B and E represent the number of basic and essential views, respectively ($B > E$).

In the eIDMA approach, the number of input depth maps is extended from E to B . While depth maps for essential views are used in their original form, those for non-essential views primarily consist of empty regions (Fig. 4.1.7a) due to the encoder-side pruning of redundant information. These gaps are filled by reprojecting depth from other available input depth maps, resulting in a more complete depth maps, such as the ones shown in Fig. 4.1.7b, which are then used as the input for the depth estimation process.

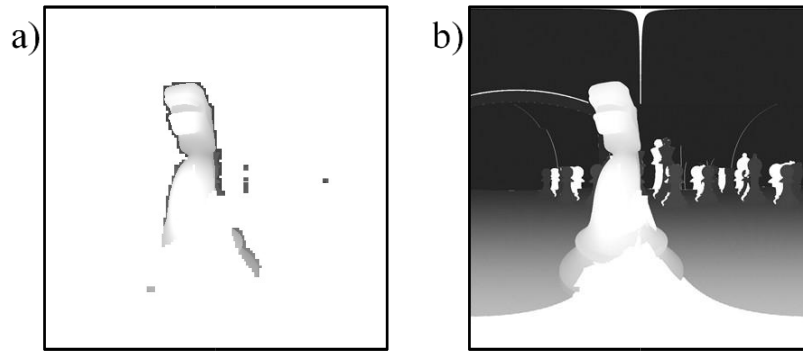


Figure 4.1.7. A depth map for non-essential view; a) recovered, b) rendered by reprojection from other views and extended using the recovered depth map.

In addition to the MIV decoder modifications, it was necessary to adapt a depth estimation method capable of integrating input depth maps. For this purpose, the Immersive Video Depth Estimation (IVDE) framework [77] was selected and modified.

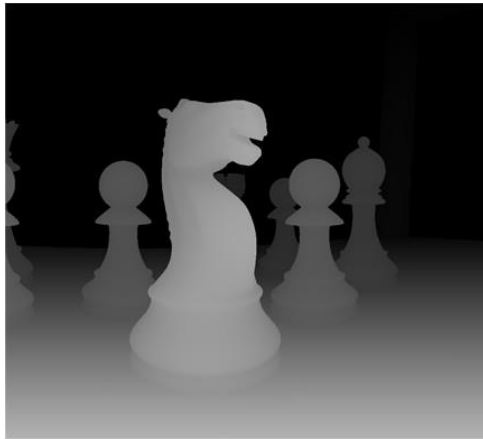
A fundamental feature of IVDE is its global depth optimisation in which depth maps for all views are estimated simultaneously. Consequently, incorporating high-quality input depth maps inherently enhances the estimation accuracy for the remaining views (Fig. 4.1.1). Furthermore, this approach reduces computational complexity, as the spatial regions that require estimation from scratch are significantly reduced.

While this framework facilitates the use of input depth maps, it assumes that their quality is sufficiently high. However, standard video codecs are not inherently designed for depth map compression and often introduce significant artefacts. To mitigate these compression-induced errors, IVDE was modified to refine the input depth maps during the optimisation process. This refinement allows for higher quantisation during depth compression. By allocating fewer bits to depth, the texture bitrate can be increased, which in turn improves the reliability of the texture-based consistency checks used during depth refinement.

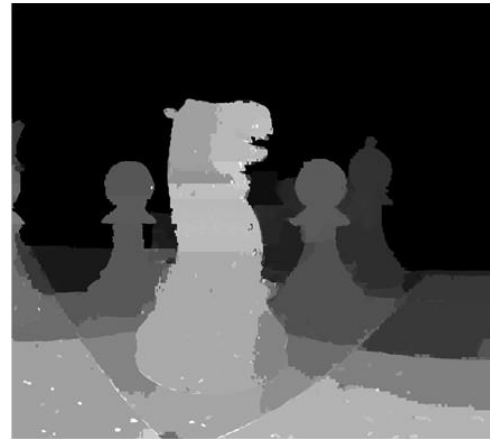
In IVDE, depth is estimated at the superpixel (segment) level, with segments defined using texture information. Initially, the minimum and maximum depth values within the input depth map of each superpixel are identified. These extrema define a constrained search range for the global graph-cut optimisation. This mechanism ensures that even if

individual pixels are degraded by compression, the search space remains anchored by more reliable adjacent values, which facilitates effective error correction.

Figure 4.1.8 illustrates a segment of the final estimated depth map that has been subjected to a high compression ratio. Although the depth map in Fig. 4.1.8(d) might appear to be of lower quality than in Fig. 4.1.8(c), it yields superior results from the synthesis perspective. This is due to its reduced blurriness, which ensures a more accurate virtual view synthesis.



a) Reference ground-truth depth map
(available in the encoder)



b) Depth map estimated in Decoder-Side Depth Estimation



c) Highly compressed input depth map in eIDMA (before IVDE)



d) Final depth map in eIDMA (after IVDE)

Figure 4.1.8. Fragment of depth map: a) reference ground-truth; b) acquired in Decoder-Side Depth Estimation; c) compressed input depth map in eIDMA; d) depth map acquired by refinement of input depth map performed in Decoder-Side Depth Estimation process.

4.2. EXPERIMENTAL METHODOLOGY

To evaluate the performance of the proposed compression methods, IDMA and eIDMA are compared against two state-of-the-art immersive video coding schemes: the standard MIV Decoder-Side Depth Estimation (DSDE) anchor [101] and MIV DSDE with Geometry Assistance (GA) SEI messages (encoder-derived features) [97].

The conditions of experiments are based on the common test conditions defined by ISO/IEC MPEG Video Coding to provide a fair comparison between methods for immersive video coding [102]. The sequences used for experiments are listed in the Table 4.2.1.

Table 4.2.1. List of test sequences. ERP – Equirectangular Projection, CG – Computer-Generated, NC – Natural Content.

Sequence	Source	Type		Resolution	Views
ClassroomVideo	[38]	Equirectangular projection	CG	4096×2048	15
Chess	[36]	Equirectangular projection	CG	2048×2048	10
Hijack	[39]	Equirectangular projection	CG	4096×2048	10
Museum	[39]	Equirectangular projection	CG	2048×2048	24
Group	[90]	Perspective, convergent	CG	1920×1080	21
Fencing	[50]	Perspective, convergent	NC	1920×1080	10
Fan	[89]	Perspective, planar	CG	1920×1080	15
Kitchen	[86]	Perspective, planar	CG	1920×1080	25
Mirror	[88]	Perspective, planar	NC	1920×1080	15
Carpark	[49]	Perspective, planar	NC	1920×1088	9
Frog	[25]	Perspective, planar	NC	1920×1080	13
Hall	[49]	Perspective, planar	NC	1920×1088	9
Street	[49]	Perspective, planar	NC	1920×1088	9
Painter	[74]	Perspective, planar	NC	2048×1088	16

The experiments were conducted using TMIV 11.0 [99], with video compression performed via the VVenC codec. Four rate points (RP) were evaluated: 3, 12, 22, and 38 Mbit/s, denoted as RP4, RP3, RP2, and RP1, respectively. Quantisation parameter (QP) values for both texture and depth atlases were independently tuned for each tested method to align with the target bitrates as closely as possible. Decoder-Side Depth Estimation was performed using the Immersive Video Depth Estimation (IVDE) framework [14, 77].

Following depth estimation, the TMIV renderer was employed to synthesise virtual views at the original camera positions of the test sequences. To evaluate the quality of these synthesised views, the IV-PSNR metric [30] was utilised.

4.3. RESULTS AND COMPARATIVE ANALYSIS

As illustrated in Fig. 4.3.1, the results for basic DSDE (without assistance) at low bitrates (RP4) align with the high performance documented in prior research [15, 25]. However, at higher bitrates (e.g., RP1 and RP2), the final quality of basic DSDE is among the lowest of all evaluated methods. This suggests that without supplementary data, the standard DSDE approach reaches a performance ceiling where additional bandwidth cannot compensate for the inherent limitations of the estimation process.

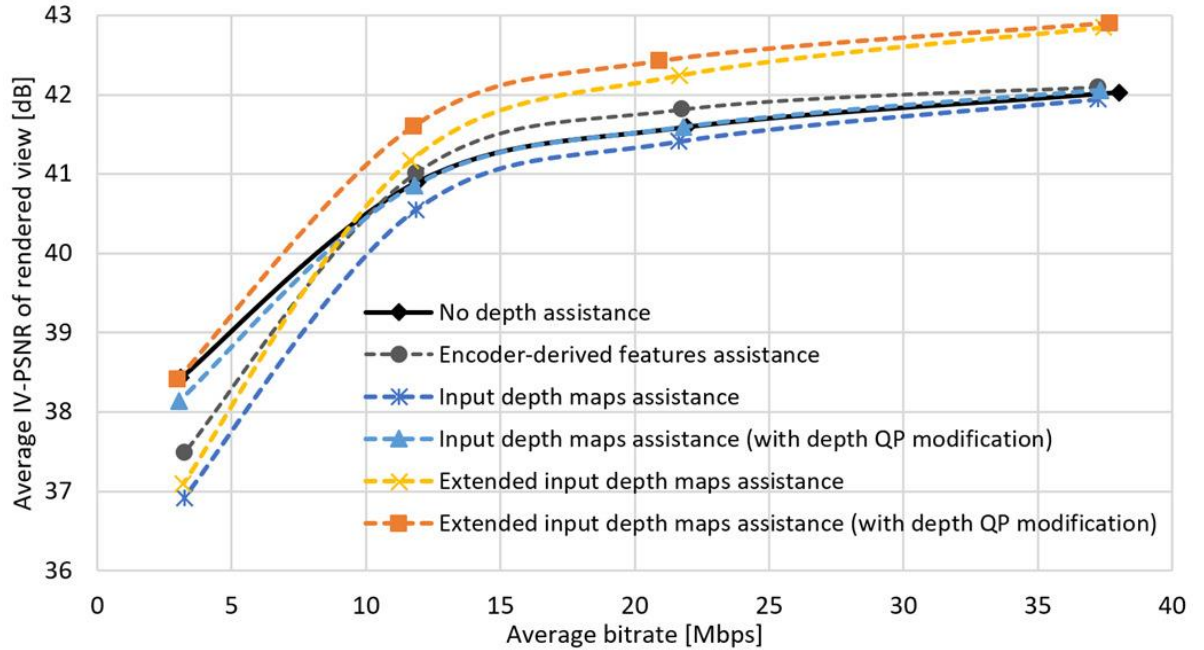


Figure 4.3.1. Average IV-PSNR of rendered views achieved by tested methods; shown bitrate include all texture, depth and metadata.

The encoder-derived features approach [97] which pioneered the use of encoder-side depth data to enhance DSDE demonstrated improved quality at medium and high bitrates compared to the basic DSDE anchor. On the other hand, at low bitrates, the metadata overhead required for these features, exceeding 20% of the total bitstream (Table 4.3.1), detrimentally affects texture encoding, thereby reducing the final quality of the rendered virtual views.

While the baseline IDMA approach initially yielded the lowest performance, the application of modified depth QP values significantly enhances quality across all bitrates. For medium and high bitrates, the average quality is comparable to that of the basic DSDE. However, as detailed in Table 4.3.2, this is accompanied by a substantial reduction in decoder-side runtime. Although a minor increase in encoder-side complexity is observed,

the overhead associated with atlas encoding remains negligible in reference to the overall video encoding pipeline.

Table 4.3.1. Distribution of bitrates used for the texture, depth and metadata.

No assistance [14]							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	38.008	0.000	0.009	38.017	99.98	0.00	0.02
RP2	21.915	0.000	0.009	21.924	99.96	0.00	0.04
RP3	11.923	0.000	0.009	11.931	99.93	0.00	0.07
RP4	3.105	0.000	0.009	3.114	99.72	0.00	0.28
Encoder-derived features assistance [97]							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	36.653	0.000	0.689	37.342	98.15	0.00	1.85
RP2	21.075	0.000	0.689	21.764	96.83	0.00	3.17
RP3	11.112	0.000	0.689	11.801	94.16	0.00	5.84
RP4	2.521	0.000	0.689	3.210	78.54	0.00	21.46
Input depth maps assistance (IDMA)							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	32.500	4.735	0.008	37.243	87.26	12.71	0.02
RP2	17.667	3.977	0.008	21.653	81.59	18.37	0.04
RP3	9.095	2.749	0.008	11.853	76.74	23.19	0.07
RP4	1.993	1.236	0.008	3.237	61.56	38.18	0.26
IDMA with depth QP modification							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	35.686	1.635	0.008	37.329	95.60	4.38	0.02
RP2	20.487	1.326	0.008	21.821	93.89	6.08	0.04
RP3	10.777	1.017	0.008	11.803	91.31	8.62	0.07
RP4	2.655	0.369	0.008	3.032	87.56	12.16	0.28
Extended input depth maps assistance (eIDMA)							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	30.543	6.877	0.018	37.437	81.58	18.37	0.05
RP2	16.370	5.277	0.018	21.665	75.56	24.36	0.08
RP3	8.136	3.494	0.018	11.648	69.85	30.00	0.15
RP4	1.668	1.514	0.018	3.199	52.13	47.31	0.55
eIDMA with depth QP modification							
Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	35.399	2.258	0.018	37.675	93.96	5.99	0.05
RP2	19.160	1.731	0.018	20.909	91.64	8.28	0.08
RP3	10.494	1.263	0.018	11.775	89.12	10.73	0.15
RP4	2.496	0.475	0.018	2.988	83.52	15.89	0.59

It should be noted that the software frameworks employed in this study (IVDE depth estimator, TMIV) are reference implementations designed for academic research and standardisation, rather than computational efficiency. Nevertheless, real-time solutions already exist for MIV bitstream decoding and computationally intensive virtual view rendering [103]. For example, the latter required approximately 20 seconds per view in the presented experiments.

Table 4.3.2. The runtime of tested methods; the percentages provide a runtime ratio compared to DSDE without assistance.

Type of assistance	Average with standard deviation of runtime per one frame [s]			
	Encoding			Decoding and rendering of a view
	Atlas encoding	Video encoding	Overall	
No assistance [14]	1.5	228.1	229.6	152.3
Features [97]	1.9 (126%)	228.1 (100%)	231.9 (101%)	83.8 (55%)
IDMA	7.7 (518%)	216.7 (95%)	227.3 (99%)	73.1 (48%)
IDMA + QP mod.	7.7 (518%)	235.0 (103%)	245.7 (107%)	74.6 (49%)
eIDMA	66.0 (4481%)	191.6 (84%)	261.7 (114%)	48.7 (32%)
eIDMA + QP mod.	66.0 (4481%)	212.1 (93%)	277.8 (121%)	48.7 (32%)

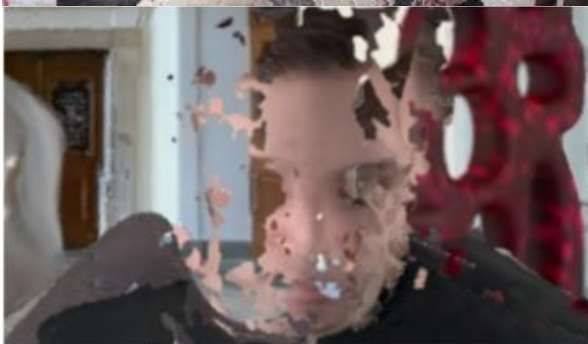
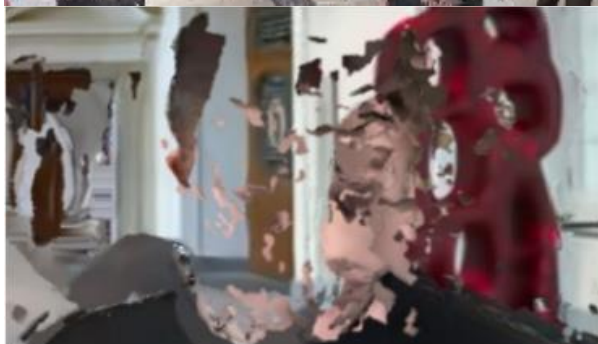
Extended Input Depth Map Assistance (eIDMA) outperforms the previously discussed methods at medium and high bitrates, while simultaneously offering the shortest decoding and rendering times. When the modified depth QP is applied, this approach yields superior results across all rate points (RP1–RP4), demonstrating efficacy of the depth refinement process integrated into the modified estimator.

A joint analysis of the virtual view quality (Fig. 4.3.2) and bitstream composition (Table 4.3.1) reveals a critical trend: for the top-performing configurations at each rate point, the bitrate fraction allocated to depth data - whether as encoded maps or metadata features - remains below 20%. This bitrate allocation strategy preserves sufficient texture quality to facilitate accurate depth estimation, resulting in higher-quality synthesised views. On the other hand, while omitting depth data entirely (as in basic DSDE) maximises texture fidelity, the resulting depth maps are prone to significant errors caused by occlusions and non-Lambertian surfaces. These inaccuracies ultimately bottleneck the final view quality, particularly at higher bitrates.

Museum

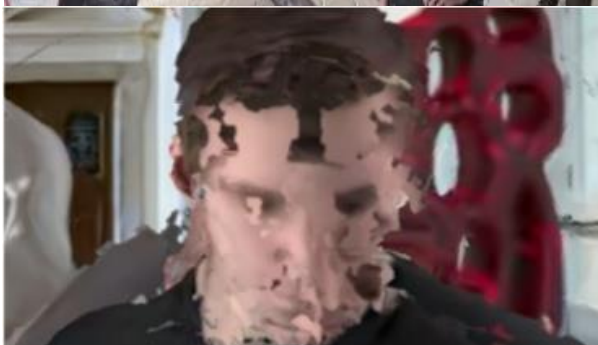
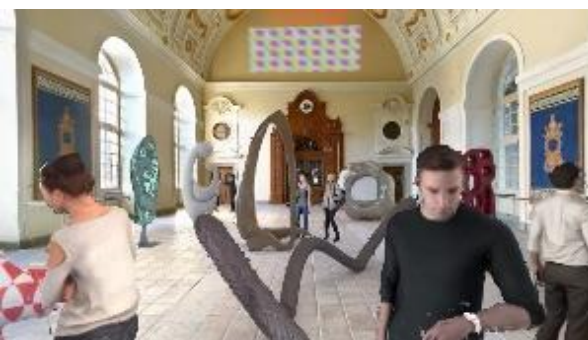
No assistance [14]

Encoder-derived features [97]



IDMA

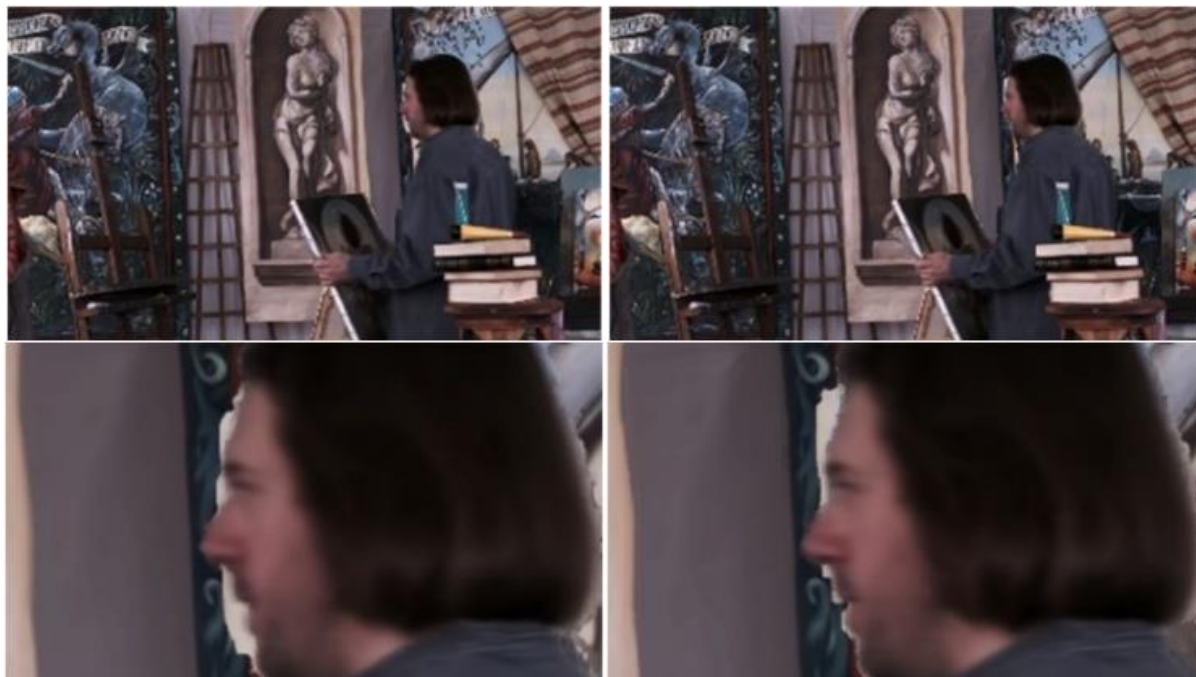
eIDMA



Painter

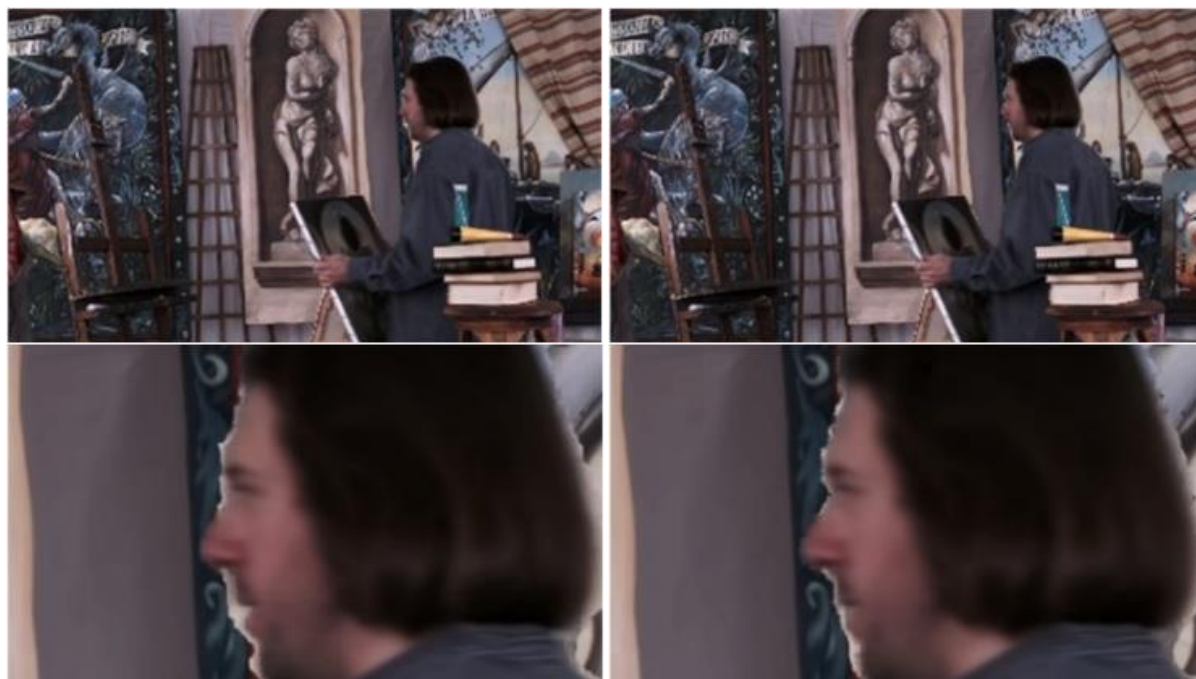
No assistance [14]

Encoder-derived features [97]



IDMA

eIDMA



Frog

No assistance [14]

Encoder-derived features [97]



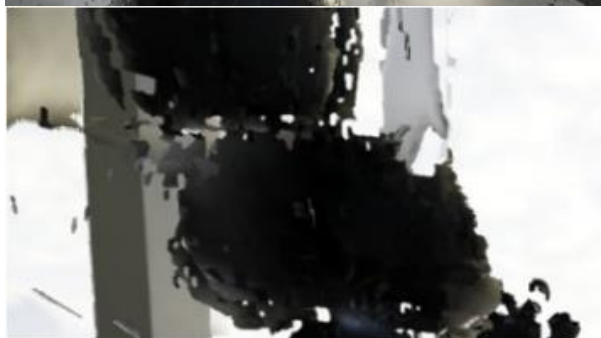
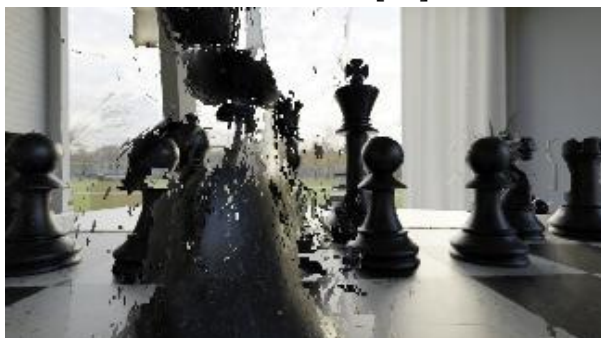
IDMA

eIDMA



Chess

No assistance [14]



Encoder-derived features [97]



IDMA



eIDMA



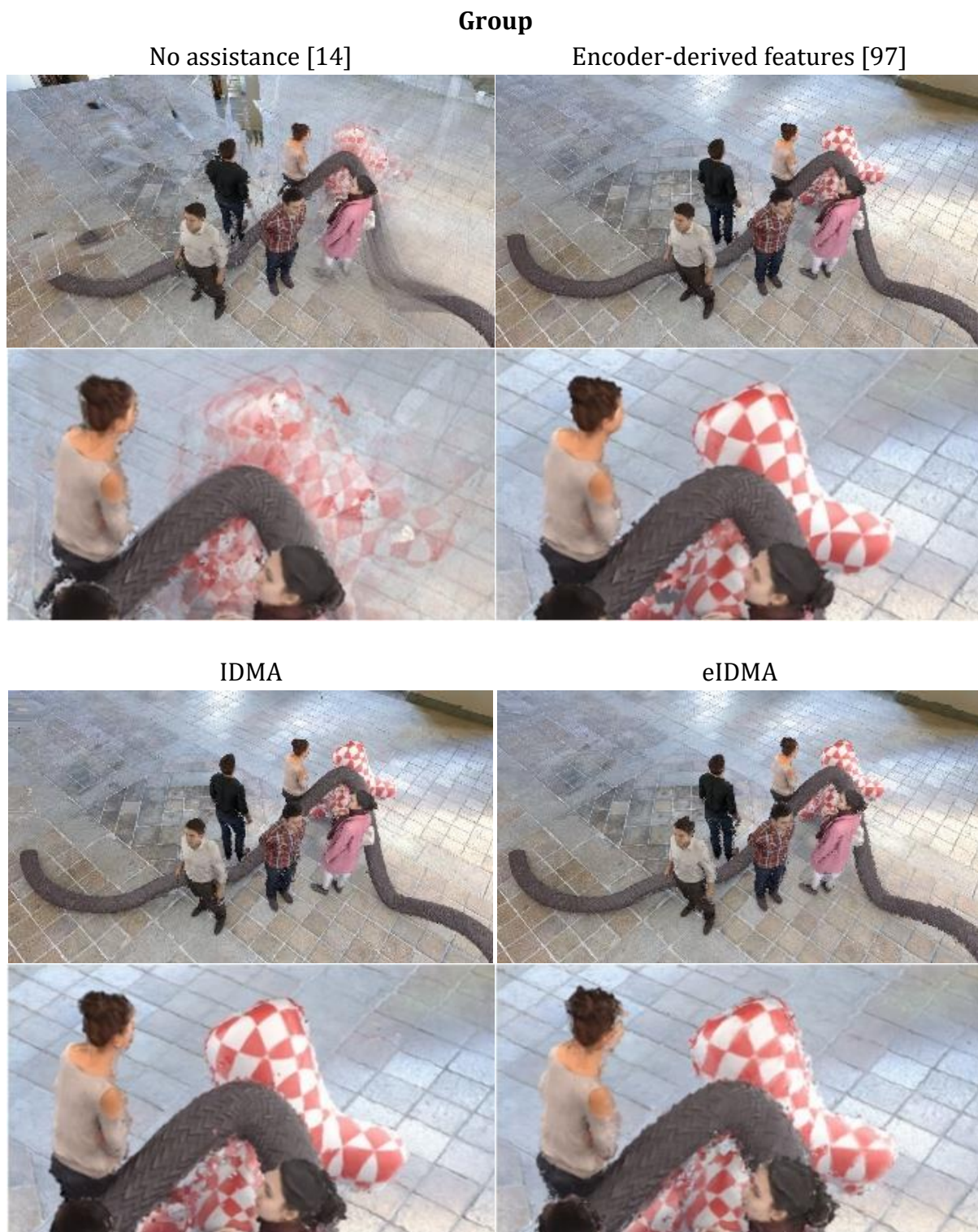


Figure 4.3.2. The subjective comparison of tested methods for fragments of the selected viewport synthesised between the positions of the input views.

As illustrated in Fig. 4.3.2, the proposed IDMA and eIDMA approaches outperform the encoder-derived features method [97] in most of the test cases.

Furthermore, both IDMA and eIDMA offer a substantial advantage in terms of computational efficiency, as their decoding processes are significantly faster than those required for the encoder-derived features approach.

When analysed separately for CG and natural content (Fig. 4.3.3 and 4.3.4, respectively), the results reveal that the fidelity of the encoder-side depth maps significantly dictates the relative performance of the evaluated methods. For CG sequences, where perfect quality ground-truth depth maps are available, the eIDMA approach consistently delivers superior quality, regardless of the depth QP configuration. On the other hand, for natural content, where encoder-side depth maps are estimated based on the captured views, the IDMA approach yields the highest performance.

Figure 4.3.5 illustrates the performance of a hybrid approach in which the proposal is chosen according to the type of handled content. The results indicate that the proposed adaptive IDMA framework consistently outperforms the alternative configurations, effectively integrating individual strengths of both methods into a single, robust solution for DSDE-based immersive video coding.

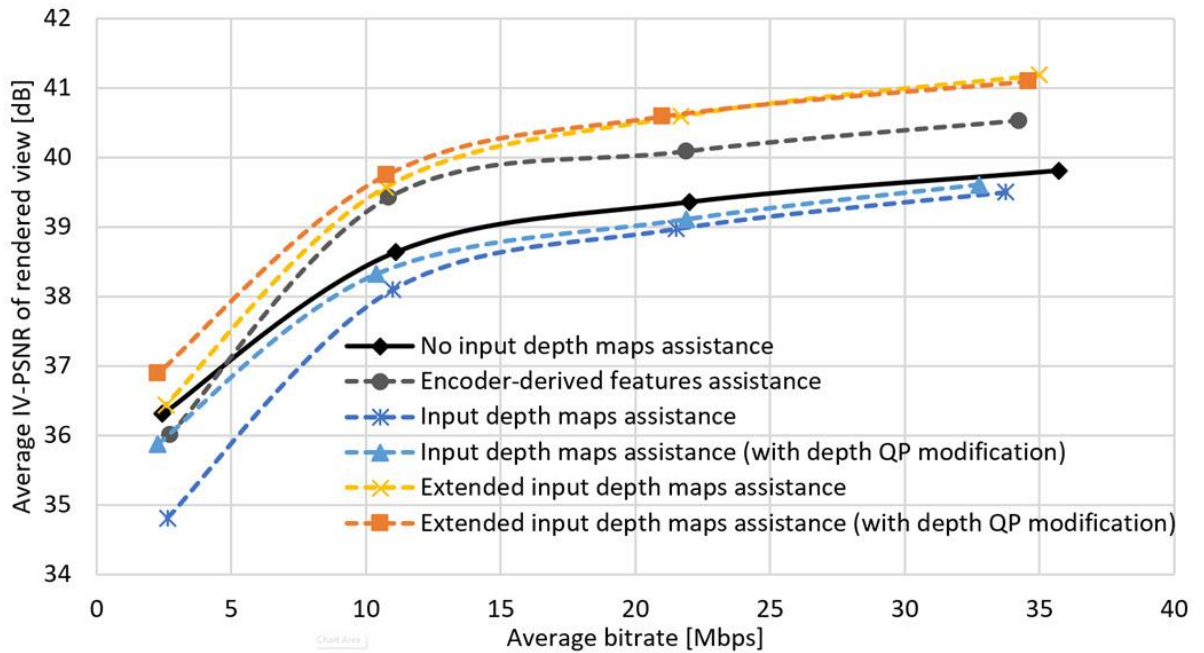


Figure 4.3.3. Results for CG sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

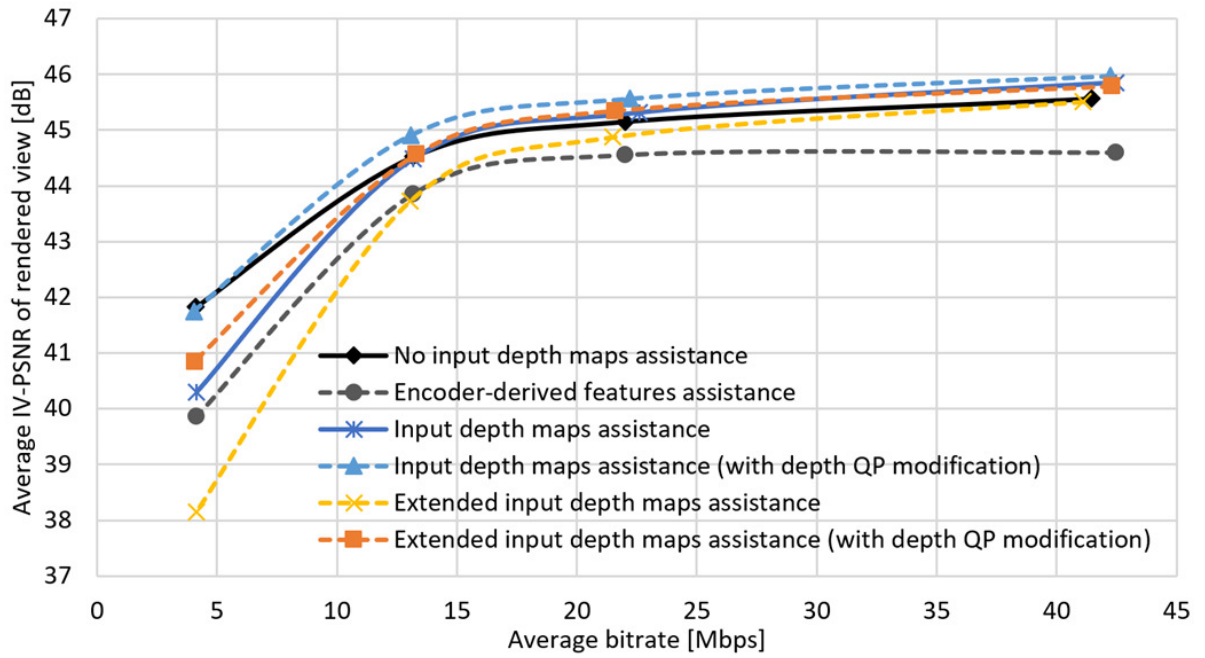


Figure 4.3.4. Results for natural sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

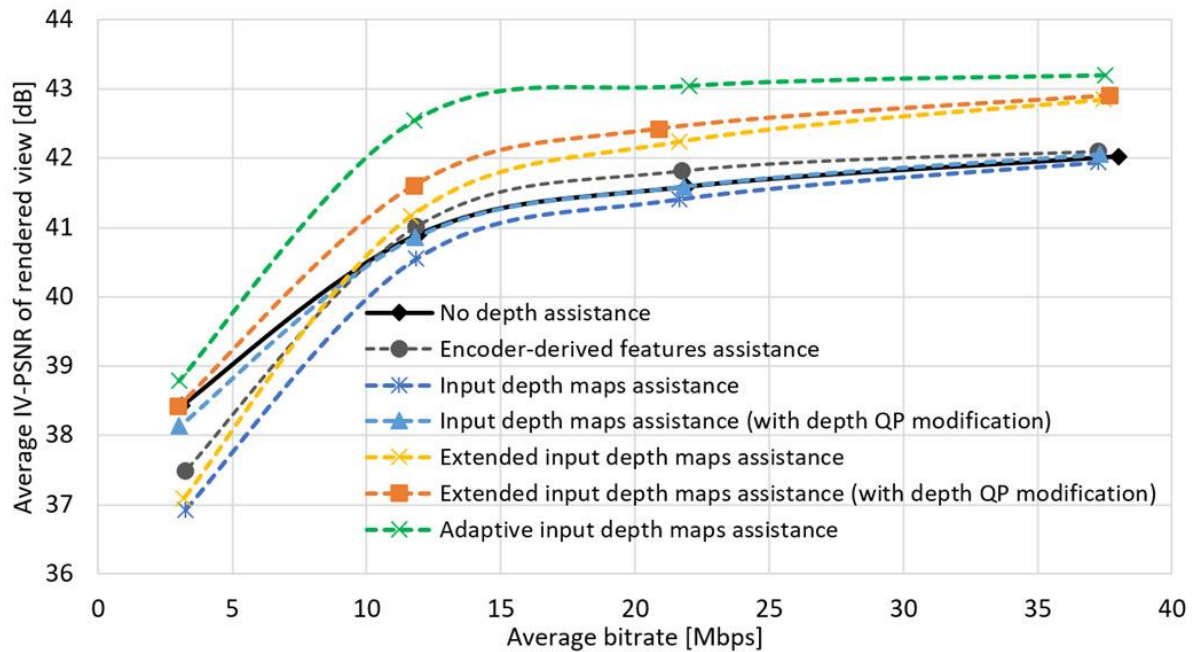


Figure 4.3.5. Average IV-PSNR of rendered views achieved by tested methods and proposed adaptive input depth maps assistance for four tested bitrates; bitrates include all texture, depth, and metadata.

The final results, summarised in Table 4.3.3, demonstrate a substantial bitrate reduction that exceeds 50% for the adaptive IDMA framework. Alongside significant gains in decoding efficiency, these results underscore the clear superiority of the proposed approach over current state-of-the-art DSDE solutions.

Table 4.3.3. *BD-Rate of tested methods versus no assistance DSDE [14].*

Type of assistance	BD-rate versus no assistance [%]
Encoder-derived features [97]	0.70
IDMA	32.70
IDMA + depth QP mod.	3.50
eIDMA	-2.70
eIDMA + depth QP mod.	-27.40
Adaptive IDMA	-53.00

In summary, the proposed approach substantially accelerates the decoding process compared to the state-of-the-art DSDE framework [14], enabling depth estimation up to three times faster. Furthermore, the integration of the input depth map assistance enhances the final quality of synthesised views by guiding the decoder-side estimator with original transmitted geometry. These results provide empirical validation for the third thesis formulated in Chapter 1:

“It is possible to enhance the quality of virtual view synthesis while maintaining the same bitrate as the state-of-the-art Decoder-Side Depth Estimation methods by leveraging additional input depth information at the decoder side.”

Flexibility, utility, and novelty of these IDMA-based methodologies were formally recognised by MPEG Video Coding experts. Consequently, these methods have been adopted into the new DSDE profile [104, 105] of the second edition of MIV standard [106], as ratified during the 140th MPEG meeting in October 2022.

The findings presented in this chapter are protected under the patent [P6] issued by United States Patent and Trademark Office. This patent encompasses view selection and labelling algorithm used in IDMA and eIDMA, a signaling flag that is used to indicate whether the depth values for a current view are present within the bitstream, or if they must be estimated. The patent also covers decoding method that uses input depth maps and depth patches from atlases in the estimation process.

5. SUMMARY OF THE DISSERTATION

This dissertation presented a cohesive set of five research publications, documenting the systematic advancement of Decoder-Side Depth Estimation (DSDE) technology for immersive video.

Chapter 2 addressed the critical process of view selection. Initial investigations established the optimal view arrangements for transmission, which were then validated within the specific context of DSDE. The findings provided empirical evidence for confirming the first thesis: *The selection of views transmitted for Decoder-Side Depth Estimation should depend on scene characteristics, specifically the proportion of occlusions.*

Chapters 3 focused on refining the depth estimation algorithm to improve its efficiency within the particularly challenging DSDE framework. Chapter 3.1 targeted the degradation caused by compression artefacts in encoded video sequences, which typically impair estimation accuracy. The results provide clear evidence confirming the second thesis: *it is possible to modify the depth estimation method to improve the quality of depth maps estimated from encoded views.* Chapter 3.2 built upon this by introducing optimisations that not only further enhance map quality but also substantially accelerate the time-intensive estimation process.

Chapter 4 detailed a novel immersive video compression method built on the MIV Geometry Absent (GA) profile. Two distinct methods were proposed, both leveraging encoder-side depth data to assist the decoder in achieving superior estimation quality and reduced latency. These developments successfully validated the third thesis: *it is possible to enhance the quality of virtual view synthesis while maintaining the same bitrate as the state-of-the-art Decoder-Side Depth Estimation methods by leveraging additional input depth information at the decoder side.*

In conclusion, all research objectives and the three formulated theses have been successfully fulfilled. The impact of this work extends beyond academic theory into international industrial applications. Throughout the period of conducting the presented research, numerous contributions were submitted by PhD candidate to the MPEG Video Coding group, where the proposed solutions were formally adopted into the MIV standard, the IVDE reference software, and the TMIV software. The industrial significance of these contributions is further underscored by the USPTO patent granted for the proposals detailed in Chapter 4.

In terms of future research directions, it would be highly beneficial to explore the integration of Gaussian splatting [107] into the framework established in this dissertation. Specifically, rather than estimating depth, a promising approach involves estimating Gaussian splats on the decoder side [108]. Consequently, the scope of this research could be extended to encompass this scenario. Building upon the investigations in Chapter 2, future work could examine whether the selection of input views has

a significant impact on the quality of the estimated Gaussian splats. Furthermore, the Point-to-block Matching method introduced in Chapter 3 could be adapted to evaluate optimal strategies for Gaussian splat estimation from compressed input views. This could involve utilising a quality metric specifically tailored to compressed data during the estimation phase to enhance reconstruction fidelity. Finally, as a potential expansion of Chapter 4, it would be valuable to investigate whether transmitting auxiliary data, such as partial depth information, could further enhance decoder-side Gaussian splatting.

REFERENCES

- [1] J. B. Jeong, S. Lee, D. Jang and E.-S. Ryu, “Towards 3DoF+ 360 Video Streaming System for Immersive Media,” *IEEE Access*, vol. 7, pp. 136399-136408, 2019,
- [2] M. Domański, O. Stankiewicz, K. Wegner and T. Grajek, “Immersive visual media — MPEG-I: 360 video, virtual navigation and beyond,” 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), Poznan, Poland, 2017,
- [3] M. Tanimoto, M. Panahpour Tehrani, T. Fujii and T. Yendo, “FTV for 3D Spatial Communication,” *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, 2012,
- [4] M. Wien, J. M. Boyce, T. Stockhammer and W.-H. Peng, “Standardization Status of Immersive Video Coding,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 5-17, 2019,
- [5] K. Müller, P. Merkle, T. Wiegand, “3-D Video Representation Using Depth Maps,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, 2011,
- [6] O. Stankiewicz, M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch and J. Samelak, “A Free-Viewpoint Television System for Horizontal Virtual Navigation,” *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2182-2195, 2018,
- [7] “Depth Map Formats Used Within MPEG 3D Technologies,” ISO/IEC JTC1/SC29/WG11 doc. MPEG2017/N16730, Geneva, Switzerland, January 2017,
- [8] R. Szelinski, “Computer Vision: Algorithms and Applications,” The University of Washington, 2022,
- [9] Y. S. Kang and Y. S. Ho, “High-quality multi-view depth generation using multiple color and depth cameras,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Singapore, pp. 1405–1410, July 2010,
- [10] S. Xiang, L. Yu, Q. Liu, and Z. Xiong, “A gradient-based approach for interference cancelation in systems with multiple Kinect cameras,” *IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 13–16, Beijing, China, May 2013,
- [11] P. Carballeira et al., “FVV Live: A Real-Time Free-Viewpoint Video System With Consumer Electronics Hardware,” *IEEE Transactions on Multimedia*, vol. 24, pp. 2378-2391, 2022,
- [12] L. Fang et al., “Estimation of virtual view synthesis distortion toward virtual view position,” *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 1961–1976, 2016,
- [13] J. M. Boyce et al., “MPEG Immersive video coding standard,” *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521–1536, 2021,

- [14] D. Mieloch, P. Garus, M. Milovanović, J. Jung, J. Y. Jeong, S. L. Ravi, B. Salahieh, "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 32, No. 9, 2022,
- [15] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan and Y.-K. Wang, "Developments International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC)," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463-1493, 2021,
- [16] Y. L. Chan, C. H. Fu, H. Chen and S. H. Tsang, "Overview of current development in depth map coding of 3D video and its future," *IET Signal Proc.*, vol. 14, no. 1, pp. 1-14, 2020,
- [17] "Test Model 9 for MPEG Immersive Video," document ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0084, Online, April 2021,
- [18] B. Bross et al. "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Tr. on Circ. and Syst. for Vid. Tech.*, vol. 31, no. 10, October 2021,
- [19] B. Salahieh, J. Boyce, F. Julien, C. Bertrand, D. Renaud, "MIV View with Decoder-Derived Depth," ISO/IEC JTC1/SC29/WG11 MPEG2019/M54492, July 2020,
- [20] D. Mieloch, A. Dziembowski, "MIV Geometry Absent for perspective content," ISO/IEC JTC1/SC29/WG11 MPEG2020/M54755, June 2020,
- [21] S. Fachada, et al. "Depth image based views synthesis with multiple reference views for virtual reality," *3DTV-Conf*, 2018,
- [22] J. Stankowski, A. Dziembowski, "Real-time CPU-based view synthesis for omnidirectional video," *30th International Conference in Central Europe on Computer Graphics, Visualisation and Computer Vision, WSCG 2022*, 2022,
- [23] V. K. M. Vadakital, A. Dziembowski, G. Lafruit, F. Thudor, G. Lee, P. R. Alface, "The MPEG Immersive Video Standard - Current Status and Future Outlook," *IEEE Multimedia* 2022, 29, 101–111, 2022,
- [24] A. Dziembowski, et al., "View selection for virtual view synthesis in free navigation systems," *International Conference on Signals and Electronic Systems* 2018, Kraków, Poland, 2018,
- [25] B. Salahieh et al., "Kermit test sequence for Windowed 6DoF Activities," Document ISO/IEC JTC1/SC29/WG11 MPEG/M43748, Ljubljana, Slovenia, July 2018,
- [26] P. Boissonade, and J. Jung, "[MPEG-I] Visual Improvement of VVS1.0.1," ISO/IEC JTC1/SC29/WG11/MPEG2019/m46263, Marrakesh, January 2019,
- [27] P. Kovacs, "BBB light-field test sequences," ISO/IEC JTC1/SC29/WG11, M35721, 2015,

- [28] "Common test conditions for MPEG immersive video," ISO/IEC JTC1/SC29/WG04MPEG VC, N0232, 2022,
- [29] Y. Sun et al., "Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video," IEEE Signal Processing Letters 24 (9), pp.1408-1412, 2017,
- [30] A. Dziembowski et al., "IV-PSNR – the objective quality metric for immersive video applications," IEEE Transactions on Circuits and Systems for Video Technology 32 (11), pp. 7575-7591, 2022,
- [31] A. De Abreu, C. Ozcinar and A. Smolic, "Look around you: Saliency maps for omnidirectional images in VR applications," 9th International Conference on Quality of Multimedia Experience (QoMEX), 2017,
- [32] V. Sitzmann et al., "Saliency in VR: How Do People Explore Virtual Environments?," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 4, pp. 1633-1642, doi: 10.1109/TVCG.2018.2793599, April 2018,
- [33] "Test Model 14 for MPEG immersive video," ISO/IEC JTC1/SC29/WG04 MPEG VCN0242, 2022,
- [34] A. Dziembowski, D. Klóska, J. Y. Jeong and G. Lee, "[MIV] Inhomogeneous view selection for omnidirectional content," ISO/IEC JTC1/SC29/WG4 MPEG 140, M60668, October 2022,
- [35] "Test Model 15 for MPEG immersive video," Document ISO/IEC JTC1/SC29/WG04MPEG VC, N0271, 2022,
- [36] L. Ilola, et al., "New test content for immersive video – Nokia Chess," Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M50787, 2019,
- [37] L. Ilola, V. K. M. Vadakital, "Improved Nokia Chess sequence," Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M57382, 2020,
- [38] B. Kroon, "3DoF+ test sequence ClassroomVideo," ISO/IEC JTC1/SC29/WG11 MPEG M42415, 2018,
- [39] R. Doré, "Technicolor 3DoF+ Test Materials," Doc. ISO/IEC JTC1/SC29/WG11 MPEG/M42349, 2018,
- [40] "Subjective Video Quality Assessment Methods for Multimedia Applications," document Rec. ITU-T P.910, ITU-T, 2008,
- [41] "Methodology for the Subjective Assessment of the Quality of Television Pictures," document Rec. ITU-R BT.500-9, ITU-R, 1998,

- [42] G. Bjøntegaard, “Calculation of average PSNR differences between RD986 curves,” ISO/IEC JTC1/SC29/WG11 MPEG M15378, Austin, TX, 2001,
- [43] “Enhanced Depth Estimation Reference Software (DERS) for Free-viewpoint Television,” ISO/IEC JTC1/SC29/WG11 MPEG/M31518, Geneva, Switzerland, 2013,
- [44] “Depth Based View Blending in View Synthesis Reference Software (VSRS),” ISO/IEC JTC1/SC29/WG11 MPEG/M37232, Geneva, Switzerland, 2015,
- [45] Z. Sun, C. Jung, “Real-time depth-image-based rendering on GPU,” International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 324–328, China, 17–19 September 2015,
- [46] J. Samelak, A. Dziembowski, D. Mieloch, “Advanced HEVC Screen Content Coding for MPEG Immersive Video,” Electronics, November 2022,
- [47] A. Wieckowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, et al., “VVenC: An Open and Optimized VVC Encoder Implementation,” IEEE International Conference on Multimedia Expo Workshops (ICMEW), Online, 5–9 July 2021,
- [48] “Manual of Immersive Video Depth Estimation,” ISO/IEC JTC1/SC29/WG04 MPEG VC/N0058, Geneva, Switzerland, 2021,
- [49] D. Mieloch, A. Dziembowski, and M. Domański, “[MPEG-I Visual] Natural Outdoor Test Sequences,” Document ISO/IEC JTC1/SC29/WG11 MPEG/M51598, Brussels, January 2020,
- [50] M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, and K. Wegner, “Multiview test video sequences for free navigation exploration obtained using pairs of cameras,” Document ISO/IEC JTC1/SC29/WG11, MPEG M38247, 2016,
- [51] “[MIV] Undistorted CBA Basketball Test Sequence for MPEG-I Visual,” ISO/IEC JTC1/SC29/WG11 MPEG2021/M58500, Geneva, Switzerland, 2021,
- [52] “[MIV] New Natural Content—MartialArts,” ISO/IEC JTC1/SC29/WG04 MPEG2023/M61949, Geneva, Switzerland, 2023,
- [53] J. Stankowski, A. Dziembowski, “Version [7.1]—[IV-PSNR: Software for immersive video objective quality evaluation],” SoftwareX 2024, 28, 101961, 2024,
- [54] M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, “Optimization of camera positions for free-navigation applications,” International Conference on Signals and Electronic Systems, ICSES 2016, Kraków, Poland, 5–7 September 2016,
- [55] “Enhanced View Synthesis Reference Software (VSRS) for Free-Viewpoint Television,” ISO/IEC JTC 1/SC 29/WG 11 M31520, Geneva, Switzerland, 2013,

- [56] A. Leontaris, P. C. Cosman, and A. R. Reibman, "Quality Evaluation of Motion-Compensated Edge Artifacts in Compressed Video," 2007 IEEE Transactions on Image Processing, vol. 16, no. 4, pp. 943-956, April 2007,
- [57] W. Sankowski, M. Włodarczyk, D. Kacperski, and K. Grabowski, "Estimation of measurement uncertainty in stereo vision system," Image and Vision Computing, vol 61, pp. 70-81, 2017,
- [58] R. Chen, F. Chen, G. Xu, et al., "Precision analysis model and experimentation of vision reconstruction with two cameras and 3D orientation reference," Scientific Reports, No. 11, 3875, 2021,
- [59] Z. Tang, R. Grompone von Gioi, P. Monasse, and J. Morel, "A Precision Analysis of Camera Distortion Models," IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2694-2704, 2017,
- [60] M. Godbolt, "Optimizations in C++ compilers," Communications of the ACM, vol. 63, no. 2 pp. 41-49, 2020,
- [61] O. Stankiewicz, M. Domański, and K. Wegner, "Analysis of noise in multi-camera systems," 3DTV Conference, Budapest, 2014,
- [62] K. Wegner, and O. Stankiewicz, "Similarity measures for depth estimation," 2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, pp. 1-4, 2009,
- [63] O. Zeglazi, M. Rziza, A. Amine, C. Demonceaux, "Structural Similarity Measurement Based Cost Function for Stereo Matching of Automotive Applications," Journal of Imaging, vol. 6, no. 8, 2020,
- [64] G. Saygili, L. Van der Maaten, E. A. Hendriks, "Adaptive stereo similarity fusion using confidence measures," Computer Vision and Image Understanding, vol. 135, pp. 95-108, 2015,
- [65] Y. Chai and X. Cao, "Stereo Matching Algorithm Based on Joint Matching Cost and Adaptive Window," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, pp. 442-446, 2018,
- [66] Y. Shin, and K. Yoon, "Adaptive spatiotemporal similarity measure for a consistent depth maps," 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta, pp. 1-4, 2017,
- [67] M. Köppel, M. B. Makhlouf, M. Müller, and P. Ndjiki-Nya, "Temporally consistent adaptive depth map preprocessing for view synthesis," 2013 Visual Communications and Image Processing (VCIP), Kuching, Malaysia, pp. 1-6, 2013,

- [68] O. Suominen, A. Gotchev, and M. Hannuksela, "Transform domain similarity measures in stereo matching," 2012 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp. 1-4, Zurich, 2012,
- [69] C. Zhang, C. He, Z. Chen, W. Liu, M. Li and J. Wu, "Edge-Preserving Stereo Matching Using Minimum Spanning Tree," IEEE Access, vol. 7, pp. 177909-177921, 2019,
- [70] H. Bay, A. Ess, T. Tuytelaars, and T. Van Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, vol. 110, Issue 3, pp. 346-359, 2008,
- [71] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," European Conference on Computer Vision, pp. 214-227, 2012,
- [72] V. Kolmogorov, and R. Zabih, "Multi-camera Scene Reconstruction via Graph Cuts," 7th European Conference on Computer Vision-Part III (ECCV '02), London, pp. 82-96, 2002,
- [73] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," 2012 IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012,
- [74] D. Doyen, G. Boisson, and R. Gendrot, "[MPEG-I Visual] New Version of the Pseudo-Rectified Technicolor painter Content," Document ISO/IEC JTC1/SC29/WG11 MPEG/M43366, Ljubljana, 2018,
- [75] Multicore Ware Inc. x265 HEVC Encoder/ H.265 Video Codec. Available on <http://x265.org>,
- [76] S. L. Ravi, M. Milovanovic, L. Morin, F. Henry, "A Study of Conventional and Learning-Based Depth Estimators for Immersive Video Transmission," IEEE 29th International Conference on Image, Video and Signal Processing (MMSP), Shanghai, China, 25-27 August 2022,
- [77] D. Mieloch, O. Stankiewicz, M. Domanski, "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," IEEE Access 2020, 8, 5760-5776, 2020,
- [78] J. Lei, J. Liu, H. Zhang, Z. Gu, N. Ling, C. Hou, "Motion and Structure Information Based Adaptive Weighted Depth Video Estimation," IEEE Transactions on Broadcasting 2015, 61, 351-362, 2015,
- [79] Y. Boykov, O. Veksler, R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Trans. Pattern Anal. Mach. Intell. 2001, 23, 1222-1239, 2001,
- [80] D. Mieloch, A. Grzelka, "Segmentation-based Method of Increasing the Depth Maps Temporal Consistency," Int. J. Electron. Telecommun. 2018, 64, 283-289, 2018,

- [81] T. Xue, A. Owens, D. Scharstein, M. Goesele, R. Szeliski, “Multiframe stereo matching with edges, planes, and superpixels,” *Image Vis. Comput.* 2019, 91, 103806, 2019,
- [82] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, “Depth map upsampling and refinement for FTV systems,” *International Conference on Signals and Electronic Systems (ICSES)*, Kraków, Poland, 11–13 September 2016,
- [83] “Common Test Conditions for MPEG Immersive Video,” ISO/IEC JTC1/SC29/WG4 MPEG2023/N0406, Geneva, Switzerland, 2023,
- [84] “A New Computer Graphics Scene, Guitarist, Suitable for MIV Edition-2,” ISO/IEC JTC1/SC29/WG04 MPEG2021/M58080, Geneva, Switzerland, 2021,
- [85] “[MIV] ERP Content Proposal for MIV ver.1 Verification Test,” ISO/IEC JTC1/SC29/WG11, MPEG2021/M58433, Geneva, Switzerland, 2021,
- [86] “Proposition of New Sequences for Windowed-6DoF Experiments on Compression, Synthesis and Depth Estimation,” ISO/IEC JTC1/SC29/WG11 MPEG2018/M43318, Geneva, Switzerland, 2018,
- [87] “[MIV] New Cadillac Content Proposal for Advanced MIV v2 Investigations,” ISO/IEC JTC1/SC29/WG4, MPEG2021/M57186, Geneva, Switzerland, 2021,
- [88] “Interdigital Mirror Content Proposal for Advanced MIV Investigations on Reflection,” ISO/IEC JTC1/SC29/WG11 MPEG2020/M55710, Geneva, Switzerland, 2021,
- [89] “Interdigital Fan Content Proposal for MIV,” ISO/IEC JTC1/SC29/WG11 MPEG/M54732, Geneva, Switzerland, 2020,
- [90] “Interdigital Group Content Proposal,” ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731, Geneva, Switzerland, 2020,
- [91] “[MIV] Dancing Sequence for Verification Tests,” ISO/IEC JTC1/SC29/WG4 MPEG2021/M57751, Geneva, Switzerland, 2021,
- [92] “[MIV] Breakfast New Natural Content Proposal for MIV,” ISO/IEC JTC1/SC29/WG4 MPEG2021/M56730, Geneva, Switzerland, 2021,
- [93] “Barn New Natural Content Proposal for MIV,” ISO/IEC JTC1/SC29/WG4 MPEG2021/M56632 Geneva, Switzerland, 2021,
- [94] A. Dziembowski et al., “The influence of a lossy compression on the quality of estimated depth maps,” in *IWWSIP 2016 Conf.*, 2016, pp. 1-4, 2016,
- [95] P. Garus et al., “Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3250-3264, 2022,

- [96] B. Szydelko, et al., “Recursive block splitting in feature-driven decoder-side depth estimation,” *ETRI Journal*, vol. 44, pp. 38– 50, 2022,
- [97] G. Clare et al., “Combination of m56626 and m56335 for Geometry Assistance SEI message,” *ISO/IEC JTC1/SC29/WG4/M56950*, 2021,
- [98] P. Garus et al., “Motion Compensation-based Low-Complexity Decoder-Side Depth Estimation for MPEG Immersive Video,” in *IEEE Int. Workshop on Multimedia Signal Proc. (MMSP)*, Shanghai, China, 2022,
- [99] “Test Model 11 for MPEG Immersive Video,” *ISO/IEC JTC1/SC29/WG4MPEG2021/N0142*, Online, October 2021,
- [100] “Text of ISO/IEC FDIS 23090-12 MPEG Immersive Video,” *ISO/IEC JTC1/SC29/WG4 MPEG2021/N0111*, Online, 2021.
- [101] “Report of MPEG immersive video CTC anchor generation,” *ISO/IEC JTC1/SC29/WG4 MPEG2022/N0234*, Online, July 2022,
- [102] “Common Test Conditions for MPEG Immersive video,” *ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0143*, Online, October 2021,
- [103] M. Chen et al., “Simplified carriage of MPEG immersive video in HEVC bitstream,” *Proc. SPIE*, vol. 11842, Art. no. 118420C, 2021,
- [104] A. Dziembowski et al., “MIV Decoder-Side Depth Estimation profile,” *ISO/IEC JTC1/SC29/WG4 MPEG VC M60667*, Mainz, October 2022,
- [105] D. Mieloch, A. Dziembowski, J. Y. Jeong and G. Lee, “On the future of decoder-side depth estimation in MPEG immersive video coding,” in *2023 Data Compression Conference (DCC)*, Snowbird, UT, USA, 2023,
- [106] “Preliminary WD4 of ISO/IEC 23090-12 MPEG immersive video Ed. 2,” *ISO/IEC JTC1/SC29/WG4 MPEG VC N0269*, Mainz, October 2022,
- [107] Y. Bao et al., “3D Gaussian Splatting: Survey, Technologies, Challenges, and Opportunities,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 7, pp. 6832-6852, 2025,
- [108] D. Mieloch, S. Perry, “A Single Atlas Is All You Need: Decoder-Side Gaussian Splatting for Immersive Video,” preprint, arXiv:2605.17002, 2026.

PUBLICATIONS OF THE AUTHOR

International journals:

1. **D. Klóska**, A. Dziembowski, A. Grzelka, D. Mieloch, "On the Selection of Transmitted Views for Decoder-Side Depth Estimation," *Applied Sciences*, Vol. 16, No. 1, pp. 72-1 - 72-18, ISSN: 2076-3417, 2026.
[M = 100, IF = 2.5]
2. B. Szydełko, D. Mieloch, A. Dziembowski, J. Stankowski, **D. Klóska**, J. Y. Jeong, G. Lee, "ECPC – versatile multicamera system calibration framework for immersive video applications," *Software X*, Vol. 26, No. 101669, 2024.
[M = 200, IF = 2.4]
3. D. Mieloch, **D. Klóska**, O. Stankiewicz, "On Alpha-Expansion-Based Graph-Cut Optimization for Decoder-Side Depth Estimation," *Applied Sciences*, Vol. 14, No. 13, 2024, pp. 5768.
[M = 100, IF = 2.5]
4. D. Mieloch, A. Dziembowski, **D. Klóska**, B. Szydełko, J. Y. Jeong, G. Lee, "A New Approach to Decoder-Side Depth Estimation in Immersive Video Transmission," *IEEE Transactions on Broadcasting*, Vol. 69, No. 4, 2023, pp. 951-965, ISSN: 1557-9611.
[M = 140, IF = 4.8]

National journals:

5. D. Mieloch, **D. Klóska**, A. Dziembowski, B. Szydełko, A. Grzelka, J. Stankowski, „Tworzenie wielowidokowych sekwencji dla wizji wszechogarniającej,” *Przegląd telekomunikacyjny*, Vol. 4, 2024, pp. 183-186.
[M = 20]
6. **D. Klóska**, B. Szydełko, D. Mieloch, „Wykorzystanie wejściowych map głębi w estymacji głębi po stronie dekodera,” *Przegląd telekomunikacyjny*, 2022, pp. 392-395.
[M = 20]
7. B. Szydełko, **D. Klóska**, A. Dziembowski, „Parametryzacja geometrii sceny dla estymacji głębi w dekodrze wizji wszechogarniającej,” *Przegląd telekomunikacyjny*, 2022, pp. 388-391.
[M = 20]

Scientific papers published in the proceedings of conferences:

8. J. Kit, **D. Klóska**, D. Mieloch, A. Dziembowski, "AI-driven enhancement of depth map consistency," 21st International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2026, Marbella, Spain, 9-11.03.2026.
[M = 70]

9. **D. Klóska**, A. Dziembowski, J. Samelak, "Versatile input view selection for efficient immersive video transmission," 31. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2023, Pilsen, Czech Republic, 2023-05-15.
[M = 70]
10. D. Mieloch, **D. Klóska**, M. Woźniak, "Point-to-Block Matching in Depth Estimation," 29th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2021, Prague, Czech Republic, 2021-05-17.
[M = 70]

ISO/IEC MPEG Documents:

11. A. Dziembowski, **D. Klóska**, B. Szydełko, D. Mieloch, G. Lee, "Selective patch margin enabling for rendering quality increase," ISO/IEC JTC1/SC29/WG4 MPEG 148, M70068, November 2024,
12. A. Dziembowski, **D. Klóska**, B. Szydełko, "EE3 M-NC5 MIV DSDE anchor generation," ISO/IEC JTC1/SC29/WG4 MPEG 148, M69530, November 2024,
13. A. Dziembowski, D. Mieloch, **D. Klóska**, J. Stankowski, B. Szydełko, G. Lee, J. Y. Jeong, "'Choreo' natural content for INVR applications," ISO/IEC JTC1/SC29/WG4 MPEG 147, M68221, July 2024,
14. **D. Klóska**, D. Mieloch, A. Dziembowski, B. Szydełko, J. Stankowski, G. Lee, "A new natural content proposal: Choreo," ISO/IEC JTC1/SC29/WG4 MPEG 146, M66990, April 2024,
15. B. Szydełko, D. Mieloch, A. Dziembowski, J. Stankowski, **D. Klóska**, J. Y. Jeong, G. Lee, "Update on the manual of Extrinsic Camera Parameters Calibration software," ISO/IEC JTC1/SC29/WG4 MPEG 146, M66989, April 2024,
16. **D. Klóska**, D. Mieloch, J. Kit, A. Śliwiński, G. Lee, "[MIV] New depth maps for selected CTC sequences," ISO/IEC JTC1/SC29/WG4 MPEG 144, m64708, October 2023,
17. D. Mieloch, B. Szydełko, A. Dziembowski, **D. Klóska**, J. Y. Jeong, G. Lee, "Versatile multicamera system calibration framework for immersive video applications," ISO/IEC JTC1/SC29/WG4 MPEG 143, m64164, July 2023,
18. D. Mieloch, A. Dziembowski, B. Szydełko, **D. Klóska**, A. Grzelka, J. Stankowski, M. Domański, G. Lee, J. Y. Jeong, "[MIV] New natural content – MartialArts," ISO/IEC JTC1/SC29/WG4 MPEG 141, m61949, January 2023,
19. Dziembowski, **D. Klóska**, J. Y. Jeong, G. Lee, "Inhomogeneous view selection for omnidirectional content," ISO/IEC JTC1/SC29/WG4 MPEG 140, M60668, October 2022,
20. D. Mieloch, A. Dziembowski, B. Szydełko, **D. Klóska**, G. Lee, J. Y. Jeong, "Decoder-side depth estimation with extended input depth assistance," ISO/IEC JTC1/SC29/WG4 MPEG 138, M59516, April 2022,

21. **D. Klóska**, D. Mieloch, A. Dziembowski, M. Domański, G. Lee, J. Y. Jeong, „Decoder-side depth estimation with input depth assistance”, ISO/IEC JTC1/SC29/WG4 MPEG 136, M58048, October 2021.

Patents:

22. J. Y. Jeong, G. Lee, D. Mieloch, M. Domański, A. Dziembowski, B. Szydełko, **D. Klóska**, “Method for decoding immersive video and method for encoding immersive video,” Patent office: USPTO, Status: accepted, Application number: US 18/096455, Filing date: 12.01.2023, Publication number: 2023/0230285, Publication date: 20.07.2023.

INVOLVEMENT IN RESEARCH PROJECTS

02.2026 – present

Efficient methods for multidimensional data processing and encoding

Role: Project Manager

03.2025 – present

Research on Immersive Video and Gaussian Splat Coding

Role: Researcher

05.2024 – 11.2024

Research on video coding techniques compliant with the second edition of the MIV standard

Role: Researcher

03.2023 – 11.2023

Research on depth estimation and video coding techniques compliant with the second edition of the MIV standard

Role: Researcher

03.2022 – 11.2022

Research on depth estimation and MIV video coding technique for non-Lambertian content

Role: Researcher

Appendix 1

Declaration of contribution to the authorship of the publications

Author contribution statement

I hereby declare that the author contributions described in Section 0.2 of the dissertation (“Analysis, extensions, and optimizations of high-efficiency video compression techniques”) accurately reflects the facts.

A handwritten signature in blue ink, reading "Dominika Klóska", enclosed within a black rectangular border.

Dominika Klóska

Appendix 2

Copies of publications included in this dissertation

A New Approach to Decoder-Side Depth Estimation in Immersive Video Transmission

Dawid Mieloch, *Member, IEEE*, Adrian Dziembowski, *Member, IEEE*,
Dominika Klóska, Błażej Szydełko, Jun Young Jeong, and Gwangsoon Lee

Abstract— This paper presents a novel approach to decoder-side depth estimation (DSDE) in immersive video transmission. The proposal improves the state-of-the-art coding approach by changing the encoding and decoding processes so that the decoder-side depth estimator can utilize partial geometry information about the scene from several available input depth maps. Proposed changes allow for significantly faster decoding and better quality of virtual viewports presented to the viewer. The paper proposes two approaches, namely input depth map assistance (IDMA) and extended IDMA (eIDMA), both compliant with the bitstream of MPEG immersive video (MIV) standard. IDMA involves sending full depth maps for a selected subset of input views, which are then used to refine and enhance the depth maps for the remaining views. In eIDMA, the decoder additionally reprojects decoded depth maps to the remaining views and enhances them with depth patches containing difficult-to-estimate areas. Proposed methods were tested under the ISO/IEC MPEG Video Coding common test conditions for MIV. The combination of two proposals, the adaptive IDMA, was shown to outperform the current state-of-the-art DSDE approach in the rendered video quality and the computational complexity of decoder. MPEG experts have appreciated the proposed approaches, which will comply with a new DSDE profile of the incoming second edition of the MIV standard.

Index Terms— depth map, immersive video, decoder-side depth estimation, video codecs, video processing

I. INTRODUCTION

The immersive video system is a system that allows the user (viewer) to virtually move in a three-dimensional scene recorded using multiple cameras or generated using computer graphics. Virtual navigation can be carried out by head-mounted displays (HMD) [1], traditional monitors with an attached control device, or using touch-sensitive tablets [2], [3].

Choosing a preferable viewpoint by the end user is associated with each viewer independently deciding where they look. Thus, in order to enable the simultaneous transmission of the

immersive video to many users, the broadcaster cannot transmit only one specific video stream. Instead, it is necessary to send a vast amount of data consisting of views' texture information from multiple cameras and some representation of the geometry information. The most commonly used representation of an acquired scene is MVD (multiview video plus depth) [4], in which the content and geometry of the scene are stored in the form of multiple views and depth maps.

In the case of such a significant increase in the amount of transferred data in immersive video systems, when compared to traditional television, typical video coders turn out to be insufficiently effective [5]. This is especially noticeable if we need to meet the constraint of pixels that can be decoded per second (*pixel rate*) in nowadays hardware decoders [6] or if the available bandwidth of the existing transmission system is not sufficient. These aspects make it very difficult to create a practical immersive video system without the use of dedicated compression algorithms which, e.g., consider the inter-view redundancy to decrease the size of the encoded bitstream. An overview of such relevant compression methods is included in Section II.

Most of immersive video compression methods assume that depth maps included in a bitstream are compressed using typical (or adapted [7]) video codecs [8]. Using video encoding for depth maps of computer-generated sequences is relatively efficient, as such depth maps do not contain errors related to depth estimation based on the texture of input views and are highly temporally consistent. On the other hand, the compression of depth maps for natural content is much more problematic. When compressed, these depth maps often require a much higher bitrate to still provide a satisfactory quality of virtual view synthesis [9]. When the required bitrate is unavailable, depth maps can contain blocking artifacts, blurring, ringing, and discontinuities, particularly around depth object boundaries. Differently than in the color image, degradation of the depth map will lead to an error in 3D scene reconstruction, causing more noticeable artifacts in the

Manuscript received March 10, 2023, revised June 13, 2023. The work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory). (*Corresponding author: Dawid Mieloch*).

Dawid Mieloch, Adrian Dziembowski, Dominika Klóska, and Błażej Szydełko are with Poznań University of Technology, 60-965 Poznań, Poland

(e-mail: [dawid.mieloch; adrian.dziembowski; dominika.kloska; blazej.szydelko]@put.poznan.pl).

Jun Young Jeong and Gwangsoon Lee are with Electronics and Telecommunications Research Institute, Daejeon, 34129 Republic of Korea (e-mail: [jyj0120; gslee]@etri.re.kr).

synthesized views [7]. Thus, reducing compression artifacts in depth maps used for immersive video applications is crucial.

An efficient depth refinement technique could be considered to improve the efficiency of depth map compression, as improvement of temporal and inter-view consistencies of depth maps can significantly increase the quality of virtual views synthesized for the final user [10]. Such refinement can be performed either as pre-processing before encoding, improving the effectiveness of depth compression [11], or after decoding, to remove compression artifacts in post-processing [12]. Any suitable pre-processing method can improve the quality of depth maps, as this step can be considered part of sequence acquisition and is independent of the encoding method used in the next step. On the other hand, the post-processing method useful for immersive video applications needs to be versatile to handle multiview videos of very diverse characteristics (e.g., types of cameras and their arrangement) but also variable compression errors for different bitrates [5]. Moreover, an additional computationally expensive step increases the time required to render a new virtual view for the user, making real-time implementations much more challenging to provide.

In order to overcome the problems with depth map compression, researchers considered a scheme called decoder-side depth estimation (DSDE) [6]. In such a compression method, the depth estimation step is moved from the process of the scene acquisition, done before encoding, to the decoder and is performed on compressed full views [13], [14]. Based on this coding method, we propose a novel method of immersive video compression with simultaneous multiview depth estimation and refinement performed in the decoder using a set of compressed depth maps included in the bitstream – Input Depth Map Assistance (IDMA).

The proposal changes the essential principle of previous approaches in encoding based on decoder-side depth estimation (described in Section II) that depth maps are not encoded in the bitstream, as they are estimated in the decoder using decoded textures of input views and camera parameters [15]. Our approach assumes that a subset of depth maps acquired in the depth estimation process occurring at the encoder side is also available in the decoder of immersive video. Then, the decoded depth maps available in the decoder are utilized in multiview depth estimation. It enables simultaneous enhancement of their quality, and the quality of depth maps estimated for remaining views, as the set of transmitted depth maps helps the estimator in the reconstruction. Moreover, as depth maps are not sent for all views, we preserve the high share of the texture of views in the bitrate, keeping their quality sufficiently high to provide good-quality depth estimation and virtual view synthesis.

We also propose a further extension of the proposal – extended IDMA (eIDMA). We modify immersive video encoding with a unique feature that allows the reprojection of decoded input depth maps to other views and the subsequent addition of supplemental depth patches (fragments) for further enhancement. An additional set of patches includes the

elements of a scene that were not visible in the transmitted neighboring views and could not be estimated from available views. These elements, however, could be recovered from the depth maps of the non-transmitted views existing at the encoder side. Therefore, such elements could be further used to assist in inferring regions that could not be reconstructed by simple depth reprojection from available views.

As discussed earlier, the existing video encoding methods (e.g., HEVC [16] or VVC [17]) are not sufficiently efficient for depth maps. However, the proposed method of modified DSDE, which includes depth refinement, was demonstrated in performed experiments to provide a significantly better quality of decoded virtual views and simultaneously decrease the computational complexity in comparison with other DSDE-based immersive video compression methods.

To summarize, the novelties of the proposals described in this paper are as follows:

- IDMA coding scheme:
 - sends only a subset of depth information, preserving a high share of texture information in the bitrate, ensuring good-quality depth estimation and virtual view synthesis,
 - runs the view labeling algorithm twice to properly select the depth maps to be sent, ensuring maximum scene coverage and minimal overlap between transmitted depth maps,
 - involves utilizing depth maps available in the decoder to improve the quality of depth maps estimated for the rest of views,
 - employs a modified depth estimation method based on global multiview optimization to refine the available input depth maps in the decoder.
- eIDMA coding scheme:
 - includes additional reprojection of available input depth maps to other views, further improving the quality of estimated depth maps,
 - in addition to the full-view depth maps in IDMA, sends the depth atlases that contain partial depth information for the rest of views,
 - feeds resulting depth maps into the modified depth estimator, which estimates new depth only for empty regions, reducing computational complexity.
- Adaptive IDMA coding scheme:
 - enhances the encoder with an automatic selection mechanism for determining the most suitable method, dynamically choosing between the eIDMA and IDMA approaches, depending on the characteristics of the content being processed.
 - further improves the efficiency and performance of the encoding process, ensuring that the possibly best results are achieved for both computer-generated and natural content.

After the review of state-of-the-art immersive video compression in Section II, Section III provides a detailed description of the proposal. Section IV explains the methodology of performed experimental tests, describes the used dataset, and, finally, includes the comprehensive comparison of the compression efficiency and the measured runtimes of tested methods. Section V concludes the paper and provides remarks on future works and forecasted applications of the proposal.

II. IMMERSIVE VIDEO COMPRESSION

A. Basic solutions

MVD simulcast encoding stands for the compression method in which each texture and depth video is encoded independently using the typical 2D video codec (such as AVC [18], HEVC [16]) without considering the spatially redundant information commonly existing at neighboring views [19]. This method has the merit of simple implementation, but it requires high-performance devices that support the simultaneous running of several video decoders and a large memory buffer for storing multiple videos for rendering. For most of the widely-used consumer devices, it is hard to satisfy these requirements, as they typically allow up to 4 simultaneous decoders for 4096×2048 videos at 30 fps [20], [21]. Therefore, for immersive video, for which the number of input views significantly exceeds these values, the MVD simulcast cannot be considered to be an optimal solution.

Unlike MVD simulcast, MV-HEVC and 3D-HEVC [22] exploit the similarity information among neighboring views for supporting inter-view prediction so that more efficient compression can be possible. However, these two tools still have restrictions that hinder their broader adoption in immersive video systems. First, they are only designed to work properly with the MVD data captured by the coplanar (or even linear) arrangement of traditional cameras with a narrow baseline, thus, the size and shape of the viewing zone are highly limited in comparison with immersive systems composed of omnidirectional cameras [23]. Moreover, these techniques are built on top of the HEVC codec, therefore, it is impossible to improve its performance by utilizing newer compression standards, such as VVC [17]. In addition, further minor drawbacks, such as the reference views that will be used to predict non-reference views, must be manually determined, causing further usability issues.

B. MPEG immersive video (MIV)

The current state-of-the-art compression technology for immersive video was developed by the ISO/IEC MPEG Video Coding group under the name MPEG immersive video (MIV) [23], [24]. The idea used in the MIV Main profile of this standard is the assumption that a certain small number of basic views, gathering most of the scene information, should be fully encoded, while supplementary information visible from non-basic (“additional”) views may be communicated in the form of small fragments (“patches”) (Fig. 1), or omitted, enabling increasing the number of basic views.

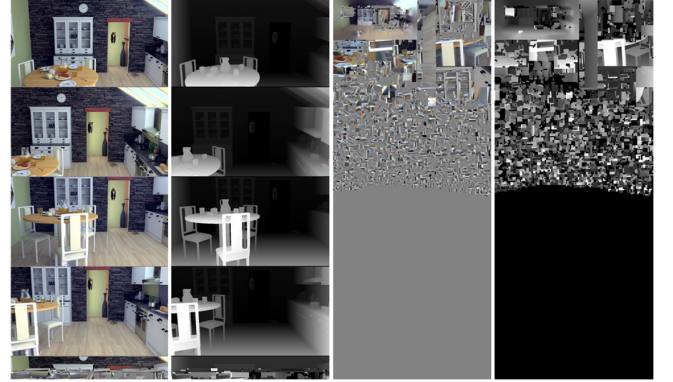


Fig. 1. Example of four atlases produced by MIV encoder running in MIV Main profile: first and third atlas contain texture information in the form of full views and patches, while the second and fourth atlases contain corresponding depth.

Unlike the previous multiview coders, which define the entire encoding process, from reading input views to creating one common bitstream, the MIV encoder (left part of Fig. 2) is a kind of pre-processing of multiview video. The encoder removes the inter-view redundancy during the pixel pruning step and changes the representation of the input data (from separate views to “atlases” containing packed information from many views). The MIV decoder (right part of Fig. 2) acts as post-processing of decoded data, recovers the input views from atlases, and produces a virtual view for the position and orientation requested by the final user.

The MIV atlases can be coded using any 2D video encoder, making the MIV a codec-agnostic technique. Thus, with MIV, it is possible to use effective yet time-consuming compression using the newest VVC encoder, as well as a less effective but much faster HEVC [16].

While this freedom in terms of the used video encoder is an unquestionable advantage of the MIV Main, the standard assumes using the same encoder both for textures and depth maps. The DSDE scheme, introduced in Section I, overcomes the problem of using the video encoder not adapted to properly compress depth maps by the depth estimation moved to the decoder. In the case of MIV, this scheme is implemented in Geometry Absent (GA) profile, which was shown to provide very high efficiency for low bitrates (smaller than 10 Mbit/s for a whole three-dimensional scene) [6].

C. Decoder-side depth estimation (DSDE) in MIV coding

In order to facilitate the decoder-side depth estimation process, which requires inter-view redundancy to estimate depth properly, the pixel pruning is not performed in the encoder for this profile (Fig. 2), therefore, produced atlases contain only full views (Fig. 3). Some first modifications of this profile, proposed in [25], shown also using partially complete views composed of patches for decoder-side depth estimation.

Naturally, DSDE requires the use of high-quality depth estimation, which is already possible in the decoder, but such estimation is usually highly time-consuming [6], [26], making it much harder to use this scheme in real-time applications. The possibility of using additional depth refinement can also be

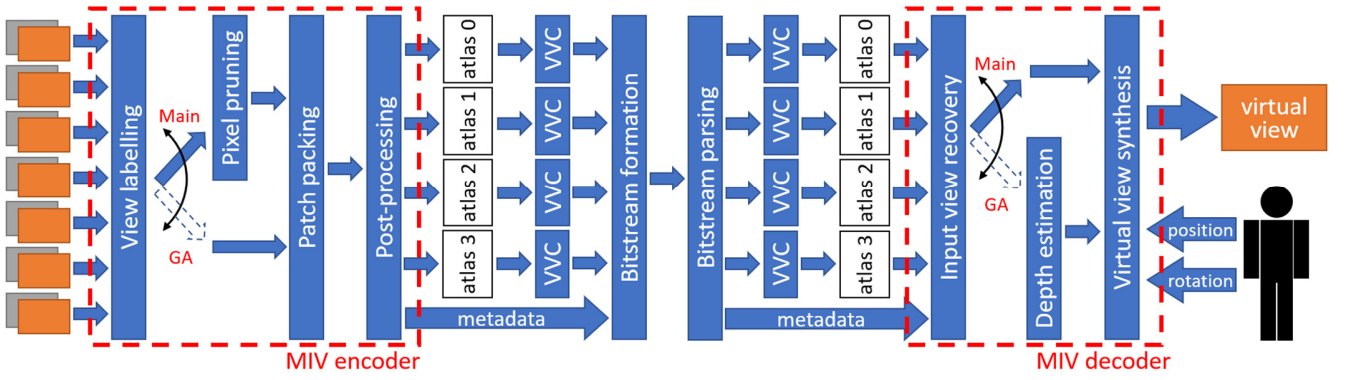


Fig. 2. The simplified scheme of MIV encoder and decoder in MIV Main and MIV Geometry Absent (GA) profiles.

considered, nevertheless, it would further increase the complexity of the decoder.

The quality of depth maps estimated in the decoder is lower than for depth maps estimated before the encoding. First, the number of views available in the decoder, due to pixel-rate constraints, is smaller than in the encoder. The depth maps quality was shown in [10] to depend on the number of views used in the estimation process. Besides that, the quality of input views is lower in DSDE due to compression-induced errors, which also affect the quality of estimated depth [14].

The further important disadvantage of DSDE is not utilizing high-quality depth maps, which usually are available on the encoder side. One of the previously proposed solutions is utilizing a set of encoder-derived features [6], [15], [27], which helps the decoder-side estimator to improve the quality of depth maps and to speed up the process. It is done by including into the bitstream information, e.g., how to narrow the range of possible depth levels which should be considered for each block of the depth map or which block can be skipped in the estimation, as the depth values from the previous frame can be copied and applied for static regions.

Unfortunately, in this scheme, it is not possible to use a depth estimator which is not adapted to utilize these features, highly narrowing the set of suitable methods. Moreover, features can use from 1 to 2 Mbit/s of available bandwidth [27], [28], so for low bitrates (less than 5 Mbit/s), the bitrate left for textures is very low, affecting the delivery of sufficient information for enabling accurate depth estimation and virtual view synthesis. Furthermore, as features are block-based, it is hard to encode more advanced structures/elements from the depth map, as using small blocks increases the bitrate significantly (changing the grid size of 128×128 pixels to 64×64 doubles the bitrate [27]).

The approach described in [29] employs depth estimation in both the encoder and decoder. The encoder-side estimation leverages motion vectors extracted from encoded texture videos to determine whether the depth map for a given block should be estimated from decoded views (similar to default DSDE) or reconstructed from previous depth maps using motion compensation. This method effectively shifts a substantial portion of the computational burden from the decoder to the

encoder, resulting in approximately 20 times faster decoding for the optimal parameter settings. Nevertheless, as the proposal assumes that motion vectors are extracted from videos encoded with AVC encoder, it breaks the assumption that any video codec can be used, so it is not compatible with MIV framework.



Fig. 3. Example of four atlases produced by MIV encoder running in MIV Geometry Absent profile: all four atlases contain full views.

III. INPUT DEPTH MAP ASSISTANCE

A. Overview

This section provides a detailed description of our two proposals for new immersive video compression methods: input depth map assistance (IDMA) and extended input depth map assistance (eIDMA).

The main idea of the proposals is shown in Fig. 4. Both of the proposed methods are based on the scheme of encoding a part of depth maps in order to improve the depth estimation performed at the decoder side, which is usually performed only on the basis of textures of available views (Fig. 4a). In the IDMA approach, the input depth maps, available for a subset of views, are simultaneously refined and used to improve the quality of depth maps estimated for other views (Fig. 4b). It is possible by utilizing the modified depth estimation method based on global multiview optimization.

In order to further improve the quality of estimated depth maps and decrease the computational complexity of their estimation, we propose the eIDMA approach (Fig. 4c), in which the set of available input depth maps is reprojected to other views for which the input depth maps are not available for all pixels. These reprojected depth maps are also fed into the

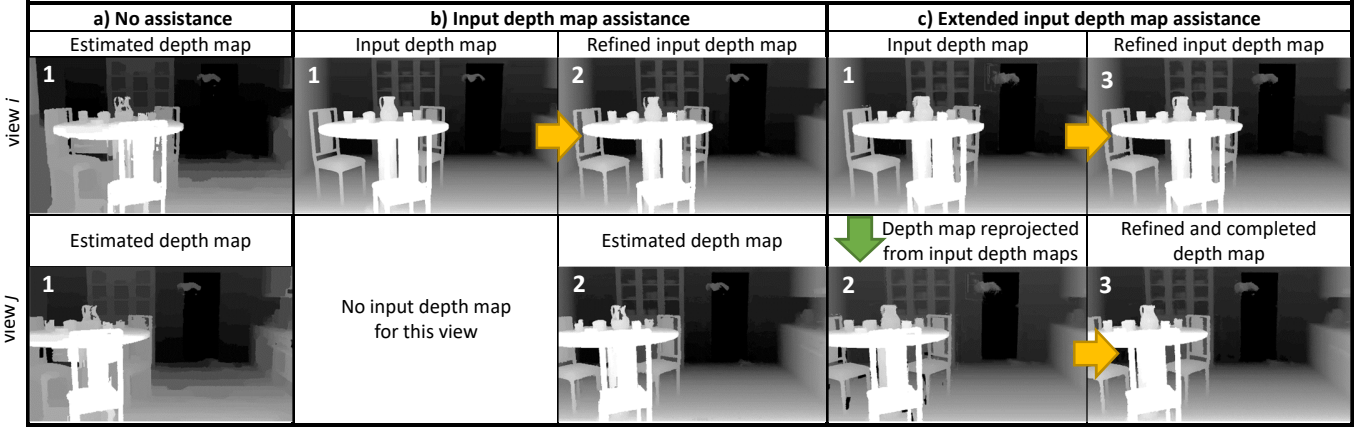


Fig. 4. Simplified overview of decoder-side depth estimation for two views, performed without assistance and for two proposed methods: a) depth maps for views i and j are estimated in the decoder (1); b) input depth map of view i decoded from the bitstream (1) is used in the joint process of estimation and refinement of depth maps for views i and j (2); c) decoded input depth map of view i (1) is reprojected to view j and becomes input depth map for view j (2), both input depth maps are used in the joint estimation and refinement process, resulting in depth maps for views i and j (3).

modified depth estimator, which refines them in the same way as input depth maps in the IDMA approach but also estimates depth for empty regions.

The proposals are based on the framework provided by the MIV compression standard [23]. Following subsections describe modifications proposed to the MIV encoder and decoder implemented in the Test Model for MPEG immersive video (TMIV) [30] and the required modifications of the depth estimator, which is treated as a part of the decoder.

B. Proposed modifications to the MIV encoder

1) Transmitted depth selection

The first step of the state-of-the-art MIV encoder is view labeling (Fig. 2). In this step, the encoder analyzes camera arrangement and derives a set of input views that carry the most non-redundant information (i.e., views with the smallest inter-view overlap). These views (called “basic views”) are packed into atlases in their entirety, while others (“additional views”) are either pruned and packed as a mosaic of smaller patches containing crucial and non-redundant depth fragments for basic views sent within the latter $M - N$ texture atlases. Both approaches are designed to work with arbitrary M and N values, but the simplest case is when N equals one, and the below description is written based on this scenario.

TABLE I
TYPES OF TRANSMITTED DATA FOR MIV MAIN AND MIV GA PROFILES.

View type	Input views	
	Basic	Additional
Transmitted data	MIV Main: texture, depth MIV GA: texture	MIV Main: texture, depth MIV GA: none

Unlike in typical approaches, where depth information is sent for all views (all basic and additional views) or not sent at all, in both proposed approaches (IDMA and eIDMA), only a subset of depth information is sent, resulting in texture atlases containing basic views and depth atlases, where the number of atlases for texture is higher than for depth. Fig. 5. shows an example of four atlases that consist of three texture and one depth atlases.

In the IDMA approach, the depth atlases contain full depth maps for views sent within the first N texture atlases among a

total number of M , while depth information for the remaining basic views is skipped. In eIDMA, the depth atlases contain, besides the same information as in IDMA, a mosaic of smaller patches containing crucial and non-redundant depth fragments for basic views sent within the latter $M - N$ texture atlases. Both approaches are designed to work with arbitrary M and N values, but the simplest case is when N equals one, and the below description is written based on this scenario.



Fig. 5: Example of four atlases produced by MIV encoder in the proposed methods: three texture atlases and one depth atlas: IDMA – only the yellow part, eIDMA – entire depth atlas (yellow and green parts).

In order to maximize profits gained by sending partial depth information, the MIV encoder has to properly select the depth maps which will be sent (additional yellow block in Fig. 6). In general, transmitted depth maps should cover the possibly largest area of the scene, and the overlap between them should be minimized. The MIV view labeling algorithm is already providing such a selection. Therefore, in both proposed IDMA solutions, the view labeler is run twice. In the first step, the basic views are selected from all input views. Next, basic views are analyzed in order to select “essential views” – views for which the depth maps are transmitted in full shape (Table II). The number of essential views may depend on the use case. In the simplest scenario, there are as many essential views as fit into an atlas.

Essential views are packed into the first texture atlas, while other basic views into the two remaining texture atlases (Fig.

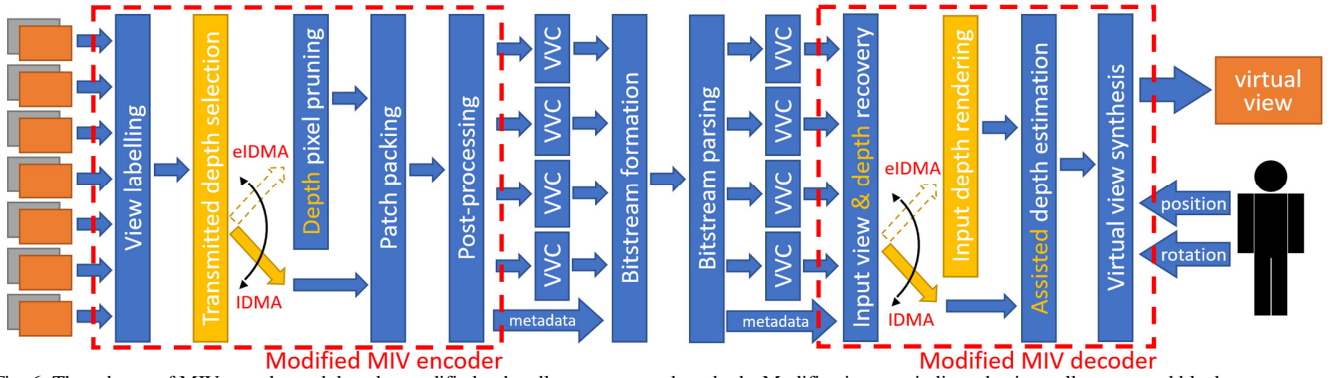


Fig. 6. The scheme of MIV encoder and decoder modified to handle two proposed methods. Modifications are indicated using yellow text and blocks.

7). In the IDMA approach, depth maps for views packed into the second and third atlas are estimated fully at the decoder side (Fig. 4b).

TABLE II
TYPES OF TRANSMITTED DATA FOR PROPOSED METHODS.

View type	Input views		
	Basic		Additional
	Essential	Non-essential	
Transmitted data	IDMA: texture, depth eIDMA: texture, depth	IDMA: texture eIDMA: texture, depth	None

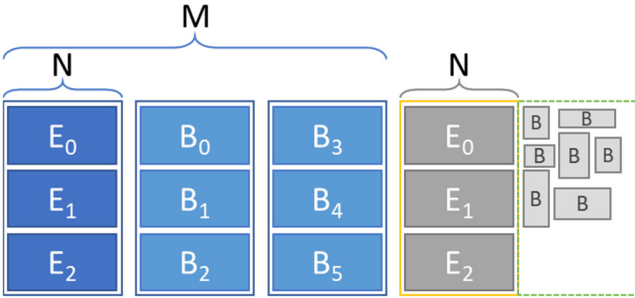


Fig. 7. Configuration of atlases in the proposed approach (blue – texture, grey – depth). M – total number of texture atlases (here – three), N – number of texture atlases for which the depth information is fully transmitted (one), E – essential views (three), B – basic non-essential views (six). The right part of the last atlas (with dotted boundary) is sent only in the eIDMA approach and contains partial depth information for all basic non-essential views.

During the experimental tests of our proposals, we identified that their efficiency is highly dependent on the characteristics of a multiview sequence being compressed. In the case of computer-generated sequences, where accurate depth maps are available, utilizing the eIDMA approach proves advantageous as it consistently delivers the highest quality. For natural content, where depth maps in the encoder are estimated based on captured views, the IDMA approach yields the best quality, as even if depth maps for these sequences are estimated using the most effective methods with optimized parameters, they cannot be considered as ground truth. Therefore, providing only partial assistance for such sequences is more preferable, as potential errors in the input depth maps will not be propagated to all views, which could occur in the case of eIDMA.

We provide an automatic selection mechanism for determining the most suitable method based on the content, dynamically choosing between the eIDMA and IDMA approaches. The required way for automatically indicating the

quality of depth maps is already available in the MIV bitstream in the form of the automatically calculated depth quality flag, described in detail in [23]. Depth accuracy is used in MIV to determine how the encoder behaves and is also signaled to the decoder. A simple assessment of the geometry is applied based on the first frame, where input views are reprojected to the position of the other views. If the reprojected geometry value is higher than the geometry value of the collocated pixel or its neighbors, it is counted as inconsistent, and the quality of the geometry is set to low. A default threshold of 0.1% is used to determine if the inconsistent pixel percentage is too high.

If the flag indicates the low quality of depth maps, then the IDMA method is chosen (as it is the best for natural content). If the quality is assessed as high, then the eIDMA method is used.

2) Depth pixel pruning

The eIDMA approach does not need to estimate depth for all the pixels of these views, and a significant part of their depth maps is created by reprojection of depth from essential views (Fig. 4c). However, in such an approach, the transmission of depth maps only for essential views is insufficient. In this case, some areas of the depth map could not be properly estimated.

An example of this problem is presented in Fig. 8, where a depth map (Fig. 8b) was rendered using only information contained in the first atlas (Fig. 8a). Pixels of this view can be divided into three sets:

1. Significant part restored properly (areas visible in views contained in the first atlas – Fig. 8a),
2. An area with no information (white area: a large part of the floor, disocclusions behind chess pieces at the back),
3. An area with wrong information (e.g., the part with the head of the knight contains background information).

Depth values for the first set were reprojected correctly and match the reference depth map presented in Fig. 8c. For pixels from the second set, there is no depth information, so the depth will be estimated at the decoder side. However, pixels from the third set already have some reprojected information, and while this information is obviously wrong, the decoder cannot determine which fragments of the reprojected depth map belong to this set.

The decision whether a pixel of the basic (non-essential)

view belongs to the first, second, or third set is taken at the encoder side by reprojection of pixels from essential views. The pixel belongs to the first set if its depth value is equal or smaller (closer to the camera) than the co-located depth of reprojected pixels. If no information is reprojected to the position of the pixel (from any essential view), it belongs to the second set. Otherwise, the pixel belongs to the third set.

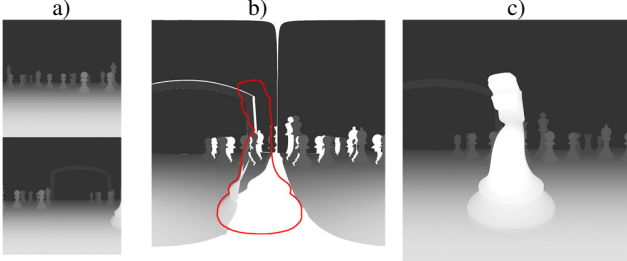


Fig. 8. Motivation for sending additional depth information. a) first depth atlas, b) depth for view rendered using information from the first atlas (silhouette of the knight is highlighted), c) reference depth for the view from the second atlas (not transmitted).

Therefore, as presented in Fig. 6, the eIDMA approach requires one additional step of depth pixel pruning, performed on depth maps for non-essential views. In this step, the modified MIV pruning algorithm is used: a pixel is pruned (removed) if it is inter-view redundant, as in MIV, but a second pruning condition is added, i.e., a pixel is pruned also if its depth can be estimated at the decoder side (on the basis of decoded textures). The only pixels which are preserved after the pruning step are the ones, which contain inestimable foreground objects, thus, the pixels for which the background from other views would be reprojected instead (e.g., the top part of the knight in Fig. 8b). These not-pruned pixels are packed into the bottom part of the depth atlas as a mosaic of patches (bottom of the fourth atlas in Fig. 5). The depth map rendered using additional depth patches is presented in Fig. 9b.

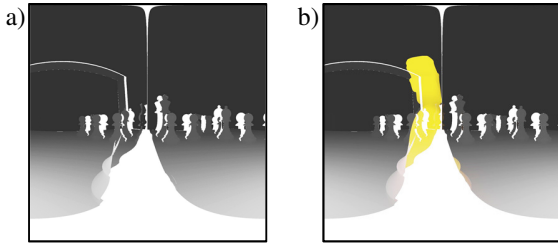


Fig. 9. a) A depth map rendered using depth maps from the first atlas and b) using depth maps from the first atlas together with additional depth patches; depth transmitted in additional patches was highlighted in yellow.

It should be noted that the height of the depth atlas in eIDMA is two times greater than for the IDMA approach. However, the resolution of depth atlases in MIV is by default reduced (twice in both directions) [23], [30], making the overall size of the depth atlas still much smaller than for each texture atlas.

C. Proposed modifications to the MIV decoder

1) Input view and depth recovery

In the first step, the modified MIV decoder (Fig. 6) unpacks received atlases and recovers all views and depth maps included

in the bitstream. This step is performed using the state-of-the-art MIV algorithm [24] and results in the recovery of all basic views and all transmitted depth maps (depth for essential views in IDMA and depth for all basic and essential views in the eIDMA approach).

All recovered basic views are fed into the modified depth estimation algorithm (described in Section III.C.3). Recovered depth maps for essential views are processed depending on the chosen approach. In the IDMA approach, the depth estimator receives B recovered input views, and E recovered input depth maps, where B and E are the numbers of basic and essential views, accordingly, and $B > E$.

2) Input depth rendering

In the eIDMA approach, the depth estimation algorithm also uses B recovered input views, but the number of input depth maps is extended from E to B . Input depth maps for essential views are fed into the depth estimator in an unchanged form. For non-essential views, the recovered depth maps are mostly empty (Fig. 10a), as they initially contained redundant information, which was pruned at the encoder side. However, these areas can be rendered by reprojection of depth from input depth maps of other views, resulting in more complete depth maps (Fig. 10b), which are fed into the depth estimation algorithm in this form.

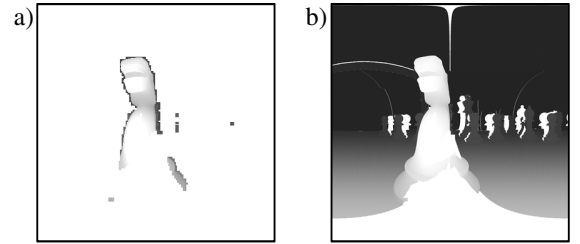


Fig. 10. A depth map for non-essential view; a) recovered, b) rendered by reprojection from other views and extended using the recovered depth map.

3) Assisted depth estimation

Besides changes in the MIV decoder, described in previous subsections, in order to implement the proposed compression methods, it was required to adapt some depth estimation method to be able to handle input depth maps. For this purpose, we decided to modify the Immersive Video Depth Estimation (IVDE [10]), which is the ISO/IEC MPEG Video Coding reference software for depth acquisition. This software has already been aligned with the MIV standard and meets the requirements imposed by the MIV on the depth estimation process [6], making it a natural choice for the scheme presented in this article.

Depth estimation in IVDE is based on cost function minimization, as defined in [10]. A minimum is determined using the graph cut algorithm [31]. Unlike other methods of depth map estimation based on graphs, in which graph vertices represent each pixel of the input views, each vertex corresponds to one superpixel [32]. To obtain cross-view consistency of the estimated depth maps, the cost of matching potentially corresponding points in adjacent views is not calculated

independently for each view but has been replaced by the cross-view matching cost defined between a pair of segments corresponding to the currently considered depth. Depth estimation is started with all superpixels assigned to the farthest depth level. The graph cut algorithm assigns points to the closer depth level in each subsequent iteration. Therefore, the estimator performs as many graph cut optimizations as there are levels of depth.

IVDE includes the method of temporal consistency enhancement in which some parts of depth maps can be copied from the previous frame and marked as unchangeable (by excluding the corresponding vertex from the optimization), while for the remaining parts, the depth estimation is performed as usual. This mechanism of excluding some vertices from the graph was utilized to handle input depth maps available in the decoder in order to use values from non-empty parts of input depth maps for the currently estimated depth map. If some area in the input map is empty (white area in Fig. 10b), then the depth for this area is estimated from scratch, and all depth levels are checked for these areas.

What is crucial in the depth estimation method implemented in IVDE, is that the process of depth optimization is global, i.e., depth maps for all views are estimated simultaneously in one common process. Therefore, if a set of high-quality depth maps is available and used as input depth maps, then the quality of depth maps for other views is also improved (Fig. 4). Moreover, the complexity of the estimation process is reduced, as the area for which the depth has to be estimated from scratch is significantly decreased.

The process described above provides utilization of input depth maps in the decoder-side depth estimation, however, it assumes that the quality of these depth maps will be sufficiently high. As discussed in Section I, the typical video codecs used for depth maps compression, also used in the MIV coding scheme, can lead to noticeable artifacts in the final, synthesized virtual views. Therefore, in order to reduce the compression-induced depth errors, the IVDE was modified to refine depth maps during the estimation process. This opens the possibility of using higher quantization when compressing depth maps – if less bits will be used for depth transmission, then the bitrate of textures will be increased, making the refinement easier to perform. The case of increasing quantization parameter for depth maps compression was also tested in experiments shown in Section IV.

In IVDE, the depth is estimated not for each pixel but for each segment (superpixel) calculated using texture information. In the beginning, for each pixel of a superpixel, it is checked what are the smallest and the largest depth values in the input depth map. These two values determine a small range of depth values for each superpixel that will be checked during the global optimization using graph cut. This way, even if the depth of some pixels is damaged during the compression, the depth values of adjacent pixels will be used to correct it.

Fig. 11 shows the fragment of the final estimated depth map

when very high compression of depth maps was used. As can be observed, errors resulting from strong compression (blurred regions in the bottom) and depth reprojection are significantly reduced. When compared to the depth map estimated in the decoder-side depth estimation scheme (without input depth maps), the proposed eIDMA scheme still provides a depth map that is more similar to the depth map available in the decoder, even if high compression is introduced.

It should be noted that the depth map presented in Fig. 11c may seem visually more plausible than the one in Fig. 11d. However, the depth map before IVDE contains artifacts, which are extremely destructive in terms of the quality of rendered views presented to the final viewer – blurred edges. Such edges imply the appearance of disturbing artifacts (i.e., ghost edges [33]) in rendered views. As a result, in research on immersive video, the direct quantitative comparison of depth maps is rarely used as it does not express the quality of virtual views presented to the final user [34], as even in the case of using ground-truth depth maps, some artifacts in view synthesis can occur [35]. Therefore, in our study, the synthesis-based quality assessment is used in experimental comparison, with the methodology described in Section IV.B.

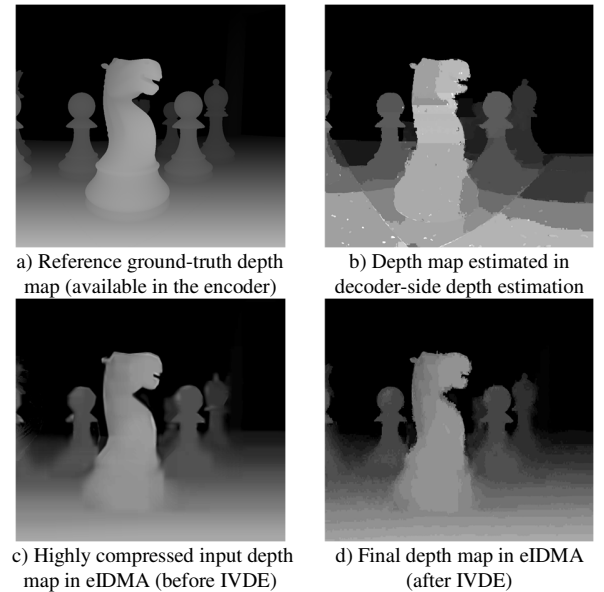


Fig. 11. Fragment of depth map: a) reference ground-truth; b) acquired in decoder-side depth estimation; c) compressed input depth map in eIDMA; d) depth map acquired by refinement of input depth map performed in decoder-side depth estimation process.

It should be noted that the proposed scheme is not restricted to be used with the described depth estimation method. While the efficiency of using the proposal with different estimators can vary, IDMA and eIDMA can be used together with any depth estimation method, similarly to the encoder-derived geometry features scheme (already tested with different estimators, e.g., IVDE [6] and DERS [13]).

D. Summary

The presented review of state-of-the-art compression schemes for immersive video shows the main disadvantages of

related work. In this section, we shortly summarize how our proposal deals with issues identified in Sections I and II:

- When the required bitrate is unavailable, depth maps can exhibit visual artifacts, blurring, and discontinuities around object boundaries. However, our approach includes depth refinement during the depth estimation process, eliminating the need for an additional computationally expensive refinement step.
- DSDE relies on high-quality depth estimation that is usually time-consuming, but our scheme significantly speeds up the estimation process as depth has to be estimated for smaller area.
- Unlike basic DSDE methods, we utilize information from high-quality depth maps typically available on the encoder side.
- The basic DSDE scheme encodes only full views in atlases, decreasing the amount of data that can be packed into atlases. In eIDMA, we use pixel pruning to allow packing of small non-redundant depth patches.
- Our approach uses standard video compression for depth maps, which is not possible when depth features are used to improve DSDE. Depth maps make it easier to represent complex structures, eliminating the need for bitrate wastage on signaling block division.
- Our scheme is based on the MIV scheme of coding by video pre-processing. Unlike previous top-performing encoders like MH-HEVC, our scheme allows for continuous improvements even after the completion of the MIV standard works.

IV. EXPERIMENTAL RESULTS

A. Overview

In order to evaluate the performance of proposed compression methods, we conducted a comparison of IDMA and eIDMA with two state-of-the-art compression schemes for immersive video: the basic MIV Decoder-Side Depth Estimation anchor [36] and the MIV DSDE with Geometry Assistance SEI (encoder-derived features assistance) [28]. Section IV.B describes the methodology of performed experiments, while Section IV.C presents their results.

B. Methodology

The conditions of experiments are based on the common test conditions defined by ISO/IEC MPEG Video Coding to provide a fair comparison between methods for immersive video coding [37]. During the test, we utilize TMIV – Test Model for MPEG immersive video 11.0 [30], which implements the MPEG immersive video coding standard. In MIV DSDE and MIV DSDE with Geometry Assistance SEI, we use publicly available unmodified encoder and decoder, while for the IDMA and eIDMA, we utilize the proposal described in Section III. The following pixel rate constraints are imposed on all configurations:

- The combined luma sample rate across all decoders shall not exceed 1,069,547,520 samples per second (as in HEVC Main10 profile level 5.2 [38]).
- Each coded video picture size shall not exceed 8,912,896 pixels (i.e., 4096×2048).
- The number of decoder instantiations shall not exceed 4.

For video compression, the VVenC codec [17], a fast implementation of VVC, is used. We use four rate points (RP) in all experiments: 3, 12, 22, and 38 Mbit/s, referred as RP4, RP3, RP2, and RP1, respectively. These values represent a practical range of bitrates used for the compression of immersive video, easily broadcasted using 5G networks [39]. The quantization parameter (QP) values used in encoding textures and depth atlases were tuned for all tested methods independently to match rate points as close as possible. All presented bitrates include summarized bitrates used for texture, depth, and other required metadata.

For depth estimation performed at the decoder side, we use Immersive Video Depth Estimation (IVDE [6], [10]), which is used in MIV experiments conducted by ISO/IEC MPEG Video Coding, as defined in common test conditions for MIV [37]. The same depth estimation method is used in all tested configurations, which ensures testing the coding scheme, not the depth estimation method itself. We use publicly available IVDE 7.0 [40], as it already provides support for encoder-derived features and input depth maps, implemented earlier by the authors of this paper.

After depth maps are estimated, the TMIV renderer is used to synthesize virtual views in the same position as all views of the used test sequences. In order to measure the quality of virtual views, the IV-PSNR metric [35] is used. The IV-PSNR values presented in the results are averaged for all views and test sequences. A brief summary of test sequences is available in Table III. In all experiments, 17 frames are used for the evaluation.

Configurations of tested compression schemes were as follows:

- MIV DSDE:
 - ❖ as in common test conditions for MIV [37],
- MIV DSDE with Geometry Assistance SEI:
 - ❖ as in the state-of-the-art approach [28] adapted in MIV [41], features calculated for a block of the maximum size of 64, quantization step for features equal to 256,
- IDMA and eIDMA: two texture-depth QP value schemes:
 - ❖ standard: $\text{depth QP} = \max(1, [-14.2 + 0.8 \text{ tex QP}])$,
 - ❖ modified: $\text{depth QP} = \max(1, 0.8 \text{ tex QP})$.

Modified depth maps QP values are introduced to test if highly compressed depth can be refined in the depth estimation, allowing more bitrate for textures. Standard depth QP values are calculated as in common test conditions [37].

Tested methods were also evaluated in terms of their computational complexity. These results are provided in the form of runtime ratio when compared to no assistance DSDE. Experiments were conducted on a set of PCs with 3rd generation

AMD Ryzen Threadrippers equipped with 128 GB of RAM.

TABLE III
LIST OF TEST SEQUENCES.

Sequence	Source	Type	Resolution	Views
ClassroomVideo	[42]	ERP	CG 4096 × 2048	15
Chess	[43]	ERP	CG 2048 × 2048	10
Hijack	[44]	ERP	CG 4096 × 2048	10
Museum	[44]	ERP	CG 2048 × 2048	24
Group	[45]	Perspective, convergent	CG 1920 × 1080	21
Fencing	[46]	Perspective, convergent	NC 1920 × 1080	10
Fan	[47]	Perspective, planar	CG 1920 × 1080	15
Kitchen	[48]	Perspective, planar	CG 1920 × 1080	25
Mirror	[49]	Perspective, planar	NC 1920 × 1080	15
Carpark	[50]	Perspective, planar	NC 1920 × 1088	9
Frog	[51]	Perspective, planar	NC 1920 × 1080	13
Hall	[50]	Perspective, planar	NC 1920 × 1088	9
Street	[50]	Perspective, planar	NC 1920 × 1088	9
Painter	[52]	Perspective, planar	NC 2048 × 1088	16

ERP – Equirectangular Projection, CG – Computer-Generated, NC – Natural Content

C. Results

The results of performed experiments are shown in Fig. 12 in the form of plots showing the average IV-PSNR of rendered views obtained using tested compression methods for different rate points (bitrates averaged for all tested sequences). For low bitrate, the results for basic DSDE (without assistance) confirmed its high quality, presented earlier in other works [5], [6]. Nevertheless, for high bitrates, the final quality of DSDE is one of the worst of the tested methods.

The use of encoder-derived features [28], the first method to utilize information from depth maps available in the encoder to improve DSDE, increased the quality for medium and high bitrates when compared to basic DSDE. On the other hand, for low bitrates, the size of metadata required to send features (more than 20% of the whole bitstream – Table IV) negatively influences the quality of encoded textures, decreasing the final quality of rendered virtual views.

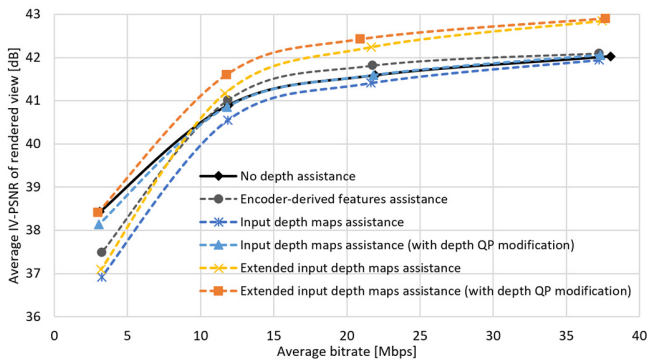


Fig. 12. Average IV-PSNR of rendered views achieved by tested methods; shown bitrates include all texture, depth, and metadata.

Input depth map assistance was shown to be the worst, however, when the modified depth QP is used, the quality increases for all bitrates. For medium and high bitrates, the average quality is comparable to basic DSDE, but, as presented in Table V, the runtime reduction for the decoder is significant. A slight increase in runtime for the encoder is also observed, but the complexity of atlas encoding in DSDE is negligible when compared to video encoding.

TABLE IV
DISTRIBUTION OF BITRATE USED FOR TEXTURE, DEPTH, AND METADATA.

No assistance [6]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	38.008	0.000	0.009	38.017	99.98%	0.00%	0.02%	
RP2	21.915	0.000	0.009	21.924	99.96%	0.00%	0.04%	
RP3	11.923	0.000	0.009	11.931	99.93%	0.00%	0.07%	
RP4	3.105	0.000	0.009	3.114	99.72%	0.00%	0.28%	
Encoder-derived features assistance [28]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	36.653	0.000	0.689	37.342	98.15%	0.00%	1.85%	
RP2	21.075	0.000	0.689	21.764	96.83%	0.00%	3.17%	
RP3	11.112	0.000	0.689	11.801	94.16%	0.00%	5.84%	
RP4	2.521	0.000	0.689	3.210	78.54%	0.00%	21.46%	
Input depth maps assistance [28]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	32.500	4.735	0.008	37.243	87.26%	12.71%	0.02%	
RP2	17.667	3.977	0.008	21.653	81.59%	18.37%	0.04%	
RP3	9.095	2.749	0.008	11.853	76.74%	23.19%	0.07%	
RP4	1.993	1.236	0.008	3.237	61.56%	38.18%	0.26%	
Input depth maps assistance with depth QP modification [28]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	35.686	1.635	0.008	37.329	95.60%	4.38%	0.02%	
RP2	20.487	1.326	0.008	21.821	93.89%	6.08%	0.04%	
RP3	10.777	1.017	0.008	11.803	91.31%	8.62%	0.07%	
RP4	2.655	0.369	0.008	3.032	87.56%	12.16%	0.28%	
Extended input depth maps assistance [28]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	30.543	6.877	0.018	37.437	81.58%	18.37%	0.05%	
RP2	16.370	5.277	0.018	21.665	75.56%	24.36%	0.08%	
RP3	8.136	3.494	0.018	11.648	69.85%	30.00%	0.15%	
RP4	1.668	1.514	0.018	3.199	52.13%	47.31%	0.55%	
Extended input depth maps assistance with depth QP modification [28]								
Rate point	Bitrate [Mbps]				Fraction [%]			
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata	
RP1	35.399	2.258	0.018	37.675	93.96%	5.99%	0.05%	
RP2	19.160	1.731	0.018	20.909	91.64%	8.28%	0.08%	
RP3	10.494	1.263	0.018	11.775	89.12%	10.73%	0.15%	
RP4	2.496	0.475	0.018	2.988	83.52%	15.89%	0.59%	

It is important to note that the software used in this study, including the IVDE depth estimator, TMIV and VVenC, were not optimized for low computational complexity but are the current implementations used for academic and standardization purposes. The real-time implementations of required processes can already be found, e.g., for MIV bitstream decoding and computationally expensive virtual view rendering [53], which in the presented experiments took about 20 seconds per view.

Extended assistance provides better quality than previously discussed methods for medium and high bitrates and has the fastest decoding and rendering time. When depth QP is modified, the method shows the best results among all tested configurations for all rate points, showing the high performance of depth refinement performed using modified depth estimator.

When analyzing the virtual view quality in Fig. 12 together with the content of the encoded bitstream in Table IV, it can be seen that for the best methods at each rate point (eIDMA with depth QP modification for all RPs and additionally eIDMA in RP1 and no depth assistance and IDMA with depth QP modification in RP4), the fraction of bitrate used for depth

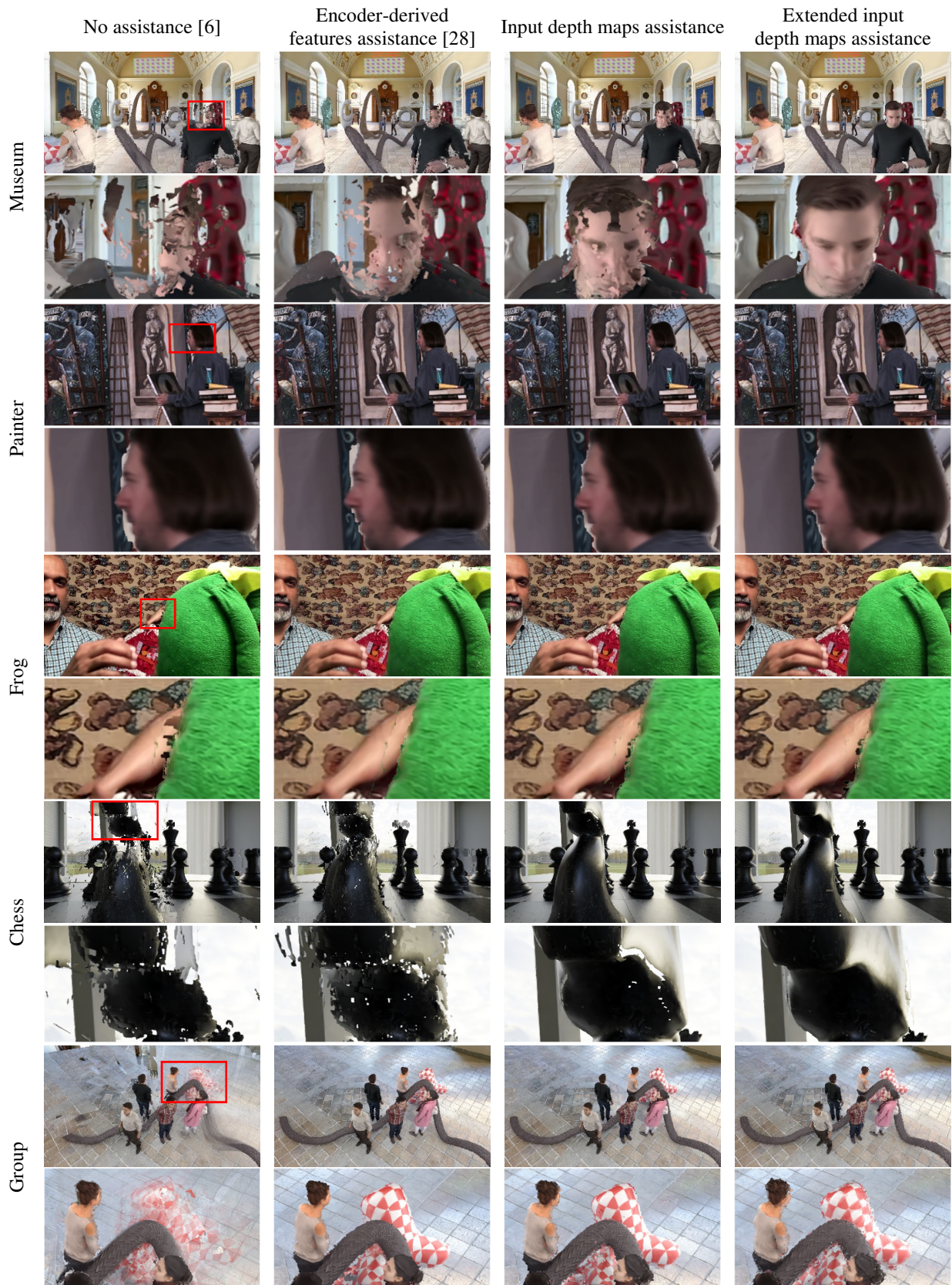


Fig. 13. The subjective comparison of tested methods for fragments of selected viewport synthesized between positions of input views.

information (in the form of encoded depth maps or features included in metadata) does not exceed 20%. It ensures a decent quality of textures, enabling estimation of better depth maps and, in the end, higher quality of synthesized views, as this step is influenced both by better textures and depth maps. When no depth information is being sent, the quality of textures is very high, however, possible significant errors in depth maps (resulting from occlusions, non-Lambertian surfaces) affect the final quality of virtual views and prevent the decoder from providing high quality, especially for high bitrates.

It can also be observed in the provided visual comparison of virtual views obtained for tested methods, presented in Fig. 13. While the quality for most of the areas in virtual views is similar for different methods, the enlarged fragments show fragments of views for which the erroneously-estimated depth maps significantly decreased the observed quality in basic DSDE. Providing encoder-derived features [28] or the proposed input depth map assistance considerably improves the reconstruction quality in these areas.

As presented in Fig. 13, in most of the cases, proposed IDMA and eIDMA approaches also outperform the encoder-derived features. The only test sequence for which the proposed method performed worse than [28] was Group. This computer-generated sequence has a relatively low number of objects, and most fragments in depth maps are flat. Therefore, ranges of possible depth levels are very similar for neighboring blocks, making it easy to compress these depth map features in this coding scheme, increasing the final quality of rendered views. However, the subjective quality gain over the proposal is negligible when compared to the poor quality of virtual views rendered using the basic DSDE approach. Moreover, the decoding process in both IDMA and eIDMA methods is significantly faster than using the encoder-derived features.

TABLE V

THE RUNTIME OF TESTED METHODS; SHOWN PERCENTAGES PROVIDE RUNTIME RATIO COMPARED TO DSDE WITHOUT ASSISTANCE.

Type of assistance	Average with standard deviation of runtime per one frame [s]			
	Encoding			Decoding & rendering of a view
	Atlas encoding	Video encoding	Overall	
No assistance [6]	1.5±0.3	228.1±104.9	229.6±104.8	152.3±62.2
Features [28]	1.9±0.6 (126%)	228.1±104.9 (100%)	231.9±105.1 (101%)	83.8±33.7 (55%)
IDMA	7.7±6.9 (518%)	216.7±101.4 (95%)	227.3±99.3 (99%)	73.1±25.5 (48%)
IDMA + QP mod.	7.7±6.9 (518%)	235.0±106.1 (103%)	245.7±103.0 (107%)	74.6±27.4 (49%)
eIDMA	66.0±60.2 (4481%)	191.6±68.1 (84%)	261.7±106.1 (114%)	48.7±12.8 (32%)
eIDMA + QP mod.	66.0±60.2 (4481%)	212.1±100.6 (93%)	277.8±120.2 (121%)	48.7±13.4 (32%)

When the objective quality results are shown independently for CG and natural content (Fig. 14 and Fig. 15, respectively), it can be seen that the quality of depth maps available in the encoder has a considerable influence on the differences between efficiencies of tested methods. In the case of CG sequences, for which the ground-truth depth maps are available,

it is beneficial to use the eIDMA approach, as it provides the best quality, independently of used depth QP values. However, for natural content, for which depth maps available in the encoder are estimated basing on the acquired views, the best quality is observed for the IDMA approach. In this case, even if estimated using the best methods with fine-tuned parameters, depth maps cannot be considered a ground-truth. Therefore, providing only partial assistance for such sequences is more favorable, as possible errors in input depth maps will not be reprojected to all views, which could be the case in eIDMA.

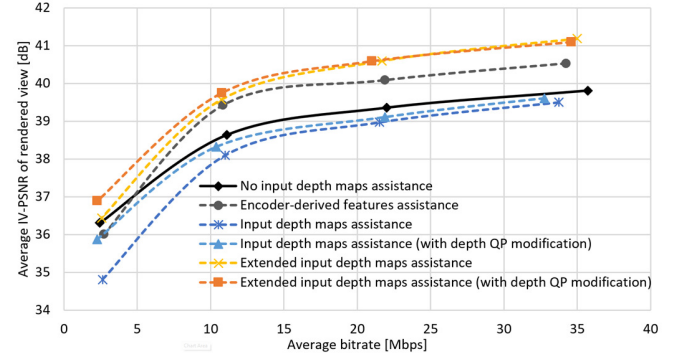


Fig. 14. Results for CG sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

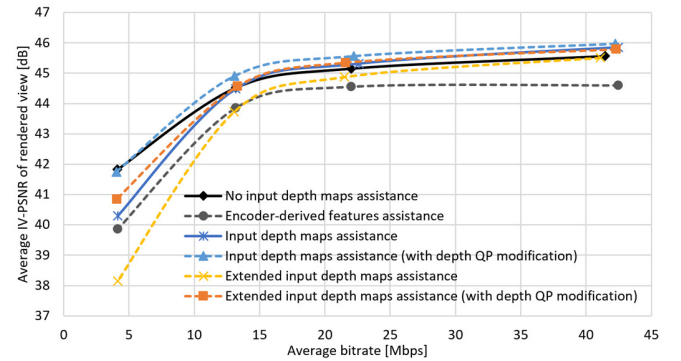


Fig. 15. Results for natural sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

Although extended assistance with modified depth QP seems to be the best compromise, the encoder was improved to automatically select the best method, depending on the currently encoded content (adaptive IDMA scheme, described in Section III.B.1).

Fig. 16 shows the performance of such a mixed method. As it can be seen, such a proposal of adaptive IDMA performs much better than other tested methods, merging two solutions into one robust solution for DSDE-based immersive video coding.

The final results are shown in Table VI. This table shows BD-rate (Bjontegaard delta [54]), i.e., the percentage change in the bitrate required to achieve the same quality for tested coding techniques in comparison with the reference method (no assistance DSDE). The results show a significant bitrate saving of over 50% for the final adaptive IDMA method. Combining this result with the faster decoding of the proposed methods shows the proposal's advantages over state-of-the-art DSDE.

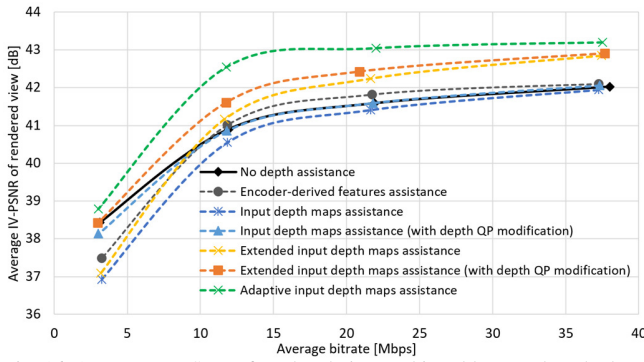


Fig. 16. Average IV-PSNR of rendered views achieved by tested methods and proposed adaptive input depth maps assistance for four tested bitrates; bitrates include all texture, depth, and metadata.

TABLE VI
BD-RATE OF TESTED METHODS VERSUS NO ASSISTANCE DSDE [6].

Type of assistance	BD-rate versus no assistance [6]
Encoder-derived features [28]	0.70%
IDMA	32.70%
IDMA + depth QP mod.	3.50%
eIDMA	-2.70%
eIDMA + depth QP mod.	-27.40%
Adaptive IDMA	-53.00%

V. CONCLUSIONS

The paper describes a novel method of increasing the efficiency of the DSDE approach in immersive video transmission. When compared to the state-of-the-art DSDE techniques, where the textures of views and camera parameters are transmitted to the decoder, which estimates the 3D scene representation from scratch, we have proposed to additionally send a subset of depth maps available at the encoder side. In the decoder, this additional geometry information assists the depth estimation process, increasing the overall efficiency.

The proposed approach significantly speeds up the decoding process when compared to the state-of-the-art DSDE approach [6], allowing for estimating depth maps even three times faster. Moreover, the proposed input depth map assistance achieves a better quality of final virtual views, as the depth estimator is guided by the original geometry transmitted to the decoder.

We proposed two ways of transmitting the input depth information within an MPEG immersive video (MIV) bitstream, meeting different requirements of the practical, immersive video systems. The advantage of the first proposal, where the depth information is sent for only a subset of transmitted input views, is a relatively fast decoding process and quality improvement over the state-of-the-art DSDE approach, visible especially for encoding of natural content. In the second, extended approach, additional depth patches for remaining views are being transmitted, exploiting pruning and packing algorithms of the MPEG immersive video encoder. Such an approach allows for an even faster decoding process and further improvement of the quality of viewports presented to the final user of the immersive video system.

Previous works have proven that the DSDE approach achieves satisfactory quality in compressed immersive video, especially for low-bitrate systems. However, the depth map

estimation process is time-consuming, and developing a real-time immersive video encoder working in the DSDE mode is challenging. Therefore, the authors believe that the proposed input depth map assistance approach is a step in the direction of the development of practical immersive video systems. Moreover, using the proposed approach, the decoder is able to reproduce a good-quality representation of the 3D scene even for challenging content, including non-Lambertian surfaces, numerous disocclusions, and areas with inestimable depth (e.g., for areas visible only in one transmitted view).

The proposal, besides changing the main principle of DSDE, that the depth maps are not available in the decoder, changes also the assumptions of immersive video coding, that there cannot be any depth in the transmitted bitstream that does not correspond to texture. Our scheme proposes more flexible encoding, e.g., by opening the possibility of using depth acquired from sensors such as Kinect [55], which are very popular among consumers [56]. This acquired depth can be easily used as an input depth map and improve the quality of depth for other views. Functionalities of the proposal enable many further improvements, e.g., much faster estimation of depth only for missing areas in reprojected depth maps, performed using deep-learning-based methods or by simple inpainting.

MPEG Video Coding experts appreciated the flexibility, usefulness, and novelty of proposed IDMA-based approaches, therefore, these encoding methods will comply with a new DSDE profile [57], [58] of the incoming second edition of the MIV standard [59], as agreed during the 140th MPEG meeting in October 2022.

REFERENCES

- [1] J.-B. Jeong, et al., "Towards 3DoF+ 360 Video Streaming System for Immersive Media," *IEEE Access*, vol. 7, pp. 136399-136408, 2019.
- [2] D. Mi et al., "Demonstrating Immersive Media Delivery on 5G Broadcast and Multicast Testing Networks," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 555-570, 2020.
- [3] M. Tanimoto et al., "FTV for 3-D Spatial Communication," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, April 2012.
- [4] A. Vetro et al., "3D-TV Content Storage and Transmission," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 384-394, 2011.
- [5] V.K.M. Vadakital et al., "The MPEG Immersive Video Standard—Current Status and Future Outlook," *IEEE MultiMedia*, vol. 29(3), 2022.
- [6] D. Mieloch et al., "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6360-6374, 2022.
- [7] Y.-L. Chan et al., "Overview of current development in depth map coding of 3D video and its future," *IET Signal Proc.*, vol. 14(1), pp. 1-14, 2020.
- [8] M. Wien et al., "Standardization Status of Immersive Video Coding," *IEEE J. Emer. and Sel. Topics in Circ. and Syst.*, vol. 9(1), pp. 5-17, 2019.
- [9] J. Samelak et al., "Advanced HEVC Screen Content Coding for MPEG Immersive Video," *Electronics*, vol. 11, no. 23, pp. 4040, 2022.
- [10] D. Mieloch et al., "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," *IEEE Access*, vol. 8, pp. 5760-5776, 2020.
- [11] D. Mieloch, A. Dziembowski, M. Domański, "Depth Map Refinement for Immersive Video," *IEEE Access*, vol. 9, pp. 10778-10788, 2021.
- [12] M. Ibrahim et al., "Depth map artefacts reduction: a review," *IET Image Processing*, vol. 14, no. 12, pp. 2630-2644, 2020.
- [13] P. Garus et al., "Bypassing Depth Maps Transmission for Immersive Video Coding," in *PCS 2019*, Ningbo, China, 2019, pp. 1-5.
- [14] A. Dziembowski et al., "The influence of a lossy compression on the quality of estimated depth maps," in *IWWSIP 2016 Conf.*, 2016, pp. 1-4.

- [15] P. Garus et al., "Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3250-3264, 2022.
- [16] G. J. Sullivan et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Tr. Circ. & Syst. for V. Tech.*, vol. 22(12), 2012.
- [17] A. Wiecek et al., "VVenc: An Open and Optimized VVC Encoder Implementation," in *ICMEW 2021 Conf.*, 2021, pp. 1-2.
- [18] G. Sullivan and T. Wiegand, "Video compression – from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, 2005.
- [19] M. Domański et al., "Universal Modeling of Monoscopic and Multiview Video Codecs with Applications to Encoder Control," in *2021 IEEE International Conference on Image Processing*, pp. 2144-2148, 2021.
- [20] 2021 TV Video Specifications. Accessed: Jan. 11, 2021. [Online]. Available: <https://developer.samsung.com/smarttv/develop/specifications/media-specifications/2021-tv-video-specifications.html>.
- [21] H.264/H.265 Video Codec Unit. Accessed: Jan. 11, 2021. [Online]. Available: xilinx.com/products/intellectual-property/v-vcu.html
- [22] G. Tech et al., "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35-49, Jan. 2016.
- [23] J. Boyce, et al., "MPEG Immersive Video coding standard," *Proceedings of the IEEE*, vol. 109, no. 9, p. 1521-1536, 2021.
- [24] "Text of ISO/IEC FDIS 23090-12 MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/N0111, Online, 2021.
- [25] M. Milovanović et al., "Patch Decoder-Side Depth Estimation in Mpeg Immersive Video," in *ICASSP 2021*, pp. 1945-1949, 2021.
- [26] S.L. Ravi et al., "A Study of Conventional and Learning-Based Depth Estimators for Immersive Video Transmission," in *IEEE Int. Workshop on Multimedia Signal Processing (MMSp)*, Shanghai, China, 2022.
- [27] B. Szydelko, et al., "Recursive block splitting in feature-driven decoder-side depth estimation," *ETRI Journal*, vol. 44, pp. 38– 50, 2022.
- [28] G. Clare et al., "Combination of m56626 and m56335 for Geometry Assistance SEI message," ISO/IEC JTC1/SC29/WG4/M56950, 2021.
- [29] P. Garus et al., "Motion Compensation-based Low-Complexity Decoder Side Depth Estimation for MPEG Immersive Video," in *IEEE Int. Workshop on Multimedia Signal Proc. (MMSp)*, Shanghai, China, 2022.
- [30] "Test Model 11 for MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0142, Online, October 2021.
- [31] Y. Boykov et al., "Fast approximate energy minimization via graph cuts," *IEEE Tr. on Pattern An. and Mach. Int.*, vol. 23(11), pp. 1222-1239, 2001.
- [32] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple noniterative clustering," in *CVPR 2017*, Honolulu, HI, USA, Jul. 2017.
- [33] A. Dziembowski et al., "Multiview synthesis – improved view synthesis for virtual navigation," in *32nd Picture Coding Symposium (PCS)*, Nürnberg, Germany, 2016.
- [34] F. Shao et al., "Depth Map Coding for View Synthesis Based on Distortion Analyses," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 106-117, March 2014.
- [35] A. Dziembowski et al., "IV-PSNR – the objective quality metric for immersive video applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, 2022.
- [36] "Report of MPEG immersive video CTC anchor generation," ISO/IEC JTC1/SC29/WG4 MPEG2022/N0234, Online, July 2022.
- [37] "Common Test Conditions for MPEG Immersive video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0143, Online, October 2021.
- [38] ISO/IEC 23008-2, Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding.
- [39] C. Colman-Meixner et al., "Deploying a Novel 5G-Enabled Architecture on City Infrastructure for Ultra-High Definition and Immersive Media Production and Broadcasting," *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 392-403, 2019.
- [40] "MPEG-I Visual / IVDE – Gitlab", available online, accessed 28.09.2022: <https://gitlab.com/mpeg-i-visual/ivde/-tree/v7.0>.
- [41] "Text of ISO/IEC DIS 23090-5 Visual Volumetric Video-based Coding and Video-based Point Cloud Compression 2nd Edition," ISO/IEC JTC1/SC29/WG7 MPEG2021/N00188, Online, July 2021.
- [42] B. Kroon, "3DoF+ Test Sequence ClassroomVideo," ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415, San Diego, April 2018.
- [43] L. Ilola et al., "New Test Content for Immersive Video – Nokia Chess," ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787, Geneva, Sep. 2020.
- [44] R. Doré, "Technicolor 3DoF+ Test Materials," ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349, San Diego, March 2018.
- [45] R. Doré, G. Briand, and F. Thudor, "InterdigitalGroup Content Proposal," ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731, Online, June 2020.
- [46] M. Domański et al., "Multiview Test Video Sequences for Free Navigation Exploration Obtained using Paris of Cameras," ISO/IEC JTC1/SC29/WG11 MPEG2018/M38247, Geneva, May 2016.
- [47] R. Doré, G. Briand, and F. Thudor, "InterdigitalFan Content Proposal for MIV," ISO/IEC JTC1/SC29/WG11 MPEG/M54732, Online, June 2020.
- [48] P. Boissonade and J. Jung "Proposition of New Sequences for Windowed-6DoF Experiments on Compression, Synthesis and Depth Estimation," ISO/IEC JTC1/SC29/WG11 MPEG2018/M43318, Ljubljana, July 2018.
- [49] R. Doré and G. Briand, "Interdigital Mirror Content Proposal for Advanced MIV Investigations on Reflection," ISO/IEC JTC1/SC29/WG11 MPEG2020/M55710, Online, January 2021.
- [50] D. Mieloch et al., "Natural Outdoor Test Sequences," ISO/IEC JTC1/SC29/WG11 MPEG2019/M51598, Brussels, January 2020.
- [51] B. Salahieh et al. "Kermit Test Sequence for Windowed 6DoF Activities," ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748, Ljubljana, July 2018.
- [52] D. Doyen et al., "Light Field Content from 16-camera Rig," ISO/IEC JTC1/SC29/WG11 MPEG2017/M40010, Geneva, January 2017.
- [53] M. Chen et al., "Simplified carriage of MPEG immersive video in HEVC bitstream," *Proc. SPIE*, vol. 11842, Aug. 2021, Art. no. 118420C
- [54] G. Bjoentegaard, "Calculation of average PSNR differences between RD-Curves," *ITU-T VCEG Meeting*, Austin, USA, 2001.
- [55] X. Ye et al., "Computational Multiview Imaging with Kinect," *IEEE Transactions on Broadcasting*, vol. 60, no. 3, pp. 540-554, 2014
- [56] A. Doumanoglou et al., "Quality of Experience for 3-D Immersive Media Streaming," *IEEE Tr. on Broadcasting*, vol. 64, no. 2, pp. 379-391, 2018.
- [57] A. Dziembowski et al., "MIV Decoder-Side Depth Estimation profile," ISO/IEC JTC1/SC29/WG4 MPEG VC M60667, Mainz, 10.2022.
- [58] D. Mieloch, A. Dziembowski, J. Y. Jeong and G. Lee, "On the future of decoder-side depth estimation in MPEG immersive video coding," in *2023 Data Compression Conference (DCC)*, Snowbird, UT, USA, 2023.
- [59] "Preliminary WD4 of ISO/IEC 23090-12 MPEG immersive video Ed. 2," ISO/IEC JTC1/SC29/WG4 MPEG VC N0269, Mainz, 10.2022.



Dawid Mieloch received his M.Sc. and Ph.D. from Poznań University of Technology in 2014 and 2018, respectively. Currently, he is an assistant professor at the Institute of Multimedia Telecommunications. He is actively involved in ISO/IEC MPEG activities, where he contributes to the development of immersive media technologies. He has been involved in several projects focused on multiview video. His professional interests also include depth estimation, and camera calibration.



Adrian Dziembowski was born in Poznań, Poland in 1990. He received the M.Sc. and Ph.D. degrees from the Poznan University of Technology in 2014 and 2018, respectively. Since 2019 he is an Assistant Professor at the Institute of Multimedia Telecommunications. He authored and coauthored over 40 articles on various aspects of immersive video, free navigation, and FTV systems. He is also actively involved in ISO/IEC MPEG activities towards MPEG immersive video coding standard.



Dominika Klóska was born in 1998. She received her B.Sc. and M.Sc. degrees from Poznań University of Technology in 2021 and 2022 respectively. Currently she works at the Institute of Multimedia Telecommunications where she is involved in ISO/IEC MPEG activities such as MPEG immersive video coding standard and the development of other immersive media technologies.



Błażej Szydełko received the master's degree in electronics and telecommunications from Poznan University of Technology (PUT), Poland in 2022. Currently, he is a research and teaching assistant at the Institute of Multimedia Telecommunications at PUT. His professional interests are related to immersive video technologies, especially multiple-camera system calibration and depth estimation.



Jun Young Jeong received his BS and MS degrees in electrical engineering in 2013 and 2016, respectively from Purdue University, West Lafayette, IN, USA. He has been a research staff in the Immersive Media Research Laboratory, ETRI, Rep. of Korea since 2016, and has primarily been involved in the development of camera systems for acquiring immersive 6DoF VR and depth estimation software by using stereo vision algorithms. His current research interests include image processing and computer vision, especially in the field of deep-learning based depth estimation.



Gwangsoon Lee received his PhD degree in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2004. He joined the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, in 2001. He is currently a Principal Researcher with Realistic-Media Research Section. His research interests include immersive video processing, light field imaging system, and three-dimensional video system.

Article

On the Selection of Transmitted Views for Decoder-Side Depth Estimation

Dominika Klóska , Adrian Dziembowski * , Adam Grzelka  and Dawid Mieloch 

Institute of Multimedia Telecommunications, Poznan University of Technology, 60-965 Poznań, Poland; dominika.kloska@put.poznan.pl (D.K.); adam.grzelka@put.poznan.pl (A.G.); dawid.mieloch@put.poznan.pl (D.M.)

* Correspondence: adrian.dziembowski@put.poznan.pl

Abstract

The selection of optimal views for transmission is critical for the coding efficiency of the MPEG Immersive Video (MIV) profile of Decoder-Side Depth Estimation (DSDE). Standard approaches, which favor a uniform camera distribution, often fail in scenes with complex geometry, leading to decreased quality of depth estimation, and thus, reduced quality of virtual views presented to a viewer. This paper proposes an adaptive view selection method that analyzes the scene's percentage of occluded regions. Based on this analysis, the encoder dynamically selects a transmission strategy: for scenes with a low occlusion ratio (smaller than 10%), a uniform layout is maintained to maximize spatial coverage; for scenes with a high occlusion ratio, the system switches to grouping cameras into stereo pairs, which are more robust for decreasing numbers of occlusions. Experiments conducted using the TMIV reference software demonstrated that this approach yields measurable quality gains (up to 2 dB BD-IVPSNR) for complex test sequences, such as MartialArts and Frog, without requiring any modifications to the decoder.

Keywords: immersive video; decoder-side depth estimation; video compression; depth maps

1. Introduction

Immersive video technologies [1] are continuously advancing, driven by the increasing demand for realistic and interactive multimedia experiences. These technologies implement various degrees of freedom, e.g., 3DoF, 3DoF+, and 6DoF [2,3]. The process of generating immersive video (Figure 1) requires using a multicamera system to register a 3D scene. Recent developments in immersive video have explored multiple configurations of multicamera systems, including linear, planar, and spherical camera arrangements.

The other component crucial for creating immersive video is the three-dimensional geometry of the registered scene, which can be obtained either from depth cameras or estimated using dedicated software. Since using depth cameras introduces a problem of interference between sensors [4], for the considerations presented in this paper, we will be assuming that depth information is obtained through the process of depth estimation.

Practical systems of immersive video [5] can enhance multicamera setups with a virtual camera placed among the real cameras, enabling dynamic and adaptive view synthesis to reflect the movement of the user in a 3D scene. In other words, a virtual camera is a viewport rendered by the system at the request of the viewer [6]. To represent all of this data, the multiview video plus depth (MVD [7]) format is usually used, as shown in Figure 2. It allows separate encoding of real views and their corresponding



Academic Editor: JungHwan Oh

Received: 21 November 2025

Revised: 12 December 2025

Accepted: 19 December 2025

Published: 20 December 2025

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

depth maps using any available video encoder (HEVC, VVC [8,9]). Because traditional encoders were not developed for data such as depth maps (as they do not resemble naturally captured videos), alternative solutions have been explored (MV-HEVC, 3D-HEVC [10]). Unfortunately, experimental data show that their versatility is highly limited [11].

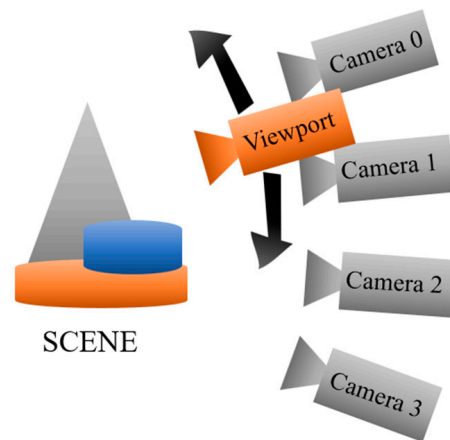


Figure 1. Scene acquisition in the immersive video system.

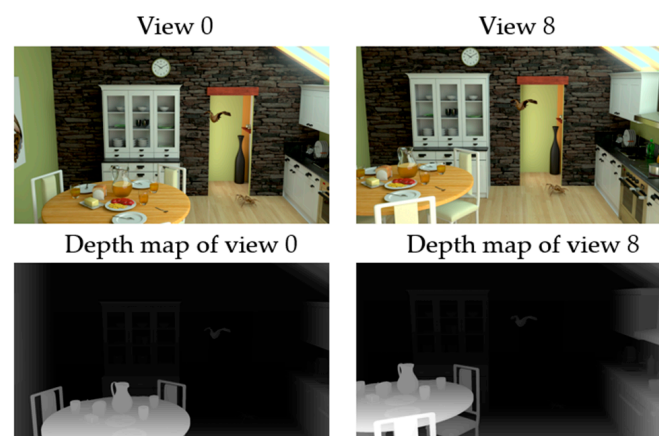


Figure 2. Multiview video plus depth format.

The latest approach to immersive video compression is the MPEG immersive video (MIV) standard [12], created by the ISO/IEC MPEG Video Coding. MIV utilizes multiview pre-processing and post-processing (Figure 3), making the resulting videos more efficient to compress with traditional video encoders. Additionally, MIV supports a range of profiles dedicated to different coding scenarios.

The MIV Main profile groups input views into a set of videos called atlases (usually four), which are shown in Figure 4. Each atlas is encoded independently. Input views, which are views captured by the multicamera system, are divided into basic and additional views. Basic views contain the most information about a scene and are usually packed into the first atlas. Additional views contain a large amount of redundant information, which is removed by the pruning process. This process creates small fragments (called patches) of the remaining view information. Patches are packed into the remaining space left in the first two atlases. The third and fourth atlases contain depth information corresponding to the first two atlases.

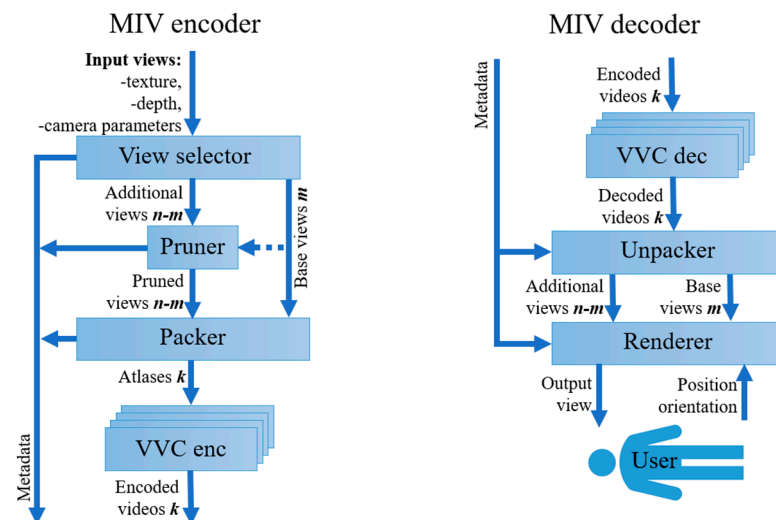


Figure 3. Simplified MIV scheme.

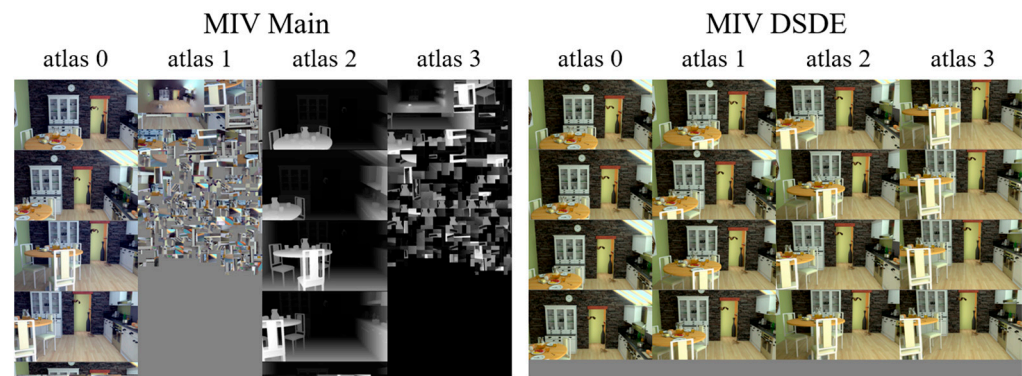


Figure 4. MIV Main and MIV DSDE atlases.

Another approach is called the MIV decoder-side depth estimation (DSDE) coding profile. Similarly to the MIV Main, this profile also utilizes atlases (Figure 4) consisting of multiple input views per atlas, but in this approach the atlases do not contain depth information. As depth is required to perform virtual view synthesis, it is estimated from the decoded views on the decoder side. In this scenario, the transmitted views should be chosen in such a way as to minimize mutual redundancy while maximizing coverage of scene content. The view selection process for the DSDE profile is further complicated by an intrinsic trade-off: increasing the number of transmitted views improves the accuracy of the depth estimates and, consequently, the quality of synthesized views, whereas a poorly chosen small subset can cause the depth estimator to fail, particularly in regions affected by occlusions. Hence, it is unclear whether the view selection method used for the MIV Main profile will remain effective for the DSDE profile, or whether selection criteria tailored to decoder-side depth estimation are required.

This paper addresses the challenge of optimal input view selection for the MIV DSDE profile, proposing a method that is adjusted to scene characteristics. We analyze how to ensure the highest possible depth-estimation quality when only a limited number of views can be transmitted, while simultaneously selecting them in a way that mitigates the negative impact of scene occlusions on the estimation process. Our proposed adaptive approach ensures the highest coding efficiency and visual quality across diverse immersive video content.

In Section 2, we review the state of the art in view selection, outline the issues that may arise during the view selection process, and provide a detailed analysis of how view

selection operates in the MIV Main profile. In Section 3, we describe our proposed view selection method for the DSDE profile, which explicitly accounts for occlusions present in the scene. Section 4 presents an overview of the experiments, and Section 5 reports and analyzes the experimental results. Section 6 contains conclusions and future work.

2. View Selection for Immersive Video

View selection for immersive video is a complex process that requires careful consideration of the specific application scenario. A variety of factors influence the choice of input views in the DSDE scenario: the selected cameras determine not only the accuracy of estimated depth maps but also the quality of the synthesized virtual views [13], and therefore, ultimately, the overall quality of the immersive experience. Key challenges to consider include scene occlusions, finite image resolution, non-Lambertian surfaces, camera layout and type (e.g., linear, omnidirectional), and the decoding hardware's computational limits. In the following sections, we survey several view selection strategies designed to address these diverse problems and use cases.

2.1. View Selection Optimized for Virtual View Synthesis

This section describes the problem of selecting the best input views for the virtual view synthesis process in free viewpoint television (FTV) systems. As such systems require real-time synthesis, it is essential to limit the number of input views in order to maintain reasonable computational time. Adding more views to view synthesis [14–16] increases the quality of virtual views; nevertheless, it also increases computational time [6]. Therefore, an effective view selection method should aim to balance the quality of synthesized virtual views with the need to minimize computational complexity.

In the simplest implementations, a virtual view is synthesized from two manually selected real views [17]. The method described in [13] deals with the problem of choosing two views that will guarantee the highest quality of the synthesized virtual view. To achieve this, three main view synthesis challenges were taken into account:

1. Occlusions: Gaps occur where real views do not overlap; choosing cameras closest to the virtual viewpoint minimizes these holes.
2. Finite resolution: Objects appear at different scales across views; projecting from the views where each object is largest preserves geometric continuity.
3. Non-Lambertian reflectance: Surface brightness varies with angle; using the two cameras nearest the virtual position ensures more consistent lighting.

Through addressing the above challenges, it was shown that the best quality of the synthesized virtual view can be obtained when using two neighboring real views (nearest left and right), and this is true for any camera arrangement. However, this research considers view selection for virtual view synthesis in a scenario where all of the real views are available. In the scenario analyzed for the purpose of this article, we have only a subset of real views available at the decoder-side for depth estimation and virtual view synthesis. Consequently, it is not always possible to select the nearest-right and nearest-left views, and it is therefore necessary to investigate which selection strategy can provide a quality level closest to the scenario described in [13].

2.2. View Selection Optimized for Transmission

This section presents the problem of optimal selection of input views transmitted within the video bitstream [18] in immersive video systems for the MIV Main profile. In this scenario, only a subset of views available at the encoder side can be transmitted to the decoder. Selecting proper real views for transmission is crucial to achieving good quality on the decoder side. The method described in the previous section is unsuitable for

this scenario, as it would require knowing which view was selected by the user prior to transmission, which is not possible.

The most straightforward approach to this scenario is using multiview simulcast coding. Unfortunately, it results in high bitrate and pixelrate [12], making this approach not valuable for a practical immersive video system.

Another approach is to extend the method described in the previous section. View selection had to be performed in such a way that would guarantee the highest average quality of the synthesized views independently of the user's viewpoint. For this purpose, Ref. [18] presents the results of a simulation of a simple practical immersive video system with the assumption of a reasonable pixelrate [12] and number of cameras [19]. The results showed that the best quality in the transmission scenario is achieved when cameras are distributed evenly; however, for omnidirectional content, there is a necessity to send input views from the horizontal axis rather than the vertical. The authors proved this through subjective tests. As noted in Section 1, the MIV DSDE profile differs materially from the MIV Main profile. While [18] describes a view selection strategy tailored to MIV Main, we contend that the DSDE scenario requires a different approach; our proposed method is presented in Section 3.

2.3. Impact of Camera Arrangement on Depth Estimation

The methods of view selection described in the previous sections did not consider one crucial process used in the creation of immersive video: depth estimation. When cameras are too far apart, fewer pixels are captured by at least two views, which negatively influences depth estimation, since any scene point must be visible in two or more images to have its depth reliably calculated, whereas occluded regions can only be interpolated or extrapolated. Moreover, larger baselines amplify lighting and reflectance discrepancies across views, complicating depth matching between views and degrading both depth maps and synthesized views.

There are other efficient camera setups [20–24], but these techniques require more input information, e.g., geometry of objects, which cannot be predicted in practical immersive video systems.

Stereopair grouping has been proposed to address these issues without increasing the number of cameras [25]. By organizing cameras into closely spaced pairs, each pair shares nearly identical viewpoints and lighting conditions, minimizing intra-pair occlusions and ensuring most scene points are visible to at least two cameras. However, the short baseline intrinsic to each pair limits depth precision. The solution is a hierarchical approach: Use long-baseline pairs (drawn from different stereo pairs) for precise depth estimation where possible, and rely on dense, short-baseline pairs to fill in occluded or poorly observed regions.

To present experimental results, the authors of [25] decided to use the PSNR metric because of its simplicity [26,27]. In the results, PSNR gains are split into baseline-adjustment and occlusion reduction. While long baselines improve depth, uniform layouts encounter a problem of a large number of occlusions in complex scenes [28,29], which causes horizontal displacements in virtual views. In typical two-step view-synthesis pipelines [30,31], occluded regions are inpainted and have lower quality. Empirical results indicated that when occlusions exceed roughly 20–25% of the scene, stereo-pair arrangements dramatically outperform uniform layouts, striking the optimal balance between depth accuracy and coverage in challenging immersive-video scenarios.

The experiments in [25] were conducted using the DERS [29] depth estimation algorithm, which is now outdated. In the present work, we employ IVDE [32], the current reference software for depth estimation, and perform experiments both on the se-

quences used in [25] and on new test sequences created specifically for research on immersive video [33].

3. The Proposal—Transmitted View Selection for DSDE

In immersive video systems with decoder-side depth estimation (DSDE), only a subset of available input views can be transmitted within the bitstream due to bitrate and pixelrate constraints [12]. Usually, a uniform selection of views is used [34] for limiting the number of transmitted views. However, such a strategy does not guarantee the highest quality on the decoder side.

The goal of the proposed method is to maximize the quality of synthesized virtual views while preserving the number of transmitted views and the overall bitrate and pixelrate.

The method relies on the analysis of occluded regions across input views. The occlusions are detected by a reprojection of 3D points between views, identifying pixels that are visible in only one view (thus, the pixels for which the depth is not estimable). In the proposed approach, the occlusion detection is implemented using the reprojection module already available in the TMIV v16.0 encoder [35]. At first, a set number of evenly spaced input views is set (in this paper, this number equals four), and each of them is reprojected to the remaining ones. For each pixel, the reprojected depth is compared with the original depth in the target view. A pixel is considered “visible” in the target view if the reprojected depth does not exceed the original depth (i.e., it is not occluded). The percentage of pixels visible in fewer than two views is accumulated. This provides a direct estimation of the occlusions ratio, i.e., the proportion of regions where depth cannot be reliably estimated.

To avoid fluctuations of the selected views within a single group of pictures (GOP)—which would significantly decrease video compression efficiency—the analysis is performed once per GOP (i.e., on the first frame of the GOP) and the resulting view selection remains fixed for the entire GOP. Because the underlying TMIV module is already highly optimized, the computation introduces only a negligible overhead. A detailed analysis of the processing time is provided in Section 5.3.

If a sequence contains a high ratio of occlusions (inestimable areas), the algorithm selects multiple camera stereopairs instead of uniformly arranged single cameras (see Figure 5). As proven in [18], such an approach reduces the number of non-estimated regions, thus increasing the overall quality of synthesized views. On the other hand, when the quantity of occlusions is lower, uniformly distributed views are chosen to maximize coverage and spatial diversity.

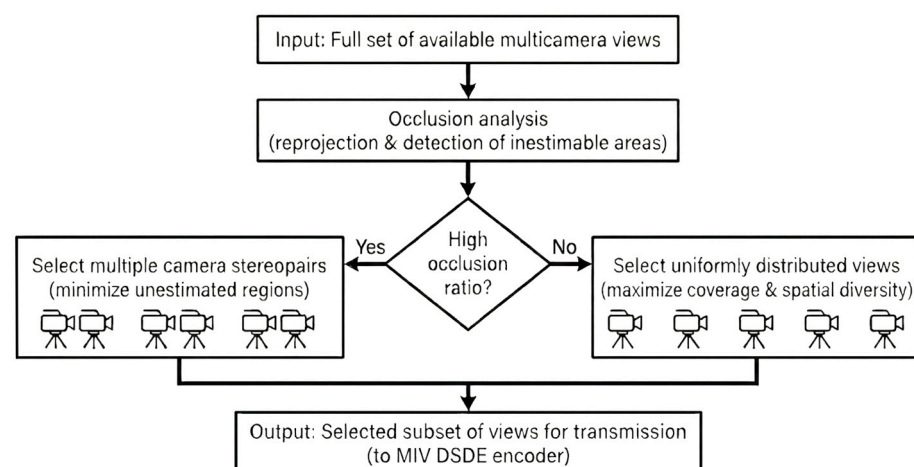


Figure 5. Scheme of the proposal.

The proposed approach is straightforward to implement within an MIV encoder (e.g., in the Test Model for MPEG immersive video reference software, TMIV [35]) and does not require any changes on the decoder side.

Moreover, by selecting views captured by stereopairs instead of evenly distributed cameras, the method can also reduce the total bitrate of an immersive video, as similar neighboring views packed into a single atlas may be compressed more efficiently (especially when screen-content coding tools are used by a video encoder [34]).

4. Overview of Experiments

To comprehensively evaluate the effectiveness of the proposed view selection method, two complementary experiments were conducted:

1. “MIV experiment”—evaluation using modern immersive video content.
2. “Supplementary experiment”—evaluation using legacy, classical FTV multiview sequences.

Both experiments employed exactly the same processing pipeline (defined in MIV common test conditions, MIV CTC [33]), using the same software:

- TMIV reference software [35] for creating atlases, view selection, and view synthesis on the decoder side.
- VVenC + VvdeC [36] for VVC [9] atlas encoding and decoding.
- IVDE reference software [32] for decoder-side depth estimation.

All the parameter settings except one were aligned with MIV CTC. The only intentional change—applied consistently in both experiments—was restricting the system to transmit only four views in total (all packed into a single atlas). This design choice reflects the goal of evaluating the influence of camera pairing and spacing in the most controlled and interpretable setting. By focusing on a minimal, one-dimensional camera layout (cameras placed along a line or an arc), the experiment isolates the effect of horizontal baselines on video coding, depth estimation, and synthesized view quality, without confounding factors introduced by multidimensional camera rigs or additional transmitted views.

The four-view configuration also represents the simplest non-trivial case in which different pairing strategies meaningfully affect view synthesis. More complex scenarios (e.g., multi-atlas setups or two-dimensional camera arrangements) are natural extensions of this study, but addressing them would require analyzing several factors at once. Our intention was to start with a clean, analyzable scenario, establishing conclusions that can later be generalized to higher-dimensional arrangements and a higher number of transmitted views.

Both experiments differ solely in the test sequences, allowing for assessing whether the proposed methodology is consistent across datasets with different capture characteristics, resolutions, and scene geometries.

4.1. MIV Experiment

The main experiment was conducted on the modern immersive video content from the MIV CTC [33]. The MIV CTC test set contains 21 miscellaneous multiview sequences. However, only six of them satisfy the requirements of this study, i.e., multicamera setups with cameras arranged approximately along a line or along an arc. Such setups are essential for analyzing how different horizontal baselines influence immersive video processing in the DSDE scenario.

To systematically evaluate the influence of camera spacing, a uniformity coefficient U was introduced:

$$U = \frac{b}{d}$$

where b is the baseline of each stereopair, and d is the distance between two stereopairs (i.e., the distance between two middle cameras within the selected four), c.f. Figure 6.

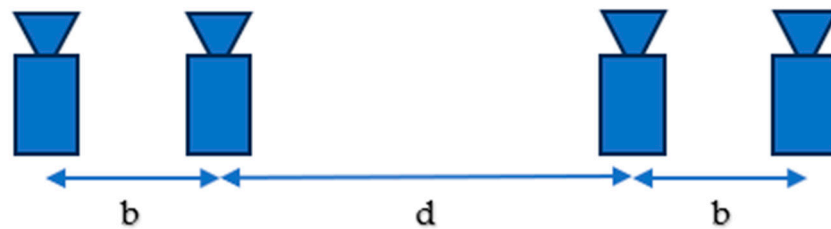


Figure 6. Analyzed camera arrangement; b —stereopair baseline (the same for both stereopairs), d —distance between two stereopairs; uniformity coefficient describes the ratio between b and d .

This parameter quantifies the uniformness of camera placement across the camera setup, where a coefficient equal to 1 corresponds to a perfectly uniform distribution (with equal distances between all neighboring cameras), and smaller values indicate non-uniform layouts with two stereopairs of cameras. A uniformity coefficient greater than 1 corresponds to a situation where the distance between two camera pairs is smaller than the baseline of each stereopair (i.e., the arrangement with a stereopair in the middle and two single cameras at each side).

The evaluation was based on the IV-PSNR metric, which measures the fidelity of the synthesized virtual views compared to reference ones, taking into account typical immersive video characteristics [37]. To assess the rate-distortion performance, BD-IVPSNR values were computed. The authors chose BD-IVPSNR instead of typical BD-rates because the RD-curves for different configurations did not always overlap. Therefore, the BD-IVPSNR metric ensured consistent comparison of the results.

Moreover, for each test sequence, occlusion maps were also generated in order to assess the proportion of scene areas not visible in multiple input views. This information was used to analyze how camera arrangement affects depth estimation and view synthesis quality for different levels of scene complexity.

The obtained results and their interpretation are discussed in Section 5, where the relationship between occlusion percentage, camera arrangement, and the quality of synthesized virtual views is analyzed in detail.

4.2. Supplementary Experiment

To validate the generality of the conclusions beyond modern immersive datasets, a supplementary experiment was performed using several classical FTV sequences: five BigBuckBunny sequences [38] (BBB Butterfly Arc, BBB Flowers Arc, BBB Rabbit Arc, BBB Butterfly Linear, and BBB Rabbit Linear), Bee [39], and three sequences from Nagoya University [40] (Champagne, Dog, and Pantomime). Although these sequences are of lower resolution and are no longer used in standardization activities, they provide diverse content characteristics and dozens of input views, allowing for analysis of multiple uniformity coefficients.

Moreover, the choice of sequences creates the possibility to partially compare the results of this research with the results presented in [25], where the authors analyzed the influence of camera pairing on depth estimation and view synthesis quality, but without the use of any video compression.

Crucially, the same experimental pipeline was used: TMIV + IVDE + VVenC/VVdeC, following MIV CTC guidelines and the same four-view, single-atlas constraint. This allowed a direct comparison of trends observed across datasets.

5. Experimental Results

5.1. MIV Experiment

Figure 7 presents a quality increase caused by camera pairing in comparison with uniform view distribution. Gain in quality is clearly visible for CBABasketball and MartialArts sequences, as well as the Frog and Fencing sequences, indicating that the camera pairing approach provides a noticeable advantage in scenes with complex geometry and frequent occlusions. The percentage of occlusions in each test sequence is presented in Table 1.

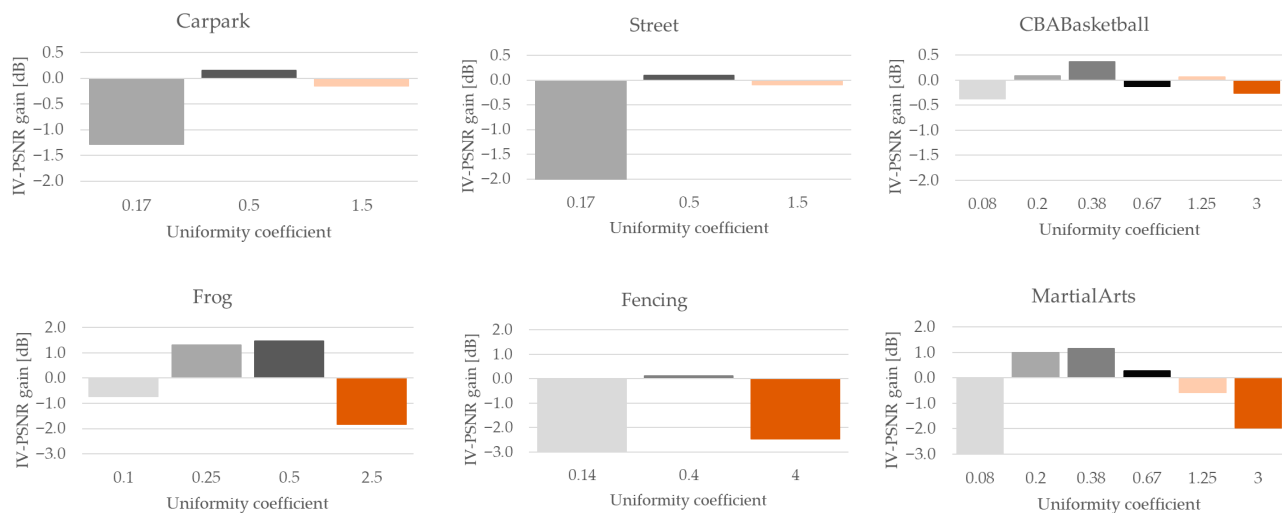


Figure 7. IV-PSNR of virtual views synthesized based on uncompressed atlases—quality increase compared to uniform view distribution (uniformity coefficient = 1); grey bars: two pairs of cameras (uniformity coefficient < 1), red bars: a pair of cameras in the middle and two single cameras at each side (uniformity coefficient > 1).

Table 1. Percentage of occluded areas in test sequences.

Sequence	Percentage of Occluded Areas
Frog	13.47%
Carpark	6.13%
Street	2.51%
Fencing	11.54%
CBABasketball	11.29%
MartialArts	21.47%

The existence of the relationship between the arrangement of selected real views and the amount of occlusions in the scene is further confirmed by the rate–distortion curves presented in Figure 8. For the sequences with more occlusions, the proposed view selection strategy consistently achieves higher IV-PSNR for a given bitrate.

Because the RD-curves for different configurations did not overlap, BD-rates could not be computed; therefore, in order to assess the average quality improvement across bitrates, the BD-IVPSNR values were used (Figure 9). Figure 9 summarizes the BD-IVPSNR gains as a function of the uniformity coefficient. The results confirm that pairing of the cameras (with a reasonable baseline—uniformity coefficient in the range [0.2, 0.7]) leads to a measurable quality increase for most of the tested sequences, especially for those with more occlusions.

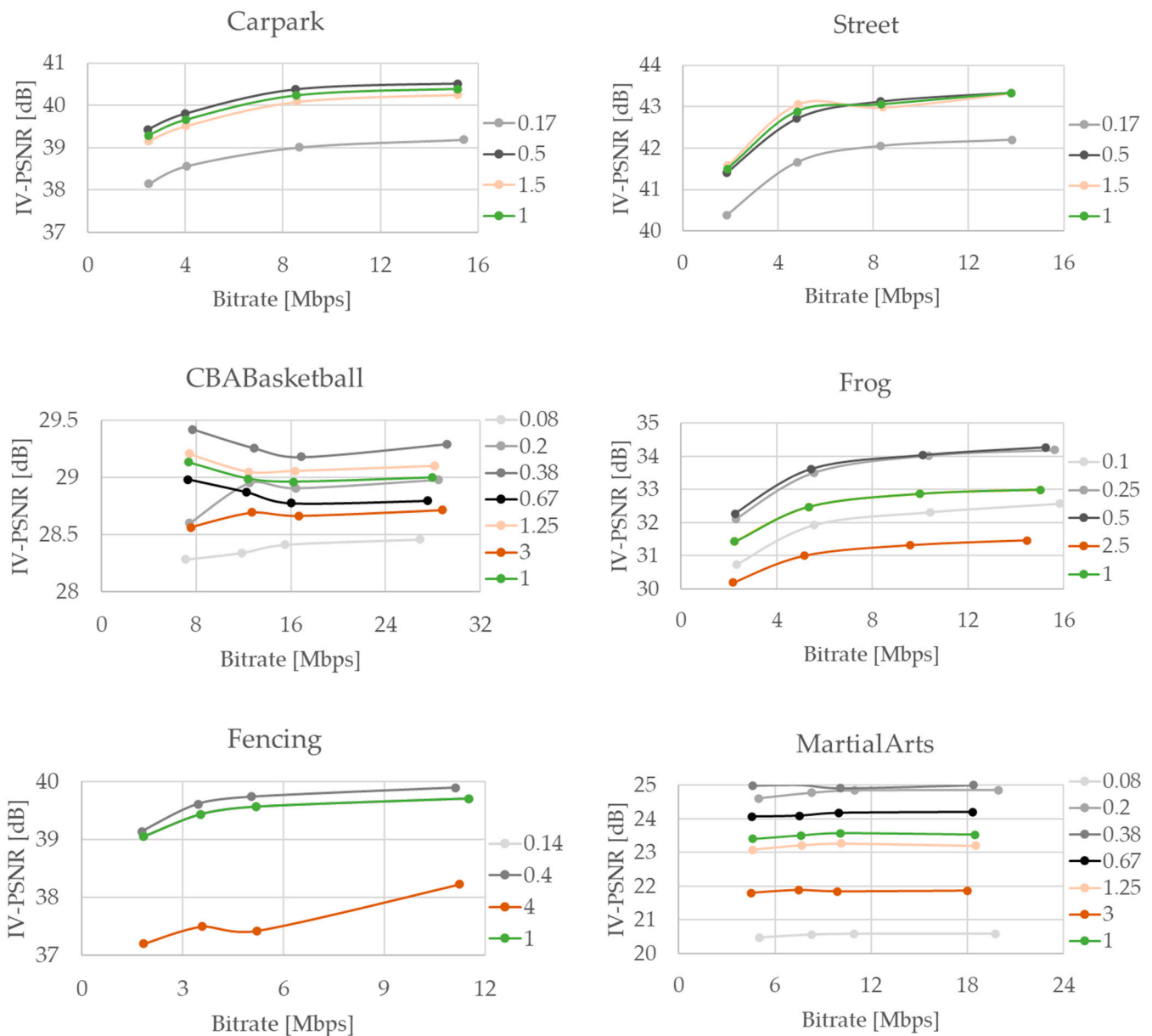


Figure 8. RD-curves of the immersive video system compared to uniform view distribution (uniformity coefficient = 1, green line); grey lines: two pairs of cameras (uniformity coefficient < 1), red lines: a pair of cameras in the middle and two single cameras at each side (uniformity coefficient > 1).

For scenes with very limited occlusions (e.g., Carpark, Street), both the IV-PSNR gains and BD-IVPSNR gains caused by camera pairing remain negligible. In such scenes, all the cameras observe nearly the same content, and an area with non-estimable depth is small for all camera arrangements. As a result, camera pairing does not introduce additional geometric cues that would noticeably improve depth estimation or view synthesis. On the other hand, uniform camera distribution—as described in Section 2.1 and [13]—minimizes problems with finite resolution of depth maps and existence of non-Lambertian reflections in the scene. Therefore, for scenes with limited occlusions, the proposed view selection method purposely selects views distributed uniformly.

The results demonstrate that the proposed view selection method brings clear benefits when the proportion of occluded regions exceeds approximately 10% (c.f., Table 1).

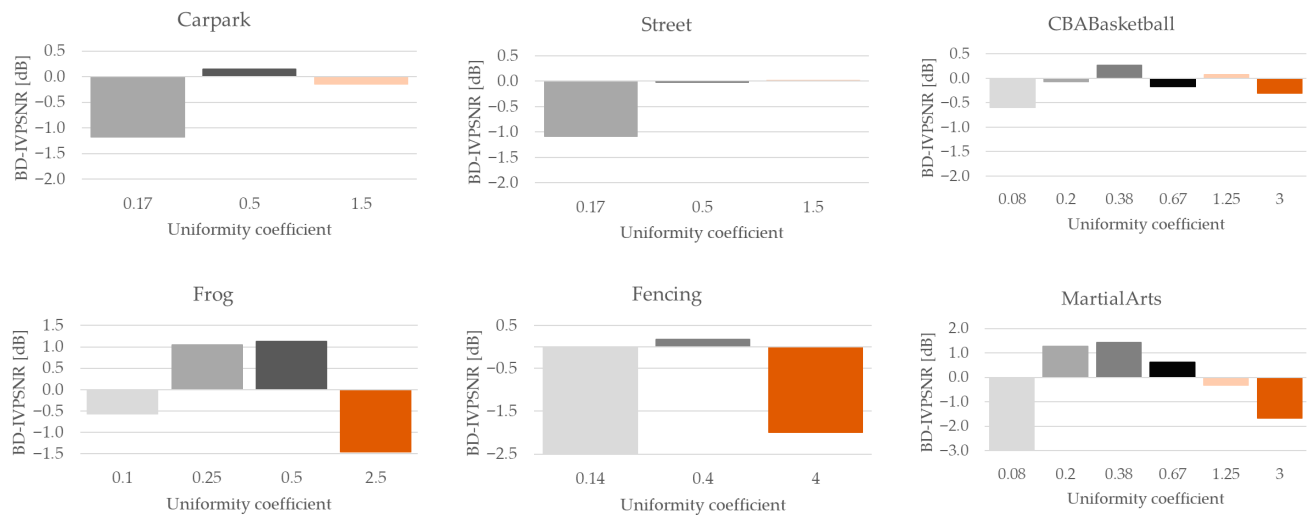


Figure 9. BD-IVPSNR of the immersive video system compared to uniform view distribution (uniformity coefficient = 1); grey bars: two pairs of cameras (uniformity coefficient < 1), red bars: a pair of cameras in the middle and two single cameras at each side (uniformity coefficient > 1).

In such cases, pair-wise camera grouping improves the reconstruction of disoccluded areas and reduces geometric distortions in synthesized views. On the other hand, for sequences with minimal occlusions (e.g., Carpark, Street), a uniform camera distribution remains sufficient, and camera pairing provides no significant quality gains (Table 2 and Figures 10 and 11).

Table 2. Gain of camera pairing (compared to the uniform camera arrangement).

Sequence	IV-PSNR Gain	BD-IVPSNR Gain
Frog	1.47 dB	1.14 dB
Carpark	0.13 dB	0.11 dB
Street	0.07 dB	−0.03 dB
Fencing	0.14 dB	0.20 dB
CBABasketball	0.37 dB	0.27 dB
MartialArts	1.15 dB	1.44 dB

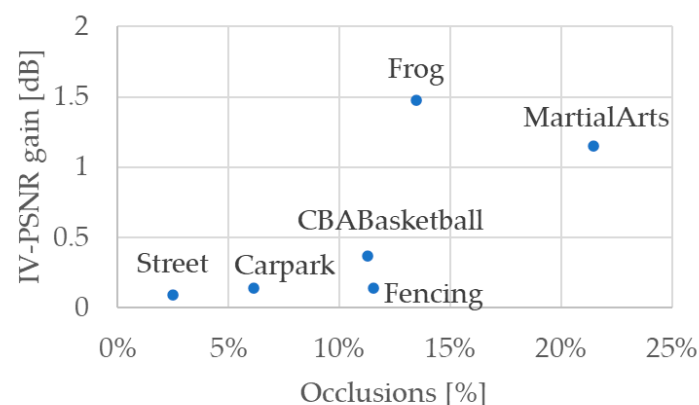


Figure 10. IV-PSNR gain from camera pairing (compared to the uniform camera arrangement) as a function of occlusions.

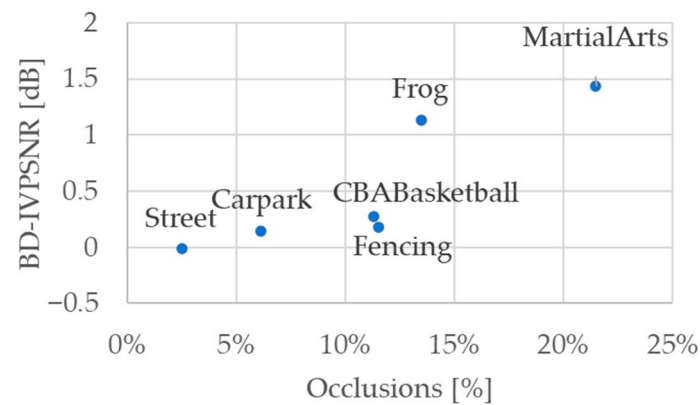


Figure 11. BD-IVPSNR gain from camera pairing (compared to the uniform camera arrangement) as a function of occlusions.

Overall, the experimental results confirm that the spatial arrangement of cameras has a measurable effect on the final quality of content presented to the user of an immersive video system. As presented, the pairwise view selection optimally balances occlusion handling and the precision of depth estimation, making it particularly suitable for complex scenes where occluded areas are a significant part of the visible content.

5.2. Supplementary Experiment

Figure 12 presents a quality increase caused by camera pairing in comparison with uniform view distribution. All sequences used in the supplementary experiment contain enough input views to provide results for ten different uniformity coefficients (including a coefficient equal to 1, representing uniform camera arrangement). Among all test sequences, there are three for which camera pairing always introduces quality loss: BBB Rabbit Arc, BBB Butterfly Arc, and Dog. As presented in Table 3, these three sequences are characterized by the lowest percentage of occlusions. For the remaining sequences, in which the occlusion ratio is higher, camera pairing with a sufficiently large baseline (uniformity coefficient > 0.5) provides gains in terms of the quality of synthesized views.

Table 3. Percentage of occluded areas in test sequences.

Sequence	Percentage of Occluded Areas
BBB Rabbit Arc	4.12%
BBB Butterfly Arc	8.71%
Dog	9.13%
BBB Rabbit Linear	15.30%
Pantomime	16.13%
BBB Butterfly Linear	16.22%
ChampagneTower	34.72%
Bee	37.11%
BBB Flowers Arc	39.34%

To more clearly illustrate the relationship between occlusions (Table 3) and the effectiveness of camera pairing (Figure 12), Figure 13 presents a scatterplot combining occlusion ratio and IV-PSNR gain. For all sequences, the results obtained for the arrangement with a uniformity coefficient of 0.7 are shown. The results confirm the outcomes from the MIV experiment: camera pairing yields consistent quality benefits when the proportion

of occluded regions exceeds approximately 10%. In opposite cases, a uniform camera arrangement outperforms the pair-wise camera layout in terms of virtual view quality.

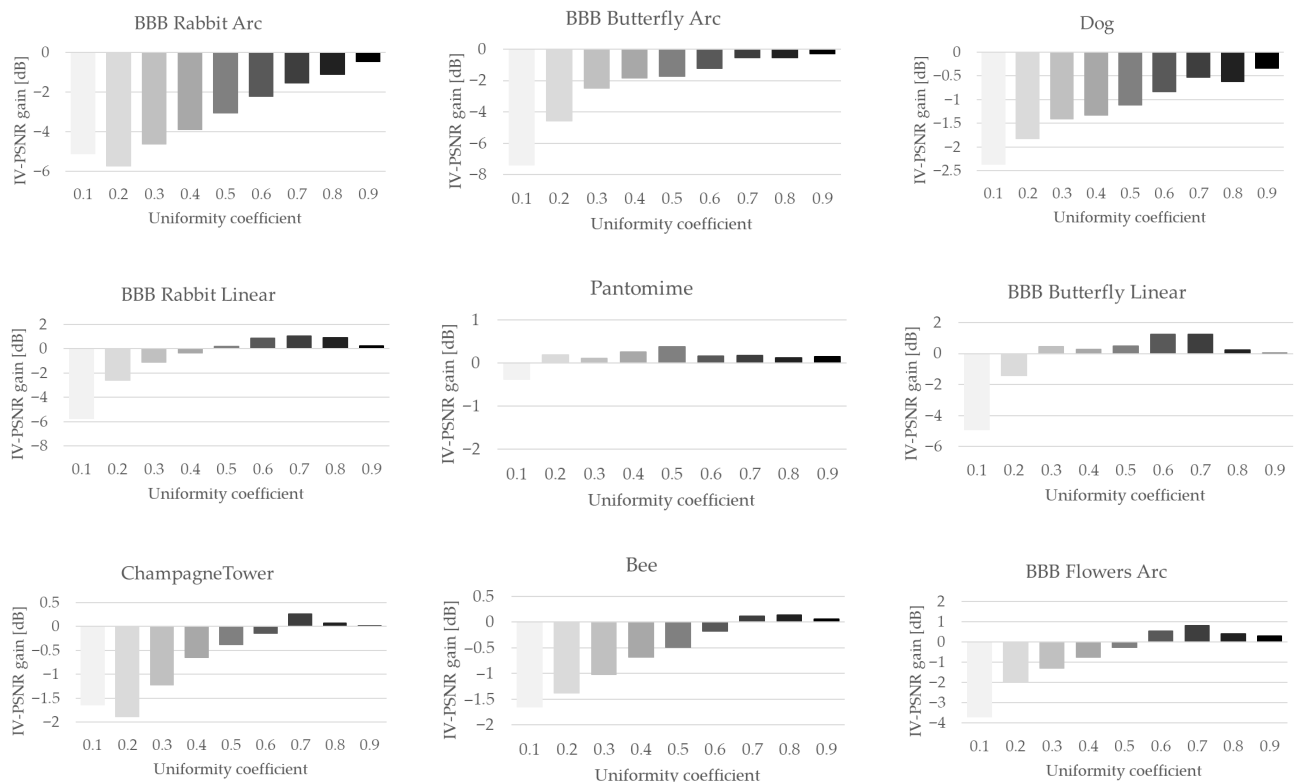


Figure 12. IV-PSNR of virtual views synthesized based on uncompressed atlases—quality increase compared to uniform view distribution (uniformity coefficient = 1); sequences sorted based on ascending ratio of occluded areas (Table 3).

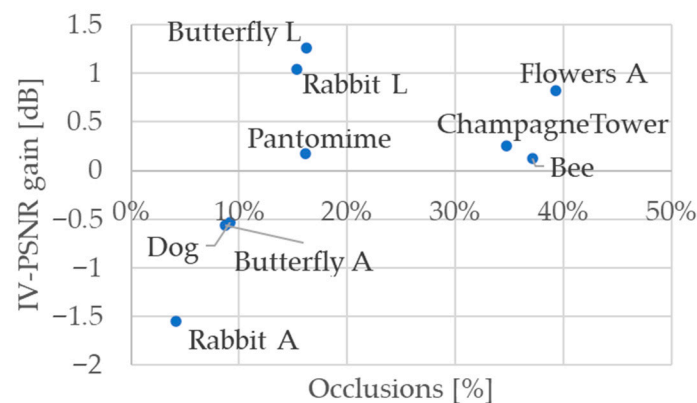


Figure 13. IV-PSNR gain from camera pairing (compared to the uniform camera arrangement) as a function of occlusions.

An additional observation concerns the three sequences with the highest occlusion ratios (above 30%), for which the quality gains from camera pairing are present but smaller than could be expected given the strong advantage of pair-wise camera setups. This behavior is consistent with the limitations of current immersive video pipelines—when the scene becomes overly complicated (the proportion of occluded areas becomes very large), the accuracy of depth estimation and view synthesis is constrained by the lack of information. In such cases, camera pairing cannot fully compensate for the severe geometric ambiguity, and the quality gains are restricted by incomplete scene information. Nevertheless, even in such challenging scenarios, the pair-wise camera layout still outperforms the uni-

form arrangement, proving that the proposed approach is beneficial for all tested content difficulty levels.

It is important to highlight that the 10% threshold—however consistent in both presented experiments—is smaller than a similar threshold reported in [25], where it exceeded 20%. The difference in the occlusion threshold used in this article, when compared with research conducted in [25], originates from the evolution of processing tools used for depth estimation and view synthesis. The study conducted in [25] relied on the DERS [29] algorithm for depth estimation and VSRS [41] software for virtual view synthesis. For the purpose of this article, the authors used IVDE v7.0 [32] depth estimation software and TMIV’s view weighting synthesizer (VWS) [35] for virtual view synthesis. Both IVDE and VWS were developed as successors to DERS and VSRS, and they contain improvements such as inter-view and temporal consistency (IVDE) and the ability to synthesize views based on more than two input views (VWS). As a result, the newer software provides more robust depth estimation, improved handling of challenging or weakly textured regions, and significantly higher-quality virtual view synthesis. These improvements reduce the sensitivity of the system to missing information and therefore shift the effective threshold at which stereo-pair grouping becomes advantageous.

In order to present the evolution of immersive video processing pipeline efficiency, we have estimated PSNR values for all tested sequences and uniformity coefficients, keeping the same methodology as in [25] (PSNR for luma component only, averaged over all virtual views). Obtained results are presented in Table 4.

Table 4. Average PSNR of luma component for different camera arrangements.

Sequence	Uniformity Coefficient									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
	PSNR of Luma Component [dB]									
BBB Rabbit Arc	26.73	27.26	28.39	29.27	30.17	31.00	31.60	32.06	32.41	32.65
BBB Butterfly Arc	27.37	29.85	31.87	32.56	32.85	33.18	33.85	33.83	33.93	34.27
Dog	22.28	22.74	23.17	23.36	23.83	24.12	24.48	24.61	24.72	25.30
BBB Rabbit Linear	23.19	24.92	26.07	26.97	27.55	28.11	28.47	28.51	28.31	28.22
Pantomime	24.07	24.57	24.69	24.96	25.72	25.93	26.04	26.32	26.27	26.47
BBB Butterfly Linear	26.29	29.11	30.65	31.25	31.67	32.86	33.17	32.46	32.17	32.25
ChampagneTower	17.70	17.73	18.47	19.01	19.04	19.20	19.37	19.66	19.48	19.44
Bee	16.30	16.63	16.92	17.17	17.31	17.51	17.62	17.70	17.57	17.58
BBB Flowers Arc	20.45	22.14	22.85	23.41	23.85	24.54	24.77	24.27	24.10	23.86

Table 5 contains differences between results gathered in Table 4 and the results reported in [25]. As presented, for most of the content, the quality is significantly higher when using the modern immersive video processing pipeline (TMIV + IVDE) than with the use of legacy depth estimation and view synthesis software (DERS + VSRS). The only exceptions are the Bee and Dog sequences (and Pantomime for several uniformity coefficients), where the multiview-based synthesis used in VWS performed worse than the simple two-view synthesis used in the VSRS algorithm.

Table 5. Difference in PSNR of the luma component between current research and research presented in [25]; positive value means that views synthesized using the current TMIV pipeline have better quality than those reported in [25].

Sequence	Uniformity Coefficient									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
	PSNR of Luma Component [dB]									
BBB Rabbit Arc	−0.08	0.48	1.46	2.17	2.73	3.26	3.70	3.97	4.13	4.22
BBB Butterfly Arc	0.68	2.33	1.75	1.46	1.90	1.98	2.30	1.89	2.10	2.12
Dog	−2.16	−1.84	−1.84	−2.12	−1.69	−1.55	−1.64	−0.99	−0.65	−0.47
BBB Rabbit Linear	1.64	2.26	2.78	3.51	3.94	4.36	4.16	4.94	4.39	4.57
Pantomime	4.22	2.44	0.52	−1.07	−0.81	−0.42	−0.54	−0.09	−0.52	−0.27
BBB Butterfly Linear	3.78	3.4	2.97	3.33	3.31	4.05	3.89	3.06	2.69	2.94
ChampagneTower	0.27	−1.18	−0.38	−0.20	0.33	0.18	0.97	1.86	1.34	1.79
Bee	−2.86	−4.05	−4.03	−3.98	−3.83	−3.50	−3.20	−2.97	−2.86	−2.46
BBB Flowers Arc	0.42	1.11	1.52	2.18	2.90	4.21	4.22	4.81	4.77	4.75
Average	0.66	0.55	0.53	0.59	0.98	1.40	1.54	1.83	1.71	1.91

Taken together, the results in Tables 4 and 5, and Table 2 from [25] illustrate how the substantial progress in depth estimation and view synthesis achieved over the past decade fundamentally changed the system’s sensitivity to occluded content, fully justifying the lower empirical occupancy threshold observed in this work.

5.3. Computational Overhead

The coding pipeline of the proposal follows the standard TMIV workflow, with all modules and settings unchanged, except for the addition of the occlusion analysis step. The proposed occlusion analysis is executed using the effective and fast reprojection module already implemented within the TMIV v16.0 software [35]. To compute the occlusion ratio, four evenly spaced input views are selected, and each view is reprojected onto the remaining ones to determine the proportion of pixels that are visible by fewer than two cameras (i.e., for which the depth is not estimable).

To maintain encoding efficiency, the occlusion analysis is performed once per GOP (only for the first frame of GOP), and the resulting view selection persists for the entire GOP. Such an approach additionally decreases the computational overhead introduced by the proposed approach.

As presented in Table 6, the time required for performing the introduced occlusion analysis is negligible when compared to the total TMIV encoding time. In both experiments, the proposal increased the computational time by less than 0.5%. Moreover, the proposal does not change the decoding time, which is crucial in any practical video system.

Table 6. Computational time of the proposed approach compared to the unmodified TMIV encoder in DSDE configuration; results averaged over all sequences.

	MIV Experiment	Supplementary Experiment
Original encoding time (per GOP)	58.25 s	27.13 s
Occlusion analysis time (per GOP)	0.255 s	0.117 s
Computational time increase	0.44%	0.43%

The presented results demonstrate that the proposed method is computationally efficient and practical for real-world immersive video encoding, providing an efficient and reliable view selection with minimal impact on processing time.

6. Conclusions

In this paper, we propose an adaptive view selection method for the MIV DSDE profile that dynamically switches between uniform camera placement and grouping them into stereo pairs. This decision is made based on an analysis of the occlusion level in the scene, performed at the encoder.

Our experimental results, obtained within the TMIV reference software, quantitatively validate this approach. We demonstrate a clear correlation between the percentage of occluded areas and the optimal camera layout. A key finding is the identification of a decision threshold: for scenes with occlusion levels exceeding approximately 10%, the stereo pair grouping strategy yields a significant and measurable gain in quality (up to 2 dB BD-IVPSNR). Below this threshold, the traditional uniform layout is sufficient or even more effective; thus, the uniform layout is automatically chosen by the proposed method.

However, it should be emphasized that this threshold is not a universal constant, but rather a consequence of the performance of the depth estimation, view synthesis, and video encoding algorithms used in an immersive video pipeline. In the modern MIV-based system, where TMIV with its efficient view weighting synthesizer [35], together with IVDE [32] for decoder-side depth estimation, is used, the threshold is equal to approximately 10%. Historically, when legacy state-of-the-art depth estimation and view synthesis were used (DERS [29] and VSRS [41], respectively), the effective threshold was significantly higher, reaching 20–25% for comparable multiview setups [25]. This evolution reflects the continuous improvement of geometric reconstruction of 3D scenes using modern immersive video processing techniques. Therefore, the empirical threshold reported in this paper should be interpreted as typical for modern MIV DSDE pipelines.

Overall, the presented results confirm that the spatial arrangement of transmitted views has a measurable impact on the quality of video watched by a viewer in a DSDE immersive video system. The proposed adaptive method effectively balances occlusion handling and geometric accuracy, making it suitable for practical use.

For future work, while this study confirmed the benefit of switching to stereo pairs, further investigation could focus on automating the selection of the optimal baseline (represented by the “uniformity coefficient”) for those pairs, potentially adapting it dynamically based on scene geometry. Furthermore, exploring alternative or combined scene analysis metrics beyond a simple occlusion percentage could lead to an even more robust and fine-grained decision model for view selection.

Author Contributions: Conceptualization, D.K. and A.D.; methodology, D.K. and A.G.; software, D.K., A.G., and D.M.; validation, D.K. and A.D.; writing, D.K., A.D., and D.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Ministry of Science and Higher Education of the Republic of Poland.

Data Availability Statement: The experiments described in Section 4 were conducted using the TMIV v16.0, <https://gitlab.com/mpeg-i-visual/tmiv>, accessed on 11 August 2025, and IVDE v7.0, <https://gitlab.com/mpeg-i-visual/ivde>, accessed on 11 August 2025, using MPEG test sequences [33,38–40].

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

References

- Wien, M.; Boyce, J.M.; Stockhammer, T.; Peng, W.-H. Standardization Status of Immersive Video Coding. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2019**, *9*, 5–17. [\[CrossRef\]](#)
- ISO/IEC JTC1/SC29/WG11MPEG/N18145; MPEG Call for Proposals on 3DoF+Visual. ISO: Brussels, Belgium, 2020.
- ISO/IEC JTC1/SC29/WG11 MPEG N17073; Requirements on 6DoF (v1). ISO: Torino, Italy, 2017.
- Xiang, S.; Yu, L.; Liu, Q.; Xiong, Z. A gradient-based approach for interference cancelation in systems with multiple Kinect cameras. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013; pp. 13–16.
- Stankiewicz, O.; Domanski, M.; Dziembowski, A.; Grzelka, A.; Mieloch, D.; Samelak, J. A free-viewpoint television system for horizontal virtual navigation. *IEEE Trans. Multimed.* **2018**, *20*, 2182–2195. [\[CrossRef\]](#)
- Fachada, S.; Bonatto, D.; Schenkel, A.; Lafruit, G. Depth image based view synthesis with multiple reference views for virtual reality. In Proceedings of the 3DTV Conference 2018, Stockholm, Sweden, Helsinki, Finland, 3–5 June 2018.
- Muller, K.; Merkle, P.; Wiegand, T. 3-D Video Representation Using Depth Maps. *Proc. IEEE* **2010**, *99*, 643–656. [\[CrossRef\]](#)
- Sullivan, G.; Ohm, J.-R.; Han, W.-J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [\[CrossRef\]](#)
- Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.-R. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [\[CrossRef\]](#)
- Tech, G.; Chen, Y.; Müller, K.; Ohm, J.-R.; Vetro, A.; Wang, Y.-K. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 35–49. [\[CrossRef\]](#)
- ISO/IEC JTC1/SC29/WG4 MPEG2023/N0341; Verification Test Report of MPEG Immersive Video. ISO: Antalya, Turkey, 2023.
- Boyce, J.; 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*, 1521–1536. [\[CrossRef\]](#)
- Dziembowski, A.; Samelak, J.; Domanski, M. View selection for virtual view synthesis in free navigation systems. In Proceedings of the International Conference on Signals and Electronic Systems, ICSES, Kraków, Poland, 10–12 September 2018.
- Dziembowski, A.; Grzelka, A.; Mieloch, D.; Stankiewicz, O.; Wegner, K.; Domanski, M. Multiview Synthesis—improved view synthesis for virtual navigation. In Proceedings of the 32nd Picture Coding Symposium (PCS), Nuremberg, Germany, 4–7 December 2016.
- Ceulemans, B.; Lu, S.-P.; Lafruit, G.; Munteanu, A. Robust multiview synthesis for wide-baseline camera arrays. *IEEE Trans. Multimed.* **2018**, *20*, 2235–2248. [\[CrossRef\]](#)
- Li, S.; Zhu, C.; Sun, M.-T. Hole filling with multiple reference views in DIBR view synthesis. *IEEE Trans. Multimed.* **2018**, *20*, 1948–1959. [\[CrossRef\]](#)
- ISO/IEC JTC1/SC29/WG11 MPEG2013/M40657; View Synthesis Reference Software (VSRS) 4.2 with Improved Inpainting and Hole Filling. ISO: Geneva, Switzerland, 2017; pp. 3–7.
- Kłóska, D.; Dziembowski, A.; Samelak, J. Versatile input view selection for efficient immersive video transmission. In Proceedings of the 31st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG, Prague, Czechia, 15 May 2023.
- ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748; Kermit Test Sequence for Windowed 6DoF Activities. ISO: Geneva, Switzerland, 2018.
- Rahimian, P.; Kearney, J.K. Optimal camera placement for motion capture systems. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 1209–1221. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chen, X.; Davis, J. An occlusion metric for selecting robust camera configurations. *Mach. Vis. Appl.* **2008**, *19*, 217–222. [\[CrossRef\]](#)
- Olague, G.; Mohr, R. Optimal camera placement for accurate reconstruction. *Pattern Recognit.* **2002**, *35*, 927–944. [\[CrossRef\]](#)
- Qian, N.; Lo, C.-Y. Optimizing camera positions for multiview 3D reconstruction. In Proceedings of the International Conference on 3D Imaging (IC3D), Liege, Belgium, 14–15 December 2015; pp. 1–8.
- Gargallo, P.; Prados, E.; Sturm, P. Minimizing the reprojection error in surface reconstruction from images. In Proceedings of the IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007.
- Domański, M.; Dziembowski, A.; Grzelka, A.; Mieloch, D. Optimization of camera positions for free-navigation applications. In Proceedings of the International Conference on Signals and Electronic Systems, ICSES 2016, Kraków, Poland, 5–7 September 2016.
- Shao, F.; Lin, W.; Jiang, G.; Yu, M.; Dai, Q. Depth map coding for view synthesis based on distortion analyses. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2014**, *4*, 106–117. [\[CrossRef\]](#)
- Wang, R.; Luo, J.; Jiang, X.; Wang, Z.; Wang, W.; Li, G.; Gao, W. Accelerating image-domain-warping virtual view synthesis on GPGPU. *IEEE Trans. Multimed.* **2017**, *19*, 1392–1400. [\[CrossRef\]](#)
- Tanimoto, M.; Panahpour, M.; Fujii, T.; Yendo, T. FTV for 3-D spatial communication. *Proc. IEEE* **2012**, *100*, 905–917. [\[CrossRef\]](#)
- ISO/IEC JTC1/SC29/WG11 MPEG/M31518; Enhanced Depth Estimation Reference Software (DERS) for Freeviewpoint Television. ISO: Geneva, Switzerland, 2013.

30. ISO/IEC JTC1/SC29/WG11 MPEG/M37232; Depth Based View Blending in View Synthesis Reference Software (VSRS). ISO: Geneva, Switzerland, 2015.
31. Sun, Z.; Jung, C. Real-time depth-image-based rendering on GPU. In Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an, China, 17–19 September 2015; pp. 324–328.
32. ISO/IEC JTC1/SC29/WG04 MPEG VC/N0058; Manual of Immersive Video Depth Estimation. ISO: Geneva, Switzerland, 2021.
33. ISO/IEC JTC1/SC29/WG04 MPEG VC/N0539; Common Test Conditions for MPEG Immersive Video. ISO: Geneva, Switzerland, 2024.
34. Samelak, J.; Dziembowski, A.; Mieloch, D. Advanced HEVC Screen Content Coding for MPEG Immersive Video. *Electronics* **2022**, *11*, 23. [\[CrossRef\]](#)
35. ISO/IEC JTC1/SC29/WG4 MPEG2021/N0084; Test Model 9 for MPEG Immersive Video. ISO: Geneva, Switzerland, 2021.
36. Wieckowski, A.; Brandenburg, J.; Hinz, T.; Bartnik, C.; George, V.; Hege, G.; Helmrich, C.; Henkel, A.; Lehmann, C.; Stoffers, C.; et al. VVenC: An Open and Optimized VVC Encoder Implementation. In Proceedings of the IEEE International Conference on Multimedia Expo Workshops (ICMEW), Online, 5–9 July 2021.
37. Stankowski, J.; Dziembowski, A. Version [7.1]—[IV-PSNR: Software for immersive video objective quality evaluation]. *SoftwareX* **2024**, *28*, 101961. [\[CrossRef\]](#)
38. ISO/IEC JTC1/SC29/WG11 M35721; [FTV AHG] Big Buck Bunny Light-Field Test Sequences. ISO: Geneva, Switzerland, 2015.
39. MPEG2013/M32995; FTV AHG: EE1 and EE2 Results with Bee by NICT Document IEC JTC1/SC29/WG11. ISO: Geneva, Switzerland, 2014.
40. ISO/IEC JTC1/SC29/WG11 M15378; 1D Parallel Test Sequences for MPEG-FTV. ISO: Archamps, France, 2008.
41. ISO/IEC JTC 1/SC 29/WG 11 M31520; Enhanced View Synthesis Reference Software (VSRS) for Free-Viewpoint Television. ISO: Geneva, Switzerland, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

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 and 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 Editors: Vladimir A. Golovko, Kurosh Madani and Anatoliy A. Sachenko

Received: 13 June 2024

Revised: 27 June 2024

Accepted: 28 June 2024

Published: 1 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

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

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

where P is the set of all points of the input view, p is a point of the input view, $\bar{d}_p$ is the currently considered depth of a point p , D_p is the data term that represents the cost of assigning the depth $\bar{d}_p$ 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 , $\bar{d}_q$ 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 $\bar{d}_p$ to the point p and depth $\bar{d}_q$ 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 $\bar{d}_p$. 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:

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

where β_0 is the smoothing coefficient provided by the user or calculated from camera parameters [4], $\bar{d}_p$ is the currently considered depth of a point p , and $\bar{d}_q$ 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 $E(\bar{d}_p)$, 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

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

where $P_{on}(\bar{d}_p)$ is the set of active nodes.

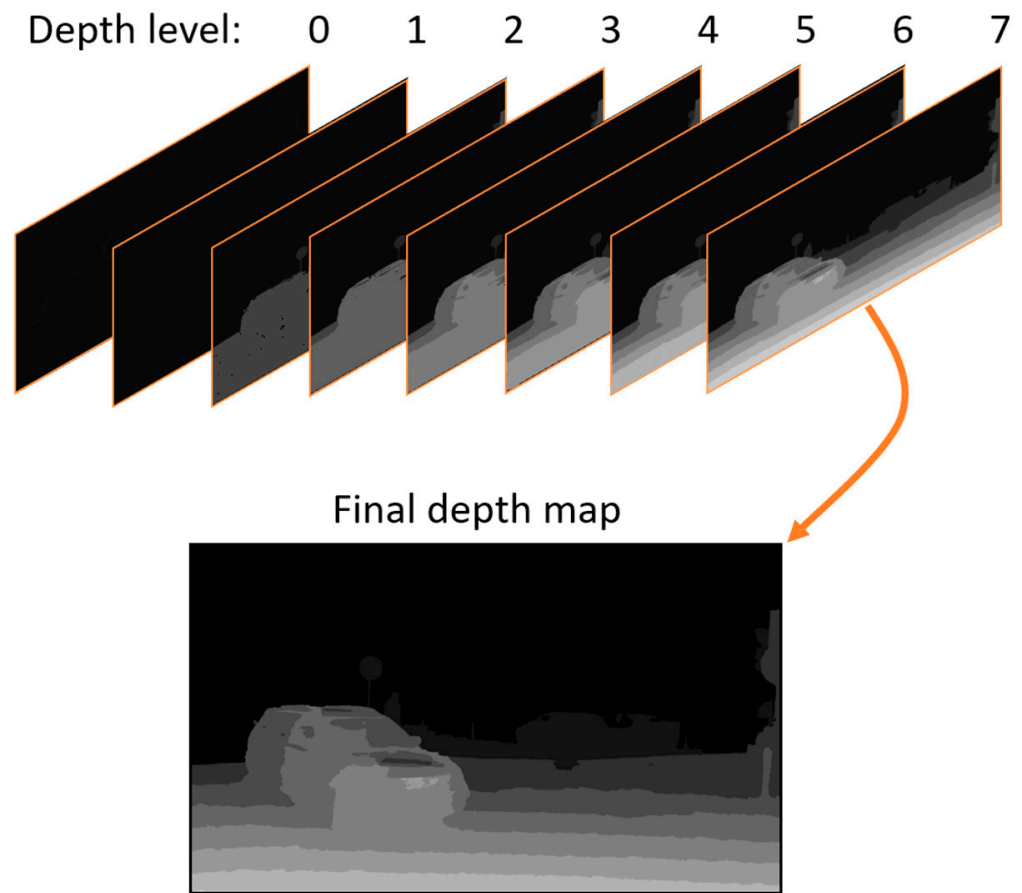


Figure 1. An example of alpha-expansion-based depth estimation.

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

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

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.

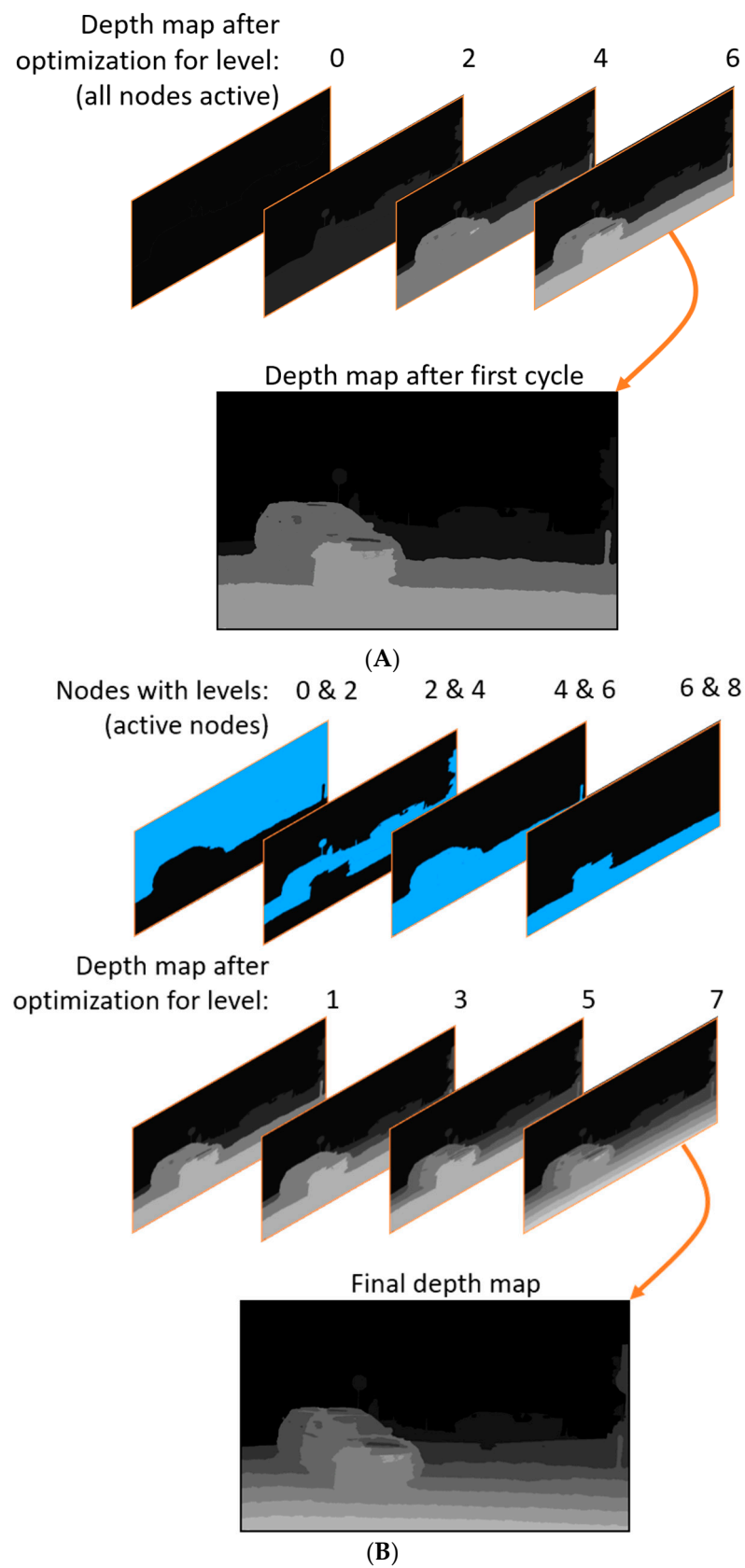


Figure 2. An example of hierarchical alpha-expansion-based depth estimation. The currently active nodes are shown as the blue area. (A) First cycle of graph cut (coarse optimization). (B) Second cycle of graph cut (refinement optimization).

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

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

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:

$$P_{on}(\bar{d}_p) = \{r : |\bar{d}_p - r| < n \wedge r = \bar{d}_q\}. \quad (6)$$

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 (Bjontegaard 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 metrics.

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

- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#) [\[PubMed\]](#)
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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.
- 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.
- 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. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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.
- 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.
- 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.
- Papadakis, N.; Caselles, V. Multi-label Depth Estimation for Graph Cuts Stereo Problems. *J. Math. Imaging Vis.* **2010**, *38*, 70–82. [\[CrossRef\]](#)
- Mieloch, D.; Grzelka, A. Segmentation-based Method of Increasing the Depth Maps Temporal Consistency. *Int. J. Electron. Telecommun.* **2018**, *64*, 283–289. [\[CrossRef\]](#)
- Mieloch, D.; Stankiewicz, O.; Domanski, M. Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation. *IEEE Access* **2020**, *8*, 5760–5776. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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.
- 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. [\[CrossRef\]](#)
- Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Xue, T.; Owens, A.; Scharstein, D.; Goesele, M.; Szeliski, R. Multiframe stereo matching with edges, planes, and superpixels. *Image Vis. Comput.* **2019**, *91*, 103806. [\[CrossRef\]](#)
- 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.
- ISO/IEC JTC1/SC29/WG4 MPEG2023/N0406; Common Test Conditions for MPEG Immersive Video. ISO: Geneva, Switzerland, 2023.
- Bjontegaard, G. Calculation of average PSNR differences between RD-Curves. In Proceedings of the ITU-T VCEG Meeting, Austin, TX, USA, 2–4 April 2001.
- ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415; 3DoF+ Test Sequence ClassroomVideo. ISO: Geneva, Switzerland, 2018.
- ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349; Technicolor 3DoF+ Test Materials. ISO: Geneva, Switzerland, 2018.
- ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787; New Test Content for Immersive Video—Nokia Chess. ISO: Geneva, Switzerland, 2020.
- 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.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Versatile input view selection for efficient immersive video transmission

Dominika Klóska Adrian Dziembowski Jarosław Samelak
dominika.kloska@put.poznan.pl adrian.dziembowski@put.poznan.pl jaroslaw.samelak@gmail.com

Institute of Multimedia Telecommunications, Poznań University of Technology
Polanka 3, 61-131 Poznań, Poland

ABSTRACT

In this paper we deal with the problem of the optimal selection of input views, which are transmitted within an immersive video bitstream. Due to limited bitrate and pixel rate, only a subset of input views available on the encoder side can be fully transmitted to the decoder. Remaining views are – in the simplest approach – omitted or – in the newest immersive video encoding standard (MPEG immersive video, MIV) – pruned in order to remove less important information. Selecting proper views for transmission is crucial in terms of the quality of immersive video system user's experience. In the paper we have analyzed which input views have to be selected for providing the best possible quality of virtual views, independently on the viewport requested by the viewer. Moreover, we have proposed an algorithm, which takes into account a non-uniform probability of user's viewing direction, allowing for the increase of the subjective quality of virtual navigation for omnidirectional content.

Keywords

Immersive video, virtual view synthesis, MPEG immersive video (MIV)

1. INTRODUCTION

A natural consequence of rapidly growing interest in immersive video and virtual reality (VR) is the demand for efficient and versatile immersive media transmission. The virtual reality technology allows the user for immersing into the scene captured by a multicamera system and virtually navigating within it (Fig. 1). Such a navigation may be restricted to several degrees of freedom (DoF). For instance, 3DoF systems allow users to rotate their head around a single pivot point, and 3DoF+ systems additionally support restricted, translational movement of user's head [MPEG19], increasing the quality of experience (QoE) when using the head-mounted display (HMD) devices. The latest, most advanced systems – 6DoF – allow users for free, unrestricted navigation within a scene [MPEG17].

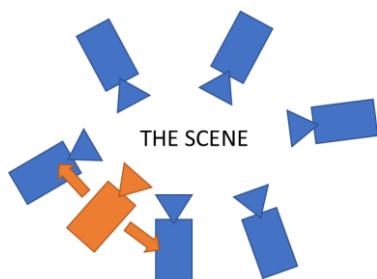


Figure 1. Idea of an immersive video system; the scene is captured by several cameras (blue), a viewer may virtually change their viewpoint (orange camera).

In order to obtain an immersive video sequence, it is required to use a multicamera system, containing even hundreds of cameras [Fuj06]. Practical systems

contain less cameras (e.g., 10 – 20 [Sta18]), but even in such a case a tremendous amount of data has to be processed and transmitted to the viewer. Moreover, the possibility of virtual immersion into the scene in the immersive video systems is provided by rendering [Fac18], [Sta22] of viewports demanded by the viewer. Such an operation requires information of the three-dimensional scene, which is typically represented in the MVD format (multiview video plus depth, Fig. 2) [Mul18]. Therefore, for each input view also a depth map should be transmitted.

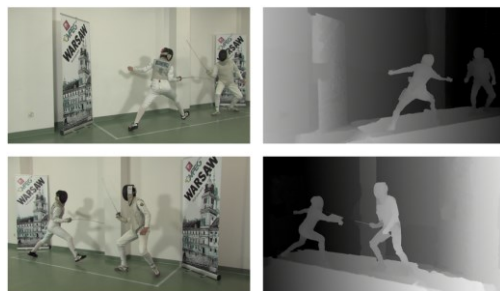


Figure 2. Sequence in MVD format.

The easiest way to address the problem of transmission of a huge amount of multiview video data would be to encode each real view (e.g., using HEVC [Sul12]) and the corresponding depth map separately – such an approach is called multiview simulcast. However, this method is not effective due to the high bitrate and pixel rate [Boy21]. Moreover, the quality of the immersive content is not satisfactory because HEVC (or any typical 2D video encoder such as the newest, VVC [Bro21]) encoder was not developed for processing depth maps. It is possible to enhance the quality of the final immersive

vision with the use of MV-HEVC and 3D-HEVC [Tec16], which are HEVC extensions dedicated to encoding 3D content. However, these methods do not guarantee versatility, because they are not adapted for encoding sequences acquired by omnidirectional cameras, or by multicamera systems where the cameras are located arbitrarily. Therefore, none of the abovementioned methods can be used in practical immersive video systems.

The simplest practical solution allowing for a significant decrease of pixel rate is to transmit only a subset of the given real views. In such an approach, in order to obtain the best possible quality, these views have to be carefully chosen.

A more sophisticated, newest approach is based on the use of the MPEG immersive video (MIV) standard [Boy21], [ISO22] which defines the compression of immersive media in a form of multiview video pre- and post-processing combined with the typical video encoder, e.g., VVC [Bro21]. The MIV encoding process can be divided into three main steps, in which the input data (n views and corresponding depth maps) are processed into k video bitstreams called “atlases”, further encoded using the VVC encoder (Fig. 3).

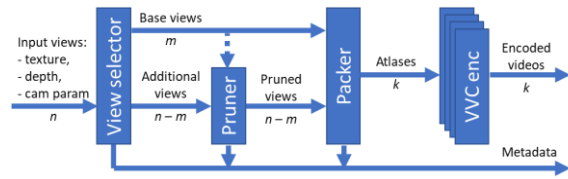


Figure 3. Simplified scheme of the MIV encoder.
Figure from [Dzi22a].

In the first step the MIV decides which views are the most important from the user’s point of view. These views are then being labeled as “base views” and are being placed in atlases in their entirety (Fig. 4A and C). Remaining views (“additional views”) contain a lot of redundant data are then pruned in order to remove the excess data. Finally, after the pruning operation additional views are packed into atlases as a form of patch mosaic (Fig. 4B and D) [Vad22].

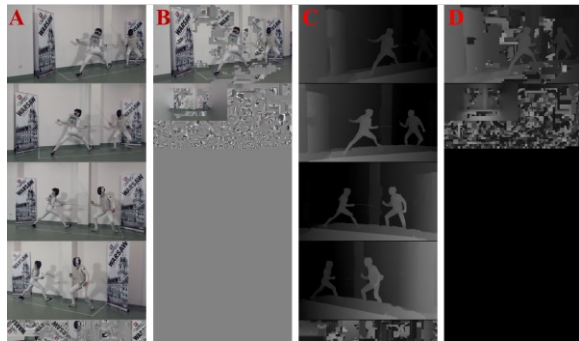


Figure 4. Four atlases produced by the MIV encoder using the MIV Main profile [Vad22]: 2 texture atlases (A, B) and 2 depth atlases (C, D).

On the decoder side, the atlases are firstly decoded using the typical video decoder (such as VVC). After the video decoding step, the views and depths stored in atlases are unpacked and then used for rendering of the views requested by user (Fig. 5).

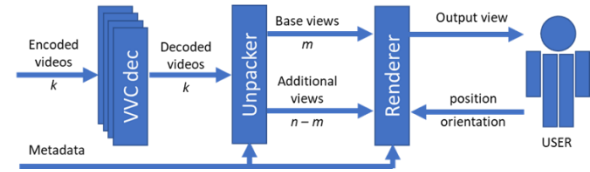


Figure 5. Simplified scheme of the MIV decoder.
Figure from [Dzi22a].

Irrespective of the immersive video coding approach, the effective input view selection is the crucial step in terms of providing the best quality and the highest coding efficiency. In this paper, we describe the view selection method which allows for efficient immersive video transmission in practical systems, where it is not possible to send all the real views to the decoder. Moreover, proposed algorithm performs efficiently both for content acquired by perspective and omnidirectional cameras, and can be used for 6DoF systems, where the user virtually immerses into the scene [Laf17].

2. INPUT VIEW SELECTION

2.1. View selection for virtual view synthesis

The proper input view selection method should provide the highest possible quality of synthesized views while preserving similar bitrate. Considerations on the influence of input view selection on the virtual view quality were described by the authors of this paper in [Dzi18], where we focused on optimizing the quality in simple free navigation systems [Sta18]. In [Dzi18], we assumed that the renderer has access to all of input views, but - in order to provide reasonable computational time - it can use only two of them for rendering purposes.

The input view choice requires addressing three problems: occlusions, finite resolution of video, and non-Lambertian surfaces; leading to the conclusion that the highest quality of rendered views can be obtained based on nearest left and nearest right input view. Obviously, in such a scenario it would be optimal to transmit these two views and skip all the others. However, a selection of these two views is possible only if the position of view requested by the viewer is known before the transmission.

2.2. View selection for immersive video transmission

In a practical immersive video system, where multiple viewers receive the same bitstream and are able to independently choose their point of view [Tan12], an assumption regarding viewer’s position known *a priori* before the transmission is invalid. Instead, it is required to choose input views in the

way, which guarantees the highest average quality of views watched by users, independently of their viewpoint.

In order to meet the requirements for immersive data transmission, where the position of a user cannot be predicted, the view selection method described in [Dzi18] has to be extended.

Taking into account the statement that the quality of synthesized view is highest when the rendering is performed on the basis of the nearest left and right real view, we have conducted a simulation. In the simulation we assumed a simple practical immersive video system with reasonable pixel rate [Boy21] and number of cameras [Sal18]:

- linear multicamera system with 13 evenly distributed cameras,
- 13 input views available at the encoder side,
- 4 input views transmitted to the decoder,
- 100 possible virtual positions of the viewer (evenly distributed too).

We assumed that the left-most and right-most input views are transmitted (in order to ensure, that for all virtual positions of the viewer there exists left and right input view). Therefore, the index of the first transmitted view was fixed to 1, and index of the fourth view was fixed to 13. Indices of remaining two input views to be transmitted were unknown, and they were iteratively changed in order to calculate their optimal position.

For each virtual position of the viewer, we calculated the total distance to the nearest left and nearest right view. The results are presented in Fig. 6, where the horizontal axis presents the index of the first real view used for virtual view synthesis, the vertical axis presents the index of the second real view.

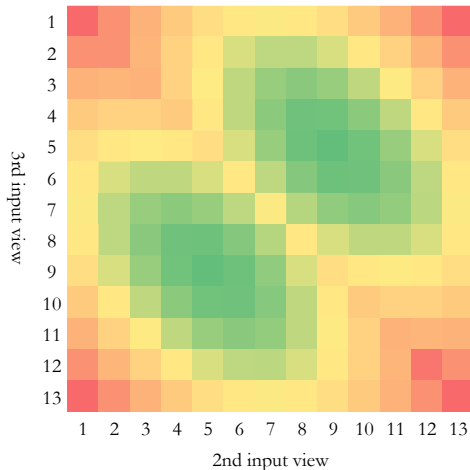


Figure 6. Total distance between nearest left and nearest right view calculated for all real views used for the experiment.

The green color in Fig. 6 indicates that the total distance between the virtual view and its transmitted

neighbors was low, red – that the distance was higher. These results show that the distance is minimized when the input views selected to be transmitted are distributed evenly. Of course, in Fig. 6 we presented only the distance measured for a simple simulation, not the quality of synthesized views. Therefore, in the next section we presented evidence for these considerations.

It should be noted that the presented model and experiment assumed a simple, linear camera arrangement. However, analogous conclusions can be taken also for more sophisticated multicamera systems.

3. EXPERIMENTAL RESULTS

3.1. Methodology

In order to prove authenticity of considerations presented in the previous section, we have performed an experiment. In the experiment, we assessed the quality of virtual views synthesized using various combinations of input views. The view synthesis was performed using the MPEG’s reference software – VVS [Dzi19].

The test set comprised of two computer-generated sequences – BBB Butterfly and BBB Flowers [Kov15]. We have decided to use these sequences, as they contain multiple (79) input views, making possible quality assessment for several viewpoints. Both sequences were captured by 79 evenly-distributed cameras placed on an arc. For each sequence, views are numbered from v6 to v84. Seven views: v6, v19, v32, v45, v58, v71, and v84 were used as input ones, while all remaining views were treated as reference for objective quality evaluation.

The quality of synthesized views was calculated using two objective quality metrics, described in the MIV Common Test Conditions (MIV CTC) [MPEG22c] and commonly used in the experiments related to immersive video: WS-PSNR [Sun17] and IV-PSNR [Dzi22b]. Both quality metrics are full-reference ones, therefore the quality was assessed by comparing input views with virtual views synthesized in the same position (the same viewpoint).

In the experiment we assumed the transmission of four input views (of the seven available). Two input views were fixed: v6 as the first input view and v84 as the fourth one. The position of second and third input views was being changed in order to define the optimal arrangement of transmitted views.

3.2. Results

Mean IV-PSNR and WS-PSNR of synthesized virtual views are presented in Tables 1 and 2. The values were averaged over 75 synthesized views (v6 to v84, excluding four views used as input ones, for

which the quality is perfect, as no synthesis is needed).

3rd input view	v19		31.32	32.06	32.32	31.34
	v32	31.32		32.10	33.16	32.57
	v45	32.06	32.10		31.78	31.59
	v58	32.32	33.16	31.78		30.74
	v71	31.34	32.57	31.59	30.74	
2nd input view						
3rd input view	v19		39.39	41.08	41.03	39.44
	v32	39.39		40.78	41.65	40.36
	v45	41.08	40.78		40.04	39.79
	v58	41.03	41.65	40.04		38.05
	v71	39.44	40.36	39.79	38.05	
2nd input view						

Table 1. Mean IV-PSNR [dB] of virtual view (averaged over 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

3rd input view	v19		24.12	24.91	24.99	23.99
	v32	24.12		24.98	25.68	25.03
	v45	24.91	24.98		24.64	24.42
	v58	24.99	25.68	24.64		23.57
	v71	23.99	25.03	24.42	23.57	
2nd input view						
3rd input view	v19		31.99	33.05	32.91	31.80
	v32	31.99		32.90	33.52	32.68
	v45	33.05	32.90		32.40	32.26
	v58	32.91	33.52	32.40		30.91
	v71	31.80	32.68	32.26	30.91	
2nd input view						

Table 2. Mean WS-PSNR [dB] of virtual view (averaged over 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

As presented, in all considered scenarios (both sequences and both quality metrics), the best average quality of synthesized views can be achieved when using views v6, v32, v58, and v84, thus evenly distributed input views.

Such a view selection provides also highest quality in a worst-case scenario (the lowest quality among all synthesized views, Table 3).

Moreover, even distribution of transmitted input views minimizes Δ IV-PSNR (difference between lowest and highest quality among all synthesized views, Table 4), making the user's experience more stable, as the perceived quality change during virtual navigation among the scene is lower.

Tables 3 and 4 present only the results obtained for the IV-PSNR metric. The WS-PSNR results were omitted, as it behaves similarly, and the best results were achieved for evenly distributed input views.

3rd input view	v19		22.95	26.94	23.50	23.40
	v32	22.95		26.94	29.72	25.72
	v45	26.94	26.94		24.41	24.17
	v58	23.50	29.72	24.41		21.41
	v71	23.40	25.72	24.17	21.41	
2nd input view						
3rd input view	v19		31.54	36.28	35.08	32.28
	v32	31.54		36.15	38.36	34.95
	v45	36.28	36.15		33.75	33.38
	v58	35.08	38.36	33.75		30.38
	v71	32.28	34.95	33.38	30.38	
2nd input view						

Table 3. Lowest IV-PSNR [dB] of virtual view (among 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

3rd input view	v19		18.99	14.99	18.74	21.14
	v32	18.99		14.99	12.52	18.82
	v45	14.99	14.99		17.83	20.37
	v58	18.74	12.52	17.83		23.15
	v71	21.14	18.82	20.37	23.15	
2nd input view						
3rd input view	v19		14.06	9.91	10.12	12.67
	v32	14.06		10.16	7.83	9.94
	v45	9.91	10.16		13.35	12.49
	v58	10.12	7.83	13.35		16.20
	v71	12.67	9.94	12.49	16.20	
2nd input view						

Table 4. Δ IV-PSNR [dB] of virtual view (among 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

4. INPUT VIEW SELECTION FOR OMNIDIRECTIONAL CONTENT

4.1. Omnidirectional content problem

All the considerations presented in previous sections assumed that the viewer can watch the scene from any viewpoint, and the probability of choosing various viewpoints is the same. However, it is not true for 6DoF and 3DoF+ [Wie19] immersive video systems, where the user virtually immerses into the scene, e.g., using the HMD device. In such a case, a typical user tends to look around in the horizontal

plane, while not focusing on floor, ceiling or the sky above [Dzi22c].

Therefore, the use of the described view selection algorithm, which chooses input views most distant to each other may result in non-optimal selection. For instance, views captured by cameras facing down or up may be selected instead of views containing essential information about the scene (Fig. 7).



Figure 7. Three views of the Chess sequence [Ilo19].

The example presented in Fig. 7. A and B are two of 7 views selected as “base views” by the MIV encoder (working under the decoder-side depth estimation – DSDE – configuration [Mie22]). Fig. 7.C presents a view, which was selected as an “additional view” and skipped despite having more important information from the viewer’s perspective.

Such a selection increases the quality of the floor and the ceiling but decreases the quality of a chess knight – which is more crucial for the viewer.

4.2. Proposed solution

Taking into account the subjective non-uniform significance of different areas of the scene, we proposed a modification of the simple view selection algorithm, which penalizes the vertical distance between cameras.

In the basic approach (e.g., the one implemented in the 14th version of the Test Model for MPEG immersive video – TMIV 14 [MPEG22a]), basic views were selected by maximization of the total distance between them. The distance between two views i and j was calculated as:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (r_{i,j}^z)^2}, \quad (1)$$

where $r_{i,j}^x$, $r_{i,j}^y$, and $r_{i,j}^z$ are distances between views i and j along three axes of the global coordinate system.

We proposed to modify (1) by addressing the non-uniform probability of viewer’s watching direction and by penalizing the vertical distance. To achieve that the camera distances calculated in the view selection process are not homogenous meaning that the vertical direction is being treated differently from horizontal directions:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (w \cdot r_{i,j}^z)^2}, \quad (2)$$

Where r^x, r^y, r^z indicate a distance between two cameras among three axes, and where w is the inhomogeneity coefficient. In the experiments described in the further part of this section, the w value was set to 0.4.

The proposed change allows for selecting input views, which carry valuable information (Fig. 8.B) instead of sending views containing plain floor or ceiling of the scene, irrelevant for the viewer (Fig. 8).

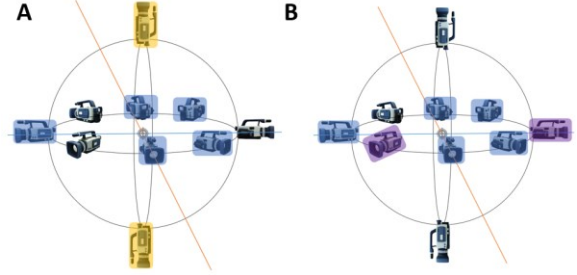


Figure 8. Sequence Chess. A – views selected for encoding by the anchor method (blue color highlights selected cameras that carry valuable information about the scene, yellow shows selected cameras that have less importance to the viewer and therefore can be omitted), B – views selected for encoding after the modification (purple color highlights cameras that were selected instead of the yellow cameras from Fig 8.A).

Cameras highlighted in blue were selected as base views for both basic and modified view selection algorithms. Besides them, the basic algorithm selected cameras facing up and down (yellow cameras in Fig. 8), while the modified algorithm – two cameras acquiring important parts of the scene. Considering the fact, that a typical viewer spends more time looking around on the horizontal plane rather than the vertical one (which contains the floor and the ceiling) [Dzi22c], we propose to send more views from the horizontal plane instead of the views facing upwards and downwards. It will have a positive influence on the final quality of the particular parts of the scene at which the user looks the majority of the time.

The proposed modification was appreciated by the experts of the ISO/IEC JTC1/SC29/WG 04 MPEG VC group and is included in the newest version of the Test Model for MIV – TMIV 15 [MPEG22b].

The influence of this view selection modification on the objective and subjective quality of the final immersive vision was described in the following subsections.

4.3. Methodology of the experiment

The experiment was conducted under the common test conditions for MPEG immersive video (MIV CTC) [MPEG22c], but the test set was limited to omnidirectional sequences only (fig 9).

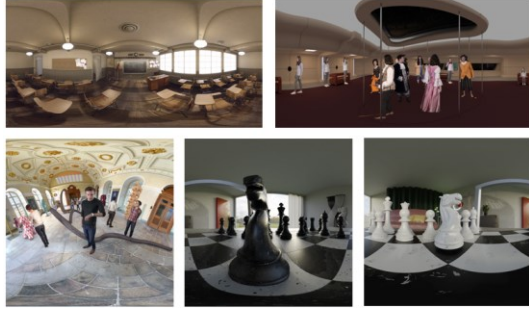


Figure 9. Omnidirectional sequences used for the experimental results, sequences: ClassroomVideo [Kro18], Hijack [Dor18], Museum [Dor18], Chess [Ilo19], and ChessPieces [Ilo20].

In the experiment, the TMIV 14 software [MPEG22a] was used. Each sequence was compressed with the use of five different rate points (RP) using the VVC encoder. The total bitrate for a sequence ranged from 2.5 Mbps (RP5) to over 20 Mbps (RP1). The quality of the synthesized virtual views was assessed using WS-PSNR and IV-PSNR objective quality metrics [Sun17], [Dzi22b].

In order to perform a thorough test, three configurations of TMIV were evaluated: MIV, MIV View and MIV DSDE. The detailed description of these configurations can be found in the publicly available MIV CTC document [MPEG22c].

Besides the objective quality measurement, also the subjective quality of rendered views was evaluated. The subjective quality assessment was performed based on pose traces [Boy21], according to the MIV CTC.

The subjective quality evaluation was done by 45 naïve viewers, watching two side-by-side videos (Pair Comparison method, Rec. ITU-T P.910 [ITU08] and ITU-R BT.500 [ITU98]). The viewers judged the quality of presented posetraces with the use of 7-number scale with values from -3 to 3.

To minimize the time duration of the subjective test, only three RP were shown to the viewers: RP1, RP3, and RP5. Moreover, subjects were assessing quality change only for sequences, for which the view selection result was different, than for unmodified TMIV14 (see Table 5).

MIV Configuration Sequence Name	MIV Main	MIV View	MIV DSDE
Classroom	×	×	×
Museum	✓	✓	✓
Hijack	×	×	✓
Chess	×	×	✓
ChessPieces	×	×	✓

Table 5. Overview of the sequences and different MIV configurations. „X” indicates the scenario in which view selection result was the same as with the unmodified TMIV 14 software.

a) Subjective quality evaluation

The results of performed subjective quality evaluation are presented in Figs. 10 and 11. In Fig. 10, an influence of the proposed method on efficiency of different MIV configurations are presented. Fig. 11 contains comparison of subjective quality change for different test sequences. The results are presented as an average quality change caused by the proposed modification and the 95% confidence interval, calculated according to ITU-R recommendations [ITU98] as:

$$CI = 1.96 \cdot \frac{SD}{\sqrt{N}}, \quad (3)$$

where CI is the confidence interval, SD – standard deviation, and N – number of viewers (in presented experiment N = 45).

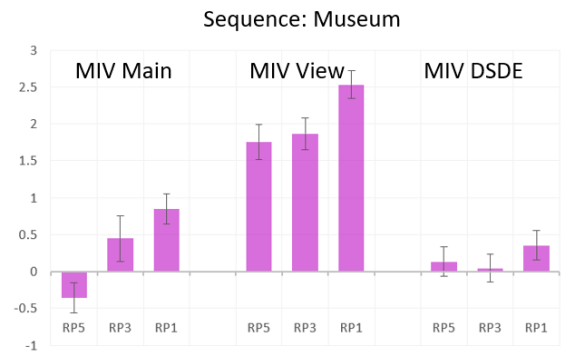


Figure 10. Subjective results for the Museum sequence in three different MIV configurations.

Subjective quality changes presented for the Museum sequence in Fig. 10 show that in almost every scenario there was a visible quality improvement. For 6 of 9 tests, the proposal allowed for achieving a statistically important quality improvement. For two tests (RP5 and RP3 in MIV DSDE configuration) the quality gain was also spotted, but it was not statistically important.

The proposal decreased the subjective quality in only one case – the heaviest compression in the MIV Main scenario. However, as presented in Fig. 12, for such a low bitrate the MIV Main cannot properly handle Museum sequence, and the quality of the content was unsatisfactory also before proposed modification.

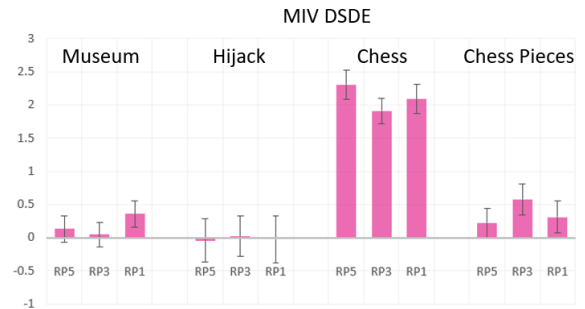


Figure 11. Subjective results for four sequences in MIV DSDE configuration.

Analysis of the results obtained for different sequences in the MIV DSDE configuration (Fig. 11) is similar to the results presented in Fig. 10. For 3 of 4 test sequences there is a quality gain induced by the proposed view selection modification. Moreover, for these sequences this gain is statistically important in 7 of 9 cases.

The only sequence for which the quality change did not occur is Hijack, but it should be noted that the proposal did not decrease the subjective quality.

b) Objective results

Objective quality results are gathered in Tables BDR and BDP, and presented as Bjøntegaard deltas [Bjo01]: BD-rate (Table 6) and BD-PSNR (Table 7).

Sequence		High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
		WS-PSNR	WS-PSNR	IV-PSNR	IV-PSNR
Museum	MIV	-3.4%	0.2%	0.9%	2.3%
	MIV View	-4.6%	-5.5%	-7.7%	-3.2%
Hijack	MIV DSDE	57.2%	32.5%	13.1%	10.2%
		---	---	---	---
		---	---	---	---
Chess		---	---	---	---
ChessPieces		---	---	---	---

Table 6. Objective metric (WS-PSNR and IV-PSNR) BD-rates obtained for 4 lowest rate points (Low-BR) and for 4 highest rate points (High-BR); “---” denotes, that the BD-rate calculation was not possible because of non-overlapping curves.

Sequence		High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
		WS-PSNR	WS-PSNR	IV-PSNR	IV-PSNR
Museum	MIV	0.4%	0.1%	-0.0%	-0.2%
	MIV View	0.2%	0.3%	0.8%	0.3%
Hijack	MIV DSDE	-1.6%	-1.3%	-0.8%	-0.6%
		-13.2%	-13.2%	-11.1%	-11.1%
		-6.4%	-6.2%	-5.9%	-5.5%
Chess		-11.3%	-11.0%	-8.3%	-7.9%
ChessPieces					

Table 7. Objective metric (WS-PSNR and IV-PSNR) BD-PSNRs obtained for 4 lowest rate points (Low-BR) and for 4 highest rate points (High-BR).

Surprisingly, presented objective results show, that in general the proposed method performs worse than the basic view selection algorithm implemented in TMIV 14. However, it has to be highlighted that results presented in Tables 6 and 7 were obtained by averaging the IV-PSNR and WS-PSNR values of all synthesized views, including the basic views. An example is shown in Table 8, where exact IV-PSNR values for all 10 views of sequence Chess are presented.

As presented in Table 8, the average IV-PSNR for non-base (i.e., “additional”) views is similar to the quality obtained for unmodified TMIV14. The only views with significant quality degradation are v0 and

v9 (i.e., views captured by cameras facing up and down), which are less important to the viewer.

View	IV-PSNR [dB]		
	TMIV 14	Proposed	delta
v0	57.00	36.17	-20.84
v1	37.84	37.99	0.15
v2	56.78	56.76	-0.02
v3	33.20	56.28	23.09
v4	56.73	56.44	-0.29
v5	56.01	55.93	-0.08
v6	38.85	56.11	17.26
v7	56.02	56.04	0.03
v8	57.08	57.18	0.10
v9	56.34	30.08	-26.26
Average (all views)			-0.69
Average (only non-base views)			-0.02

Table 8. IV-PSNR of synthesized views, RP1, similar bitrate for both approaches (21.3 Mbps for TMIV 14 and 20.8 Mbps for proposed); Chess sequence, MIV DSDE configuration.

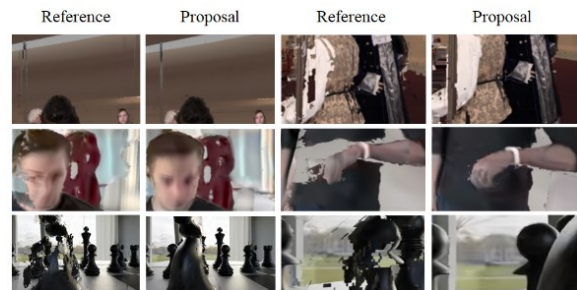


Figure 12. Visual comparison of posetraces generated with the use of original (reference) and proposed view selection method; sequences (from top): Hijack, Museum, and Chess.

5. CONCLUSIONS

This paper deals with problems on view selection for immersive video and its influence on the quality of final immersive vision on the decoder side.

Firstly, we have conducted an experiment to assess which views from multiview sequences should be selected into the virtual view synthesis process in order to obtain the best quality possible. Received results proved that the view selection algorithm should select views that are evenly distributed.

Moreover, in the paper we proposed an algorithm, which increases the subjective quality of virtual navigation by taking into account a non-uniform probability of choosing the viewing direction. We have noticed, that a typical user usually chooses to watch the scene in the horizontal plane, while the top and bottom parts of the omnidirectional scene are less important. This proposal was appreciated by the ISO/IEC MPEG VC experts, and is included in the reference software [MPEG22b] for the MPEG immersive video coding standard [ISO22].

The proposed approach is based on observations on the behavior of a typical user, without thorough

statistical analysis. Moreover, a correlation between optimal view selection and scene characteristics should be taken into consideration. Therefore, the topic of view selection for immersive video transmission will be studied further in our future research.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7. REFERENCES

- [Bjo01] Bjøntegaard, G. Calculation of average PSNR differences between RD986 curves. ISO/IEC JTC1/SC29/WG11 MPEG M15378, Austin, TX, 2001.
- [Bro21] Bross B. et al. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE T. on Circuits and Systems for Video Technology* 31 (10), pp. 3736-3764, 2021.
- [Boy21] Boyce J. et al. MPEG Immersive Video coding standard. *Proceedings of the IEEE* 109 (9), pp. 1521-1536, 2021.
- [Dor18] Doré R. Technicolor 3DoF+ Test Materials. Doc. ISO/IEC JTC1/SC29/WG11 MPEG/M42349, 2018.
- [Dzi18] Dziembowski A. et al. View selection for virtual view synthesis in free navigation systems. *International Conference on Signals and Electronic Systems* 2018, Kraków, Poland, 2018.
- [Dzi19] Dziembowski A. et al. Virtual view synthesis for 3DoF+ video. *Picture Coding Symposium, PCS* 2019, 2019.
- [Dzi22a] Dziembowski A. et al. Spatiotemporal redundancy removal in immersive video coding. *Journal of WSCG*, vol. 30, no. 1-2, pp. 54-62, 2022.
- [Dzi22b] Dziembowski A. et al. IV-PSNR – the objective quality metric for immersive video applications. *IEEE T. on Circuits and Systems for Video Technology* 32 (11), pp. 7575-7591, 2022.
- [Dzi22c] Dziembowski, A., Klóska, D., Jeong, J.Y., and Lee, G. [MIV] Inhomogeneous view selection for omnidirectional content. ISO/IEC JTC1/SC29/WG4 MPEG 140, M60668, 10.2022.
- [Fac18] Fachada S. et al. Depth image based view synthesis with multiple reference views for virtual reality. *3DTV-Conf*, 2018.
- [Fuj06] Fujii T. et al. Multipoint measuring system for video and sound – 100-camera and microphone system, ICME conference, 2006.
- [Ilo19] Ilola L. et al. New test content for immersive video – Nokia Chess. Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M50787, 2019.
- [Ilo20] Ilola L., Vadakital V.K.M. Improved NokiaChess sequence. Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M57382, 2020.
- [ISO22] Standard ISO/IEC FDIS 23090-12. Information technology – Coded representation of immersive media – Part 12: MPEG Immersive video. 2022.
- [ITU98] Methodology for the Subjective Assessment of the Quality of Television Pictures, document Rec. ITU-R BT.500-9, ITU-R, 1998.
- [ITU08] Subjective Video Quality Assessment Methods for Multimedia Applications, document Rec. ITU-T P.910, ITU-T, 2008.
- [Kov15] Kovacs, P. BBB light-field test sequences. ISO/IEC JTC1/SC29/WG11, M35721, 2015.
- [Kro18] Kroon B. 3DoF+ test sequence ClassroomVideo, ISO/IEC JTC1/SC29/WG11 MPEG M42415, 2018.
- [Mie22] Mieloch D. et al. Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video. *IEEE T. on Circ. and Systems for Video Tech.* 32 (9), pp. 6360-6374, 2022.
- [MPEG17] Requirements on 6DoF (v1), ISO/IEC JTC1/SC29/WG11 MPEG N17073, 2017.
- [MPEG19] MPEG Call for Proposals on 3DoF+ Visual, ISO/IEC JTC1/SC29/WG11 MPEG/N18145 2019.
- [MPEG22a] Test Model 14 for MPEG immersive video, ISO/IEC JTC1/SC29/WG04 MPEG VC N0242, 2022.
- [MPEG22b] Test Model 15 for MPEG immersive video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0271, 2022.
- [MPEG22c] Common test conditions for MPEG immersive video. ISO/IEC JTC1/SC29/WG04 MPEG VC, N0232, 2022.
- [Mul18] Müller K. et al. 3-D Video Representation Using Depth Maps. *Proceedings of the IEEE* 99 (4), pp. 643-656, 2011.
- [Sal18] Salahieh, B. et al. Kermit test sequence for Windowed 6DoF activities. ISO/IEC JTC1/SC29/WG11 MPEG M43748, Ljubljana, Slovenia, 2018.
- [Sta18] Stankiewicz O. et al. A free-viewpoint television system for horizontal virtual navigation. *IEEE Transactions on Multimedia* 20 (8), pp. 2182-2195, 2018.
- [Sta22] Stankowski J., Dziembowski A. Real-time CPU-based view synthesis for omnidirectional video. 30. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG* 2022, 2022.

- [Sul12] Sullivan G. et al. Overview of the High Efficiency Video Coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology 22 (12), pp. 1649-1668, 2012.
- [Sun17] Sun Y. et al. Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video. IEEE Signal Processing Letters 24 (9), pp. 1408-1412, 2017.
- [Tan12] Tanimoto M. et al. FTV for 3-D Spatial Communication. Proceedings of the IEEE, vol. 100, no. 4, pp. 905-917, 2012.
- [Tec16] Tech G. et al. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding. IEEE T. on Circuits and Systems for Video Technology 26 (1), pp. 35-49, 2016.
- [Vad22] Vadakital V.K.M. et al. The MPEG immersive video standard – current status and future outlook. IEEE MultiMedia 29 (3), 2022.
- [Wie19] Wien M. et al., Standardization Status of Immersive Video Coding, IEEE J. on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 5-17, 2019.

Point-to-Block Matching in Depth Estimation

Dawid Mieloch

dawid.mieloch@put.poznan.pl

Dominika Klóska

Magdalena Woźniak

Institute of Multimedia Telecommunications, Poznań University of Technology
Polanka 3, 61-131 Poznań, Poland

ABSTRACT

In this paper, the novel correspondence search method called point-to-block matching was proposed. Recently, in many proposed multimedia systems, depth estimation is performed on compressed input views. To address this problem and increase the quality of such depth maps, in the proposal, a point in a view is not compared simply with a point in another view, but to the most similar point in a small block surrounding it. The introduction of this method in the depth estimation process is beneficial to the quality of depth maps, as it decreases the influence of small shifts in images, caused e.g., by encoding-related errors introduced to input views. The method was implemented in one of the state-of-the-art depth estimation methods and tested in series of experiments. Based on the comparison of synthesized virtual views with the input views, the proposal increases the quality of estimated depth maps in most of the tested configurations.

Keywords

Depth maps, depth estimation, inter-view correspondence, similarity metric, virtual view synthesis.

1. INTRODUCTION

The methods of creating natural three-dimensional content are recently under constant development, as the emergence of new applications of immersive media systems such as free-viewpoint television [Tan12] and virtual reality systems can be easily seen [Dom17] [Laf16].

One of the most common representations of such multimedia content is the use of depth maps that correspond to each view captured by a camera system and show the distance from the acquired 3D point to the camera [Mül11]. Depth maps can be used to synthesize virtual views, so the user can freely move and see an acquired three-dimensional scene from any viewpoint.

In its essence, the depth estimation methods are based on the inter-view correspondence search. The result and efficiency of this search are dependent on a similarity of a point in one view of the scene to a point that most probably shows the same part of the scene in another view [Tip13]. Unfortunately, this search can be very difficult due to factors that we will summarize below.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Recently, in many proposed multimedia systems, the depth estimation is proposed to be performed using compressed input views, e.g., in the representation server that sends the requested virtual view to the user [Dzi16], or directly, in the client-side decoder of the multi-view image encoder [Gar19]. Therefore, the depth estimation methods should take into account that some encoding-related errors can be present in used videos.

The comparison of uncompressed and compressed input view can be seen in Figure 1, while Figure 2 shows the difference between uncompressed and compressed images, unveiling artifacts introduced by the high compression. The main differences are clearly visible in highly-textured areas and all edges visible in the scene.

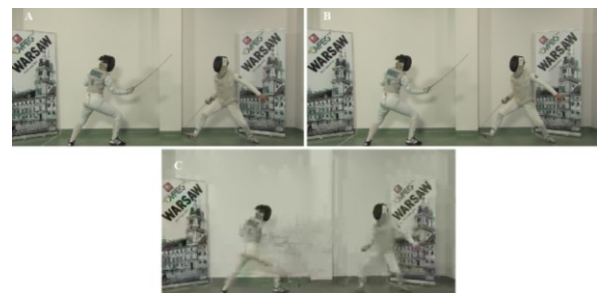


Figure 1. Input view of Fencing sequence [Dom16] without compression (A) and with high (B) / low (C) compression.

The presence of high-frequency harmonics is necessary for the correct representation of the edges in the image, unfortunately, due to coding, when the

transform coefficients are quantized, high-frequency components are notched out, which in turn leads to blurring or, in the worst case, an edge shift in the image. Such shift can also occur in highly compressed videos because of the high share of temporally predicted parts of videos [Leo07].



Figure 2. The logarithm of difference between compressed and uncompressed input view of Fencing sequence for high compression. The contrast of the image was increased to increase the visibility of differences.

Depth maps are of course most often estimated before any compression of the representation of a scene. In this case, the input views are usually uncompressed, so the compression-related distortions do not influence the quality of depth maps. On the other hand, this quality can be decreased due to low accuracy of camera parameters (understood as the accuracy of camera distortion models [Tan17] and intrinsic and extrinsic camera parameters [San17] [Che21]), and the use of floating-point representation [God20] and rounding errors.

Other causes of depth estimation errors are related to inter-view correspondence search can be seen also in noise present in input views of natural sequences [Sta14], as such noise influences the original color of points.

In order to address all listed problems, we propose the novel correspondence search method called point-to-block matching. In the proposal, a point in a view is not compared simply with a collocated point in another view, but to the most similar point in a small block surrounding it.

The introduction of such a degree of freedom in the estimation can be very beneficial on the quality of depth maps, as it decreases the influence of small shifts in images. Moreover, the proposal also can decrease the noise-induced errors, as the proposed method lets us find a less noised similar point in the nearest neighborhood.

To fully test the proposal, the point to block matching was implemented in one of the publicly available state-of-the-art methods of depth estimation [Mie20a] and tested in 10 different configurations to find the most proper applications in which the proposed matching should be used.

The organization of the paper is as follows: Section 2 shows the description of the most relevant similarity measurement methods, with emphasis on their usefulness in the case of use of compressed views in the depth estimation process; Section 3 describes in details the proposed inter-view matching method; Section 4 includes the results of the comparison of the proposed point-to-block matching with the traditional point-to-point matching.

In the end, Section 5 summarizes the paper and includes conclusions drawn from the performed experiments.

2. RELATED WORKS

Some studies were already performed in order to evaluate the efficiency of the existing matching or similarity measurement methods in various cases, e.g., for noised views. Study [Weg09] has shown that a mixture of census or rank transforms with the simple sum of absolute differences (SAD) or sum of squared differences (SSD) can be very efficient for such multi-view sequences. [Zeg20] proves that the use of structural similarity performs very well when radiometric distortions are present in processed input views.

Authors of [Say15] indicate that similarity measurement should be adapted to the local features of input views. It can be used to change the weight of used mixture similarity metrics and lead to better estimation in problematic areas, such as low-texture regions or occluded parts of views. Another method that tries to adapt to processed views is [Cha18], but in this proposal, the size and shape of the matching window become dependent on the characteristics of the part of the image.

Other methods were proposed to improve the efficiency of inter-view matching by introducing temporal information into the similarity metric [Shi17]. The temporal stability of depth maps is crucial for the quality of the final view synthesis. On the other hand, such stability can be ensured on other stages of the depth estimation process, e.g., by post-processing filtering [Köp13].

An interesting method is proposed by [Suo12]. Its authors propose to perform the matching of views in the transform domain. Such an approach results in exposure-independent estimation, very valuable when multi-view systems are used. However, if the encoding of input views would be performed as simulcast compression (independent in each view, e.g., performed in each camera as post-processing step), the distribution of frequencies could significantly vary in compressed views, negatively influencing the efficiency of transform domain matching.

Matching methods base on minimum spanning tree, such as [Zha19], show high robustness for complex scenes, however, no qualitative results of efficiency for noised or compressed images are provided. This method uses also SURF [Bay08] and KAZE [Alc12] features detectors, which are translation invariant, so they could be also invariant to compression-induced errors, however, these detectors are used only for calculation of disparity range, not depth maps themselves.

To the best knowledge of the authors, contrary to the proposal presented in Section 3, none state-of-the-art method of inter-view correspondence search takes specifically into account the compression-induced artifacts present in the encoded input views.

3. PROPOSED INTER-VIEW MATCHING METHOD

The problem of the depth map estimation can be in most cases presented as a cost (goal) function minimization [Kol02]. In its most commonly used form, the cost function is defined as:

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

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

The proposed method changes the data term D_p , which is responsible for the correspondence between points in different views. D_p usually uses some similarity metric (e.g., one of metrics described in Section 2) between a point p and a point p' , which is a point in neighboring view that corresponds to the point p for the considered depth of a point ($\bar{d}_p$).

In the proposal, the data term compares the point p with points in a small block of points R that surround point p' and chooses the point that gives the lowest matching error (highest similarity) to point p . So, if the SAD metric would be used, then:

$$D_p(\bar{d}_p) = \min(SAD(p, r)), \quad r \in R. \quad (2)$$

Therefore, if 3×3 block would be used as R , the $D_p(\bar{d}_p)$ would be equal to the smallest of 9 SAD values, calculated for the central point p' and its 1- and 2- pixel neighborhood, respectively.

Note that any similarity measure can be used with the proposal. Therefore, such a method can enhance

almost any depth estimation method with robustness to small shifts in images caused e.g., by image compression and decrease the noise-induced errors.

The drawback of the proposal can be seen in the small increase of the computational complexity, as the similarity is now performed not once for each point, but R^2 times, nevertheless, the calculation of similarity can be easily parallelized.

As the proposal was implemented in the depth estimation software provided in [Mie20a], the Graph Cut method [Kol02] was used to minimize the function (1).

4. EXPERIMENTAL RESULTS

The performed experiments were performed for different configurations of the modified depth estimation software [Mie20a]:

1. Standard parameters:
 - 256 levels of depth,
 - 50,000 superpixels per view,
 - uncompressed input views,
 - all views used in the estimation.
2. Decreased number of depth levels:
 - 2.1. 16 levels of depth,
 - 2.2. 128 levels of depth.
3. Changed number of superpixels [Ach12]:
 - 3.1. 5000 superpixels per view,
 - 3.2. 100000 superpixels per view.
4. Compression of input views:
 - 4.1. Low compression,
 - 4.2. Medium compression,
 - 4.3. High compression.
5. Decreased number of used cameras:
 - 5.1. 70% of cameras,
 - 5.2. 20% of cameras.

All configurations were tested 3 times: for the unmodified depth estimation and the point-to-point matching performed in 3×3 and 5×5 blocks. The use of larger blocks is possible, but as preliminary tests have shown, does not have a large influence on the estimated depth maps.

The quality of estimated depth maps directly affects the final product, which is the most faithful representation of the three-dimensional scene. Nevertheless, in the case of using depth maps for the view synthesis purposes, such faithfulness is understood more as the ability to provide virtual views of high quality, rather than the low absolute error in comparison with ground-truth depth [Fan16]. Moreover, such ground-truth depth maps are not available for natural multi-view sequences, making such comparison not possible.

Therefore, in order to test the proposal, a set of estimated depth maps was used for the synthesis of

virtual views which were placed in the same position as the real cameras that captured a scene. The synthesized views were in the end compared with the original input views. The objective comparison was performed using IV-PSNR [MPEG20], which is the PSNR-based metric used by the ISO/IEC MPEG Video Coding to evaluate immersive video quality. The calculated IV-PSNR was averaged for all views in each test sequence (listed in Table 1).

The experiments were performed for 5 multiview test sequences that vary in the number of input views and their arrangements.

Sequence name	Number of views	Resolution	Source
Carpark	9	1920×1088	[Mie20b]
Fencing	10	1920×1080	[Dom16]
Frog	13	1920×1080	[Sal18]
Painter	16	2048×1088	[Doy18]
Street	9	1920×1088	[Mie20b]

Table 1. Test sequences used in experiments.

The virtual view synthesis software used in the experiments is VVS 2.0 with its default configuration [Boi19], i.e., the synthesis of each virtual view was performed using 4 nearest views (and corresponding depth maps).

The software used for compression of input views is x265 [X265] which is the implementation of the MPEG HEVC encoder. The values of the quality parameter *crf* were equal to 0 (low compression, near-lossless), 26 (medium compression), and 51 (high compression).

The following five subsections include the description of acquired results for each test sequence. Color green in tables denotes that the quality of the proposal (point-to-block matching) was higher than for unmodified depth estimation. The last subsection contains a summary of performed experiments.

Carpark

Table 2. shows the average IV-PSNR of synthesized views for the Carpark sequence.

For 8 out of 10 the point-to-block matching 3×3 provides a gain in the quality, making this option the best for this sequence. On the other hand, the highest gain (more than 3 dB) can be seen for the number of superpixels decreased to 5000 for 5×5 block.

Fencing

For the Fencing sequence, the results, presented in Table 3, show that the proposed matching provides a gain in the quality mainly for the compressed input views. This sequence is the only one with a non-planar camera arrangement, which can suggest that in such a

case, the use of point-to-block matching is not beneficial when uncompressed input views are used.

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	37.09	37.19	37.11
16 levels of depth	36.09	36.13	36.13
128 levels of depth	37.07	37.20	37.16
5000 superpixels	32.28	32.39	35.55
100000 superpixels	38.28	38.35	38.21
Low compression	37.10	37.23	37.14
Medium compression	37.36	37.28	37.18
High compression	34.08	34.17	34.20
70% of cameras	36.19	36.21	35.95
20% of cameras	35.21	35.17	35.21

Table 2. Results of experiments for the Carpark sequence.

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	40.24	40.13	39.90
16 levels of depth	35.40	35.55	35.72
128 levels of depth	40.56	40.23	40.05
5000 superpixels	37.87	37.77	37.71
100000 superpixels	40.89	40.87	40.55
Low compression	40.49	40.10	39.98
Medium compression	38.19	39.03	39.14
High compression	34.09	36.19	36.18
70% of cameras	33.80	39.65	39.06
20% of cameras	38.41	38.40	38.43

Table 3. Results of experiments for the Fencing sequence.

Frog

In the Frog sequence, the best results were obtained for Point-to-block matching in 3×3 block (Table 4).

One of the possible explanations to results observed for Frog can be a very high amount of noise present in this sequence. Table 5 shows the standard deviation of noise estimated for each sequence (from [Dzi20]). As it can be seen, Frog has more than twice the amount of noise than other sequences. Therefore, in configurations without the compression of input views, the point-to-block matching provides a gain in most cases, as it decreases the influence of noise on the inter-view matching.

On the other hand, compression of video highly reduces the noise of sequences, even for low compression. In this sequence, objects have very detailed, rich textures (see Figure 3), so the changes in the position of edges in compressed views do not

occur so often as e.g., in the Fencing sequence, where the foreground and background objects have similar color characteristics. We can see that a high gain in the quality in Frog cannot be seen until high compression is applied.

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	38.95	39.00	38.54
16 levels of depth	32.63	32.97	33.05
128 levels of depth	38.99	39.01	38.66
5000 superpixels	37.27	36.97	36.53
100000 superpixels	39.07	39.16	38.62
Low compression	39.04	38.98	38.47
Medium compression	38.99	38.92	38.33
High compression	33.90	34.26	33.88
70% of cameras	38.05	37.89	37.58
20% of cameras	33.28	33.31	33.08

Table 4. Results of experiments for the Frog sequence.

Sequence	σ_Y	σ_U	σ_V
Carpark	10.29	4.23	4.57
Fencing	12.83	3.38	3.83
Frog	26.77	5.59	5.62
Painter	10.17	3.89	3.89
Street	9.85	3.84	4.21

Table 5. Estimated noise standard deviation for luma (σ_Y) and chromas (σ_U and σ_V) [Dzi20]. Evaluated sequences have 10 bits per sample.

Painter

Painter is the only sequence that shows decreased quality in almost all cases of estimation with the point-to-block matching (Table 6). This sequence is the only sequence with lightfield-like arrangement of cameras (matrix of cameras).

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	40.55	40.18	39.57
16 levels of depth	39.04	39.12	39.13
128 levels of depth	40.63	40.36	39.73
5000 superpixels	39.51	39.14	38.70
100000 superpixels	40.64	40.24	39.47
Low compression	40.68	40.17	39.86
Medium compression	39.87	39.66	39.04
High compression	36.32	35.85	35.58
70% of cameras	40.22	39.84	39.66
20% of cameras	40.31	40.12	39.70

Table 6. Results of experiments for the Painter sequence.

Street

Results for the Street sequence (Table 7) clearly benefit from using the proposed point-to-block matching for all tested configurations of depth estimation.

The gain of quality is very high in most cases (up to 3 dB). As it can be seen in Figure 4, most of the differences between synthesis that used depth maps with point-to-point matching and point-to-block matching are still visible mainly on the edges of objects.

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	37.34	40.23	40.09
16 levels of depth	36.14	39.09	39.03
128 levels of depth	37.23	40.23	39.98
5000 superpixels	31.99	32.75	32.83
100000 superpixels	38.34	40.43	39.95
Low compression	37.37	40.35	40.06
Medium compression	40.24	40.08	40.53
High compression	32.52	34.59	34.49
70% of cameras	35.34	37.99	38.04
20% of cameras	38.97	38.32	40.10

Table 7. Results of experiments for the Street sequence.

Summary of experiments

In order to summarize the performed experiments, the results were averaged for all sequences and presented in Table 8.

Depth estimation configuration	Average IV-PSNR [dB] of synthesized view		
	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	38.83	39.35	39.04
16 levels of depth	35.86	36.57	36.61
128 levels of depth	38.90	39.41	39.12
5000 superpixels	35.78	35.80	36.26
100000 superpixels	39.44	39.81	39.36
Low compression	38.94	39.37	39.10
Medium compression	38.93	38.99	38.84
High compression	34.18	35.01	34.87
70% of cameras	36.72	38.32	38.06
20% of cameras	37.24	37.06	37.30

Table 8. Results of experiments averaged for all sequences.

Although the results vary for different sequences, as was presented in previous subsections, these results show that for the used set of test sequences the point-to-block matching in 3×3 block was on average better than standard point-to-point matching in all but one



Figure 3. Comparison of virtual views synthesized using depth maps estimated using point-to-point matching (left column), point-to-block matching with 3×3 block (center), and with 5×5 block (right). Test sequences (top to bottom): Painter, Frog, Fencing, Carpark, Street.

tested configurations. Most significant gains can be observed for the reduced number of cameras, high compression, and levels of depth reduced from 256 to 16.

5. CONCLUSIONS

This paper presents the novel method of the inter-view correspondence search called point-to-block matching. In the proposal, a point in a view is not compared simply with a point in another view, but

with the most similar point in a small block surrounding it.

Series of performed experiments based on the comparison of synthesized virtual views with the input views showed that on average the proposal increases the quality of estimated depth maps in almost all configurations. The results for individual sequences seem to confirm that the method provides the best results for highly compressed input views or when the

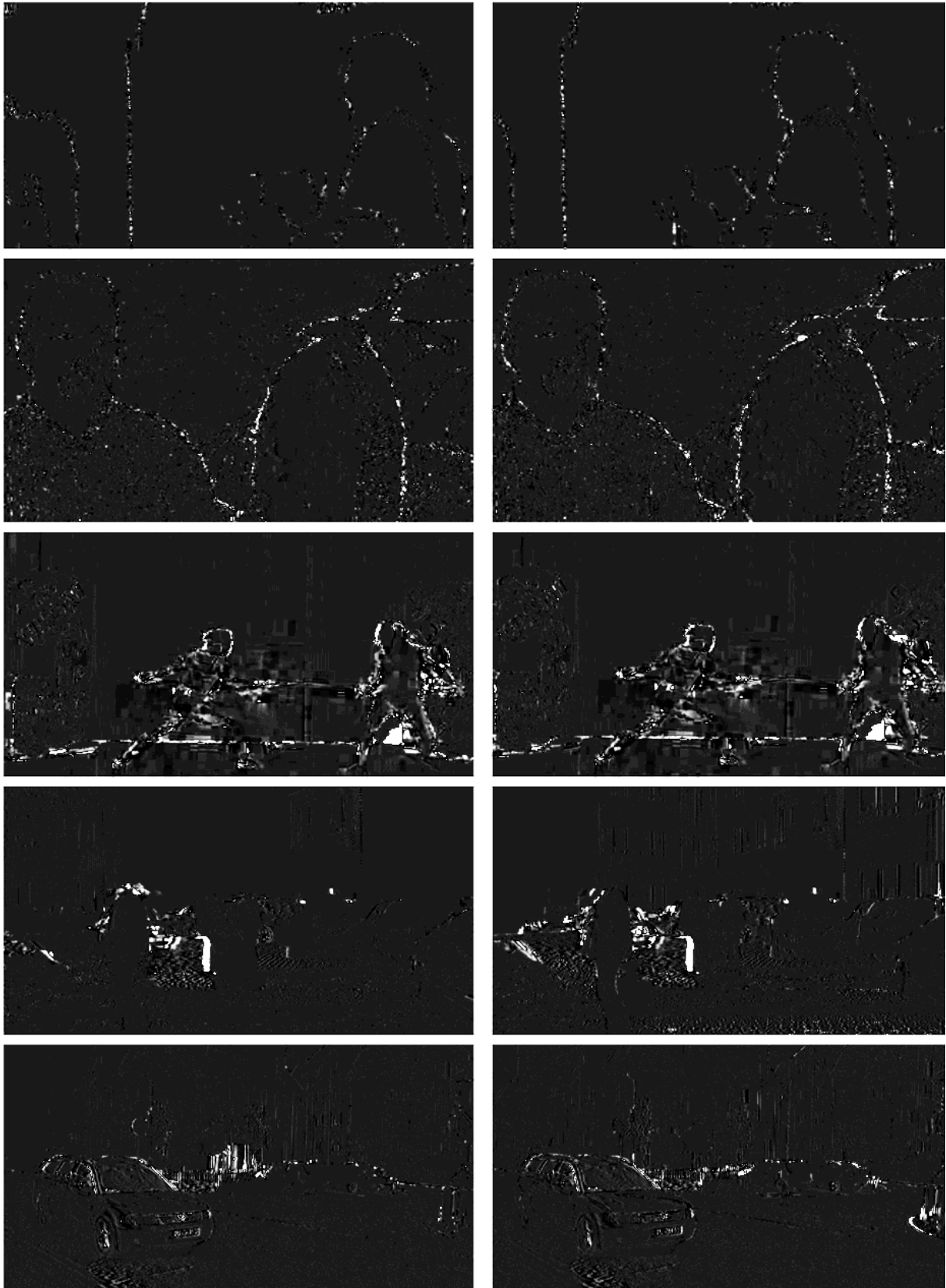


Figure 4. Logarithm of differences between virtual views synthesized using depth maps estimated using point-to-point matching and point-to-block matching with 3×3 block (left) / 5×5 block (right). Test sequences (top to bottom): Painter, Frog, Fencing, Carpark, Street. The contrast of the image was increased to increase the visibility of differences.

amount of noise present in a multi-view sequence is significant.

A gain in quality was also observed for all sequences in cases of decreased number of depth levels tested during the estimation process. Such estimation is much less computationally expensive, which indicates that the method could also be very effective in visual systems with limited resources (e.g., a typical PC with two internet cameras). Such systems almost always operate using compressed input video streams, therefore, the abovementioned efficiency of the proposal in such cases further increases its usability in such real-life configuration.

The need for further research on depth estimation methods adapted to be used with compressed input views can be seen also in the current state of the development of the new MPEG Immersive Video codec [Boy21] for virtual and augmented reality applications. This soon-to-be video encoding standard includes Geometry Absent profile [MPEG21] which can be used to send a subset of input views to the decoder, where the depth estimation process is performed to synthesize any viewport of the three-dimensional scene.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7. REFERENCES

- [Ach12] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., and Süsstrunk S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. 2012 IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012.
- [Alc12] Alcantarilla P. F., Bartoli A., and Davison A. J. KAZE features. Proceedings of European Conference on Computer Vision, pp. 214-227, 2012.
- [Bay08] Bay H., Ess A., Tuytelaars T., and Van Gool T. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, vol. 110, Issue 3, pp. 346-359, 2008.
- [Boi19] Boissonade P., and Jung J. [MPEG-I Visual] Improvement of VVS1.0.1. ISO/IEC JTC1/SC29/WG11/MPEG2019/m46263, Marrakesh, Jan. 2019.
- [Boy21] Boyce J., Doré R., Dziembowski A., Fleureau J., Jung J., Kroon B., Salahieh B., Vadakital V., and Yu L. MPEG Immersive Video Coding Standard. Proceedings of the IEEE, early access. 2021.
- [Cha18] Chai Y. and Cao X. Stereo Matching Algorithm Based on Joint Matching Cost and Adaptive Window. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, pp. 442-446, 2018.
- [Che21] Chen R., Chen F., Xu G., et al. Precision analysis model and experimentation of vision reconstruction with two cameras and 3D orientation reference. Scientific Reports, No. 11, 3875, 2021.
- [Dom16] Domański M., Dziembowski A., Grzelka A., Mieloch D., Stankiewicz O., and Wegner K. Multiview test video sequences for free navigation exploration obtained using pairs of cameras. Document ISO/IEC JTC1/SC29/WG11, MPEG M38247, 2016.
- [Dom17] Domański M., Stankiewicz O., Wegner K. and Grajek T. Immersive visual media – MPEG-I: 360 video, virtual navigation and beyond. 2017 IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), Poznań, pp. 1-9, 2017.
- [Doy18] Doyen D., Boisson G., and Gendrot R. [MPEG-I Visual] New Version of the Pseudo-Rectified Technicolorpainter Content. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43366, Ljubljana, 2018.
- [Dzi16] Dziembowski A., Domański M., Grzelka A., Mieloch D., Stankowski J., and Wegner K. The influence of a lossy compression on the quality of estimated depth maps. 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, pp. 1-4, 2016.
- [Dzi20] Dziembowski A., Mieloch D., Stankiewicz O., Nikonowicz J., Domański M. Noise modeling in TMIV. Document ISO/IEC JTC1/SC29/WG11 MPEG/M54895, Online, June 2020.
- [Fan16] Fang L., Xiang Y., Cheung N. M., and Wu F. Estimation of virtual view synthesis distortion toward virtual view position. 2016 IEEE Transactions on Image Processing, vol. 25, no. 5, pp. 1961-1976, May 2016.
- [Gar19] Garus P., Jung J., Maugey T., and Guillemot C. Bypassing Depth Maps Transmission for Immersive Video Coding. 2019 Picture Coding Symposium (PCS), Ningbo, pp. 1-5, 2019.
- [God20] Godbolt M. Optimizations in C++ compilers. Communications of the ACM, vol. 63, no. 2 pp. 41-49, 2020.
- [Kol02] Kolmogorov V., and Zabih R. Multi-camera Scene Reconstruction via Graph Cuts. Proceedings of the 7th European Conference on Computer

- Vision-Part III (ECCV '02), London, pp. 82-96, 2002.
- [Köp13] Köppel M., Makhlouf M. B., Müller M., and Ndjiki-Nya P. Temporally consistent adaptive depth map preprocessing for view synthesis. 2013 Visual Communications and Image Processing (VCIP), Kuching, Malaysia, pp. 1-6, 2013.
- [Laf16] Lafruit G., Domański M., Wegner K., Grajek T., Senoh T., Jung J., Kovács P., Goorts P., Jorissen L., Munteanu A., Ceulemans B., Carballeira P., García S., and Tanimoto M. New visual coding exploration in MPEG: Super-MultiView and Free Navigation in Free viewpoint TV. 2016 Proceedings of the Electronic Imaging Conference: Stereoscopic Displays and Applications, San Francisco, pp. 1-9, 2016.
- [Leo07] Leontaris A., Cosman P. C., and Reibman A. R. Quality Evaluation of Motion-Compensated Edge Artifacts in Compressed Video. 2007 IEEE Transactions on Image Processing, vol. 16, no. 4, pp. 943-956, Apr. 2007.
- [Mie20a] Mieloch D., Stankiewicz O., and Domański M. Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation. 2020 IEEE Access, vol. 8, pp. 5760-5776, 2020.
- [Mie20b] Mieloch D., Dziembowski A., and Domański M. [MPEG-I Visual] Natural Outdoor Test Sequences. Document ISO/IEC JTC1/SC29/WG11 MPEG/M51598, Brussels, Jan. 2020.
- [MPEG20] Software manual of IV-PSNR for Immersive Video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0013, Online, Oct. 2020.
- [MPEG21] Test Model 8 for MPEG Immersive Video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0050, Online, Jan. 2021.
- [Mül11] Müller K., Merkle P., and Wiegand T. 3-D Video Representation Using Depth Maps. 2011 Proceedings of the IEEE, vol. 99, no. 4, pp. 643-656, 2011.
- [Sal18] Salahieh B., Marvar B., Nentedem M., Kumar A., Popvic V., Seshadrinathan K., Nestares O. and Boyce J. Kermit test sequence for Windowed 6DoF Activities. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43748, Ljubljana, Slovenia, Jul. 2018.
- [San17] Sankowski W., Włodarczyk M., Kacperski D., and Grabowski K. Estimation of measurement uncertainty in stereo vision system, Image and Vision Computing, vol 61, pp. 70-81, 2017.
- [Say15] Saygili G., Van der Maaten L., Hendriks E. A. Adaptive stereo similarity fusion using confidence measures. Computer Vision and Image Understanding, vol. 135, pp. 95-108, 2015.
- [Shi17] Shin Y., and Yoon K. Adaptive spatiotemporal similarity measure for a consistent depth maps. 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta, pp. 1-4, 2017.
- [Sta14] Stankiewicz O., Domański M., and Wegner K. Analysis of noise in multi-camera systems. 3DTV Conference 2014, Budapest, 2014.
- [Suo12] Suominen O., Gotchev A., and Hannuksela M. Transform domain similarity measures in stereo matching. 2012 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), Zurich, pp. 1-4, 2012.
- [Tan12] Tanimoto M., Tehrani M. P., Fujii T., and Yendo T. FTV for 3-D Spatial Communication. 2012 Proceedings of the IEEE, vol. 100, no. 4, pp. 905-917, 2012.
- [Tan17] Tang Z., Grompone von Gioi R., Monasse P., and Morel J. A Precision Analysis of Camera Distortion Models. IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2694-2704, 2017.
- [Tip13] Tippetts B., Jye Lee D., Lillywhite K., and Archibald J. Review of stereo vision algorithms and their suitability for resource-limited systems. Journal of Real-Time Image Processing, vol. 11, no. 1, pp. 5-25, 2013.
- [Weg09] Wegner K., and Stankiewicz O. Similarity measures for depth estimation. 2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, pp. 1-4, 2009.
- [X265] MulticoreWare Inc. x265 HEVC Encoder/H.265 Video Codec. Available on <http://x265.org>
- [Zeg20] Zeglazi O, Rziza M, Amine A, Demonceaux C. Structural Similarity Measurement Based Cost Function for Stereo Matching of Automotive Applications. Journal of Imaging, vol. 6, no. 8, 2020.
- [Zha19] Zhang C., He C., Chen Z., Liu W., Li M. and Wu J. Edge-Preserving Stereo Matching Using Minimum Spanning Tree. IEEE Access, vol. 7, pp. 177909-177921, 2019.