

---

# 3D-LMVIC: Learning-based Multi-View Image Compression with 3D Gaussian Geometric Priors

---

Yujun Huang<sup>\*1</sup> Bin Chen<sup>\*2</sup> Niu Lian<sup>2</sup> Xin Wang<sup>1</sup> Baoyi An<sup>3</sup> Tao Dai<sup>4</sup> Shu-Tao Xia<sup>1</sup>

## Abstract

Existing multi-view image compression methods often rely on 2D projection-based similarities between views to estimate disparities. While effective for small disparities, such as those in stereo images, these methods struggle with the more complex disparities encountered in wide-baseline multi-camera systems, commonly found in virtual reality and autonomous driving applications. To address this limitation, we propose 3D-LMVIC, a novel learning-based multi-view image compression framework that leverages 3D Gaussian Splatting to derive geometric priors for accurate disparity estimation. Furthermore, we introduce a depth map compression model to minimize geometric redundancy across views, along with a multi-view sequence ordering strategy based on a defined distance measure between views to enhance correlations between adjacent views. Experimental results demonstrate that 3D-LMVIC achieves superior performance compared to both traditional and learning-based methods. Additionally, it significantly improves disparity estimation accuracy over existing two-view approaches.

## 1. Introduction

The rapid advancement of 3D applications has led to an explosion of multi-view image data across various fields, including virtual reality (VR) (Anthes et al., 2016), augmented reality (AR) (Schmalstieg & Hollerer, 2016), visual simultaneous localization and mapping (vSLAM) (Mokssit et al., 2023), 3D scene understanding (Dai et al., 2017), au-

<sup>\*</sup>Equal contribution <sup>1</sup>Shenzhen International Graduate School, Tsinghua University, Shenzhen, China <sup>2</sup>School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China <sup>3</sup>Huawei Technologies Company Ltd., Shenzhen, China <sup>4</sup>Department of Software Engineering, Shenzhen University, Shenzhen, China. Correspondence to: Bin Chen <chenbin2021@hit.edu.cn>.

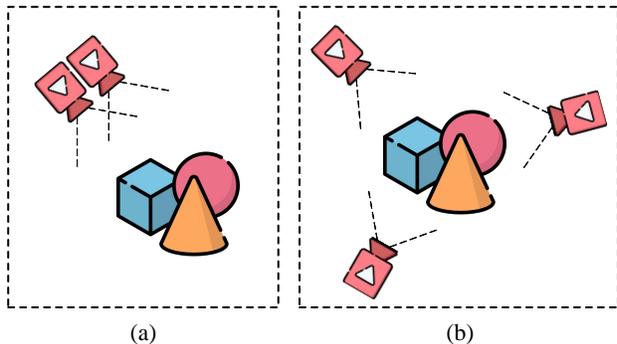


Figure 1. Illustrations of camera systems. (a) A stereo camera configuration. (b) A wide-baseline multi-camera configuration.

tonomous driving (Chen et al., 2017), and medical imaging (Hosseinian & Arefi, 2015). In particular, applications like VR and AR, which rely on high-quality multi-view visual content to create immersive experiences, generate a massive volume of data that poses significant challenges for storage and transmission. This makes the development of efficient compression techniques crucial for managing the increasing data demands in these fields.

As illustrated in Figure 1, unlike the commonly studied stereo camera systems, 3D applications often rely on wide-baseline multi-camera systems to capture global scene information (Xu et al., 2020; Yan et al., 2024). In such scenarios, the spatial positions and viewing angles of cameras differ significantly compared to stereo setups, resulting in large disparities between images captured from different views. Existing disparity estimation methods typically rely on finding similar local regions in the image domain to estimate disparities. However, this approach faces significant challenges when dealing with complex and large disparities.

Current multi-view coding standards, such as H.264-based MVC (Vetro et al., 2011) and H.265-based MV-HEVC (Hanuksela et al., 2015), have been developed to compress multi-view media by extending their respective base standards and exploiting redundancies across multiple views. These standards employ disparity estimation to calculate positional differences of objects between views, aiding in the prediction of pixel values. However, these methods rely on manually designed modules, limiting the system’s ability to fully leverage end-to-end optimization.

Learning-based single image compression has seen remarkable advancements (Ballé et al., 2017; 2018; Minnen et al., 2018), inspiring extensions of these methods to multi-view image coding (Deng et al., 2021; Lei et al., 2022; Zhang et al., 2023; Liu et al., 2024). A central challenge in these extensions lies in the accurate estimation of disparities across different views. For example, Deng et al. (2021; 2023) employ a simple 3x3 homography matrix for disparity estimation, which, while efficient, struggles with complex scene disparities. Alternatively, Ayzik & Avidan (2020); Huang et al. (2023) utilize patch matching method to align the reference view with the target view. This approach is effective for horizontal or vertical view shifts but falls short when addressing non-rigid deformations caused by view rotations. Similarly, Zhai et al. (2022) assume that disparity occurs only along the horizontal axis in their stereo matching method, which suffices for stereo images but is inadequate for more complex view transformations where disparity is not limited to the horizontal axis. Some methods leverage cross-attention mechanisms for implicit alignment (Wödlinger et al., 2022; Zhang et al., 2023; Liu et al., 2024). For instance, Zhang et al. (2023) enhance the target view’s representation by multiplying its query with the reference view’s key and value, effectively incorporating reference view features into the target view. However, these methods primarily establish correlations between two views by 2D projection similarities, without considering the 3D spatial relationships between the views and the captured objects.

Building on prior investigation, we propose a novel learning-based multi-view image compression framework with 3D Gaussian geometric priors (3D-LMVIC), which employs 3D-GS as a geometric prior to guide disparity estimation between views. Specifically, 3D-GS generates a depth map for each view, providing precise spatial information at the pixel level. This enables accurate correspondence between views, allowing the compression model to effectively fuse features from reference views. Due to positional and angular disparities between views, images generally do not fully overlap, and merging non-overlapping regions may introduce noise. To address this, we design a mask based on the 3D Gaussian geometric prior to identify overlapping regions, ensuring more accurate feature fusion. Additionally, since depth maps are required during decoding, we propose a depth map compression model to efficiently reduce geometric redundancy across views, incorporating a cross-view depth prediction module to capture inter-view geometric correlations. Finally, recognizing the importance of field of view (FoV) overlap in redundancy reduction, we introduce a multi-view sequence ordering method to address the issue of low overlap between adjacent views in unordered sequences. This method defines and proves a distance measure between view pairs to guide the ordering of view sequences.

- We propose a learning-based multi-view image compression framework with 3D Gaussian geometric priors (3D-LMVIC), which utilizes 3D Gaussian geometric priors for precise disparity estimation between views, thereby enhancing multi-view image compression efficiency. Additionally, we design a mask based on these priors to identify overlapping regions between views, effectively guiding the model to retain useful cross-view information.
- We also present a depth map compression model aimed at reducing geometric redundancy across views. Additionally, we define and prove a distance measure between views, upon which a multi-view sequence ordering method is proposed to improve the correlation between adjacent views.
- Experimental results show that our framework surpasses both traditional and learning-based multi-view image coding methods in compression efficiency. Moreover, our disparity estimation method demonstrates greater visual accuracy compared to existing two-view disparity estimation methods.

## 2. Related Works

**Single Image Coding.** Traditional image codecs, such as JPEG (Wallace, 1992), BPG (Bellard, 2014), and VVC (Bross et al., 2021), employ manually designed modules like DCT, block-based coding, and quadtree plus binary tree partitioning to balance compression and visual quality. These methods, however, do not achieve end-to-end joint optimization, limiting their performance.

In recent years, learning-based image compression methods have integrated autoencoders with differentiable entropy models to enable end-to-end optimization of rate-distortion loss. Early works, such as Ballé et al. (2017; 2018), introduced generalized divisive normalization (GDN) (Ballé et al., 2016) and proposed factorized and hyperprior entropy models. Subsequent research (Minnen et al., 2018; He et al., 2021; Jiang et al., 2023) incorporated autoregressive structures into entropy models, resulting in more accurate probability predictions. These advancements have laid the foundation for learning-based multi-view image coding.

**Multi-view Image Coding.** Traditional multi-view image codecs, such as MVC (Vetro et al., 2011) and MV-HEVC (Hannuksela et al., 2015), extend H.264 and H.265, respectively, by incorporating inter-view correlation modeling to eliminate redundant information between different views. However, these modules are manually designed, potentially limiting their ability to fully exploit cross-view information.

Learning-based multi-view image coding primarily focuses on stereo image coding (Deng et al., 2021; Lei et al., 2022;

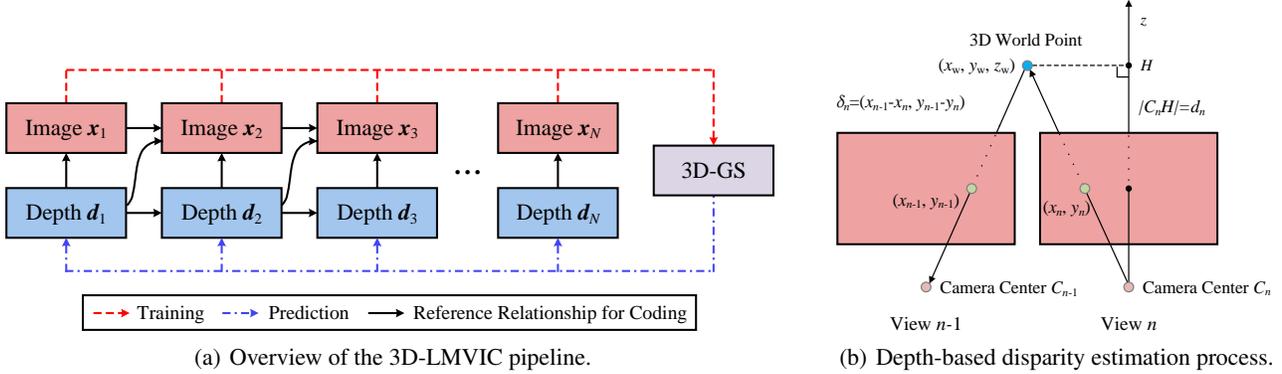


Figure 2. Overall Pipeline.

Wödlinger et al., 2022; Zhai et al., 2022; Deng et al., 2023; Liu et al., 2024) and distributed image coding (Ayzik & Avdan, 2020; Huang et al., 2023; Zhang et al., 2023). These methods either rely on finding explicit pixel coordinate correspondences between views or use attention-based implicit correspondence modeling to capture inter-view correlations. However, they model inter-view correlations based solely on two-dimensional view images, which may not fully reflect the correspondences in the original three-dimensional space.

**3D Gaussian Splatting.** 3D Gaussian Splatting (Kerbl et al., 2023; Hamdi et al., 2024) introduces a differentiable point-based rendering technique that represents 3D points as Gaussian functions (mean, variance, opacity, color) and projects these 3D Gaussians onto a view to form an image. This differentiable point-based rendering function allows for the backward update of the attributes of the 3D Gaussians, ensuring that their geometrical and textural properties match the original 3D scene. This approach inspired us to utilize 3D Gaussian Splatting to obtain geometric priors of the original 3D scene, aiding in the task of multi-view image compression.

### 3. Proposed Method

Figure 2(a) shows the overall pipeline of 3D-LMVIC. Given a set of multi-view image sequences  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\}$ , a 3D-GS is trained to estimate depth map  $\mathbf{d}_n$  for each image  $\mathbf{x}_n$ . Both  $\mathbf{x}_n$  and  $\mathbf{d}_n$  are compressed, with the coding reference relationships indicated by black solid arrows in the figure. Prior to compressing the image  $\mathbf{x}_n$ , it is necessary to compress  $\mathbf{x}_{n-1}$ ,  $\mathbf{d}_{n-1}$ , and  $\mathbf{d}_n$ . The disparity relationship between the  $(n-1)$ -th and  $n$ -th views is inferred from the reconstructed depth maps  $\hat{\mathbf{d}}_{n-1}$  and  $\hat{\mathbf{d}}_n$ . Subsequently, based on the estimated disparity relationship, as well as the extracted features of the  $(n-1)$ -th view,  $\mathbf{x}_n$  is compressed. When compressing the depth map  $\mathbf{d}_n$ , the model employs the predicted depth map derived from  $\hat{\mathbf{d}}_{n-1}$  as a reference. The same neural network architecture and

model parameters are used consistently across all views for both image compression and depth map compression.

The remainder of this section is structured as follows: Section 3.1 elaborates on the method for depth map estimation for a given view using the 3D-GS and the estimation of inter-view disparities. Section 3.2 covers the compression model for both images and depth maps, as well as the multi-view sequence ordering method.

#### 3.1. 3D-GS Based Depth and Disparity Estimation

##### 3.1.1. DEPTH ESTIMATION

For an image  $\mathbf{x}_n \in \mathbb{R}^{W \times H \times 3}$  with spatial dimensions  $W$  and  $H$ , we aim to derive a depth map  $\mathbf{d}_n \in \mathbb{R}^{W \times H}$ , representing the  $z$ -axis coordinates of each pixel’s corresponding 3D world point in the camera coordinate system. This depth map facilitates the estimation of disparities between different views.

In the context of the 3D-GS framework, consider a set of  $M$  ordered 3D points projected along a ray from the camera through a pixel. The rendered pixel color  $c$  can be expressed as:

$$c = \sum_{i=1}^M T_i \alpha_i c_i, \quad \text{with } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (1)$$

Here,  $c_i$  and  $\alpha_i$  represent the color and opacity (density) of the point, respectively, derived from the point’s 3D Gaussian properties. The factor  $T_i$  denotes the transmittance along the ray, indicating the fraction of light reaching the camera without being occluded.

In (1),  $T_i$  serves as a weight for the contribution of each point’s color to the pixel’s final color, diminishing from 1 to 0 as  $i$  increases due to cumulative absorption. To estimate the depth of a pixel  $d$ , we adopt a median depth estimation approach. Specifically, the depth is determined as the depth of the first point where  $T_i$  drops below 0.5:

$$d = z_{i^*}, \text{ where } i^* = \min\{i \mid T_i < 0.5\}. \quad (2)$$

It is worth noting that the original 3D-GS (Kerbl et al., 2023) employs a weighted averaging approach, using  $T_i \alpha_i$  as the weight for each 3D Gaussian along the ray to compute depth. In contrast, alignment experiments in Section 4.3 demonstrate that the median depth estimation approach achieves better alignment performance.

### 3.1.2. DISPARITY ESTIMATION

Next, we aim to estimate the disparity  $\Delta_n \in \mathbb{R}^{W \times H \times 2}$  between views based on the estimated depth map. This disparity represents the pixel-wise shift of each 3D world point’s projection across different views. Disparity estimation captures the geometric relationships between views, facilitating the modeling of inter-view correlations.

Figure 2(b) illustrates the depth-based disparity estimation. To estimate the disparity, a pixel  $(x_n, y_n)$  in the  $n$ -th view is back-projected into 3D space using the depth  $d_n$  to obtain the world coordinates  $(x_w, y_w, z_w)$ . This 3D world point is then projected into the  $(n-1)$ -th view to obtain the corresponding pixel coordinates  $(x_{n-1}, y_{n-1})$ . The transformations involved are as follows:

$$\begin{aligned} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} &= V_n^{-1} \cdot \text{aug} \left( K^{-1} d_n \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \right), \\ d'_{n-1} \begin{bmatrix} x_{n-1} \\ y_{n-1} \\ 1 \end{bmatrix} &= K \cdot \text{deaug} \left( V_{n-1} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \right), \end{aligned} \quad (3)$$

where  $K \in \mathbb{R}^{3 \times 3}$  denotes the camera intrinsic matrix, and  $V_n, V_{n-1} \in \mathbb{R}^{4 \times 4}$  are the extrinsic matrices corresponding to the  $n$ -th and  $(n-1)$ -th views, respectively. The camera parameters are calibrated using SfM (Schonberger & Frahm, 2016).  $d'_{n-1}$  represents the depth of the 3D world point in the camera coordinate system of the  $(n-1)$ -th view.  $\text{aug}$  denotes the operation of augmenting a vector by adding an additional dimension with a value of 1 as its final element. Conversely,  $\text{deaug}$  refers to the operation of reducing a vector by removing its last dimension. The resulting disparity  $\delta_n = (x_{n-1} - x_n, y_{n-1} - y_n)$  for each pixel is then compiled into the disparity map  $\Delta_n$ .

Finally, we define a mask  $x_{n,m} \in \mathbb{R}^{W \times H}$  to determine whether the disparity estimation is meaningful, i.e., whether the information from the reference pixel is relevant or merely noise. The mask’s criteria are as follows:

1. The projected pixel must reside within the valid image region in the  $(n-1)$ -th view.
2. The corresponding 3D world point must lie in the positive  $z$ -half-space of the  $(n-1)$ -th view’s coordinate system.
3. No occlusion must exist along the line of sight, i.e.,  $d'_{n-1}$  from (3) must be less than the estimated depth along the ray in the  $(n-1)$ -th view.

This can be formulated as:

$$x_{n,m}[i, j] = \begin{cases} 1 & \text{if } 0 < \Delta_n[i, j, 0] + i + 0.5 < W \text{ and} \\ & 0 < \Delta_n[i, j, 1] + j + 0.5 < H \text{ and} \\ & 0 < d'_{n-1}[i, j] < \text{Warp}(d_{n-1}, \Delta_n)[i, j], \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where  $d'_{n-1} \in \mathbb{R}^{W \times H}$  represents the tensor containing the depth values  $d'_{n-1}$  for each pixel, and  $\text{Warp}(\cdot, \cdot)$  denotes the warping operation based on the given disparity. Appendix A outlines the algorithmic process for disparity and mask estimation.

## 3.2. Compression Framework for Images and Depth Maps

### 3.2.1. IMAGE COMPRESSION MODEL

As shown in Figure 3, the disparity extractor *DISE* utilizes reconstructed depth maps  $\hat{d}_{n-1}$  and  $\hat{d}_n$  to extract multi-scale disparities and feature masks. The reference feature extractor *RFE* generates multi-scale reference features from the reconstructed image  $\hat{x}_{n-1}$  and its intermediate reconstruction features  $\{\mathbf{f}_{n-1}^i \mid i = 1, 2, 3\}$ . Subsequently, the image encoder *IE* and decoder *ID* incorporate the reference features, aligned using the extracted disparities, into the backbone network. This process is formalized as:

$$\begin{aligned} \mathbf{y}_n &= IE(\mathbf{x}_n, DISE(\hat{d}_{n-1}, \hat{d}_n), RFE(\hat{x}_{n-1}, \{\mathbf{f}_{n-1}^i\})), \\ \hat{\mathbf{y}}_n &= Q(\mathbf{y}_n), \\ \hat{\mathbf{x}}_n &= ID(\hat{\mathbf{y}}_n, DISE(\hat{d}_{n-1}, \hat{d}_n), RFE(\hat{x}_{n-1}, \{\mathbf{f}_{n-1}^i\})). \end{aligned} \quad (5)$$

For entropy coding, we utilize the hyperprior entropy model (Ballé et al., 2018) and the quadtree partition-based entropy model (QPEM) (Li et al., 2023). The hyperprior entropy model transforms  $\mathbf{y}_n$  into a hyperprior representation  $\mathbf{z}_n$ . The quantized hyperprior representation  $\hat{\mathbf{z}}_n$  is then used to accurately model the probability distribution of  $\hat{\mathbf{y}}_n$ . The conditional probability distribution  $p_{\hat{\mathbf{y}}_n | \hat{\mathbf{z}}_n}$  is defined as:

$$p_{\hat{\mathbf{y}}_n | \hat{\mathbf{z}}_n}(\hat{\mathbf{y}}_n | \hat{\mathbf{z}}_n) \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2). \quad (6)$$

**Disparity extractor.** As illustrated in Figure 3, we firstly employ the disparity estimation (DPE) module to derive the

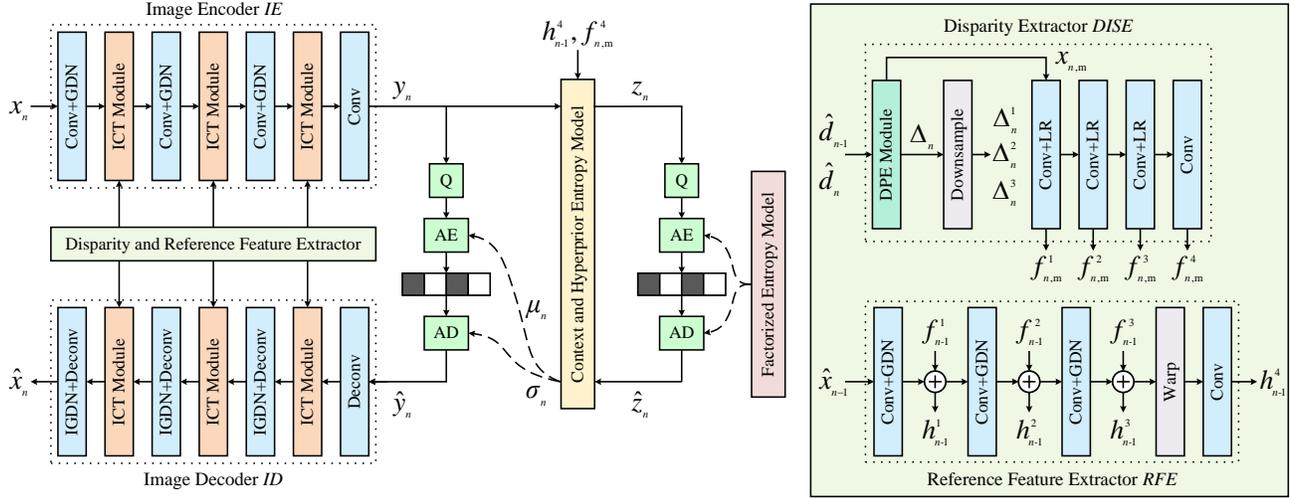


Figure 3. The architecture of the proposed image compression model. ‘LR’ represents the Leaky ReLU activation function, ‘Q’ denotes the quantization operation, and ‘AE’/‘AD’ refer to the arithmetic encoder/decoder, respectively.

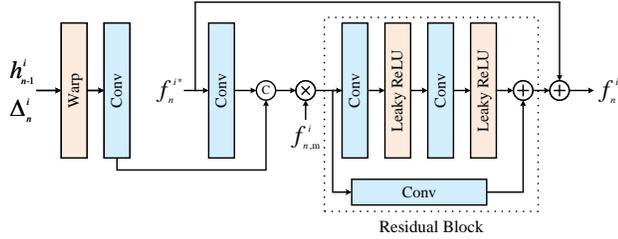


Figure 4. Illustration of the proposed image context transfer module.

disparity map  $\Delta_n$  and the corresponding mask  $x_{n,m}$ , using  $\hat{d}_{n-1}$  and  $\hat{d}_n$ , following the method outlined in Section 3.1. Subsequently,  $\Delta_n$  undergoes a series of downsampling operations to produce multi-scale disparity maps  $\{\Delta_n^i \mid i = 1, 2, 3\}$ , which will facilitate multi-scale feature alignment. The mask  $x_{n,m}$  is further processed by the disparity mask extractor to extract feature masks  $\{f_{n,m}^i \mid i = 1, 2, 3, 4\}$ .

**Reference feature extractor.** The reference feature extractor takes  $\hat{x}_{n-1}$ ,  $\{f_{n-1}^i \mid i = 1, 2, 3\}$ , and  $\Delta_n^3$  as inputs to extract multi-scale reference features  $\{h_{n-1}^i \mid i = 1, 2, 3, 4\}$ , as shown in Figure 3.

**Image context transfer module.** To incorporate the reference feature  $\{h_{n-1}^i \mid i = 1, 2, 3\}$  obtained from the  $(n-1)$ -th view into the image backbone encoder and decoder, enhancing feature representation, we introduce the image context transfer (ICT) module. As depicted in Figure 4, the module enhances the input feature  $f_n^{i*}$  from the backbone network by leveraging the aligned reference feature  $h_{n-1}^i$  via  $\Delta_n^i$ . By applying feature masks, the module filters relevant information and refines the features, ultimately producing the output feature  $f_n^i$  through a residual enhancement process.

### 3.2.2. DEPTH MAP COMPRESSION MODEL

The compression and decompression of the depth map  $d_n$  leverage  $\hat{d}_{n-1}$  as a reference. Initially,  $\hat{d}_{n-1}$  is processed by the depth prediction extractor *DEPE*, which generates multi-scale depth prediction features and corresponding feature masks. Subsequently, the depth encoder *DE* and decoder *DD* integrate these extracted features and masks into the backbone network. This process is formalized as:

$$\begin{aligned} y_{d_n} &= DE(d_n, DEPE(\hat{d}_{n-1})), \\ \hat{y}_{d_n} &= Q(y_{d_n}), \\ \hat{d}_n &= DD(\hat{y}_{d_n}, DEPE(\hat{d}_{n-1})). \end{aligned} \quad (7)$$

The entropy coding scheme incorporates both the hyperprior entropy model and the QPEM. The latent representation  $y_{d_n}$  is transformed into a hyperprior representation  $z_{d_n}$  using the hyperprior entropy model. Similar to the image compression model, the quantized hyperprior representation  $\hat{z}_{d_n}$  is used to model the probability distribution of  $\hat{y}_{d_n}$ . Additional details about the depth map compression model are provided in Appendix B.

### 3.2.3. MULTI-VIEW SEQUENCE ORDERING

Given the significant impact of FoV overlap between adjacent views on inter-view correlations, we propose a multi-view sequence ordering method to alleviate the issue of insufficient overlap in unordered sequences. We define a distance metric to evaluate inter-view overlap and employ a greedy algorithm to find an improved sequence.

In (3), if  $V_{n-1}V_n^{-1} = I$ , then  $(x_n, y_n) = (x_{n-1}, y_{n-1})$ . This indicates that each pixel in the  $n$ -th view lies within the valid image area of the  $(n-1)$ -th view, indicating high overlap. Thus, for any two views  $i$  and  $j$ , we measure

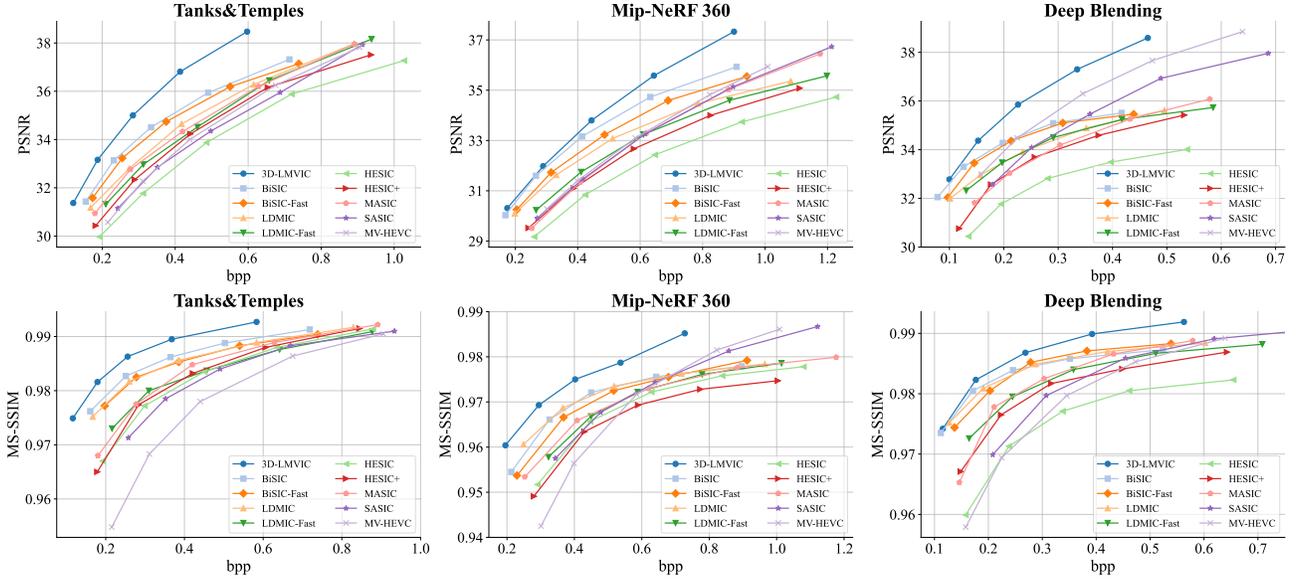


Figure 5. Rate-distortion curves of the proposed method compared with baselines.

overlap by the proximity of  $V_i V_j^{-1}$  to the identity matrix:

$$\mathcal{D}_V(i, j) = \|V_i V_j^{-1} - I\|. \quad (8)$$

Appendix C proves that  $\mathcal{D}_V(i, j)$  is a distance metric for both the 2-norm and Frobenius norm. The Frobenius norm is utilized in our experiments. After determining pairwise distances, a greedy algorithm is employed, starting from an initial sequence with only one view, iteratively selecting the view closest to the last view in the sequence.

### 3.2.4. TRAINING LOSS

For each training step, a randomly selected subsequence of length 4 from a multi-view sequence serves as the training sample. The training loss comprises the distortion losses for both the reconstructed image and depth map, as well as the estimated compression rates for the encoded image and depth map:

$$L = \sum_{n=s}^{s+3} w_{n-s+1} \left[ \lambda_{\text{img}} D(\mathbf{x}_n, \hat{\mathbf{x}}_n) + \lambda_{\text{dep}} \text{MSE}(\mathbf{d}_n, \hat{\mathbf{d}}_n) + R(\hat{\mathbf{y}}_n) + R(\hat{\mathbf{z}}_n) + R(\hat{\mathbf{y}}_{d_n}) + R(\hat{\mathbf{z}}_{d_n}) \right], \quad (9)$$

where  $D(\cdot, \cdot)$  denotes the distortion,  $\text{MSE}(\cdot, \cdot)$  represents the mean squared error (MSE), and  $R(\cdot)$  indicates the estimated compression rates. The hyperparameters  $\lambda_{\text{img}}$  and  $\lambda_{\text{dep}}$  control the contributions of the image and depth map distortion losses, respectively. The weights  $\{w_i \mid i = 1, 2, 3, 4\}$  adjust the influence of each view on the overall training loss.

## 4. Experiments

### 4.1. Experimental Setup

**Datasets.** We evaluate our model on three multi-view image datasets: Tanks&Temples (Knapitsch et al., 2017), Mip-NeRF 360 (Barron et al., 2022), and Deep Blending (Hedman et al., 2018). Further details on the datasets are provided in Appendix D.

**Benchmarks.** We compare our approach against several baselines, including traditional multi-view codec: MV-HEVC (Hannuksela et al., 2015); learning-based multi-view image codecs: two variants of HESIC (Deng et al., 2021), MASIC (Deng et al., 2023), SASIC (Wödlinger et al., 2022), two variants of LDMIC (Zhang et al., 2023), and two variants of BiSIC (Liu et al., 2024); as well as the 3D-GS compression method: HAC (Chen et al., 2024). Further details on the baseline configurations are provided in Appendix D.

**Metrics.** Image reconstruction quality is measured using peak signal-to-noise ratio (PSNR) and multi-scale structural similarity index (MS-SSIM) (Wang et al., 2003). Bitrate is expressed in bits per pixel (bpp). In addition to plotting RD curves, the Bjøntegaard Delta bitrate (BDBR) is calculated to quantify the average bitrate savings across varying reconstruction qualities. Lower BDBR values indicate better performance.

**Implementation Details.** The model was trained using five different configurations of  $(\lambda_{\text{img}}, \lambda_{\text{dep}})$ :  $((256, 64), (512, 128), (1024, 128), (2048, 128), (4096, 128))$  when the image distortion loss is MSE, and  $((8, 64), (16, 128), (32, 128), (64, 128), (128, 128))$  when using MS-SSIM. The weights  $w_i$  for four consecutive

Table 1. BDBR comparison of different methods relative to MV-HEVC.

Methods	Tanks&Temples		Mip-NeRF 360		Deep Blending	
	PSNR	MS-SSIM	PSNR	MS-SSIM	PSNR	MS-SSIM
HAC	636.81%	350.72%	374.20%	294.42%	673.57%	418.85%
HESIC	12.66%	-26.29%	28.41%	-6.18%	85.38%	3.91%
HESIC+	-4.85%	-30.42%	9.48%	-5.11%	32.5%	-19.14%
MASIC	-12.57%	-34.19%	3.26%	-9.11%	43.6%	-9.33%
SASIC	3.39%	-18.59%	2.40%	-3.70%	24.64%	-9.48%
LDMIC-Fast	-8.56%	-27.76%	1.72%	-6.21%	24.25%	-23.31%
LDMIC	-16.27%	-44.33%	-13.12%	-25.39%	16.88%	-41.94%
BiSIC-Fast	-26.59%	-42.93%	-20.61%	-23.23%	-8.24%	-41.80%
BiSIC	-30.89%	-49.96%	-29.87%	-30.75%	-15.46%	-48.47%
3D-LMVIC	<b>-47.48%</b>	<b>-63.69%</b>	<b>-34.69%</b>	<b>-40.25%</b>	<b>-27.31%</b>	<b>-54.15%</b>

Table 2. Average alignment quality (PSNR, MS-SSIM) of different alignment methods on the Train scene of the Tanks&amp;Temples dataset.

Metrics	Methods								
	HT	PM	SPyNet	PWC-Net	FlowFormer++	3D-GS	COLMAP	MVSFormer++	Proposed
PSNR	15.16	17.94	16.12	17.59	18.08	17.36	14.32	15.31	<b>18.14</b>
MS-SSIM	0.5435	0.7633	0.6289	0.7707	0.7863	0.7410	0.7446	0.5544	<b>0.8053</b>

views were set to (0.5, 1.2, 0.5, 0.9) as referenced from Li et al. (2023). The model was trained for 300 epochs with an initial learning rate of  $10^{-4}$ , which was progressively decayed by a factor of 0.5 every 60 epochs.

## 4.2. Experimental Results

**Coding performance.** Figure 5 presents the rate-distortion curves of the compared methods, while Table 1 summarizes the BDBR of each codec relative to MV-HEVC. Across the three datasets, the proposed 3D-LMVIC consistently outperforms the baselines in both PSNR and MS-SSIM, demonstrating its effectiveness in reducing inter-view redundancy. For instance, on the Tanks&Temples dataset, 3D-LMVIC achieves a BDBR reduction of 16.59% for PSNR and 13.73% for MS-SSIM compared to BiSIC. The BDBR of HAC is relatively higher, likely due to the inclusion of 3D scene information in addition to 2D image representations. Appendix F provides examples of visual comparisons. Appendix G presents an analysis of computational complexity. Appendix H includes supplementary experiments on coding performance.

## 4.3. Alignment Experiments

To evaluate the effectiveness of the proposed 3D Gaussian geometric priors-based alignment method, we conducted alignment experiments on the Train scene from the Tanks&Temples dataset. The baselines for comparison include:

1. Alignment methods commonly used in learning-based multi-view image codecs, such as Homography Transfor-

2. Optical flow estimation methods, such as SPyNet (Ranjan & Black, 2017), PWC-Net (Sun et al., 2018), and FlowFormer++ (Shi et al., 2023).

3. Depth map estimation methods, including original 3D-GS (Kerbl et al., 2023), COLMAP (Schonberger & Frahm, 2016; Schönberger et al., 2016) and MVSFormer++ (Chenjie Cao & Fu, 2024).

Alignment quality was assessed by computing PSNR and MS-SSIM between the aligned reference view images and the target view images. Table 2 summarizes the average alignment quality and runtime for each method. The proposed method outperformed the baselines in both PSNR and MS-SSIM, indicating its effectiveness in capturing complex disparities between views. Figure 6 provides visual comparisons, demonstrating that the proposed method achieves closer alignment with the target view images. Appendix D further investigates the relationship between the mask defined in (4) and the ghosting artifacts introduced during alignment.

Alignment quality was assessed by computing PSNR and MS-SSIM between the aligned reference view images and the target view images. Table 2 summarizes the average alignment quality and runtime for each method. The proposed method outperformed the baselines in both PSNR and MS-SSIM, indicating its effectiveness in capturing complex disparities between views. Figure 6 provides visual comparisons, demonstrating that the proposed method achieves closer alignment with the target view images. Appendix D further investigates the relationship between the mask defined in (4) and the ghosting artifacts introduced during alignment.

## 4.4. Ablation Study

**Codec components.** To assess the contribution of codec components, we performed ablation experiments on the Tanks&Temples dataset. The rate-distortion curves are shown in Figure 7. Specifically, we evaluated the following baselines: (1) *Separate*: encoding and decoding without cross-view information; (2) *Concatenation*: direct feature concatenation from reference view without alignment; (3)

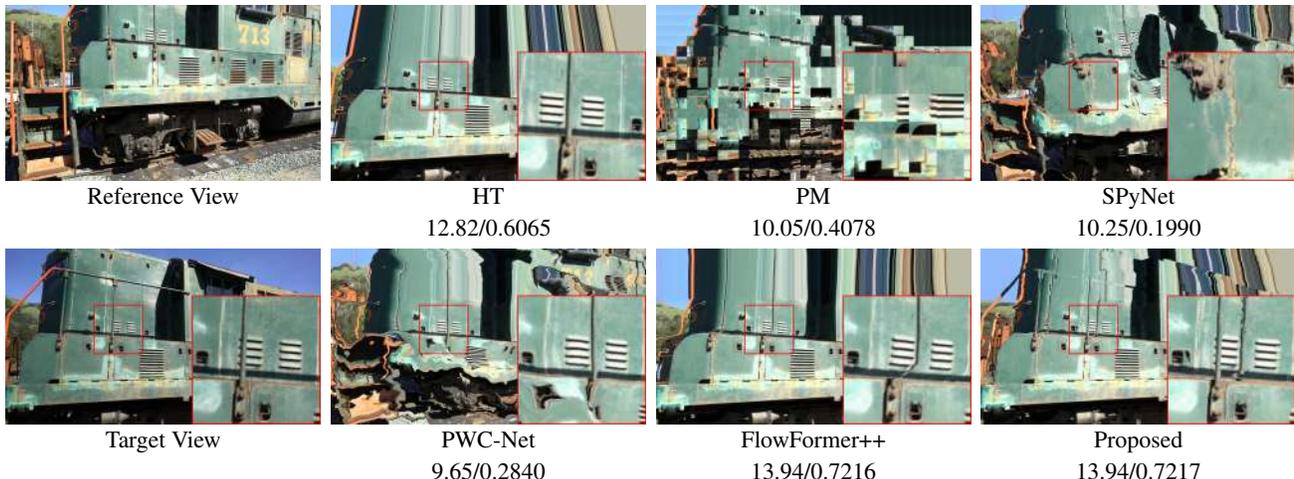


Figure 6. Visual comparison of different alignment methods on an adjacent view pair in the Train scene of the Tanks&Temples dataset. Alignment quality is reported as PSNR/MS-SSIM.

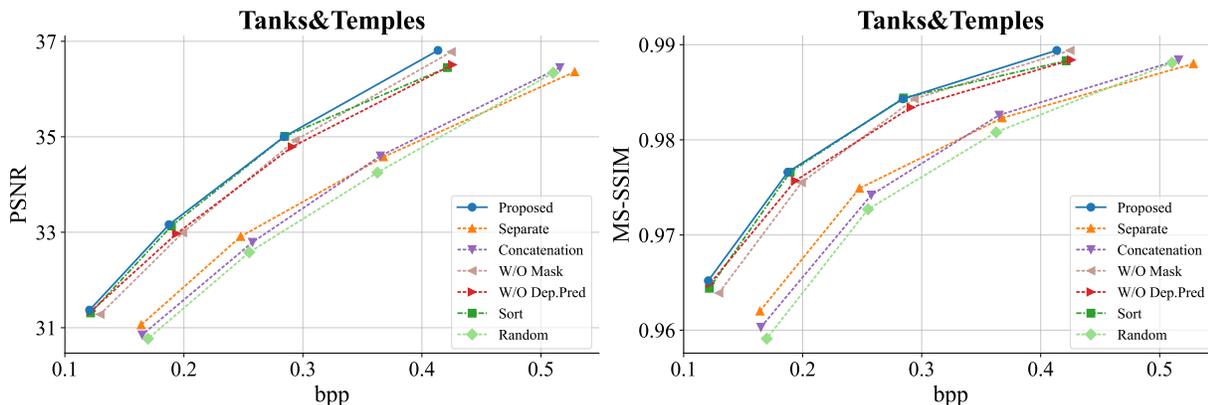


Figure 7. Rate-distortion curves of different ablation baselines on the Tanks&Temples dataset.

*W/O Mask*: removal of both image and depth mask; (4) *W/O Dep.Pred*: excluding depth prediction in the depth map compression model. These baselines resulted in bitrate increases of 41.07% (44.24%), 42.75% (47.52%), 7.19% (8.47%), and 8.03% (8.02%) for PSNR (MS-SSIM), respectively, compared to the proposed method. The experimental results validate the effectiveness of the proposed components.

**Multi-view sequence ordering.** As illustrated in Figure 7, we evaluated two baselines to assess the effectiveness of the proposed multi-view sequence ordering method: (1) *Sort*: sequences are ordered using the proposed method; (2) *Random*: sequences are randomly ordered. The *Random* baseline led to a 42.4% (50.64%) increase in bitrate for PSNR (MS-SSIM) compared to *Sort*. Furthermore, *Sort* exhibited only a 3.76% (2.97%) bitrate increase for PSNR (MS-SSIM) compared to the manually sorted sequences in the Tanks&Temples dataset. These results demonstrate the effectiveness of the proposed ordering method for unsorted multi-view sequences, achieving performance close to that of manual sorting.

## 5. Conclusion

In this paper, we present 3D-LMVIC, a novel learning-based multi-view image coding framework incorporating 3D Gaussian geometric priors. This framework exploits these geometric priors to estimate complex disparities and masks between views for effectively utilizing reference view information in the compression process. Additionally, we propose a depth map compression model designed to compactly and accurately represent the geometry of each view, incorporating a cross-view depth prediction module to capture inter-view geometric correlations. Moreover, we introduce a multi-view sequence ordering method for unordered sequences, enhancing the overlap between adjacent views by defining an inter-view distance measure to guide the sequence ordering. Experimental results confirm that 3D-LMVIC surpasses existing learning-based coding schemes in compression efficiency while achieving accurate disparity estimation.

## Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grant 62171248, 62301189, the project of Peng Cheng Laboratory (PCL2023A08), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), and Shenzhen Science and Technology Program under Grant KJZD20240903103702004, JCYJ20220818101012025, GXWD20220811172936001.

## Impact Statement

This work introduces 3D-LMVIC, a novel learning-based framework for multi-view image compression that leverages 3D Gaussian geometric priors to enable more accurate disparity estimation and efficient inter-view redundancy reduction. The proposed method demonstrates significant improvements over both traditional and learning-based baselines in compression efficiency and alignment quality across diverse 3D scene datasets. This advancement is especially valuable for applications requiring scalable and high-quality multi-view data processing, such as immersive virtual and augmented reality, autonomous driving, and 3D reconstruction. By incorporating geometric priors into the learning pipeline, this work contributes to bridging the gap between geometric scene understanding and data-driven compression. We believe our method offers a promising direction for further research in geometry-aware compression, though care should be taken to evaluate generalization to in-the-wild multi-view scenes with dynamic content.

## References

- Anthes, C., García-Hernández, R. J., Wiedemann, M., and Kranzlmüller, D. State of the art of virtual reality technology. In *2016 IEEE aerospace conference*, pp. 1–19. IEEE, 2016.
- Ayzik, S. and Avidan, S. Deep image compression using decoder side information. In *European Conference on Computer Vision*, pp. 699–714, 2020.
- Ballé, J., Laparra, V., and Simoncelli, E. P. Density modeling of images using a generalized normalization transformation. In *International Conference on Learning Representations*, 2016.
- Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.
- Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.
- Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., and Hedman, P. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.
- Bellard, F. Bpg image format. <https://bellard.org/bpg/>, 2014.
- Bross, B., Wang, Y.-K., Ye, Y., Liu, S., Chen, J., Sullivan, G. J., and Ohm, J.-R. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.
- Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.
- Chen, Y., Wu, Q., Cai, J., Harandi, M., and Lin, W. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, 2024.
- Chenjie Cao, X. R. and Fu, Y. Mvsformer++: Revealing the devil in transformer’s details for multi-view stereo. In *International Conference on Learning Representations (ICLR)*, 2024.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.
- Deng, X., Yang, W., Yang, R., Xu, M., Liu, E., Feng, Q., and Timofte, R. Deep homography for efficient stereo image compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1501, 2021. doi: 10.1109/CVPR46437.2021.00154.
- Deng, X., Deng, Y., Yang, R., Yang, W., Timofte, R., and Xu, M. Masic: Deep mask stereo image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- Hamdi, A., Melas-Kyriazi, L., Mai, J., Qian, G., Liu, R., Vondrick, C., Ghanem, B., and Vedaldi, A. Ges: Generalized exponential splatting for efficient radiance field rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19812–19822, June 2024.
- Hannuksela, M. M., Yan, Y., Huang, X., and Li, H. Overview of the multiview high efficiency video coding (mv-hevc) standard. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 2154–2158, 2015. doi: 10.1109/ICIP.2015.7351182.

- He, D., Zheng, Y., Sun, B., Wang, Y., and Qin, H. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14771–14780, 2021.
- Hedman, P., Philip, J., Price, T., Frahm, J.-M., Drettakis, G., and Brostow, G. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6), dec 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275084. URL <https://doi.org/10.1145/3272127.3275084>.
- Hosseinian, S. and Arefi, H. 3d reconstruction from multi-view medical x-ray images—review and evaluation of existing methods. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, 40:319–326, 2015.
- Huang, Y., Chen, B., Zhang, J., Han, Q., and Xia, S.-T. Compressive sensing based asymmetric semantic image compression for resource-constrained iot system. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pp. 877–882, 2022.
- Huang, Y., Chen, B., Qin, S., Li, J., Wang, Y., Dai, T., and Xia, S.-T. Learned distributed image compression with multi-scale patch matching in feature domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4322–4329, 2023.
- Jiang, W., Yang, J., Zhai, Y., Ning, P., Gao, F., and Wang, R. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 7618–7627, 2023.
- Kerbl, B., Kopanas, G., Leimkuehler, T., and Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.
- Knapitsch, A., Park, J., Zhou, Q.-Y., and Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.
- Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.
- Lei, J., Liu, X., Peng, B., Jin, D., Li, W., and Gu, J. Deep stereo image compression via bi-directional coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19669–19678, 2022.
- Li, J., Li, B., and Lu, Y. Neural video compression with diverse contexts. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22616–22626, 2023. doi: 10.1109/CVPR52729.2023.02166.
- Liu, X., Chen, B., Liu, Z., Wang, Y., and Xia, S.-T. An exploration with entropy constrained 3d gaussians for 2d video compression. In *The Thirteenth International Conference on Learning Representations*.
- Liu, Z., Zhang, X., Shao, J., Lin, Z., and Zhang, J. Bidirectional stereo image compression with cross-dimensional entropy model. In *European Conference on Computer Vision*, 2024.
- Minnen, D., Ballé, J., and Toderici, G. Joint autoregressive and hierarchical priors for learned image compression. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 10794–10803, 2018.
- Mokssit, S., Licea, D. B., Guermah, B., and Ghogho, M. Deep learning techniques for visual slam: A survey. *IEEE Access*, 11:20026–20050, 2023.
- Qin, S., Chen, B., Huang, Y., An, B., Dai, T., and Xia, S.-T. Perceptual image compression with cooperative cross-modal side information. *arXiv e-prints*, pp. arXiv–2311, 2023.
- Qin, S., Wang, J., Zhou, Y., Chen, B., Luo, T., An, B., Dai, T., Xia, S., and Wang, Y. Mambavc: Learned visual compression with selective state spaces. *arXiv preprint arXiv:2405.15413*, 2024a.
- Qin, S., Zhou, Y.-M., Wang, J.-P., Chen, B., An, B.-Y., Dai, T., and Xia, S.-T. Progressive learning with visual prompt tuning for variable-rate image compression. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 1767–1773. IEEE, 2024b.
- Ranjan, A. and Black, M. J. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- Schmalstieg, D. and Hollerer, T. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016.
- Schonberger, J. L. and Frahm, J.-M. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.
- Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K. C., See, S., Qin, H., Dai, J., and Li, H. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF Conference*

- on *Computer Vision and Pattern Recognition (CVPR)*, pp. 1599–1610, June 2023.
- Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- Vetro, A., Wiegand, T., and Sullivan, G. J. Overview of the stereo and multiview video coding extensions of the h. 264/mpeg-4 avc standard. *Proceedings of the IEEE*, 99(4):626–642, 2011.
- Wallace, G. K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
- Wang, Z., Simoncelli, E. P., and Bovik, A. C. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. Ieee, 2003.
- Wödlinger, M., Kotera, J., Xu, J., and Sablatnig, R. Sasic: Stereo image compression with latent shifts and stereo attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 661–670, 2022.
- Xia, Y., Huang, Y., Chen, B., Wang, G., Wang, H., and Wang, Y. Fca-net: Accelerating stereo image compression through cascade alignment of side information. *Pattern Recognition*, 168:111799, 2025. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2025.111799>. URL <https://www.sciencedirect.com/science/article/pii/S0031320325004595>.
- Xu, L., Su, Z., Han, L., Yu, T., Liu, Y., and Fang, L. Unstructuredfusion: Realtime 4d geometry and texture reconstruction using commercial rgb-d cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2508–2522, 2020. doi: 10.1109/TPAMI.2019.2915229.
- Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., and Li, X. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604, June 2024.
- Zhai, Y., Tang, L., Ma, Y., Peng, R., and Wang, R. Disparity-based stereo image compression with aligned cross-view priors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 2351–2360, 2022.
- Zhang, X., Shao, J., and Zhang, J. Ldmic: Learning-based distributed multi-view image coding. In *International Conference on Learning Representations*, 2023.

## A. Disparity and Mask Estimation Algorithm

### Algorithm 1 Disparity and Mask Estimation

**Input:** Depth estimation function  $GSDE$ , intrinsic matrix  $K$ , extrinsic matrices  $V_n$  and  $V_{n-1}$

**Output:** Disparity map  $\Delta_n$ , mask  $x_{n,m}$

$d_n \leftarrow GSDE(K, V_n)$

$d_{n-1} \leftarrow GSDE(K, V_{n-1})$

$\Delta_n, d'_{n-1} \leftarrow \text{DisparityEstimation}(d_n, K, V_n, V_{n-1})$

$x_{n,m} \leftarrow \text{MaskEstimation}(\Delta_n, d'_{n-1}, d_{n-1})$

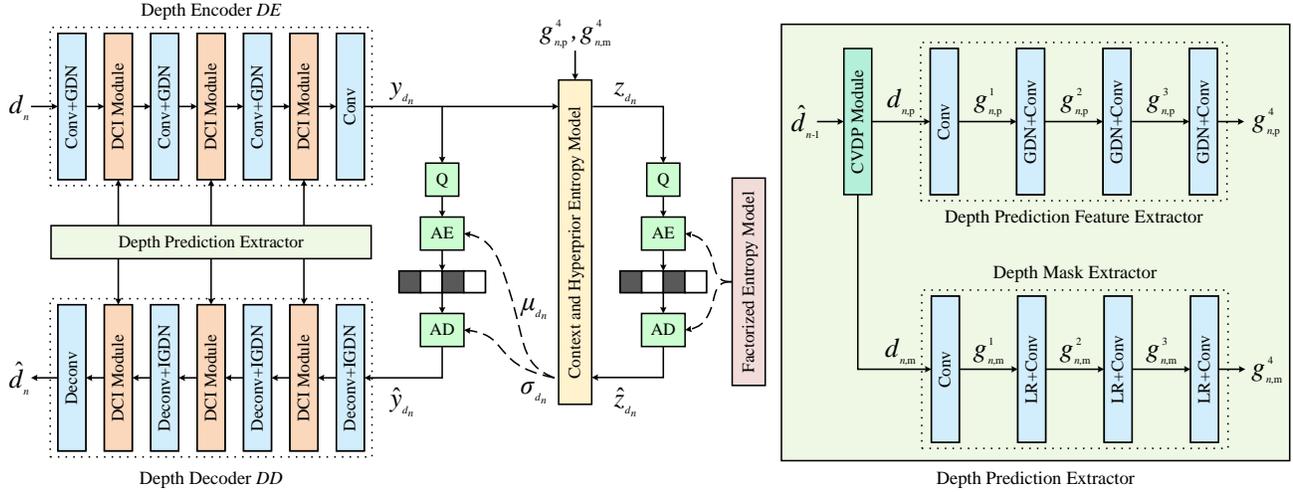


Figure 8. The architecture of the proposed depth map compression model. 'LR' represents the Leaky ReLU activation function, 'Q' denotes the quantization operation, and 'AE'/'AD' refer to the arithmetic encoder/decoder, respectively.

## B. Supplementary Information for the Depth Map Compression Model

As illustrated in Figure 8, during the compression and decompression of  $d_n$ ,  $\hat{d}_{n-1}$  is initially processed by the depth prediction extractor, which extracts multi-scale depth prediction features and associated feature masks. These extracted features and masks are then integrated into the depth backbone encoder and decoder via the depth context integration (DCI) module. Detailed explanations of the depth prediction extractor and the DCI module are provided in the subsequent content.

**Depth prediction extractor.** As illustrated in Figure 8, we first utilize the proposed cross-view depth prediction (CVDP) module to predict the depth map  $d_{n,p} \in \mathbb{R}^{W \times H}$  and the associated mask  $d_{n,m} \in \mathbb{R}^{W \times H}$  for the  $n$ -th view, based on  $\hat{d}_{n-1}$ . Specifically, for each pixel  $(x_{n-1}, y_{n-1})$  in the  $(n-1)$ -th view and its corresponding reconstructed depth  $\hat{d}_{n-1}$ , the CVDP module determines the corresponding pixel coordinates  $(x_n, y_n)$  and the depth prediction  $d'_n$  in the  $n$ -th view using the method described in (3). The depth at the nearest grid point  $(\lfloor x_n - 0.5 \rfloor, \lfloor y_n - 0.5 \rfloor)$  is then set to  $d'_n$ :

$$d_{n,p}[\lfloor x_n - 0.5 \rfloor, \lfloor y_n - 0.5 \rfloor] = d'_n. \quad (10)$$

This cross-view depth prediction is applied to each pixel in the  $(n-1)$ -th view to construct  $d_{n,p}$ . If multiple pixel coordinates map to the same grid point, the depth prediction for that point is set to the minimum of these predicted depths. Additionally, if a grid point has no corresponding pixel coordinates, its depth prediction value is set to 0. The mask  $d_{n,m}$  indicates whether each grid point has at least one corresponding pixel coordinate, with values set to 1 where a correspondence exists and 0 otherwise.

Subsequently,  $d_{n,p}$  is fed into the depth prediction feature extractor to produce multi-scale depth prediction features, denoted as  $\{g_{n,p}^i \mid i = 1, 2, 3, 4\}$ . Concurrently, the mask  $d_{n,m}$  is processed by the depth mask extractor to derive the associated multi-scale feature masks  $\{g_{n,m}^i \mid i = 1, 2, 3, 4\}$ .

**Depth Context Integration Module.** Each DCI module integrates the input features  $\mathbf{g}_n^{i*}$  from the backbone network with  $\mathbf{g}_{n,p}^i$  through channel-wise concatenation, followed by element-wise multiplication with  $\mathbf{g}_{n,m}^i$  to produce the output feature  $\mathbf{g}_n^i$ :

$$\mathbf{g}_n^i = (\mathbf{g}_n^{i*} \oplus \mathbf{g}_{n,p}^i) \odot \mathbf{g}_{n,m}^i, \quad (11)$$

where  $\oplus$  denotes channel-wise concatenation and  $\odot$  denotes element-wise multiplication.

## C. Proof of $\mathcal{D}_V(i, j)$ as a Distance Measure for 2-Norm and Frobenius Norm

### C.1. Proof for 2-Norm

#### C.1.1. DEFINITION

**Definition C.1.** For  $u = (A, B)$  and  $v = (C, D)$ , where  $A, C \in \mathbb{R}^{n \times m}$  and  $B, D \in \mathbb{R}^{n \times l}$ , we define  $(u, v)_2 = \|AC^T + BD^T\|_2$ . For any scalar  $\alpha$ ,  $\alpha u = (\alpha A, \alpha B)$ . Additionally,  $u + v = (A + C, B + D)$ .

#### C.1.2. LEMMA

**Lemma C.2.** For any  $u = (A, B)$  and  $v = (C, D)$  as defined in Definition C.1, the following inequality holds:

$$(u, v)_2 \leq \sqrt{(u, u)_2(v, v)_2}$$

*Proof.* For any real number  $t$ , we have:

$$\begin{aligned} (u + tv, u + tv)_2 &= \|(A + tC)(A + tC)^T + (B + tD)(B + tD)^T\|_2 \\ &\leq \|AA^T + BB^T\|_2 + t\|AC^T + BD^T\|_2 + t\|CA^T + DB^T\|_2 + t^2\|CC^T + DD^T\|_2 \\ &= (u, u)_2 + t(u, v)_2 + t(v, u)_2 + t^2(v, v)_2 \\ &= (u, u)_2 + 2t(u, v)_2 + t^2(v, v)_2 \end{aligned}$$

The right-hand side of the last equation can be viewed as a quadratic expression in  $t$  and is greater than or equal to  $(u + tv, u + tv)_2$ , which is non-negative. Therefore, the discriminant of this quadratic must be non-positive:

$$(2(u, v)_2)^2 - 4(u, u)_2(v, v)_2 \leq 0$$

Thus, we obtain:

$$(u, v)_2 \leq \sqrt{(u, u)_2(v, v)_2}$$

□

#### C.1.3. THEOREM

**Theorem C.3.**  $\mathcal{D}_V(i, j) = \|V_i V_j^{-1} - I\|_2$  is a distance metric.

*Proof.* We need to prove that  $\mathcal{D}_V(i, j)$  satisfies non-negativity, symmetry, and the triangle inequality.

**Non-negativity:** Since  $\mathcal{D}_V(i, j)$  is a norm, it is non-negative. Additionally, as the extrinsic matrices for different views are distinct,  $V_i \neq V_j$  for  $i \neq j$ .  $\mathcal{D}_V(i, j) = 0$  if and only if  $V_i V_j^{-1} - I = 0$ , which holds only when  $V_i = V_j$ , i.e.,  $i = j$ .

**Symmetry:** The extrinsic matrix  $V_i$  can be represented as  $V_i = \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix}$ , where  $R_i \in \mathbb{R}^{3 \times 3}$  is a rotation matrix<sup>1</sup> and  $t_i \in \mathbb{R}^{3 \times 1}$  is a translation vector. We have:

$$\mathcal{D}_V(i, j) = \|V_i V_j^{-1} - I\|_2$$

<sup>1</sup>A rotation matrix is an orthogonal matrix, meaning its inverse is equal to its transpose.

$$\begin{aligned}
 &= \left\| \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_j^T & -R_j^T t_j \\ 0 & 1 \end{pmatrix} - I \right\|_2 \\
 &= \left\| \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \right\|_2 \\
 &= \left\| \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix}^T \right\|_2^{\frac{1}{2}} \\
 &= \|2I - R_j R_i^T - R_i R_j^T + R_i R_j^T t_j t_j^T R_j R_i^T - t_i t_j^T R_j R_i^T - R_i R_j^T t_j t_i^T + t_i t_i^T\|_2^{\frac{1}{2}} \\
 &= \|R_j R_i^T (2I - R_j R_i^T - R_i R_j^T + R_i R_j^T t_j t_j^T R_j R_i^T - t_i t_j^T R_j R_i^T - R_i R_j^T t_j t_i^T + t_i t_i^T) R_i R_j^T\|_2^{\frac{1}{2}} \\
 &= \|2I - R_j R_i^T - R_i R_j^T + t_j t_j^T - R_j R_i^T t_i t_j^T - t_j t_i^T R_i R_j^T + R_j R_i^T t_i t_i^T R_i R_j^T\|_2^{\frac{1}{2}} \\
 &= \mathcal{D}_{\mathcal{V}}(j, i)
 \end{aligned}$$

The fourth equality follows from the fact that for any matrix  $A$ ,  $\|A\|_2 = \|AA^T\|_2^{\frac{1}{2}}$ . The sixth equality is due to the orthogonality of  $R_i$  and  $R_j$ , and the invariance of the 2-norm under orthogonal transformations. The final equality holds because interchanging the indices  $i$  and  $j$  in the expression on the right-hand side of the fifth equality leads to the same expression as  $\mathcal{D}_{\mathcal{V}}(j, i)$ , which matches the right-hand side of the seventh equality.

**Triangle inequality:** For views  $i$ ,  $j$ , and  $k$ , define  $A_{i,j} = R_j^T - R_i^T$ ,  $B_{i,j} = -R_j^T t_j + R_i^T t_i$ , and similarly for  $A_{j,k}, B_{j,k}, A_{k,i}, B_{k,i}$ . Let  $u_{j,k} = (A_{j,k}, B_{j,k})$  and  $u_{k,i} = (A_{k,i}, B_{k,i})$ . Starting from the fourth equation in the symmetry proof, we proceed as follows:

$$\begin{aligned}
 \mathcal{D}_{\mathcal{V}}(i, j) &= \left\| \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix}^T \right\|_2^{\frac{1}{2}} \\
 &= \|(R_i R_j^T - I)(R_i R_j^T - I)^T + (-R_i R_j^T t_j + t_i)(-R_i R_j^T t_j + t_i)^T\|_2^{\frac{1}{2}} \\
 &= \|R_i^T ((R_i R_j^T - I)(R_i R_j^T - I)^T + (-R_i R_j^T t_j + t_i)(-R_i R_j^T t_j + t_i)^T) R_i\|_2^{\frac{1}{2}} \\
 &= \|(R_j^T - R_i^T)(R_j^T - R_i^T)^T + (-R_j^T t_j + R_i^T t_i)(-R_j^T t_j + R_i^T t_i)^T\|_2^{\frac{1}{2}} \\
 &= \|A_{i,j} A_{i,j}^T + B_{i,j} B_{i,j}^T\|_2^{\frac{1}{2}} \\
 &= \|(A_{j,k} + A_{k,i})(A_{j,k} + A_{k,i})^T + (B_{j,k} + B_{k,i})(B_{j,k} + B_{k,i})^T\|_2^{\frac{1}{2}} \\
 &= \|A_{j,k} A_{j,k}^T + B_{j,k} B_{j,k}^T + A_{k,i} A_{k,i}^T + B_{k,i} B_{k,i}^T + A_{j,k} A_{k,i}^T + B_{j,k} B_{k,i}^T + A_{k,i} A_{j,k}^T + B_{k,i} B_{j,k}^T\|_2^{\frac{1}{2}} \\
 &\leq (\|A_{j,k} A_{j,k}^T + B_{j,k} B_{j,k}^T\|_2 + \|A_{k,i} A_{k,i}^T + B_{k,i} B_{k,i}^T\|_2 + 2\|A_{j,k} A_{k,i}^T + B_{j,k} B_{k,i}^T\|_2)^{\frac{1}{2}} \\
 &= (\mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2(u_{j,k}, u_{k,i})_2)^{\frac{1}{2}} \\
 &\leq \left( \mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2\sqrt{(u_{j,k}, u_{j,k})_2 (u_{k,i}, u_{k,i})_2} \right)^{\frac{1}{2}} \\
 &= (\mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2\mathcal{D}_{\mathcal{V}}(j, k)\mathcal{D}_{\mathcal{V}}(k, i))^{\frac{1}{2}} \\
 &= \mathcal{D}_{\mathcal{V}}(j, k) + \mathcal{D}_{\mathcal{V}}(k, i)
 \end{aligned}$$

The second inequality follows from Lemma C.2. □

## C.2. Proof for Frobenius Norm

### C.2.1. DEFINITION

**Definition C.4.** For  $u = (A, B)$  and  $v = (C, D)$  as defined in Definition C.1, we define  $(u, v)_F = \text{tr}(AC^T + BD^T)$ .

## C.2.2. LEMMA

**Lemma C.5.** For  $u$  and  $v$  as defined in Definition C.4, the following inequality holds:

$$(u, v)_F \leq \sqrt{(u, u)_F (v, v)_F}.$$

*Proof.* The method of proof is analogous to that used in Lemma C.2. By leveraging the properties of the trace and following a similar reasoning process, the result is derived.  $\square$

## C.2.3. THEOREM

**Theorem C.6.**  $\mathcal{D}_V(i, j) = \|V_i V_j^{-1} - I\|_F$  is a distance metric.

*Proof.* We need to prove that  $\mathcal{D}_V(i, j)$  satisfies non-negativity, symmetry, and the triangle inequality.

**Non-negativity:** The proof follows a similar approach to that of Theorem C.3, so we omit the details here.

**Symmetry:**

$$\begin{aligned} \mathcal{D}_V(i, j) &= \|V_i V_j^{-1} - I\|_F \\ &= \left\| \begin{pmatrix} R_i & t_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_j^T & -R_j^T t_j \\ 0 & 1 \end{pmatrix} - I \right\|_F \\ &= \left\| \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \right\|_F \\ &= \text{tr} \left( \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix}^T \right)^{\frac{1}{2}} \\ &= \text{tr} (2I - R_j R_i^T - R_i R_j^T + R_i R_j^T t_j t_j^T R_j R_i^T - t_i t_j^T R_j R_i^T - R_i R_j^T t_j t_i^T + t_i t_i^T)^{\frac{1}{2}} \\ &= (\text{tr}(2I) - \text{tr}(R_j R_i^T) - \text{tr}(R_i R_j^T) + \text{tr}(R_i R_j^T t_j t_j^T R_j R_i^T) - \text{tr}(t_i t_j^T R_j R_i^T) - \text{tr}(R_i R_j^T t_j t_i^T) + \text{tr}(t_i t_i^T))^{\frac{1}{2}} \\ &= (\text{tr}(2I) - \text{tr}(R_j R_i^T) - \text{tr}(R_i R_j^T) + \text{tr}(t_j t_j^T) - \text{tr}(R_i^T t_i t_j^T R_j) - \text{tr}(R_j^T t_j t_i^T R_i) + \text{tr}(t_i t_i^T))^{\frac{1}{2}} \\ &= (\text{tr}(2I) - \text{tr}(R_i R_j^T) - \text{tr}(R_j R_i^T) + \text{tr}(t_i t_i^T) - \text{tr}(R_j^T t_j t_i^T R_i) - \text{tr}(R_i^T t_i t_j^T R_j) + \text{tr}(t_j t_j^T))^{\frac{1}{2}} \\ &= \mathcal{D}_V(j, i) \end{aligned}$$

The fourth equality holds because, for any matrix  $A$ , we have  $\|A\|_F = \text{tr}(AA^T)^{\frac{1}{2}}$ . The sixth equality is a result of the linearity of the trace operator. The seventh equality follows from the cyclic property of the trace, for instance,  $\text{tr}(R_i R_j^T t_j t_j^T R_j R_i^T) = \text{tr}(t_j t_j^T R_j R_i^T R_i R_j^T) = \text{tr}(t_j t_j^T)$ .

**Triangle Inequality:** For views  $i, j$ , and  $k$ , we follow the same definitions of  $A_{i,j}, B_{i,j}, A_{j,k}, B_{j,k}, A_{k,i}, B_{k,i}, u_{j,k}$ , and  $u_{k,i}$  as in the proof of the triangle inequality in Theorem C.3. Starting from the fourth equality in the proof of symmetry, we have:

$$\begin{aligned} \mathcal{D}_V(i, j) &= \text{tr} \left( \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_i R_j^T - I & -R_i R_j^T t_j + t_i \\ 0 & 0 \end{pmatrix}^T \right)^{\frac{1}{2}} \\ &= \text{tr} ((R_i R_j^T - I)(R_i R_j^T - I)^T + (-R_i R_j^T t_j + t_i)(-R_i R_j^T t_j + t_i)^T)^{\frac{1}{2}} \\ &= \text{tr} (R_i^T ((R_i R_j^T - I)(R_i R_j^T - I)^T + (-R_i R_j^T t_j + t_i)(-R_i R_j^T t_j + t_i)^T) R_i)^{\frac{1}{2}} \\ &= \text{tr} ((R_j^T - R_i^T)(R_j^T - R_i^T)^T + (-R_j^T t_j + R_i^T t_i)(-R_j^T t_j + R_i^T t_i)^T)^{\frac{1}{2}} \\ &= \text{tr} (A_{i,j} A_{i,j}^T + B_{i,j} B_{i,j}^T)^{\frac{1}{2}} \\ &= \text{tr} ((A_{j,k} + A_{k,i})(A_{j,k} + A_{k,i})^T + (B_{j,k} + B_{k,i})(B_{j,k} + B_{k,i})^T)^{\frac{1}{2}} \end{aligned}$$

$$\begin{aligned}
 &= \text{tr} \left( A_{j,k} A_{j,k}^T + B_{j,k} B_{j,k}^T + A_{k,i} A_{k,i}^T + B_{k,i} B_{k,i}^T + A_{j,k} A_{k,i}^T + B_{j,k} B_{k,i}^T + A_{k,i} A_{j,k}^T + B_{k,i} B_{j,k}^T \right)^{\frac{1}{2}} \\
 &= \left( \text{tr} \left( A_{j,k} A_{j,k}^T + B_{j,k} B_{j,k}^T \right) + \text{tr} \left( A_{k,i} A_{k,i}^T + B_{k,i} B_{k,i}^T \right) + 2 \text{tr} \left( A_{j,k} A_{k,i}^T + B_{j,k} B_{k,i}^T \right) \right)^{\frac{1}{2}} \\
 &= \left( \mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2(u_{j,k}, u_{k,i})_F \right)^{\frac{1}{2}} \\
 &\leq \left( \mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2\sqrt{(u_{j,k}, u_{j,k})_F (u_{k,i}, u_{k,i})_F} \right)^{\frac{1}{2}} \\
 &= \left( \mathcal{D}_{\mathcal{V}}(j, k)^2 + \mathcal{D}_{\mathcal{V}}(k, i)^2 + 2\mathcal{D}_{\mathcal{V}}(j, k)\mathcal{D}_{\mathcal{V}}(k, i) \right)^{\frac{1}{2}} \\
 &= \mathcal{D}_{\mathcal{V}}(j, k) + \mathcal{D}_{\mathcal{V}}(k, i)
 \end{aligned}$$

The third equality holds because the trace is invariant under similarity transformations.  $\square$

## D. Experimental Details

**Datasets.** Our evaluation is conducted on three multi-view image datasets: Tanks&Temples, Mip-NeRF 360, and Deep Blending. Tanks&Temples consists of 21 diverse indoor and outdoor scenes, ranging from sculptures and large vehicles to complex large-scale environments, with intricate geometry and varied lighting conditions. Mip-NeRF 360 includes 9 scenes—5 outdoor and 4 indoor—captured in unbounded settings, allowing for 360-degree camera rotations and capturing content at varying distances. From the Deep Blending dataset, we selected 9 representative scenes that span indoor, outdoor, vegetation-rich, and nighttime environments. For all datasets, 90% of the images in each scene were allocated for training, with the remaining 10% used for testing.

**Benchmarks.** We assess the coding performance of MV-HEVC using the HTM-16.3 software<sup>2</sup>. The learning-based multi-view image codecs used as baselines, along with our proposed method, are trained under the same conditions on a shared training set and evaluated on a common test set. For the 3D Gaussian Splatting compression method (HAC), we train the 3D Gaussian representations on each scene’s test data and measure the reconstruction quality of the rendered images. The bpp is determined by dividing the size of the compressed 3D Gaussian file by the total number of pixels in the test images.

**Implementation Details.** We utilize the Adam optimizer for training with a batch size of 2. To facilitate data augmentation and optimize memory usage, each image is randomly cropped to  $256 \times 256$ . Correspondingly, the principal point in the intrinsic matrix  $K$  is adjusted to reflect the new crop. The intrinsic matrix  $K$  is given by:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix},$$

where  $f_x$  and  $f_y$  represent the focal lengths along the x and y axes, respectively, and  $c_x$  and  $c_y$  are the principal point coordinates. If the top-left corner of the crop is located at  $(p_x, p_y)$  in the original image, the updated intrinsic matrix  $K'$  becomes:

$$K' = \begin{pmatrix} f_x & 0 & c_x - p_x \\ 0 & f_y & c_y - p_y \\ 0 & 0 & 1 \end{pmatrix}.$$

**Ablation study details.** To implement *Separate*, we set the reference view images, predicted depth maps, and masks to full-zero tensors, with  $\lambda_{\text{dep}}$  set to zero. In *Concatenation*, alignment operations in the ICT modules are removed. For *W/O Mask*, we eliminate all mask-related multiplications in the ICT and DCI modules. In *W/O Dep.Pred*, the predicted depth maps are replaced with full-zero tensors. For both *Sort* and *Random*, sequences in the training and test sets are reordered accordingly.

<sup>2</sup><https://vcgit.hhi.fraunhofer.de/jvet/HTM/-/tags>



Figure 9. Visual examples of proposed alignment method and the mask from (4).

Table 3. Complexity of learning-based image codecs evaluated on images with the resolution as 978×546 in the Tanks&Temples dataset.

Codecs	MACs Enc.	MACs Dec.	Params Enc.	Params Dec.	Time Enc.	Time Dec.	Memory
HESIC+	48.16G	134.31G	17.18M	15.1M	4.35s	10.73s	2248M
MASIC	65.62G	511.34G	32.03M	30.73M	4.38s	10.78s	5202M
SASIC	91.80G	438.09G	3.57M	4.44M	0.06s	0.09s	4498M
LDMIC-Fast	37.49G	94.43G	7.73M	11.15M	0.11s	0.09s	1168M
LDMIC	30.91G	87.84G	7.73M	11.15M	4.24s	10.63s	1096M
BiSIC-Fast	1880G (Enc.+Dec.)		85.9M (Enc.+Dec.)		-	-	3552M
BiSIC	1770G (Enc.+Dec.)		78.21M (Enc.+Dec.)		-	-	3006M
3D-LMVIC	479.43G	436.16G	41.92M	36.87M	0.19s	0.18s	3164M

## E. Supplementary Alignment Experiments

Figure 9 shows visual examples of proposed alignment method along with the corresponding masks. Notably, ghosting artifacts due to occlusion, such as those involving the iron bars and the edge of the train shell, are effectively identified by the mask, aiding the codec in filtering out irrelevant information when merging features from the reference view.

## F. Visualization

In Figure 10, we present examples from the Tanks&Temples dataset to visually compare the performance of LDMIC, BiSIC, and 3D-LMVIC. The results demonstrate that 3D-LMVIC preserves more texture details and achieves higher reconstruction quality for elements like branches, humans, and text, while consuming fewer bits.

## G. Complexity Analysis

Table 3 summarizes the Multiply-Accumulate Operations (MACs), model parameters, coding speed, and memory usage of eight learning-based image codecs. These evaluations were conducted on a platform with an Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz and a GPU containing 10,752 parallel processing cores. The neural network components were executed on the GPU, while entropy coding was performed on the CPU.

Due to the absence of separate encoder and decoder implementations in the open-source code of BiSIC, we measured only its overall computational complexity. The proposed 3D-LMVIC demonstrates computational complexity within an acceptable range, comparable to the SOTA BiSIC and slightly better than BiSIC-Fast. Specifically, 3D-LMVIC achieved encoding and decoding times of 0.19s and 0.18s, respectively, ranking it among the faster methods.



Figure 10. Visual Comparison of LDMIC, BiSIC, and 3D-LMVIC on the Tanks&Temples Dataset. Compression performance is reported as bpp/PSNR/MS-SSIM.

Table 4. BDBR of 3D-LMVIC relative to HEVC.

Methods	Tanks&Temples		Mip-NeRF 360		Deep Blending	
	PSNR	MS-SSIM	PSNR	MS-SSIM	PSNR	MS-SSIM
3D-LMVIC	-20.69%	-40.75%	-14.48%	-22.06%	-17.29%	-43.06%

While the MACs of the 3D-LMVIC encoder are relatively high, they remain lower than those of BiSIC, which employs a symmetric encoder-decoder structure. For BiSIC, we estimate that the MACs for its encoder and decoder each account for approximately half of the total MACs. Additionally, the inclusion of a depth map codec in 3D-LMVIC contributes to the higher MACs and model parameter count.

## H. Supplementary Coding Performance

We present a supplementary comparison of the coding performance between the proposed 3D-LMVIC and the HEVC video coding standard. The multi-view sequences are treated as a single video and compressed using HEVC with the *lowdelay\_P* configuration and YUV444 input format. HEVC’s coding efficiency is evaluated using the HM-18.0 software<sup>3</sup>. Table 4 reports the BDBR of 3D-LMVIC relative to HEVC. On the three datasets, 3D-LMVIC consistently surpasses HEVC in both PSNR and MS-SSIM, demonstrating its effectiveness in reducing inter-view redundancy in multi-view sequences.

<sup>3</sup><https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tags>