Available online at https://jcupt.bupt.edu.cn

The Journal of China Universities of Posts and Telecommunications

April 2025, 32(2): 1 – 17

Ultra-low storage NeRF-based semantic compression and reconstruction architecture for static object videos

Zhang Zhang, Dong Chen(✉)

School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract

The exponential growth of cultural heritage documentation videos calls for new compression methods that preserve critical details while reducing storage. For static scenes, traditional frame-based compression methods struggle with the trade-off between semantic redundancy and detail preservation. To improve compression efficiency, a novel dual-mode semantic compression framework for static object videos based on neural radiance fields (NeRF) was proposed in this paper. By integrating semantic segmentation with COLMAP technology, the proposed system decouples the video stream into two semantic layers, which are the central object containing critical details and the dynamic background rich in semantic redundancy, respectively. In the proposed dual-mode framework, the focus-priority (FP) mode is designed for scenarios with high-efficiency demands, where only the NeRF-based neural representation of the primary object is preserved and compressed. For scenarios that require additional environmental context, the panorama-compatible (PC) mode synchronously compresses the H.264-encoded background streams and the primary object streams to reconstruct the full scene. Experimental results on single-artifact video data demonstrate that the proposed framework achieves a storage reduction of 20% compared with conventional methods, thus providing a flexible and controllable solution for the compression of cultural heritage documentation videos.

Keywords static object video, neural radiance fields (NeRF), semantic compression, video compression

1 Introduction

In today's digital era, the generation and dissemination of video data have experienced exponential growth. In the field of cultural heritage preservation, high-fidelity visual recording has become a core necessity. The preservation, restoration, and exhibition of artifacts involve vast amounts of visual data, which not only encapsulate rich historical information but also document the appearance, texture, and other intricate details of cultural relics, providing an irreplaceable digital foundation for historical authenticity preservation and virtual restoration.

However, the ever-growing volume of video data in cultural heritage preservation presents significant challenges in terms of storage, transmission, and management. This problem mirrors a prevalent pattern anticipated in future networked systems, where video content is projected to constitute the predominant data

Special Issue: The 27th Annual Meeting of The China Association for Science and Technology

Corresponding author: Dong Chen, E-mail: Dongchen@bupt.edu.cn

DOI: 10.19682/j.cnki.1005-8885.2025.0018

traffic component[1]. Such development underscores the urgent need for intelligent frameworks capable of supporting efficient data processing and allocation mechanisms. Traditional video compression techniques struggle to achieve high compression rates while preserving critical details, often leading to information loss and quality degradation. Current mainstream video compression standards, such as H.264 and H.265[2-3], employ two-dimensional (2D) compression methods that encode each frame pixel-wise. These approaches process entire frames indiscriminately, resulting in inefficient handling of static object regions and introducing substantial semantic redundancy. When applied to scenarios where static artifacts are the primary subjects and dynamic backgrounds serve as secondary elements, these methods reveal substantial efficiency bottlenecks. On one hand, the uniform quantization strategy across entire frames leads to the loss of high-frequency details in static regions. On the other hand, random noise and low-semantic information in dynamic backgrounds generate inefficient bitstreams. This "undifferentiated compression" paradigm results in storage systems being burdened with redundant data, forcing users to compromise between compression rates and detail fidelity. Although existing research has attempted to optimize bitrate allocation through region of interest (ROI) encoding, its support is limited to coarse rectangular region partitioning.

Contemporary video compression research is undergoing a paradigm shift from conventional pixel-fidelity standards to semantic-aware coding paradigms. Emerging semantic-based compression methods such as semantic-mining-then-compression (SMC++)[4] and controllable video compression with multimodal generative models (M3-CVC)[5] utilize end-to-end architectures to capture cross-frame semantic correlations via masked modeling and multimodal fusion. While these methods successfully reduce bitrates by eliminating semantically redundant features in general video content, their reliance on scene-level semantic abstraction inherently neglects explicit object-level modeling. This limitation becomes particularly detrimental when processing cultural heritage materials characterized by static objects of high semantic significance, such as historically preserved ceramic vessels that demand pixel-accurate preservation of morphological details.

In recent years, a novel 3-dimensional (3D) scene modeling and rendering technique known as NeRF[6] has emerged in the fields of computer vision and computer graphics, offering a new technical foundation to surpass existing compression paradigms. By leveraging multi-layer perceptron (MLP) to learn from multiple 2D images captured from different viewpoints, NeRF models the propagation of light in 3D space, enabling the synthesis of photorealistic renderings from arbitrary viewpoints. In other words, NeRF implicitly stores 3D object information within an MLP model. Subsequent advancements, such as instant neural graphics primitives (Instant-NGP)[7], accelerate spatial feature querying through Hash encoding, while context-based NeRF compression (CNC)[8] introduces quantization-aware training to reduce model storage overhead. These developments have transformed NeRF from a rendering tool into a potential high-efficiency compression medium. Such techniques are particularly suitable for multi-view videos of static objects, as they convert temporal redundancy into compact spatial representations.

Building on these technological advancements, a dual-mode NeRF-based semantic compression framework for static object videos is proposed, enabling object-level semantic compression. Unlike semantic-based compression methods that rely on temporal modeling or general-purpose features, NeRF offers a natural advantage in static object centric scenarios by converting temporal redundancy into compact spatial representations. Its neural representation implicitly captures fine-grained geometry and appearance from multi-view inputs, making it particularly suitable for high-fidelity artifact modeling. The core innovation of this approach lies in effectively leveraging the unique semantic structure of static object videos: central artifacts typically possess stable geometric characteristics and high information value, whereas background regions exhibit low semantic density and dynamic variation. To achieve this goal, the framework

first applies boundary refine iterative attention artificial intelligence (BRIA AI)'s remove background version 2.0 (RMBG-2.0) segmentation model[9] to decouple the semantic components of the video stream. For the extracted central object, COLMAP[10] is employed to estimate camera parameters, and NeRF is used to perform implicit 3D modeling. By integrating contextual models with Hash encoding, a compact neural representation is achieved.

For background processing, a mean color space normalization preprocessing strategy is proposed, where the segmented central object region is replaced with its spatiotemporal average color to generate a pseudo-static sequence with low entropy characteristics. The segmentation edges are then smoothed using Gaussian blur before encoding with H.264. The mean color space normalization preprocessing enables the intra-frame prediction module of the H.264 encoder to establish more optimal reference pixel relationships while preserving the original average color values to ensure visual consistency during scene reconstruction. The proposed framework allows the H.264 encoder to fully leverage its intra-frame prediction and motion compensation advantages. Compared to directly encoding the original background, this strategy significantly reduces encoding complexity while ensuring scene coherence by retaining the average color parameters.

The proposed framework provides two switchable compression modes. In the FP mode, the system transmits only the neural representation of the central object, achieving maximum compression efficiency. When the PC mode is enabled, the system synchronously transmits the preprocessed background sequence encoded via H.264. Both modes share the same representation system, allowing users to flexibly choose the appropriate mode based on the application scenario. Users can opt for either multi-view rendering of the central object only or full reconstruction of the original scene, including the background environment. This design overcomes the rigidity of traditional compression methods and, for the first time in the field of cultural heritage preservation, enables controllable object-level semantic compression.

The remainder of this paper is structured as follows. Sect. 2 introduces related technologies. Sect. 3 presents the proposed semantic compression and reconstruction system architecture, detailing its technical components. Sect. 4 describes the experimental setup and showcases the simulation comparison results. Sect. 5 concludes the study.

2 Related work

2.1 Video coding and decoding

Video encoding and decoding technology serves as the core of modern multimedia communication systems, aiming to reduce storage and transmission bandwidth through efficient compression techniques while maintaining video quality. The classical H.264 advanced video coding (H.264/AVC) standard[2] significantly reduces video data redundancy by employing intra-frame prediction, inter-frame prediction, and motion compensation based on macroblock partitioning, marking a milestone in video compression technology. Building upon this foundation, H.265 high efficiency video coding (H.265/HEVC)[3] further optimizes the coding unit structure by introducing the coding tree unit (CTU) and adopting highly efficient entropy coding techniques such as context-adaptive binary arithmetic coding (CABAC), achieving approximately a 50% reduction in bitrate compared to H.264 at the same quality level[11].

However, these standards face fundamental limitations in compressing videos containing static objects. Firstly, motion estimation based on pixel similarity struggles to capture the cross-frame geometric invariance of static objects, leading to redundant encoding. Secondly, existing codecs lack explicit modeling of video semantic structures, and ROI encoding only supports coarse rectangular region partitioning, making it unsuitable for the precise compression of irregularly shaped high-value objects such as cultural artifacts. Recent studies have attempted to overcome these challenges. For instance, end-to-end deep learning-based coding frameworks[12]

extract high-level semantic features using neural networks. However, their optimization objectives remain constrained by 2D pixel-level reconstruction quality. Various semantic communication-based video coding methods have been proposed[13], significantly enhancing video compression efficiency. Nevertheless, these approaches rely on data-driven implicit semantic extraction, which struggles to effectively utilize object-level prior knowledge in static object videos. These limitations indicate that current methods still contain unresolved deep semantic redundancy, highlighting the urgent need for a novel compression paradigm that integrates prior semantic guidance with neural representations.

2.2 NeRF

Traditional NeRF relies on MLP for implicit scene representation, requiring multiple MLP queries per pixel to generate images, leading to high computational costs and slow rendering speeds[6]. Moreover, as the complexity of the scene increases, the model needs to store more feature embeddings, significantly increasing storage requirements. To address the issue of high storage demands, researchers have proposed a series of storage compression technologies, such as binary radiance fields (BiRF), which uses binary representation to quantize feature embeddings, effectively reducing storage space occupation and achieving high-quality rendering in static scenes[14]. Vector quantized radiance fields (VQRF) further compresses feature grids through vector quantization technology, eliminating redundant features and streamlining storage requirements[15]. Although these methods have achieved certain effects in storage compression, they often face a decline in rendering quality, especially in high-resolution scenes[16].

On the other hand, researchers have proposed hybrid representation methods that combine explicit representations, such as voxel grids and Hash tables, with implicit MLP representations, effectively reducing computational burdens and enhancing rendering speeds, effectively promoting the application of NeRF in 3D modeling. Instant-NGP, as a significant breakthrough in this field, uses multi-resolution Hash embedding technology, storing sparse multi-resolution feature grids in Hash tables, reducing reliance on MLP, and achieving real-time rendering. Although Hash embedding significantly improves computational efficiency, the storage demand for high-resolution feature grids remains substantial[7]. Hybrid neural radiance fields (Hyb-NeRF) proposes a multi-resolution hybrid encoding method that uses memory-efficient learnable positional features at coarse resolutions and Hash grid encoding at fine resolutions, achieving a balance between rendering efficiency and storage costs. Compared to Instant-NGP, Hyb-NeRF exhibits greater compactness in storage requirements while maintaining high rendering quality[17]. CNC introduces context modeling technology for efficient compression of multi-resolution Hash embeddings, utilizing level-wise context models (LWCMs) and dimension-wise context models (DWCMs) to accurately predict the probabilities of feature embeddings, thereby reducing the redundancy of stored information[8].

2.3 Semantic-based compression methods

In recent years, semantic-based compression has emerged as a promising alternative to traditional video coding paradigms focused on pixel-level fidelity. These methods aim to preserve high-level features relevant to downstream tasks, enabling more intelligent and goal-oriented transmission. SMC + +[4] adopts a masked modeling strategy with a blueprint-guided transformer to exploit semantic redundancy and improve temporal alignment. M3-CVC[5] introduces a multimodal compression framework that leverages large multimodal models and conditional diffusion networks to achieve text-guided, ultra-low-bitrate video reconstruction.

In addition, an RNN-based compression framework[18] combines convolutional layers, long short-term memory (LSTM) units, and generalized divisive normalization (GDN) for efficient feature extraction and iterative reconstruction, outperforming traditional codecs like H.264 and H.265. A study on 360° neural video compression[19] further emphasizes the importance of spatial context modeling and projection format selection in immersive scenarios.

While these approaches perform well in dynamic or

general-purpose videos, they are mainly designed for scene-level semantic abstraction and often lack fine-grained control. In particular, they provide limited support for object-level compression, which is critical for applications like cultural heritage preservation that require accurate retention of geometry and texture in static central objects. This underscores the need for structure-aware and object-centric semantic compression frameworks.

2.4 COLMAP

As a benchmark open-source multi-view 3D reconstruction toolchain, COLMAP[10] provides a reliable 3D perception foundation for cross-disciplinary research areas such as cultural heritage digitization and medical imaging modeling through its modular architecture. This toolchain integrates structure-from-motion (SfM) and multi-view stereo (MVS) techniques, enabling the recovery of high-precision camera parameters and scene geometry from unstructured image collections. Its output data format has become the standardized input interface for emerging representation methods such as NeRF[10].

In the practice of cultural heritage digitization, COLMAP demonstrates unique engineering value. Its robust feature matching algorithm effectively addresses the sparse correspondence problem on low-texture surfaces, while its parallel optimization framework for bundle adjustment ensures numerical stability in large-scale scene reconstruction[20]. These characteristics make COLMAP the preferred tool for high-precision 3D modeling of cultural heritage. Notably, the camera pose parameters generated by COLMAP provide crucial geometric constraints for NeRF, allowing NeRF to achieve implicit 3D reconstruction without relying on depth sensors[21].

This study inherits and extends this technological paradigm by extracting temporally consistent camera parameter sequences from multi-view videos of static objects using COLMAP, thereby establishing a geometry-aware foundation for subsequent neural compression model training. Compared to directly using calibration equipment, this video stream-based dynamic pose estimation approach not only retains the flexibility of handheld capturing but also ensures reconstruction accuracy through motion trajectory smoothing constraints.

2.5 Semantic segmentation

In the field of image processing, semantic segmentation plays a pivotal role by enabling pixel-level scene understanding, such as identifying different objects and boundaries in urban street images. segmentation network (SegNet)[22], as one of the early pioneers, employs an encoder-decoder architecture to progressively restore spatial information, achieving outstanding segmentation performance. In recent years, mask region-based convolutional neural network (Mask R-CNN)[23], an extension of faster region-based convolutional neural network (Faster R-CNN), has introduced an additional branch within the object detection framework to predict segmentation masks for each ROI, accomplishing both object detection and precise segmentation.

DeepLab V3 +[24] builds upon its predecessors by integrating atrous convolution and multi-scale feature extraction, significantly enhancing segmentation accuracy, especially in complex scenes. Kirillov et al.[25] introduced the segment anything model (SAM), renowned for its powerful zero-shot segmentation capability. By leveraging pretraining on large-scale datasets, SAM can rapidly adapt to diverse segmentation tasks and deliver exceptional performance.

RMBG-2.0, developed by BRIA AI[9], is based on the bilateral reference network (BiRefNet) architecture and integrates both global semantic information and local gradient information, enabling precise identification and segmentation of foreground and background elements in images. In the task of foreground-background separation, RMBG-2.0 has achieved state-of-the-art (SOTA) performance, making it an advanced model in the field of image background removal.

3 System model

3.1 Preliminaries

In the field of computer graphics, rendering refers to the process of converting 3D models into 2D images,

while inverse rendering is the process of recovering 3D information from 2D images. Traditional rendering methods primarily focus on surface rendering, such as creating 3D objects through surface representations like polygonal meshes or non-uniform rational b-splines (NURBS) surfaces. However, volume rendering is a different approach that considers the scene as a volume composed of many particles, each with its own density and color. NeRF technology is based on the concept of volume rendering, representing the scene as a continuous volume field where each point has its density and color.

To generate images from this representation, rays must be calculated through the camera's intrinsic and extrinsic parameters, and volume rendering is performed along these rays. The camera intrinsic matrix $\boldsymbol{K}$ can be represented as

$$\boldsymbol{K}=\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where f_x and f_y are the focal lengths along the image's x and y axes, respectively. c_x and c_y are the coordinates of the principal point on the image plane. The camera extrinsic matrix $[\boldsymbol{R}|\boldsymbol{t}]$ includes the rotation matrix $\boldsymbol{R}\in\mathbb{R}^{3\times3}$ and the translation vector $\boldsymbol{t}\in\mathbb{R}^3$, which are used to transform points from the world coordinate system to the camera coordinate system. Specifically, a point $\boldsymbol{P}_{\mathrm{w}}$ in the world coordinate system can be transformed into a point $\boldsymbol{P}_{\mathrm{c}}$ in the camera coordinate system using

$$\boldsymbol{P}_{\mathrm{c}}=\boldsymbol{R}\boldsymbol{P}_{\mathrm{w}}+\boldsymbol{t} \tag{2}$$

Subsequently, after normalization and projection through the intrinsic matrix , the point in the camera coordinate system is projected $\boldsymbol{K}$ onto the image plane

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=\boldsymbol{K}\begin{bmatrix} P_{cx} \\ P_{cy} \\ P_{cz} \end{bmatrix} \tag{3}$$

where (u,v) are the pixel coordinates on the image plane.

Next, to generate an image, rays are calculated from each pixel. The ray's origin is the camera's optical center, and its direction is determined by the pixel coordinates, converted into a vector $\boldsymbol{d}$ in the camera coordinate system. The parametric equation of the ray can be expressed as

$$\boldsymbol{r}(\tau)=\boldsymbol{o}+\tau\boldsymbol{d} \tag{4}$$

where $\boldsymbol{o}$ is the position of the camera's optical center, τ is a parameter along the ray, and $\boldsymbol{d}$ is the direction vector of the ray. Specifically, for a pixel (u,v) on the image plane, its corresponding ray direction $\boldsymbol{d}$ can be calculated using

$$\boldsymbol{d}=\boldsymbol{K}^{-1}\begin{bmatrix} u-c_x \\ v-c_y \\ f_x \end{bmatrix} \tag{5}$$

where $\boldsymbol{K}^{-1}$ is the inverse of the intrinsic matrix.

After calculating the rays, volume rendering is performed along the rays to synthesize the image. The core of volume rendering is to calculate the color of each pixel by integrating the product of color, density, and transmittance along the ray's path.

$$C(\boldsymbol{r})=\int_{p_{\mathrm{n}}}^{p_{\mathrm{f}}}T(p)\sigma(p)\boldsymbol{c}(p)\,\mathrm{d}p \tag{6}$$

where $T(p)=\exp\left(-\int_{p_{\mathrm{n}}}^{p}\sigma(s)\,\mathrm{d}s\right)$ is the transmittance from the ray's origin to point p, $\sigma(p)$ is the volume density at point p, $\boldsymbol{c}(p)$ is the color radiance at point p, p_{n} and p_{f} are the near and far bounds of the ray, respectively.

The core principle of NeRF lies in training an MLP network to predict the color and density of each particle in space. This network accepts the spatial coordinates $\boldsymbol{s}\in\mathbb{R}^3$ and view direction $\boldsymbol{d}\in\mathbb{R}^3$ of a particle, predicting the volume density $\sigma(\boldsymbol{s},\boldsymbol{d})\in\mathbb{R}$ and color $\boldsymbol{c}(\boldsymbol{s},\boldsymbol{d})\in\mathbb{R}^3$ of that particle. Therefore, for each pixel, sampling along the ray from the camera origin through the pixel center is performed, and the color and density of some particles along the ray are predicted based on the MLP model, allowing the prediction of the pixel's color.

During the training process, multiple pixel points are sampled from the dataset, and rays through these pixels are calculated using the corresponding camera intrinsic and extrinsic parameters to find the intersection points with particles in the scene. The MLP network then predicts the color and density at these intersection points. These predicted values are subsequently used for volume rendering to calculate the predicted color of the pixel. The system aims to minimize the difference

between the rendered image and the actual observed image, achieving this by minimizing the mean squared error (MSE) loss function.

$$L = \sum_{s_{\text{ray}}} \| C_{\text{r}} - C_{\text{t}} \|^2 \tag{7}$$

where C_{r} is the color predicted by the network, s_{ray} denotes the set of rays, and C_{t} is the color of the pixel in the input image. The MLP network weights are updated through backpropagation. In this way, NeRF can learn and reconstruct a continuous volume representation of the entire scene from a limited number of viewpoints, allowing the synthesis of high-quality images from new viewpoints.

Building upon NeRF technology, Instant-NGP introduces a hybrid representation method that significantly enhances rendering speed and reduces computational burden by combining explicit representations, such as voxel grids and Hash tables, with implicit MLP representations. In Instant-NGP, the input to the MLP network differs from the traditional NeRF approach, employing multi-resolution Hash encoding to optimize the encoding of input features. That is, for given spatial coordinates $\boldsymbol{s} \in \mathbb{R}^3$ and view direction $\boldsymbol{d} \in \mathbb{R}^3$ of a particle, Instant-NGP obtains its feature representation F through multi-resolution Hash tables.

In the proposed framework, the spatial domain is divided into grids across multiple levels of resolution. The resolution of the grid at level l is defined as

$$N_l = \lfloor N_{\min} b^l \rfloor \tag{8}$$

$$b = \mathrm{e}^{\frac{\ln N_{\max} - \ln N_{\min}}{L-1}} \tag{9}$$

where $N_{\min}$ represents the coarsest grid resolution, $N_{\max}$ represents the finest grid resolution, and L is the number of resolution levels. Using this approach, the grid resolution grows geometrically, allowing the method to capture both global features and fine-grained local details of the scene. For a given input coordinate $\boldsymbol{s} \in \mathbb{R}^3$, it is first normalized and quantized into the grid of the current resolution level as

$$\left.\begin{aligned} \lfloor x_l^{\text{i}} \rfloor &= \lfloor x N_l \rfloor \\ \lceil x_l^{\text{i}} \rceil &= \lceil x N_l \rceil \end{aligned}\right\} \tag{10}$$

where x denotes the normalized and quantized spatial coordinate, and x_l^{i} is integer coordinate in the current resolution grid. The quantized coordinates are then mapped to indices in the Hash table using the following Hash function.

$$h(x) = \left(\bigoplus_{d_{\text{f}}=1}^{D} x_{d_{\text{f}}} \pi_{d_{\text{f}}} \right) \bmod T \tag{11}$$

where $\oplus$ denotes the bitwise XOR operation, $x_{d_{\text{f}}}$ is the integer coordinate in the d_{f}th dimension, $\pi_{d_{\text{f}}}$ is a prime number associated with each dimension to reduce interdimensional correlations, and T is the size of the Hash table, which determines the maximum storage capacity. This Hash function provides a pseudo-random mapping, significantly reducing storage conflicts and ensuring efficient feature retrieval.

To obtain a continuous feature representation for a spatial point, feature vectors from the Hash table are interpolated linearly. The resulting feature vector for the spatial point is computed as a weighted sum of the feature vectors at the adjacent grid vertices.

$$\boldsymbol{f}_{\text{sum}} = \sum_{l} w_l \boldsymbol{f}_l \tag{12}$$

$$w_l = x_l^{\text{i}} - \lfloor x_l^{\text{i}} \rfloor \tag{13}$$

where $\boldsymbol{f}_l$ is the feature vector of an adjacent vertex, and w_l is the interpolation weight for that vertex. The interpolated feature vectors from all resolution levels are concatenated to form the final input vector to the MLP.

$$\boldsymbol{F} = \gamma(\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_l, \ldots, \boldsymbol{f}_L, \boldsymbol{d}) \tag{14}$$

where $\boldsymbol{f}_l$ is the interpolated feature vector at level l, and $\boldsymbol{d}$ is the auxiliary input representing the viewing direction. $\gamma(\cdot)$ is defined as a generalized feature aggregation operator that replaces traditional concatenation for enhanced flexibility and clarity. This multi-resolution Hash encoding enables Instant-NGP to directly query input features from explicitly stored feature vectors, significantly reducing the size of the MLP and thereby improving rendering speed and computational efficiency.

Additionally, Instant-NGP introduces an occupancy grid to further optimize ray sampling efficiency. In traditional NeRF, sampling is performed uniformly across the entire scene, which often leads to redundant computations in empty or low-density regions. To address this, Instant-NGP divides the scene into fixed-sized cubic voxels, with each voxel storing a Boolean value indicating its occupancy status. The occupancy

status is determined by

$$O(p)=\begin{cases}1; & \sigma(p)>\sigma_{\text{th}}\\ 0; & \text{otherwise}\end{cases} \tag{15}$$

where $O(p)$ is the occupancy status of the voxel, $\sigma(p)$ is the density of points within the voxel, and σ_{th} is the density threshold. During training, the occupancy grid is dynamically updated based on the density distribution. Ray tracing then only samples points within occupied voxels, skipping unoccupied regions, thereby reducing computational overhead.

Despite these improvements, Instant-NGP still relies on floating-point feature storage, which results in high memory demands. To address this limitation, BiRF introduces a storage-efficient radiance field representation based on binary feature encoding. BiRF constrains feature values to either $+1$ or -1 as

$$\theta'=\operatorname{sgn}\theta=\begin{cases}+1; & \theta\geqslant 0\\ -1; & \text{otherwise}\end{cases} \tag{16}$$

where θ represents the real-valued feature parameters, and θ' denotes the binarized feature parameters. During training, BiRF does not directly optimize the binary parameters θ'. Instead, it retains real-valued parameters θ for gradient updates. Since the sign function is almost everywhere non-differentiable, BiRF employs a straight-through estimator to propagate gradients, enabling effective training of binary feature parameters. This approach significantly reduces storage requirements while preserving high-quality scene reconstruction.

3.2 Dual-mode semantic compression framework

The proposed dual-mode compression framework realizes efficient compression and controllable reconstruction of still-life video through hierarchical semantic decoupling and neural-traditional hybrid coding strategy. As shown in Fig. 1, the system includes the following core processing modules: encoder module and decoder module.

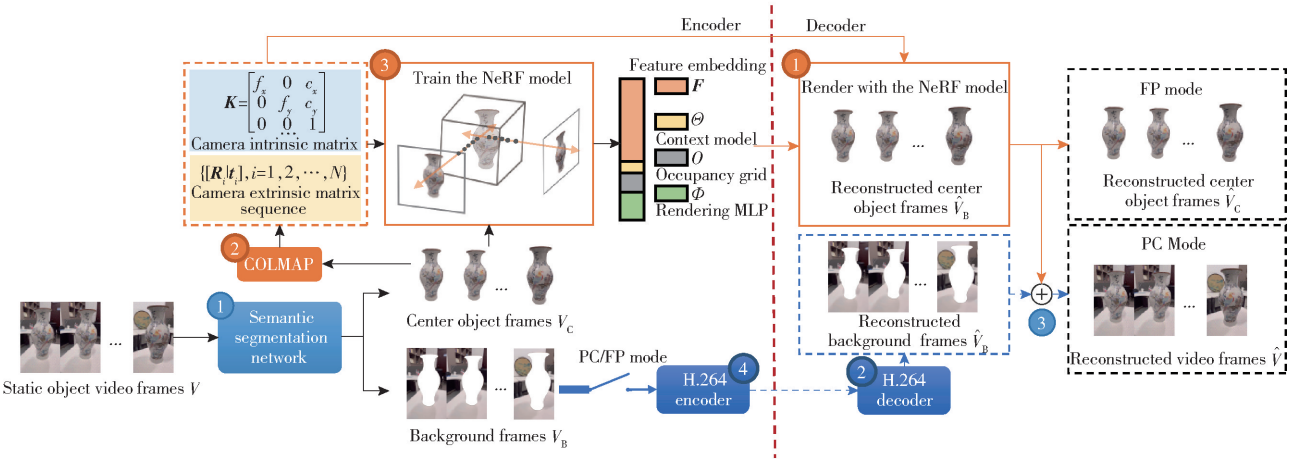

Fig. 1 Framework of the proposed dual-mode semantic compression

1) Encoder module

Input Video $V=\{I_i\}_{i=1}^{N}$ consist of N frames, where I_i denotes the ith frame of the video.

Step 1 Segment the video frames into the central object part $V_{\text{C}}=\{I_i^{\text{C}}\}$ and the background part $V_{\text{B}}=\{I_i^{\text{B}}\}$.

Step 2 Based on the video frames of the central object part V_{C}, use the COLMAP algorithm to predict the camera intrinsic matrix $\boldsymbol{K}$ and sequence of camera extrinsic parameters $\{[\boldsymbol{R}_i|\boldsymbol{t}_i], i=1,2,\ldots,N\}$.

Step 3 Establish a cultural relic video dataset and train the NeRF model. Its parameter set includes: feature embeddings $\boldsymbol{F}$, parameters $\boldsymbol{\Theta}$ of the context model, occupancy grid information O, parameters $\boldsymbol{\Phi}$ of the rendering MLP model.

Step 4 Differentiate the background video according to the working mode. If the FP mode is enabled, the background video part will be discarded without encoding. If the PC mode is enabled, the space-time average color standardization will be performed on the background video part, and then H.264 encoding will be performed to generate the background bitstream.

Output Feature embeddings $\boldsymbol{F}$, parameters $\boldsymbol{\Theta}$ of the context model, occupancy grid information O, parameters $\boldsymbol{\Phi}$ of the rendering MLP model, camera intrinsic parameters $\boldsymbol{K}$, a sequence of camera extrinsic parameters $\{[\boldsymbol{R}_i \mid \boldsymbol{t}_i], i = 1, 2, \ldots, N\}$ for N video frames, and the bitstream of background video frames E_{B} (if the PC mode is enabled).

2) Decoder module

Input Feature embeddings $\boldsymbol{F}$, parameters $\boldsymbol{\Theta}$ of the context model, occupancy grid information O, parameters $\boldsymbol{\Phi}$ of the rendering MLP model, camera intrinsic parameters $\boldsymbol{K}$, a sequence of camera extrinsic parameters $\{[\boldsymbol{R}_i \mid \boldsymbol{t}_i], i = 1, 2, \ldots, N\}$ for N video frames, and the bitstream of background video frames E_{B} (if the PC mode is enabled).

Step 1 Based on F, $\boldsymbol{\Theta}$, O, $\boldsymbol{\Phi}$, camera intrinsic parameters $\boldsymbol{K}$, and the sequence of camera extrinsic parameters $\{[\boldsymbol{R}_i \mid \boldsymbol{t}_i], i = 1, 2, \ldots, N\}$ for N video frames, recover the NeRF model, perform rendering, and reconstruct the video frames of the central object part $\hat{V}_{\mathrm{C}}$.

Step 2 If the PC mode is enabled and the encoded results of background video frames are received, use the H.264 algorithm to decode and reconstruct the video frames of the background part $\hat{V}_{\mathrm{B}}$, then proceed to the next step. Otherwise, directly output the video frames $\hat{V}_{\mathrm{C}}$, and the decoding process ends.

Step 3 Based on the reconstructed video frames of the central object part $\hat{V}_{\mathrm{C}}$ and the background part $\hat{V}_{\mathrm{B}}$, concatenate to reconstruct the final video frames $\hat{V}$.

Output Reconstructed video frames $\hat{V}_{\mathrm{C}}$ or $\hat{V}$.

The innovation of the architecture is embodied in two aspects. Firstly, the multi-view video of the central object is compressed into a compact implicit representation through NeRF. Secondly, the dual-mode design allows users to flexibly switch between FP mode and PC mode according to the application scenario, thus realizing the dynamic balance between storage efficiency and scene integrity.

3.3 Semantic segmentation and background coding

To achieve semantic decoupling of the video stream, the proposed framework uses the RMBG-2.0 model for pixel-level object segmentation. The RMBG-2.0 model is based on the BiRefNet architecture, which works in collaboration with the localization module (LM) and the reconstruction module (RM) to achieve high-precision segmentation. The LM generates coarse target location information. For an input video frame sequence $I \in \mathbb{R}^{N\times 3\times H\times W}$, were N denotes the number of frams, 3 corresponds to red, green, blue (RGB) color channels and $H \times W$ represents the spatial resolution of each frame. The model first uses a transformer encoder to extract multi-scale features with spatial resolutions ranging from 1/4 to 1/32. The first three feature layers are processed through lateral connections in the decoder, and the final encoded feature undergoes global average pooling and a fully connected layer to generate a semantic heatmap.

The RM module innovatively introduces a bilateral reference mechanism, where both the original resolution image and the gradient prior information are used as reference signals. Through feature concatenation and channel attention, this approach enhances fine details. This design effectively solves the aliasing effects that traditional segmentation methods often introduce along the edges of artifacts (such as the patterns on bronze items and crack patterns in ceramics), enabling precise reconstruction. Using the semantic segmentation provided by RMBG-2.0, the video stream is effectively decoupled into two semantic layers: the central object and the background, laying the foundation for NeRF dataset creation for the central object and efficient encoding of the background.

For the segmented background, a mean color space normalization preprocessing strategy is prosposed to optimize H.264 encoding efficiency. Specifically, after removing the central object, the system statistically computes the spatiotemporal average color of the background and replaces the original central object region with this value. For the segmented background sequence $\{I_i^{\mathrm{B}}\}_{i=1}^{N}$, the system first calculates the spatial mean color $\overline{C}_i$ for each background region I_i^{B} in the ith frame.

$$\overline{C}_i = \frac{1}{|M_{\mathrm{B}}^i|} \sum_{(x,y)\in M_{\mathrm{B}}^i} I_i^{\mathrm{B}}(x,y) \tag{17}$$

where M_B^i represents the background mask region in the ith frame. The distribution of the mean colors across all

frames is then analyzed, and the most frequently occurring color $\overline{C}$ is selected as the filling baseline. Finally, the segmented central region of each frame is uniformly filled with $\overline{C}$, generating a pseudo-static background sequence $\{\hat{I}_i^{\mathrm{B}}\}_{i=1}^{N}$

Additionally, Gaussian Blur is applied to the segmentation edges to prevent visual artifacts caused by hard segmentation. The preprocessed background is then encoded using H.264. Since the resulting video frames exhibit low entropy characteristics, the complexity of H.264 encoding is significantly reduced. Specifically, the large, uniform color blocks facilitate the construction of DC-mode intra-frame predictions, and the stability of colors across time effectively reduces motion estimation computational overhead. These properties enable the H.264 encoder to fully leverage its intra-frame prediction and motion compensation capabilities, significantly reducing the encoded data size. During encoding, the mean color distribution of each frame are retained as metadata, which occupies an extremely small amount of data.

3.4 Camera parameter prediction and NeRF model training and rendering

After completing semantic segmentation, the proposed framework utilizes the COLMAP technique to estimate camera parameters and further construct a NeRF dataset based on the segmented central object video V_{C}. First, the segmented central object video frames V_{C} are used as input, and the central object region is extracted from each frame. Since the background has been removed, COLMAP can focus exclusively on the feature detection and matching of the central object, thereby improving the accuracy of camera parameter estimation.

Specifically, COLMAP first employs the scale-invariant feature transform (SIFT) algorithm to detect key points in each frame, denoted as $P_{\mathrm{k}} = \{p_{il}\}$, where p_{il} represents the lth key point in frame I_i. The system then establishes correspondences between frames through feature matching, forming a feature correspondence set.

$$M = \{(p_{ik}, p_{jl})\} \tag{18}$$

where k and l denote the indices of the detected key points within the ith and jth frames, respectively.

Next, based on the feature matching results, the system applies incremental SfM to estimate the camera intrinsic matrix $\boldsymbol{K}$ and the sequence of camera extrinsic parameters.

$$\{[\boldsymbol{R}_i | \boldsymbol{t}_i], i = 1, 2, \ldots, N\} \tag{19}$$

Simultaneously, a sparse 3D point cloud is generated.

Finally, bundle adjustment is performed to jointly optimize the camera parameters and the 3D point cloud, minimizing reprojection errors to ensure the accuracy of camera parameters.

After obtaining the camera parameters, this framework further constructs the NeRF training dataset. First, the segmented central object video frames are aligned with their corresponding camera parameters to generate the necessary NeRF input data. Each frame's central object region serves as the NeRF training view, while the camera parameters define the geometric transformations of the viewpoints. Next, the preprocessed image frames and camera parameters are organized into the NeRF training dataset, which includes the image frames V_{C}, the camera intrinsic parameters $\boldsymbol{K}$, the camera extrinsic parameters $\{[\boldsymbol{R}_i | \boldsymbol{t}_i], i = 1, 2, \ldots, N\}$ as well as image resolution information and other metadata.

In the proposed framework, a voxel grid approach, integrated Hash encoding with an implicit MLP representation is adopted to construct the NeRF model. As illustrated in Fig. 2, this method stores features using multi-resolution 3D-2D hybrid Hash encoding, enabling efficient capture of global scene features and high-frequency details, while significantly reducing computational and storage overhead. This approach employs a 3D-2D hybrid representation, where the 3D voxel grid is responsible for capturing global scene features, while 2D intersection planes project 3D coordinates onto the xy, yz, and xz planes to supplement local high-frequency details. Furthermore, at each spatial scale, voxel grids with progressively increasing resolutions are designed to simultaneously capture global scene features and high-frequency details, thereby enhancing spatial precision while maintaining a compact storage representation.

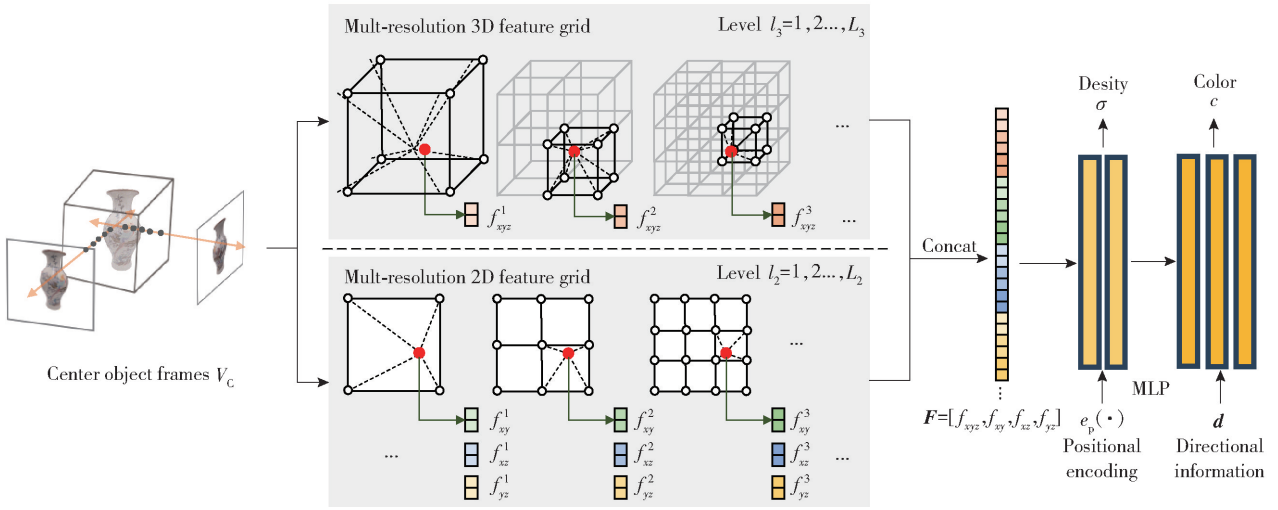

Fig. 2 Illustration of the multi-resolution 3D-2D hybrid Hash encoding

Notably, to reduce memory consumption, this method employs binary quantization encoding. Specifically, for a given spatial coordinate $\boldsymbol{p}_c \in \mathbb{R}^3$, the 3D feature value is computed using trilinear interpolation as

$$f_{xyz} = \{\Lambda(\boldsymbol{p}_c, \operatorname{sgn} \boldsymbol{\Phi}_\theta^{(l_3,xyz)})\}_{l_3=1}^{L_3} \tag{20}$$

where $\Lambda(\cdot)$ is defined as a generalized interpolation operator that abstracts trilinear or bilinear interpolation depending on the feature structure, $\boldsymbol{\Phi}_\theta^{(l_3,xyz)}$ represents the binary-quantized feature stored in the Hash table at the l_3th resolution level of the 3D voxel grid, L_3 is the number of resolution levels in the 3D voxel grid, and f_{xyz} is the feature extracted from the 3D voxel grid.

Next, the coordinates are projected onto the xy, yz, and xz planes, $p_{cxy} = (x, y)$, $p_{cxz} = (x, z)$, $p_{cyz} = (y, z)$, then, the feature values are obtained using bilinear interpolation.

$$f_{xy} = \{\Lambda(p_{cxy}, \operatorname{sgn} \boldsymbol{\Phi}_\theta^{(l_2,xy)})\}_{l_2=1}^{L_2} \tag{21}$$

where $\boldsymbol{\Phi}_\theta^{(l_2,xy)}$ represents the binary-quantized feature at the l_2th resolution level of the 2D feature plane, with L_2 denoting the number of resolution levels in the 2D voxel grid. Similarly, f_{xy} represents the feature extracted from the xy plane, while the extraction process for the other planes follows the same approach.

Finally, the features from both the 3D voxel grid and the 2D feature planes are concatenated into a complete feature vector.

$$\boldsymbol{F} = [f_{xyz}, f_{xy}, f_{xz}, f_{yz}] \tag{22}$$

Ultimately, the feature vector $\boldsymbol{F}$ is input into a shallow MLP network to predict the volume density σ and radiance color c of the particle

$$(\sigma, c) = \eta_d(e_p(\boldsymbol{p}_c), \boldsymbol{F}, \boldsymbol{d}) \tag{23}$$

where $\eta_d(\cdot)$ is defined as a generalized prediction module that maps encoded spatial and directional features to volume density and color. $e_p(\boldsymbol{p}_c)$ denotes the positional encoding of the input coordinate, and $\boldsymbol{d}$ represents the viewing direction.

Furthermore, to improve compression efficiency, entropy coding techniques are employ to model the distribution of binary-quantized features (values of -1 and $+1$). By reducing the information entropy of the embedded features, this approach effectively reduces the storage overhead.

Specifically, assuming that the feature values at each voxel follow a Bernoulli distribution, the probability of the lth feature value $\phi_l \in \{-1, +1\}$ can be expressed as

$$\Pr_l = \Pr(\phi_l = +1) = 1 - \Pr(\phi_l = -1) \tag{24}$$

The corresponding information content is given by

$$C_1(\Pr_l | \phi_l) = -\left(\frac{1+\phi_l}{2} \operatorname{lb} \Pr_l + \frac{1-\phi_l}{2} \operatorname{lb}(1-\Pr_l)\right) = \begin{cases} -\operatorname{lb} \Pr_l; & \phi_l = +1 \\ -\operatorname{lb}(1-\Pr_l); & \phi_l = -1 \end{cases} \tag{25}$$

By accumulating feature values across different dimensions and resolution levels of the voxel grid, the total entropy loss L_{en} can be computed. Finally, the

model's total loss function L_{T} is composed of both the reconstruction loss and the entropy loss, formulated as

$$L_{\mathrm{T}} = L_{\mathrm{MSE}} + \lambda L_{\mathrm{en}}, \tag{26}$$

where L_{MSE} represents the mean squared error (MSE) loss at the pixel level, and λ is a weighting factor that balances compression efficiency and reconstruction quality. By adjusting λ, the model can control the compression rate and its corresponding impact on reconstruction accuracy.

However, directly modeling the global probability Pr_l disregards the spatial correlation between features. Therefore, a hierarchical contextual model and a dimension-aware dependency model are integrated to dynamically estimate the probability Pr_l for each feature, thereby further reducing entropy.

The hierarchical contextual model adaptively captures fine-grained dependency relationships between multi-resolution features based on their spatial correlations. For a given voxel feature ϕ_l at the current resolution level, its probability estimation relies on the interpolated feature values from the l preceding decoded levels.

Specifically, assuming the current hierarchical level is $L_{\mathrm{C}} = 3$, the model retrieves features from the three preceding levels ($l = 1, 2, 3$) at the same spatial location and performs linear interpolation to obtain the contextual feature vector $\boldsymbol{f}_{\mathrm{c}}$. Additionally, the model introduces the global frequency statistic v_+ (i.e., the probability of feature values being $+1$ across the entire dataset as auxiliary information. The concatenation of contextual features and frequency statistics is then fed into a lightweight two-layer MLP, $\eta_{\mathrm{c}}(\cdot)$, which outputs the probability estimation Pr_l for the current feature.

$$\mathrm{Pr}_l = \eta_{\mathrm{c}}(\gamma(\boldsymbol{f}_{\mathrm{c}}, v_+)) \tag{27}$$

To prevent decoding order conflicts, the model strictly follows a hierarchical decoding procedure, ensuring that only previously decoded layers are used as context.

The dimension-aware contextual model further strengthens the correlation between 2D plane features and 3D voxel features. The model first projects high-resolution 3D voxel features onto the xy, yz, and xz planes, then aggregates the count of effective voxels (i.e., those with feature values of $+1$) within each 2D grid cell, thereby generating the projected voxel feature (PVF).

$$f_{\mathrm{PV}xy}(z) = \sum_{z} \sigma(z) \tag{28}$$

$$f_{\mathrm{PV}yz}(x) = \sum_{x} \sigma(x) \tag{29}$$

$$f_{\mathrm{PV}xz}(y) = \sum_{y} \sigma(y) \tag{30}$$

where $\sigma(\cdot)$ represents the density of the voxel. This process employs an occupancy grid to filter out ineffective spatial regions, retaining only areas relevant to the scene surface representation.

The projected feature map $\eta_{\mathrm{dim}}(\cdot)$, serves as a prior context, guiding the probability estimation of 2D plane features. For example, for the xy plane feature $\phi_\theta^{(l_2,xy)}$, its probability estimation incorporates the corresponding 3D projected feature $\boldsymbol{f}_{\mathrm{dim}}$.

$$\mathrm{Pr}_{l_2,xy} = \eta_{\mathrm{dim}}(\gamma(\boldsymbol{f}_{\mathrm{dim}}, v^{(l_2,xy)})) \tag{31}$$

where $v^{(l_2,xy)}$ represents the global frequency statistics at the current 2D hierarchical level. This design allows 2D features to fully leverage the structural information in 3D space, enhancing the accuracy of probability estimation.

This method adopts a hybrid architecture that integrates a 3D voxel grid with a 2D orthogonal plane grid. The 3D voxel grid utilizes a multi-resolution hierarchical structure (from coarse to fine) to extract the global geometric and radiance features of the scene, while the 2D plane grid captures high-frequency spatial details efficiently by leveraging projections at high resolutions.

This design addresses the spatial discretization problem of 3D features while also reducing Hash collisions caused by high-resolution 3D voxel feature encoding through 2D plane regularization, thereby enhancing the efficiency of feature storage. Additionally, the hierarchical structure ensures that features from coarser levels act as priors, guiding the probability estimation at finer levels. By employing low bitrate quantization, this method minimizes the memory footprint of feature extraction while dynamically adjusting feature probability estimation based on global frequency statistics at each level.

Meanwhile, the dimension-aware contextual model strengthens the correlation between high-resolution 3D voxel features and their 2D projections using PVF. This design effectively leverages 2D spatial structures to guide probability estimation for 2D plane features, thereby eliminating the loss of information caused by treating 2D and 3D features independently.

Finally, the compressed bitstream of the central object video consists of five major components. The first component is the 3D and 2D feature embeddings, which are first predicted using the hierarchical contextual model and then entropy-coded with arithmetic coding (AE) to reduce redundancy. The second component consists of the MLP parameters, which are stored using a 10-bit quantization to balance precision and storage efficiency. The third component contains the contextual model parameters, which are stored in float32 format to preserve the accuracy of probability estimation. The fourth component is the occupancy grid, where the parameters are binary representations of valid scene regions, further compressed using AE coding to minimize storage overhead. The fifth component includes the camera intrinsic and extrinsic parameters, containing both the camera intrinsic matrix and the extrinsic parameter sequence $\{[\boldsymbol{R}_i \mid \boldsymbol{t}_i], i = 1, 2, \ldots, N\}$. Based on these components, the decoder reconstructs the central object video, ensuring accurate rendering and efficient compression.

4 Simulation results

4.1 Experimental environment and dataset

In the proposed dual-mode semantic compression framework, the experimental settings is referenced to Ref. [8], with specific parameters as follows. For 3D embeddings, there are 12 levels with resolutions ranging from 16 to 512. For 2D embeddings, the resolutions range from 128 to 1 024 across 4 levels. The maximum number of feature vectors per level are set to 2^{19} and 2^{17} for 3D and 2D, respectively. The resolution of the occupancy grid is 128, and the number of context levels L_C is set to 3. The structure of the rendering MLP is the same as that in Ref. [7] but with a width of 160. During training, varies λ from 10^{-3} to 8×10^{-3}, and the feature vector dimension d_f is set to 1, 2, 4, or 8 to achieve different bitrates. The dataset used in the experiments was a manually captured video dataset of cultural relics. The video resolution is 720 × 1 280 pixels, recorded in RGB three-channel color mode. During the shooting process, the camera equipment rotated 360° around the cultural relics to ensure that the details of the relics were fully captured from all angles.

4.2 Evaluation metrics

To assess the quality of video transmission, the following metrics are used: peak signal-to-noise ratio (PSNR), structural similarity (SSIM), learned perceptual image patch similarity (LPIPS), and a newly proposed metric, PSNR-CENTER, which is a PSNR variant that evaluates image quality only in the central region. PSNR is an objective metric used to measure the difference between two images, primarily for evaluating the effects of image compression, transmission, or reconstruction algorithms. A higher PSNR value indicates greater similarity between the two images and less quality loss. Based on the concept of signal-to-noise ratio (SNR) from information theory, PSNR converts the assessment of image quality into the ratio of signal (original image) to noise (distortion part). The calculation formula for PSNR is

$$\eta_{\mathrm{PSNR}} = 10\lg \frac{V_{\mathrm{pmax}}^2}{\varepsilon_{\mathrm{MSE}}} \tag{32}$$

where V_{pmax} represents the maximum possible pixel value (for example, for an 8-bit image, V_{pmax} is 255), and $\varepsilon_{\mathrm{MSE}}$ is the mean squared error (MSE) between the original and reconstructed images. SSIM is a full-reference image quality evaluation metric used to measure the similarity between two images. Unlike MSE and PSNR, SSIM is more in line with human visual perception. LPIPS is a deep learning-based image quality evaluation metric used to assess the perceptual similarity between two images. Compared to traditional metrics, LPIPS is more in line with human

perception. A lower LPIPS value indicates greater perceptual similarity between the two images.

In addition, to more accurately evaluate the reconstruction quality of the central object under semantic compression scenarios, a region-specific metric termed PSNR-CENTER is proposed. Unlike conventional PSNR, which calculates pixel-wise fidelity over the entire frame, PSNR-CENTER restricts the computation to the segmented region corresponding to the object of interest. This region is extracted using semantic segmentation techniques and reflects the most semantically valuable part of the scene, particularly critical in cultural heritage applications. By isolating the evaluation to this target area, PSNR-CENTER offers a more precise and task-relevant assessment of object-level reconstruction quality, especially under extremely low bitrate conditions.

4.3 Experiment results

The performance of the proposed dual-mode semantic compression framework is evaluated under FP mode and PC mode, respectively. In FP mode, where only the segmented central object is encoded, the proposed framework was compared with both H.264 and H.265 by adjusting parameters for each scheme. The H.264 and H.265 schemes generated varying bitrates by adjusting group of pictures (GOP) size, $G \in \{20, 25\}$ and constant rate factor (CRF) parameters $C \in \{42, 44, 46, 48, 50\}$. The proposed framework controls compression rates by tuning the λ parameter and varying the feature dimensions $d_f \in \{1, 2\}$.

Fig. 3 demonstrates that the proposed framework outperforms both H.264 and H.265 in term of PSNR, SSIM, and LPIPS metrics, especially at ultra-low bitrates. For example, at a similar PSNR level, $\eta_{\text{PSNR}} \approx 33.5$ dB, the proposed framework achieves a bitrate of 0.002 5 bit per pixel (BPP), compared to 0.004 4 BPP for H.264 and 0.004 0 BPP for H.265, indicating up to 43.0% and 37.5% bitrate savings, respectively. Visual quality, as measured by LPIPS and SSIM, also shows significant improvement. The proposed method preserves geometric consistency and high-frequency details more effectively due to multi-resolution features and context modeling.

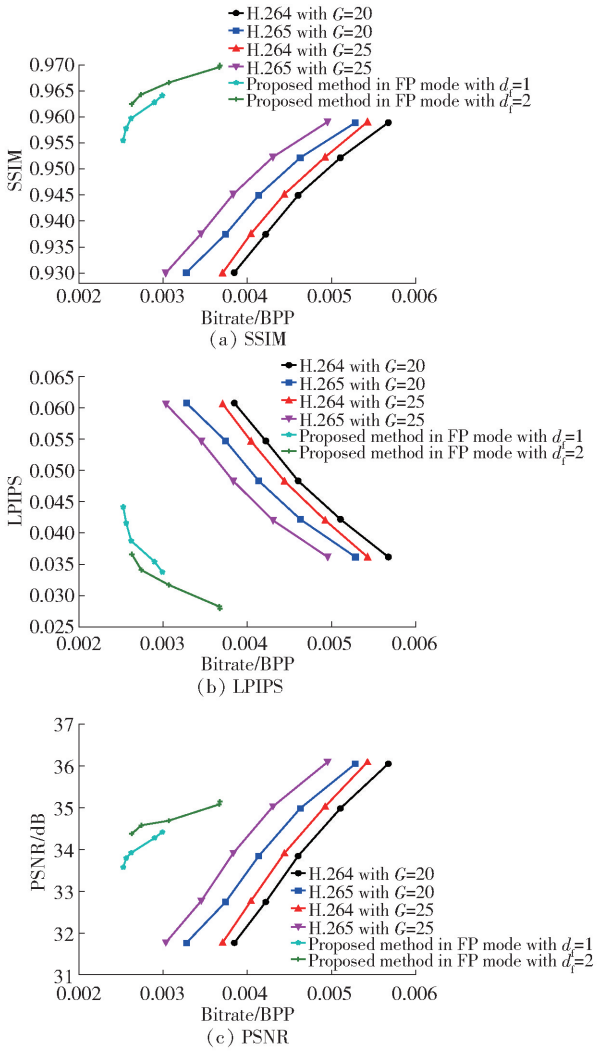

Fig. 3 Comparison results between the proposed model in FP mode and traditional codecs at different bitrates

For PC mode, full-scene compression performance is evaluated by reconstructing the central object via NeRF and encoding the background using H.264. H.264 directly compresses full videos by adjusting GOP size $G \in \{20, 25, 30\}$ and CRF parameters $C \in \{38, 40, 42, 44, 46\}$. The proposed method operated in PC mode, reconstructing the central object via NeRF and compressing the background using H.264 at $G \in \{30, 60, 90\}$, $C \in \{42, 44, 46, 47, 48, 49, 50\}$, with fixed $\lambda = 0.008$ and feature dimension $d_f = 2$. Fig. 4 shows that the proposed method achieves comparable LPIPS and SSIM scores to H.264 at the same bitrate, with slightly lower overall PSNR but significantly higher PSNR-CENTER (e.g., +2.0 dB at bitrate less than 0.009 BPP). This highlights the method's ability to

maintain high fidelity for the central object even at ultra-low bitrates. At equivalent PSNR-CENTER levels, the proposed method reduces the average bitrate by 17% (0.009 4 BPP vs. 0.011 3 BPP). Visual comparisons further confirm that H.264 suffers from edge blurring and texture loss on the central object. Fig. 5 compares the visual reconstruction results and quantitative metrics under different compression schemes. The proposed FP mode ($\lambda=0.001$, $d_f=2$) achieves the highest PSNR of 35.08 dB at bitrate of 0.003 7 BPP, significantly outperforming H.264 (PSNR is 31.10 dB) at the same bitrate. In PC mode, the proposed method yields a superior PSNR-CENTER of 36.05 dB compared to H.264's 33.35 dB at a bitrate of 0.007 8 BPP, indicating better preservation of central object details. While the proposed method retains structural details via NeRF's implicit representation.

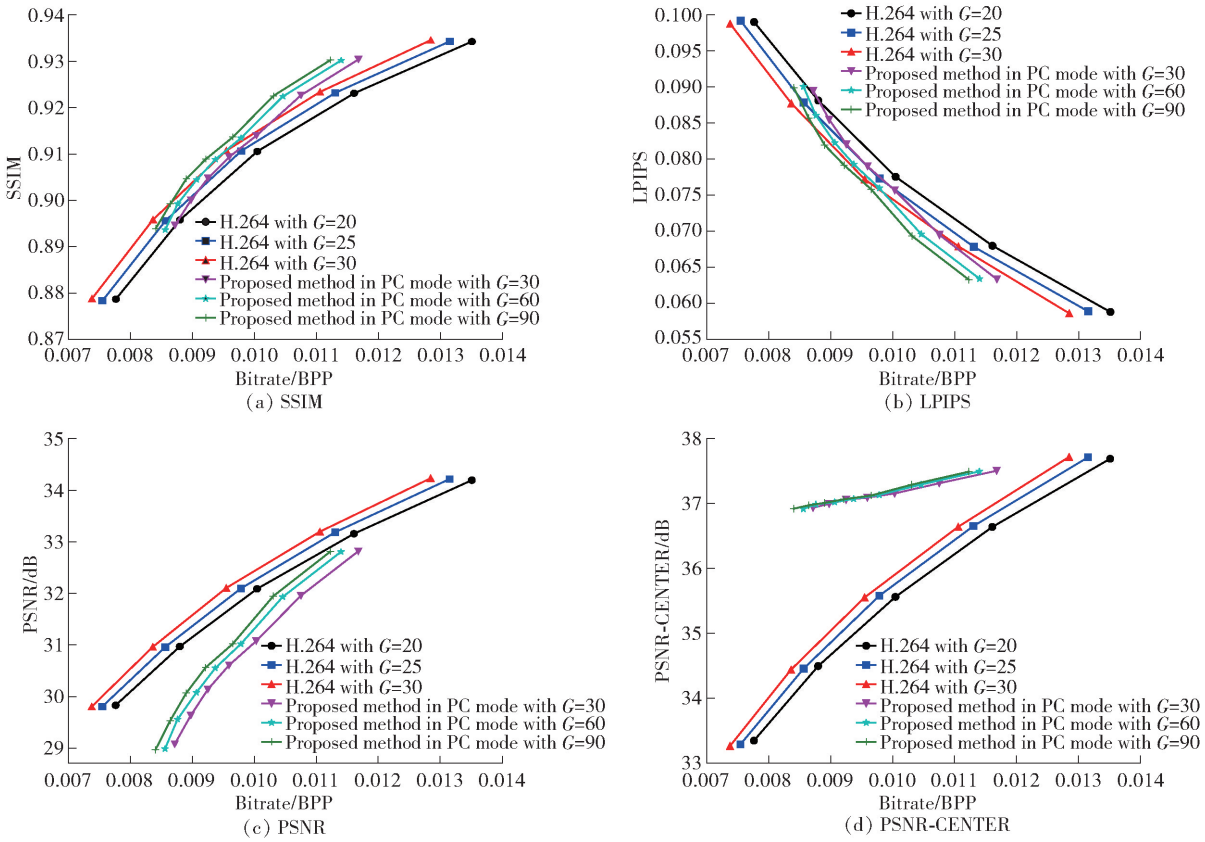

Fig. 4 Comparison results of the proposed method in PC mode and H.264 in different bitrates

(a) Original image (b) Proposed method in FP mode (c) H.264 with G=25, C=50 (d) Proposed method in PC mode (e) H.264 with G=20, C=46

Fig. 5 Examples of reconstruction results

These experiments validate that the proposed dual-mode semantic compression framework effectively minimizes redundant bit allocation. By leveraging 3D-2D hybrid feature grids and hierarchical context models, the method ensures robust reconstruction quality for the central object even at ultra-low bitrates (e. g. , 0. 002 5 BPP), demonstrating its superiority over traditional video codecs in scenarios prioritizing region-of-interest preservation.

4.4 Computational complexity analysis

For a typical 10 s video with 30 frames, COLMAP-based SfM takes approximately 1 h, while the segmentation stage, based on deep neural inference, requires about 1 min. NeRF training, using a sample size of 150 000 and 40 000 training steps, takes around 2 h. During testing, NeRF rendering shares the same inference framework as CNC, introducing no extra overhead. Background video is encoded with H. 264, requiring similar time as traditional codecs. Post-processing steps such as concatenation are negligible in time consumption. Hence, the overall computational cost of the proposed method is moderate and acceptable for offline processing scenarios such as cultural heritage archiving.

5 Conclusions and future work

This paper presents a dual-mode semantic compression framework for static object videos, addressing the limitations of traditional codecs in balancing fidelity and efficiency. By decoupling scenes into a central object (encoded via NeRF's implicit 3D representation) and a background (compressed with lightweight H. 264), the method achieves 43% lower bitrates (down to 0. 002 5 BPP) than H. 264 at equivalent quality, while eliminating artifacts like blockiness and color distortion. For full-scene applications, the PC mode reduces storage by 20% and outperforms H. 264 in region-of-interest fidelity. The framework enables flexible reconstruction offering a scalable solution for scenarios demanding ultra-low bitrates and semantic-aware compression. This work advances digital preservation paradigms, particularly for static cultural artifacts, by harmonizing neural representations with adaptive encoding strategies.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2022YFB2902100).

References

[1] ZHOU Y Q, LIU L, WANG L, et al. Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching. Digital Communications and Networks, 2020, 6(3): 253 – 260.

[2] Advanced video coding for generic audiovisual services. ITU-T Recommendation H. 264. 2003.

[3] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1649 – 1668.

[4] TIAN Y, LU G, ZHAI G T. SMC++: masked learning of unsupervised video semantic compression. arXiv Preprint, arXiv: 2406. 04765, 2024.

[5] WAN R, ZHENG Q, FAN Y B. M3-CVC: controllable video compression with multimodal generative models. Proceedings of the 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'25), 2025, Apr 6 – 11, Hyderabad, India. Piscataway, NJ, USA: IEEE, 2025: 5p.

[6] MILDENHALL B, SRINIVASAN P P, TANCIK M, et al. NeRF: representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 2021, 65(1): 99 – 106.

[7] MÜLLER T, EVANS A, SCHIED C, et al. Instant neural graphics primitives with a multiresolution Hash encoding. ACM Transactions on Graphics, 2022, 41(4): Article 102.

[8] CHEN Y H, WU Q Y, HARANDI M, et al. How far can we compress instant-NGP-based NeRF?. Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'24), 2024, Jun 16 – 22, Seattle, WA, USA. Piscataway, NJ, USA: IEEE, 2024: 20321 – 20330.

[9] ZHENG P, GAO D H, FAN D P, et al. Bilateral reference for high-resolution dichotomous image segmentation. arxiv Preprint, arxiv:2401. 03407, 2024.

[10] SCHONBERGER J L, FRAHM J M. Structure-from-motion revisited. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016, Jun 27 – 30, Las Vegas, NV, USA. Piscataway, NJ, USA: IEEE, 2016: 4104 – 4113.

[11] BOSSEN F, BROSS B, SUHRING K, et al. HEVC complexity and implementation analysis. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1685 – 1696.

[12] TUNG T Y, GÜNDÜZ D. DeepWiVe: deep-learning-aided wireless video transmission. IEEE Journal on Selected Areas in Communications, 2022, 40(9): 2570 – 2583.

[13] WANG S X, DAI J C, LIANG Z J, et al. Wireless deep video semantic transmission. IEEE Journal on Selected Areas in Communications, 2023, 41(1): 214 – 229.

[14] SHIN S, PARK J. Binary radiance fields. Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS'23), 2023, Dec 10 – 16, New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates Inc, 2023: 55919 – 55931

[15] LI L Z, SHEN Z, WANG Z S, et al. Compressing volumetric radiance fields to 1 MB. Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23), 2023, Jun 17 – 24, Vancouver, Canada. Piscataway, NJ, USA: IEEE, 2023: 4222 – 4231

[16] XING Y K, YANG Q, YANG K F, et al. Explicit-NeRF-QA: a quality assessment database for explicit NeRF model compression. arXiv Preprint, arXiv:2407.08165, 2024.

[17] WANG Y F, GONG Y, ZENG Y. Hyb-NeRF: a multiresolution hybrid encoding for neural radiance fields. Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV'24), 2024, Jan 3 – 8, Waikoloa, HI, USA. Piscataway, NJ, USA: IEEE, 2024: 3677 – 3686.

[18] MONTAJABI Z, GHASSAB V K, BOUGUILA N. Recurrent neural network-based video compression. Proceedings of the 21st IEEE International Conference on Machine Learning and Applications (ICMLA'22), 2022, Dec 12 – 14, Nassau, Bahamas. Piscataway, NJ, USA: IEEE, 2022: 925 – 930.

[19] REGENSKY A, BRAND F, KAUP A. Analysis of neural video compression networks for 360-degree video coding. Proceedings of the 2024 Picture Coding Symposium (PCS'24), 2024, Jun 12 – 14, Taichung, China. Piscataway, NJ, USA: IEEE, 2024: 5p.

[20] ÖNBERGER J L, ZHENG E L, FRAHM J M, et al. Pixelwise view selection for unstructured multi-view stereo. Proceedings of the 14th European Conference on Computer Vision (ECCV'16): Part III, 2016, Oct 11 – 14, Amsterdam, Netherlands. LNCS 9907. Berlin, Germany: Springer, 2016: 501 – 518.

[21] ZHANG K, RIEGLER G, SNAVELY N, et al. NERF++: analyzing and improving neural radiance fields. arXiv Preprint, arXiv:2010.07492, 2020.

[22] BADRINARAYANAN V, KENDALL A, CIPOLLA R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481 – 2495.

[23] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer vision (ICCV'17), 2017, Oct 22 – 29, Venice, Italy. Piscataway, NJ, USA: IEEE, 2017: 2980 – 2988.

[24] CHEN L C, ZHU Y K, PAPANDREOU G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. Proceedings of the 15th European Conference on Computer Vision (ECCV'18): Part V, 2018, Sep 8 – 14, Munich, Germany. LNCS 11211. Cham, Switzerland: Springer Nature Switzerland AG, 2018: 833 – 851.

[25] KIRILLOV A, MINTUN E, RAVI N, et al. Segment anything. Proceedings of the 19th IEEE/CVF International Conference on Computer Vision (ICCV'23), 2023, Oct 2 – 6, Paris, France. Piscataway, NJ, USA: IEEE, 2023: 4015 – 4026.

(Editor: Wang Xuying)