VISTA: Video Transmission over A Semantic Communication Approach

Chengsi Liang*, Xiangyi Deng†, Yao Sun*, Runze Cheng*, Le Xia*, Dusit Niyato‡, and Muhammad Ali Imran*
*James Watt School of Engineering, University of Glasgow, Glasgow, UK
†Glasgow College, University of Electronic Science and Technology of China, Chengdu, China
‡School of Computer Science and Engineering, Nanyang Technological University, Singapore
Email: Yao.Sun@glasgow.ac.uk

***Abstract*—Video transmission over ultra-reliable and low-latency communication (URLLC) is a promising trend to support various multimedia services. However, in view of the rapid surge in video content demand along with a stringent resolution requirement, there is an unprecedented burden on wireless networks with limited yet precious bandwidth resources. In this paper, we propose a VIdeo transmission framework over Semantic communicaTion Approach (VISTA), where semantics rather than all bits of a video should be transmitted with the aim of reducing bandwidth consumption while keeping a high visual perception. Specifically, the semantic segmentation module in VISTA is first developed to classify and encode the dynamic and static segments in source video separately. Next, the semantic location graphs (SLGs) are built to describe semantics and location relations among the detected dynamic objects. Through the joint source-channel coding (JSCC) module which is adaptive to different channel conditions, the encoded semantic features and SLGs are transmitted over wireless channel. Finally, the video is recovered at the receiver end based on the distorted semantic features and SLGs with the assistance of frame interpolation module. Simulation results demonstrate that VISTA outperforms two benchmarks in terms of required bandwidth reduction and robustness against channel noise, further satisfying the requirements on URLLC.**

I. Introduction

With the prosperity of multimedia services, it was witnessed that video streaming has occupied approximately 82 percent of all Internet traffic in 2022 [1] to cover a wide range of applications including live streaming, virtual/augmented/mixed reality, virtual meeting, etc. To further improve the quality of services (QoS) for users, ultra-reliable and low-latency communication (URLLC) is required, especially for real-time video applications such as augmented reality (AR) and virtual reality (VR). However, since the traditional wireless video transmission focuses on video compression and recovery via image pixels encoding and decoding, which consumes unprecedented amount of wireless spectrum and transmission time. Additionally, it may fail to achieve a satisfactory visual perception due to unstable wireless channel condition.

Fortunately, semantic communication (SemCom) [2]–[5] concerning with the meaning of source information rather than bits/symbols themselves, has been recently deemed as a great revolution of communication system. Generally, transmitter in SemCom system first extracts and encodes the semantic information from the source adapting to wireless channel condition, then transmits semantic information wirelessly, finally meaning of the source is recovered at the receiver aiming to minimize semantic error. In this way, SemCom is expected to dramatically reduce the amount of delivered bits, thus greatly saving the wireless resources consumption. Furthermore, considering that adjacent frames are tightly coupled at the semantic level, SemCom could achieve high robustness especially under poor wireless channels by exploiting decoder to correctly recover blurry video pixels based on semantics.

Despite many superiorities of SemCom-enabled video transmission, there are several inevitable challenges. It should be first noted that static and dynamic objects may coexist in multiple consecutive video frames, where the semantics implicit in each static object between different frames are normally identical and the change process of each dynamic object in consecutive frames should be regular. Hence, how to realize efficient semantic representation and reconstruction for consecutive frames is the first nontrivial challenge. Besides, signal attenuation and distortion in wireless channels may impose severe semantic ambiguity on transmitted videos and further greatly affect the final rendered video quality, thus the second challenge should be how to take into account different channel status in SemCom-enabled video transmission. A few pioneering works on DL-based video transmission in SemCom have been recently presented [6]–[8]. The authors in [6] design a deep joint source-channel coding framework aiming at transmitting semantics of the whole video over arbitrary wireless channels. [7] focuses on transmitting keypoints for semantic video conferencing and proposes an incremental redundancy hybrid automatic repeat-request framework to adapt varying channels. Furthermore, [8] discusses the prospect of URLLC in semantic VR delivery between the mobile edge computing server and VR users.

In responding to the aforementioned challenges, in this paper, we propose a novel VIdeo transmission framework over Semantic communicaTion Approach, named VISTA. VISTA has three modules, where the semantic segmentation module and frame interpolation module account for semantic encoding and decoding respectively, while the joint source-channel coding (JSCC) module is for SNR-adaptive wireless transmission. The main contributions of this work are summarized as followed:

- A semantic segmentation module is developed at the transmitter, where it first detects and recognizes the

dynamic objects and static background for each frame in the source video. Then, a semantic location graph (SLG) is built to describe the locations and relationships for all dynamic objects and accurately extract semantics.
- VISTA separates each video frame into environment (image of static background) and behavior segments (image of all the dynamic objects), which requires only one-frame environment and several key behavior segments to be fed in JSCC module for wireless transmission.
- A frame interpolation module is developed at the receiver end to accurately recover the video based on the segments and semantics with the help of SLG.
- We test the performance of VISTA over a real video dataset, and the results demonstrate its superiorities in terms of transmitted data volume, video processing time and video quality (especially under low SNR scenario) compared with other two benchmarks.

II. Video Transmission Framework in VISTA

To achieve SemCom video transmission, VISTA is proposed in this work. We start with the video transmission framework in VISTA in this section. Generally, the semantic coding model is to extract (at the sender side) and restore (at the receiver side) semantic information of transmitted videos, while the channel coding model takes into account different physical channel conditions for accurate semantics delivery. A shared knowledge base (KB) [9] is assumed between transmitter and receiver during video delivery.

Consider a source video composed of T sequential frames, i.e., $\mathbf{s} = \left\{s^1, \ldots, s^T\right\} \in \mathbb{R}^{H\times W\times T}$, where H and W respectively denote the height and width of a frame. These frames are first fed in the convolutional semantic-encoder to distill the textual semantic information $\mathbf{g}$. In addition, the semantic-encoder divides the source video into two parts: environment (static background) $\mathbf{s}_e$ and behavior segments (dynamic objects) $\mathbf{s}_b$ individually. Thus, the encoded frames can be written as $\hat{\mathbf{s}} = \{\mathbf{s}_e, \mathbf{s}_b, \mathbf{g}\}$ under the semantic-encoder network $\mathcal{S}(\cdot)$ with parameter set α_s, i.e.,

$$\hat{\mathbf{s}} = \{\mathbf{s}_e, \mathbf{s}_b, \mathbf{g}\} = \mathcal{S}\left(\mathbf{s}; \alpha_s\right). \tag{1}$$

The encoded frames $\hat{\mathbf{s}}$ then flow into JSCC module for SNR-adaptive wireless transmission. In this module, source-encoder $\mathcal{E}$ and channel-encoder $\mathcal{C}$ with parameter sets α_ϵ and α_c generate the symbols $\mathbf{x}$ to be transmitted,

$$\mathbf{x} = \mathcal{C}\left(\mathcal{E}\left(\hat{\mathbf{s}}; \alpha_\epsilon\right); \alpha_c\right). \tag{2}$$

At the receiver side, $\mathbf{y}$ is denoted as the received symbols for the input $\mathbf{x}$ over the wireless channel with additive noise w, i.e.,

$$\mathbf{y} = h * \mathbf{x} + w, \tag{3}$$

where h denotes the channel gain. $\mathbf{y}$ is then fed to the channel-decoder $\mathcal{C}^{-1}$ and source-decoder $\mathcal{E}^{-1}$ sequentially to reconstruct the environment $\tilde{\mathbf{s}}_e$ and behavior segments $\tilde{\mathbf{s}}_b$ with the help of the semantics. The decoded frames $\tilde{\mathbf{x}}$ is presented as

$$\tilde{\mathbf{x}} = \{\tilde{\mathbf{s}}_e, \tilde{\mathbf{s}}_b\} = \mathcal{E}^{-1}\left(\mathcal{C}^{-1}\left(\mathbf{y}; \beta_c\right); \beta_\epsilon\right). \tag{4}$$

where β_c and β_ϵ denote the parameters of channel-decoder and source-decoder networks, respectively.

Finally, the recovered video $\tilde{\mathbf{s}}$ should be constructed as per the two parts of segments $\tilde{\mathbf{s}}_e$ and $\tilde{\mathbf{s}}_b$. The semantic-decoder network and its parameters are given as $\mathcal{S}^{-1}$ and β_s. Thus, the final recovered video is expressed as

$$\tilde{\mathbf{s}} = \mathcal{S}^{-1}\left(\tilde{\mathbf{x}}; \beta_s\right). \tag{5}$$

In this work, the ultimate goal is minimizing the semantic ambiguity of the recovered video. We use average peak signal-to-noise ratio (PSNR) [10], a popular video quality metric, to measure the differences between the recovered and original video frames. In detail, for the t-th frame with the size $m \times n$, the mean squared error (MSE) between the original frame s^t and the recovered one $\tilde{s}^t$ is calculated as

$$MSE^t = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[s^t\left(i,j\right) - \tilde{s}^t\left(i,j\right)\right]. \tag{6}$$

Thus, the average of PSNR of the original and recovered video is expressed as

$$PSNR = \frac{1}{T}\sum_{t=1}^{T} 10 \cdot log_{10}\left(\frac{{I_{max}}^2}{MSE^t}\right), \tag{7}$$

where ${I_{max}}^2$ represents the maximum pixel value of the frame.

III. SLG-based Transceiver Design in VISTA

Based on the above video transmission framework, let us illustrate how to design the transceiver in VISTA. As depicted in Fig.1, there are three modules in VISTA, semantic segmentation, JSCC and frame interpolation, where semantic segmentation along with JSCC encoder is deployed at the transmitter side, while frame interpolation with JSCC-decoder at the receiver side. Notably, several SLGs are established in semantic segmentation and utilized in frame interpolation. These modules are trained separately with different loss functions to achieve the goal of PSNR minimization for the recovered video. In the following, we will discuss how to construct and train the neural networks in three modules.

A. Semantic Segmentation Module

Semantic segmentation module is deployed at the transmitter to recognize and distill the dynamic objects from video. Four tasks should be performed in this module, object detection, trajectory prediction, SLG construction and frame sampling. Generally, we first bound all objects using rectangular boxes in each frame and differ the dynamic objects from static background via velocity testing. The semantic information of each dynamic object is able to be extracted by the means of category recognition. However, the occlusion caused by overlapping objects in the video will affect the accuracy of position detection and semantics extraction. Thus, we predict the trajectory of each object for the continuous frames in the second task. After trajectory prediction, an SLG is designed to assist in reserving the estimated positions and semantics of all dynamic objects in each frame. Finally, we

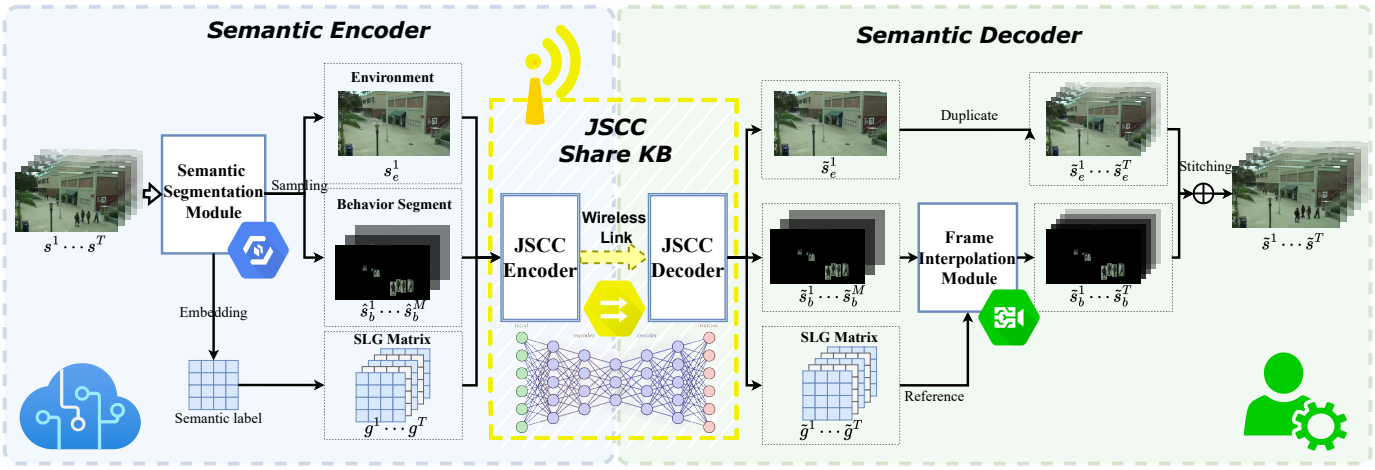

Fig. 1. The diagram of transceiver in VISTA.

sample the frames and send them along with SLGs to JSCC module. Let us below illustrate the design of the four tasks separately.

Object detection: Borrowing the idea from [11], we apply a conventional network to outline bounding boxes using features of the entire frame. Specifically, we initialize several bounding boxes and they are projected to enlarge and shift dynamically until all the objects are bounded with the optimized confidence scores. In this way, each bounding box is associated with six predictions: 2D-coordinates (u, v) of the center for the object, the width and height (w, h) of the box containing relative to the whole image of the object, object category l and the associated confidence score c.

Trajectory prediction: After getting these objective detections, we should guarantee that every dynamic objects can be captured completely. Therefore, we deploy the trajectory prediction module to track the dynamic objects when they are occluded. The input of the network of trajectory prediction is the images of dynamic objects and the five predictions of the corresponding bounding boxes. We first define an "observation" of a bounding box as $z = [u, v, w, h, c]^\top$. Moreover, we employ the Kalman filter (KF) to generate the state $q = [u, v, a, r, \dot{u}, \dot{v}, \dot{a}]^\top$, where a is the bounding box scale (area), r is the width-to-height ratio of the bounding box, and the other three variables ($\dot{u}$, $\dot{v}$ and $\dot{a}$) are the related time derivatives.

Next, we utilize an observation-centric tracker [12] with the object movement. Specifically, since a non-linear motion can be regarded as a synthesis of many small-scale linear motions in a reasonably short time [12], we calculate the velocity consistency (momentum) to gain the accurate velocity value and direction. Then, for an untracked object, an observation-centric online smoothing strategy is performed through a virtual trajectory $\hat{\mathbf{z}}^t$ starting from its last occurrence and ending at the re-associated observation, which is denoted as

$$\hat{\mathbf{z}}^t = \mathcal{T}_v\left(\mathbf{z}^{t_1}, \mathbf{z}^{t_2}, t\right), t_1 < t < t_2, \tag{8}$$

where $\mathbf{z}^{t_1}$ is the the last observation before being untracked, $\mathbf{z}^{t_2}$ is observation triggering the re-association, and $\mathcal{T}_v()$ represents the network of virtual trajectory. Along this virtual trajectory, the status at t_1 is recalled back to check the filter parameters. Thus, the refreshed state $\hat{q}^t$ is estimated as

$$\hat{q}^t = \mathbf{F}^t \hat{q}^{t-1} + \mathbf{K}^t\left(\hat{z}^t - \mathbf{H}^t \mathbf{F}^t \hat{q}^{t-1}\right), \tag{9}$$

where $\mathbf{K}^t$ denotes the KF matrix, $\mathbf{F}^t$ and $\mathbf{H}^t$ denote the state transition and observation model respectively. With the instruction of $\hat{q}^t$, for the t-th frame in the video, we update the bounding boxes of dynamic objects and use the behavior segments $\hat{s}_b^t$ to represent the images of all estimated boxes covering. Moreover, the rest of this frame is represented by the environment $\hat{s}_e^t$.

SLG construction: Aiming at locating the dynamic objects and illustrating the association between their location and semantics, we deliver an SLG to concatenate the classes and location from the refreshed states $\hat{q}$. With respect to a frame containing B boxes, the set of object categories is $\mathbf{l} = \{l_1, \ldots, l_B\}$, and the 2D-coordinates set are $\hat{\mathbf{u}} = \{\hat{u}_1, \ldots, \hat{u}_B\}$ and $\hat{\mathbf{v}} = \{\hat{v}_1, \ldots, \hat{v}_B\}$. Thus, the SLG $g^t \in \{g^1, \ldots, g^T\}$ of the t-th frame can be represented as

$$g^t = \{\mathbf{l}, \hat{\mathbf{u}}, \hat{\mathbf{v}}\}. \tag{10}$$

Frame sampling: According to the results of trajectory prediction, we split the whole video into environment and behavior segments and transmit them separately. Since the environment is fixed, it is supposed that only the environment of the first frame s_e^1 needs to be transmitted. It is also thriftless for encoder to cope with behavior segments in the whole video, so that we sample them every T_s frames and denote $M = \lceil T/T_s \rceil$ samples as $\hat{\mathbf{s}}_b' = \left\{\hat{s}_b^1, \hat{s}_b^{T_s+1}, \ldots, \hat{s}_b^T\right\}$. The output $\hat{s}^t$ of the t-th frame is illustrated as

$$\hat{s}^t = \begin{cases} \{s_e^1, \hat{s}_b^1, g^1\}, & t = 1, \\ \{\hat{s}_b^t, g^t\}, & t = nT_s + 1, n = \{1, \ldots, M-1\}, \\ g^t, & otherwise. \end{cases} \tag{11}$$

Generally, the overall output for all frames after semantic-encoder is composed of environment of the first frame, behavior segments from the sample frames and SLGs of all frames.

B. JSCC Module

As illustrated, all the extracted semantic segments along with an SLG should be transmitted through a wireless channel. In VISTA, we employ an SNR-adaptive JSCC module which can configure its parameters depending on the SNR of the channel [13]. Its overall structure can be described as source-encoder, channel-encoder, channel-decoder and source-decoder. In more detail, the features $\mathbf{f} = \left\{f_e^1, \mathbf{f}_b\right\}$ are first extracted from the input of environment and behavior segments (s_e^1 and $\hat{\mathbf{s}}_b'$) via several conventional layers and some of them are activated to be transmitted first. After getting $\mathbf{f}$, the channel-encoder produces two groups of length-L features. The first group with the length G_s contains either active or inactive features selected by a policy network $\mathcal{P}$, while the following G_n groups are always active without selection. The selection for each input is conducted by a binary mask W_i, where can only be 0 or 1. The total number of active groups is demonstrated as $\tilde{G} = G_n + \sum_{i=1}^{G_s} W_i$. All the active features are passed through the power normalization network to generate complex-valued transmission symbols $\left\{x_e^0, \hat{\mathbf{x}}_b'\right\} \in \mathbb{C}^{G\times L/2}$ with unit average power using the first half of features as the real part and the other half as the imaginary part. Moreover, the textual SLGs $\mathbf{g} = \left\{g^1, \ldots, g^T\right\}$ are encoded to bits $\mathbf{x}_g$ and transmitted directly. In a word, the total encoded symbols are represented by $\mathbf{x} = \left\{x_e^0, \hat{\mathbf{x}}_b', \mathbf{x}_g\right\}$.

Next, $\mathbf{y} = \left\{y_e^1, \hat{\mathbf{y}}_b', \mathbf{y}_g\right\}$ is received as $\mathbf{x}$ should be transmitted over the wireless channel model in (3), where y_e^1, $\hat{\mathbf{y}}_b'$ and $\mathbf{y}_g$ denote the transmitted symbols of environment, behavior segments, and SLG, respectively. Then, $\mathbf{y}$ is fed to the channel-decoder and source-decoder sequentially to reconstruct the environment $\tilde{s}_e^1$, behavior segments $\tilde{\mathbf{s}}_b'$ and SLGs $\tilde{\mathbf{g}}$. Specifically, $\tilde{s}_e^1$ and $\tilde{\mathbf{s}}_b'$ are recovered through several convolutional layers while $\tilde{\mathbf{g}}$ are decoded to text directly.

It is worth noting that the SNR value is the part of input fed to the policy network and the SNR adaptive network leveraged in channel-encoder, channel-decoder [14]. Particularly, for the SNR adaptive network, the features in one frame are first pooled averagely across diverse feature channels (different from the wireless channels) of a neural network and then concatenated with the SNR value. Next, the results are received by two multi-layer perceptrons to produce the factors for channel-wise scaling and addition. In this way, we adjust the network of transceiver in JSCC module depending on SNR value.

C. Frame Interpolation Module

After receiving the environment and behavior segments of sample frames, we complement them and combine the results to rebuild the video with the help of SLGs $\tilde{\mathbf{g}}$ in the semantic-decoder. In more detail, we make T copies of the one-frame environment and generate the sequence of environment as $\tilde{\mathbf{s}}_e = \left\{\tilde{s}_e^1, \ldots, \tilde{s}_e^1\right\}$ at first. Then, according to the behavior segments $\tilde{\mathbf{s}}_b' = \left\{\tilde{s}_b^1, \ldots, \tilde{s}_b^M\right\}$ of M sample frames, we utilize Transformer for frame interpolation with the inspiration of Video frame interpolation with Transformer (VFIformer) [15], aiming at predicting the behavior segments for all the remaining frames. Consider the behavior segments $\tilde{s}_b^1$ and $\tilde{s}_b^2$ of the two adjacent sample frames, and the intermediate frame is denoted as $\tilde{s}_b^t$.

A convolutional network called flow estimator is utilized to obtain the optical flows $O^{t\to1}$ and $O^{t\to2}$. Additionally, the images w_b^1 and w_b^2 are restored as per the features $\tilde{f}_i^1$ and $\tilde{f}_i^2$ which are warped by $O^{t\to1}$ and $O^{t\to2}$ respectively. Further, the semantic decoder includes Transformer blocks (TFB) and each TFB consists of convolutional layers and several Transformer layers (TFL) with Cross-Scale Window-based Attention (CSWA) network which is a state-of-the-art attention mechanism. For the i-th TFB, its output feature f_i^t is formulated as

$$f_i^t = TFB_i\left(f_{i-1}^t, \tilde{f}_i^1, \tilde{f}_i^2\right), \tag{12}$$

where f_{i-1}^t is the output of (i-1)-th TFB.

Then, the intermediate frame $\tilde{s}_b^t$ is generated by a soft mask H and an image residual $\Delta\tilde{s}_b^t$ (from flow errors and occlusion) in the decoder as follows:

$$\tilde{s}_b^t = H \odot w_b^1 + (1 - H) \odot w_b^2 + \Delta\tilde{s}_b^t, \tag{13}$$

where $\odot$ signifies the Hadamard product. It is worth noting that the interpolation is under the guidance of SLG. In other word, the prediction of behavior segments in the intermediate frames is limited in the bounding boxes provided by SLG.

In terms of model training, the loss should be evaluated from three aspects. The first is reconstruction loss, which compares the recovered behavior segments s_{gt}^t and its ground-truth $\tilde{s}_b^t$ in t-th frame as

$$\mathcal{L}_{rec} = \left\|s_{gt}^t - \tilde{s}_b^t\right\|_1. \tag{14}$$

Next, the census loss [16] is robust to illumination changes, which is defined as the soft Hamming distance between census-transformed [17] image patches of s_{gt}^t and $\tilde{s}_b^t$. The last one is distillation loss for supervising the estimated flows explicitly,

$$\mathcal{L}_{dis} = \left\|\tilde{O}^{t\to1} - O^{t\to1}\right\|_1 + \left\|\tilde{O}^{t\to2} - O^{t\to2}\right\|_1, \tag{15}$$

where $\tilde{O}^{t\to1}$ and $\tilde{O}^{t\to2}$ are derived from a pretrained flow estimation network [18].

As a result, the total loss is presented as

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{css}\mathcal{L}_{css} + \lambda_{dis}\mathcal{L}_{dis}, \tag{16}$$

where $\mathcal{L}_{rec}$, $\mathcal{L}_{css}$ and $\mathcal{L}_{dis}$ correspond to the reconstruction loss, census loss and distillation loss with their weights λ_{rec}, λ_{css} and λ_{dis} respectively.

After frame interpolation, the behavior segments are estimated as the combination of the sample frames and intermediate frames. The recovered video $\tilde{\mathbf{s}}$ is the synthesis of the behavior segments $\tilde{\mathbf{s}}_b$ and the copies of the environment $\tilde{\mathbf{s}}_e$, which can be expressed by

$$\tilde{\mathbf{s}} = \tilde{\mathbf{s}}_b \oplus \tilde{\mathbf{s}}_e. \tag{17}$$

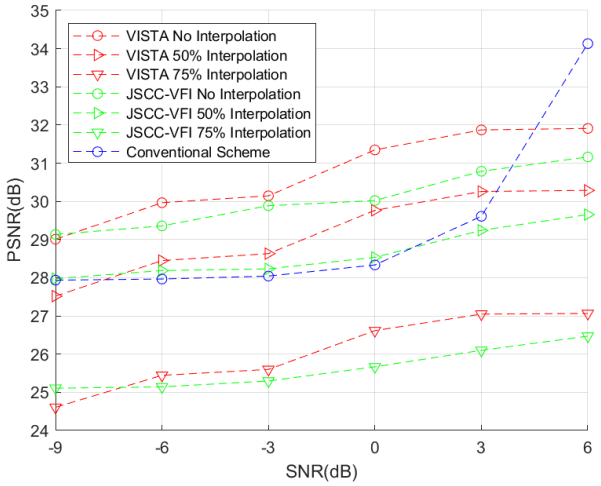

Fig. 2. PSNR performance of recovered video frames versus varying SNRs.

Herein, we stitch $\tilde{\mathbf{s}}_e$ and $\tilde{\mathbf{s}}_b$ via $\oplus$ and maintain $\tilde{\mathbf{s}}_b$ as their overlapping parts.

IV. Simulation Results and Discussions

In this section, we conduct simulations to evaluate the performance of the proposed VISTA framework in comparison with two different benchmarks: 1) A JSCC integrated with VFIformer scheme (JSCC-VFI), which first employs a single deep neural network to transmit video frames over wireless channels without any awareness of semantics and then uses the powerful Transformer model for behavior segments interpolation; 2) A conventional bit-oriented communication scheme (Conventional scheme) [19], in which all pixels of each video frame should be encoded into bits based on the prescribed coding rule (low density parity check in our simulations) for precise transmission.

We test the performance of VISTA on an open video dataset [20]. For the simulation settings, the OC-SORT structure is first leveraged for object segmentation of video frames, which keeps consistent with the setup given in [12]. Besides, the parameters in JSCC-related channel encoding and decoding networks are proceeding as those in [13], where the wireless channel model is simulated as an additive white Gaussian noise channel with SNR values varying from -9 to 6 dB. Moreover, the architecture details of VFIformer-related networks can refer to [15]. Note that the Adam optimizer is adopted to train the VISTA with an initial learning rate of 5×10^{-4}, and all subsequent simulations are implemented in a computer with six CPU cores and Inter Core i7 processor, where the main software environment is Python 3.9.

Fig. 2 first shows the PSNR performance under varying SNRs from -9 to 6 dB, where three differing interpolation proportions 0, 50%, and 75% are considered. It can be seen that the PSNR of all schemes increases with SNR, which is because the higher SNR leads to less impairment of transmitted semantic features so as to render a more accurate frame recovery. Meanwhile, we can see a better PSNR of VISTA at a higher interpolation proportion. This trend is attributed to the fact that using fewer behavior frames for transmission means that more compressed features could be lost between consecutive dynamic objects, thereby resulting in a worse PSNR performance. Furthermore, it can be found that in the same condition without interpolation, the PSNR of VISTA can always outperform the conventional scheme when SNR is lower than 3 dB. Such a performance gain of VISTA can be credited to its accurate semantic calibration function provided by SLG, which sufficiently guarantees high reliability of video transmission even in low-SNR conditions. For further visual comparisons, we exhibit a specific frame of the video in Fig. 3, where the frames of all different situations of Fig. 2 along with their corresponding original frame are presented at an SNR of 0 dB. Similar to the conclusions in Fig. 2, we can obviously see that all objects in the frames of VISTA are well recovered, while presenting a higher image quality than the conventional scheme.

Next, we test the processing time for a total of 20 consecutive video frames with different interpolation proportions in

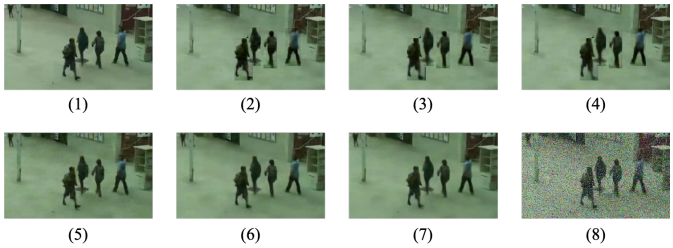

Fig. 3. Visual comparison on a video frame sample from VIRAT dataset [20], where (1) the original frame, (2) the frames restored via VISTA with no interpolation, (3) VISTA with 50% interpolation, (4) VISTA with 75% interpolation, (5) JSCC-VFI with no interpolation, (6) JSCC-VFI with 50% interpolation, (7) JSCC-VFI with 75% interpolation, and (8) the conventional scheme with no interpolation are all presented.

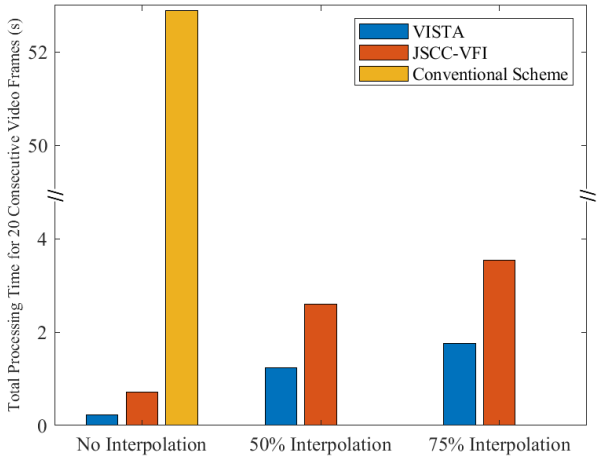

Fig. 4. Total processing time for 20 consecutive video frames under different interpolation proportions.

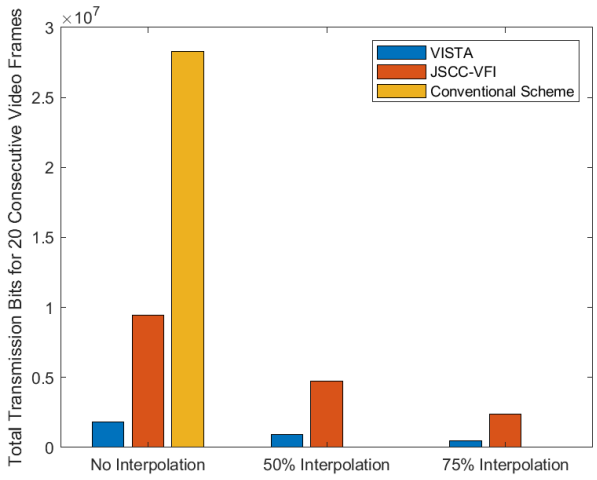

Fig. 5. Total transmission bits consumed for 20 consecutive video frames under different interpolation proportions.

Fig. 4, which is precisely obtained based on the aforementioned computer configuration. It can be seen that without interpolation, the proposed VISTA only needs 0.22s processing time, which saves around 2.63 s/frame compared with the conventional scheme and saves nearly half the time of JSCC-VFI at any proportion of interpolation. This is because only behavior segments of a few video frames need to be processed in VISTA thanks to the used SLG mechanism, thereby fewer pixels are required to be encoded and decoded so as to save a significant processing time. In addition, a higher proportion of interpolation leads to a higher processing time, since more intermediate frames should be sampled and interpolated.

Finally, Fig. 5 demonstrates the number of required bits for transmitting 20 consecutive video frames with the same three interpolation proportions. It is implied that the proposed VISTA indeed shows its amazing superiority in communication resource saving, whose bit consumption is only 6.4% of the conventional scheme and 19.2% of the JSCC-VFI scheme when no interpolation is set. Moreover, we can see that the amount of transmitted bits of VISTA decreases as the interpolation proportion improves. Such a trend is easy to understand because the core semantics are more compressed at the high interpolation proportion, enabling these video frames to be sent with fewer bits.

V. Conclusions

In this paper, we propose a SemCom-enabled wireless video transmission framework, named VISTA. A novel transceiver in VISTA is designed to perform semantic encoding and decoding, where we construct a semantic location graph to work in conjunction with several neural networks for video semantics extraction and recovery. Simulation results show an excellent reduction in transmitted bits with no compromise (even an improvement when SNR is below 3 dB) on video quality and time consumed in transmission. This original work is expected as a pioneer in exploiting SemCom in wireless video transmission to significantly alleviate the bandwidth shortage in future communication systems.

References

[1] A. Ahmad, A. B. Mansoor, A. A. Barakabitze, A. Hines, L. Atzori, and R. Walshe, "Supervised-learning-Based QoE Prediction of Video Streaming in Future Networks: A Tutorial with Comparative Study," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 88–94, 2021.

[2] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep Learning Enabled Semantic Communication Systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[3] Z. Weng and Z. Qin, "Semantic Communication Systems for Speech Transmission," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[4] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, 2021.

[5] L. Xia, Y. Sun, D. Niyato, X. Li, and M. A. Imran, "Wireless Semantic Communication: A Networking Perspective," *arXiv preprint arXiv:2212.14142*, 2022.

[6] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless Deep Video Semantic Transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 214–229, 2023.

[7] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless Semantic Communications for Video Conferencing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 230–244, 2023.

[8] L. Xia, Y. Sun, C. Liang, D. Feng, R. Cheng, Y. Yang, and M. A. Imran, "WiserVR: Semantic Communication Enabled Wireless Virtual Reality Delivery," *arXiv preprint arXiv:2211.01241*, 2022.

[9] E. C. Strinati and S. Barbarossa, "6G Networks: Beyond Shannon Towards Semantic and Goal-Oriented Communications," *Computer Networks*, vol. 190, p. 107930, 2021.

[10] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] J. Cao, X. Weng, R. Khirodkar, J. Pang, and K. Kitani, "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking," *arXiv preprint arXiv:2203.14360*, 2022.

[13] M. Yang and H.-S. Kim, "Deep Joint Source-Channel Coding for Wireless Image Transmission with Adaptive Rate Control," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5193–5197.

[14] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless Image Transmission Using Deep Source Channel Coding With Attention Modules," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2315–2328, 2021.

[15] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video Frame Interpolation with Transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3532–3542.

[16] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[17] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," in *European conference on computer vision*. Springer, 1994, pp. 151–158.

[18] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989.

[19] Z. Cai, J. Hao, P. Tan, S. Sun, and P. Chin, "Efficient Encoding of IEEE 802.11n LDPC Codes," *Electronics Letters*, vol. 42, no. 25, p. 1, 2006.

[20] K. Corona, K. Osterdahl, R. Collins, and A. Hoogs, "MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1060–1068.