IBC2022

AN ULTRA-LOW BITRATE VIDEO CONFERENCING SYSTEM WITH FLEXIBLE VIRTUAL ACCESS PATTERNS

Jun Xu, Zhiyu Zhang, Jun Ling, Haoyong Li, Bingcong Lu, Li Song

Institute of Image Communication and Network Engineering,
Shanghai Jiao Tong University, China

ABSTRACT

The demand for remote work and online entertainment is surging annually, placing heightened challenge on bandwidth usage and experience quality of applications such as video conferencing. Video codecs in traditional video conferencing systems typically utilize a block-based hybrid coding architecture, which often have sub-optimal rate distortion performance and computational resource consumption in these scenarios. In addition, in low bit rate scenarios due to low bandwidth networks, traditional codecs may lead to a disastrous experience. In this paper, we propose an ultra-low bitrate video conferencing system with flexible virtual access patterns. Conventional video codecs are partially or fully replaced to get ultra-low bitrate while ensuring a smooth communication experience. Furthermore, the three access patterns, face encoding, realistic and virtual avatar, can be driven with either video or audio modality and generate videos in different domain, providing a possible future video conferencing paradigm. The video captured by camera is not necessarily transmitted, protecting privacy of the users. Experiments demonstrate the excellent rate distortion quality and real-time performance of the proposed system.

INTRODUCTION

Since the emergence of Coronavirus pandemic in 2020, the industry and academia have seen substantial growth rates in terms of increased consumption and accelerated innovation topics, from remote collaboration to online entertainment. While the burst of demand for video conferencing and live entertainment presents massive opportunities, it also poses hungry demand for bandwidth. Conventional video systems typically encode captured video with a block-based hybrid encoding scheme, which is versatile and stable. However, for some specific scenarios such as video conference or virtual avatar in live streaming, block-based coding scheme is insufficient to decrease the semantic redundancy. We have the following findings and analyses:

(1) In video conferencing scenario, the video to be encoded is mainly about talking faces in a fixed background. General video encoding schemes, such as High Efficiency Video Coding (HEVC/H.265) [1], Versatile Video Coding (VVC/H.266) [2], and AV1 [3], is designed for arbitrary videos and focus on recovering pixel-level fidelity. However, we argue that the general codec is sub-optimal in video scenarios for the following reasons. First, the background images in video conferencing human faces are often static and

keep unchanged while the audiences will focus on the face region during the conference. Second, human faces commonly share similar structures and semantic meanings (e.g., eyes, mouth, and nose, etc.), which provide the opportunity to recover face details from less semantic cues by learning priors from face datasets. Recently, deep learning methods [4]–[8] have generation capability based on abridged information, promising potential in face video compression. These methods typically use some sparse representations like key points in place of some or all of the video frames, and use deep learning technique to recover these frames before rendering. In industry, NVIDIA has also released platforms and suites for audio and video communication with Artificial Intelligence (AI), such as NVIDIA Maxine [9], where the AI video compression solution also transports key points of the face at the sender and generates a reconstruction at the receiver with AI methods.

(2) As the animation industry has made a sharp increase in the market, the Virtual YouTuber (VTuber) have also grown rapidly. Many VTubers have shown their commercial value in the live streaming market and have a large number of fans on numerous social platforms such as YouTube, Niconico and Bilibili. In addition to live streaming, using virtual avatars to access video conferencing is becoming a new trend. Standardization organizations have also paid attention to such trends. Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) publishes the second version of Multimodal Conversation (MPAI-MMC) standard [10], where one use case is Avatar-Based Videoconference (ABV) [11]. The video to be encoded in virtual video conferencing and live streaming is just a virtual avatar doing slight movements in fixed scenes, which is similar to video conferencing except the faces. Furthermore, the virtual avatar itself is driven by some key points extracted from the human actor, which takes less data volume than encoded frames. Therefore, transferring key points and rendering them in real time on the client side may be an attractive solution to reduce bandwidth consumption.

(3) With the rise of new concepts such as metaverse, besides face encoding and virtual avatar, a realistic avatar is also a possible replacement of user’s real face in video conferences in the future. In this scenario, the conference system allows the user to animate any predefined avatar or his/her own face image via her real-time facial dynamics or even only his/her audios. To enable various access patterns, we propose an effective method for photo-realistic talking face rendering in video conferencing system. Our proposed method utilizes either the videos captured by a camera or the audio signal recorded by a recorder to synthesize virtual avatar or realistic talking face videos. It is worth mentioning that both the modalities of visual signal or audio signal are acceptable. With the proposed pipeline, one can join a conference with his/her voice and transfer the synthesized videos, which abolishes the necessity of real-time camera recording and ensure the user privacy not to be violated.

To overcome the sub-optimality of traditional encoding schemes in the above scenarios, and to explore new forms of entertainment, this paper proposes an ultra-low bitrate video conferencing system with various virtual access patterns. In this paper, we combine the latest developments in face video compression, virtual avatar and realistic talking face generation method with real time communication (RTC) to provide a practical video conferencing system.

The main contribution of this paper are as follows:

- This paper combines face video compression, virtual avatar and realistic face rendering with RTC. To the best of our knowledge, we make novel attempt to explore a new paradigm for video conferencing system which is not seen in prior studies.
- The proposed video conferencing system provides acceptable and better results than conventional video codec schemes under ultra-low bitrate constraints, meet the needs of real-time communication in bandwidth-constrained network environments.
- Our developed prototype system is ready for practical deployment and will soon be released on https://github.com/sjtu-medialab/virtualConference.

The remainder of this paper is organized as follows. We first discuss related works in the following section. Next, we demonstrate the architecture of the whole system and describe each of the modules in detail. Then, we conduct adequate experiments to indicate the performance of the system in bitrate and latency aspects. Finally, we summarize the entire work and discuss the future directions on our video conferencing system.

RELATED WORK

Video Conference and Protocol

Since the COVID-19 pandemic, the demand for Real-Time Communication (RTC) applications has increased rapidly. Especially, online video conference became the most important way for people to communicate, work and study [12]. Users can access the video conferencing system in various ways, such as mobile devices, personal computer, and room systems. While with the expansion of application scale and the computing pressure arisen by advanced features, most systems provide cloud-based services, instead of traditional on-premises video system. From the perspective of multimedia technologies, the video conferencing system realizes the seamless exchange of audio, video and content through audio and video coding, quality optimization, encryption, transmission and many other modules. To realize good Quality of Experience (QoE) in complex network environment, video conferencing system has strict requirements of latency and bandwidth utilization. Therefore, transmission scheme, especially the choice of protocol is of vital importance.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the two most widely used transport-layer protocols. While for video conferencing system, timeliness is more important than reliability. Thus, although TCP can guarantee sequenced and reliable transmission, its high-latency caused by head-of-line blocking and simple retransmission scheme makes TCP obsolete by RTC applications. Most modern video conferencing systems, such as WebRTC-based ones [13], commonly apply UDP-based protocols: Real-Time Transport Protocol (RTP) [14] and its variant Secure Real-time Transport Protocol (SRTP) [15]. As representative enterprise in the video conference industry, Zoom reportedly proposed a custom extension of RTP [16].

Recently, another UDP-based protocol Quick UDP Internet Connections (QUIC) [17] has aroused widespread concern because of its excellent performance and high flexibility. QUIC only needs one Round-Trip Time (RTT) to establish a reliable and secure connection, which is much more efficient than TCP's three-way handshake. Moreover, QUIC has the ability to handle stream-multiplexing and connection migration, which further improved transmission performance under unstable circumstances. Besides, QUIC has pluggable congestion control module, which makes the congestion control algorithm upgrading very convenient.

The prospect of QUIC is so bright that there are many studies on adopting it into real-time video streaming, such as RTP over QUIC [18] and extension of unreliable transmission [19].

Face Encoding

Conventional video encoding techniques utilize some manually designed schemes to eliminate redundant information. Based on hybrid video compression framework, many conventional video encoding methods have been proposed like HEVC and VVC. Moreover, VVC represents the most advanced conventional method.

With the development of deep learning techniques, many generation methods achieve considerable advances in talking face generation which can be applied to talking face video compression. Feng et al. [4] proposed a generative video compression framework based on FSGAN [20] which achieve a low bit rate around 1 KB/s. FOMM [21] uses key points and Jacobians to represent sparse motion which then is used to animate talking face. Omitting the Jacobians, Tang et al. [6] only relies on key points to characterize motion and propose a hybrid compression scheme for face video which achieves better quality and lower bit rate. Oquab et al. [5] designs a mobile-capable architecture based on FOMM while the quality may not be satisfactory. Combining 3D information, Wang et al. [7] proposes a framework that can generate free-view talking face video, while the training process is too hard, consuming significant time and computing resources. Based on FOMM, Konuko et al. [8] utilizes one raw frame as reference frame and add generated frames to reference frame pool, which may cause error accumulation.

Realistic Talking Face Generation

Realistic talking face generation aims at generating talking faces that match the input conditions including audio, facial landmarks, segmentation maps, or text. In the past years, plenty of methods have been proposed to achieve realistic talking face generation. One branch of methods utilizes audio as input to synthesize face videos. Guo et al. [22] proposes a NeRF-based method that takes audio speech and facial parameters ad input to generate talking face. However, NeRF works in modelling static scene but fails to deal with dynamic motions in talking videos. Consequently, [22] easily synthesize videos that suffer from jittering issues. Thies et al. [23] proposes to estimate the expressions from audio feature and renders photo-realistic features from neural textures with a simple U-Net. Wav2Lip [24] synthesizes lips from audio and background images. On another branch of methods that use text, facial landmarks, or segmentation maps as input to generate the target face images. Chen et al. [25] regresses facial landmarks to render facial images. Xue et al. [26] proposes adopting the face segmentation map to animate face images. However, the reconstruction quality of such method heavily depends on the accuracy of segmentations, and is not practical for realistic talking face generation. The text-based method [27] generates talking videos with text input. However, to bridge the gap between text and image, [27] needs to optimize the renderer in dozens of hours of training videos for single person.

Audio Encoding

Conventional audio codecs combine traditional coding tools such as linear prediction techniques and modified discrete cosine transform, to deliver high coding efficiency over different content types, bitrates and sampling rates, while ensuring low-latency for real-time audio communications. Opus[28], EVS[29], and USAC[30] are state-of-the-art (SOTA) conventional audio codecs.

End-to-end neural audio codecs rely on data-driven methods to learn efficient audio representations, instead of relying on handcrafted signal processing components. Lyra [31] is a generative model that encodes quantized mel-spectrogram features of speech, which are decoded with an auto-regressive WaveGRU model to achieve impressive results at 3 kbps. SoundStream [32] is a novel neural audio codec which relies on a model architecture composed by a fully convolutional encoder/decoder network and a residual vector quantizer, which are trained jointly end-to-end. As is reported in [32], it achieves SOTA results in all bitrates.

SYSTEM ARCHITECTURE

Overview

The proposed system provides three access patterns, which are face encoding, virtual avatar and realistic avatar. These three workflows share a similar system architecture, but differ in the way each module is processed and the data flows.

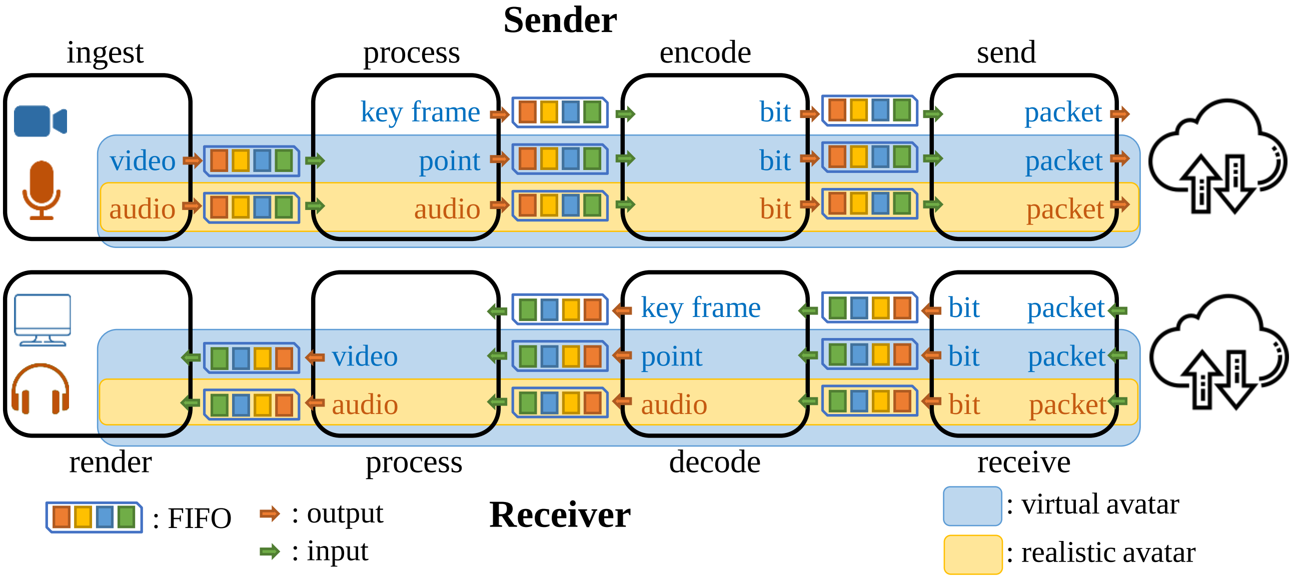


Figure 1 System architecture for face encoding workflow. The blue and yellow boxes indicate the parts of the workflow for virtual avatar and realistic avatar, respectively.

The system architecture is shown in Figure 1, which is actually the system architecture of face encoding workflow. There are three data flows in the system, key frame, point and audio. All of the three are used in the face encoding workflow. The latter two are used in the virtual avatar workflow while the realistic avatar workflow needs only audio flow, as shown in the blue and yellow boxes in Figure 1.

The modules are connected to each other via some FIFOs in which data flows through. For clarity of presentation, different data flows use different FIFOs in the figure, which is not the necessary the case in the actual system.

Face Encoding Workflow

Face encoding workflow provides a similar experience to traditional video conferencing, but with significant bitrate savings at the same quality because of the generation-based face encoding scheme.

Ingestion and rendering

The ingest module in the sender is responsible for interacting with the camera and the microphone to obtain video and audio data. The data is then put into the FIFOs in frames and passed to the subsequent modules.

The rendering module in the receiver is responsible for displaying the acquired video while playing the acquired audio.

Processing

To encode face videos with the generation-based scheme, the method in [6] is used in our system. The procedure is shown in Figure 2.

In the encoder, according to key frame frequency *f*, video frames are divided into key frames and non-key frames. Specifically, the first, $(f+1)^{\text{th}}$, $(2f+1)^{\text{th}}$, ⋯, frames are key frames while others are non-key frames. For key frames, the data is passed into the video encoding module, while non-key frames are fed to Key Point Detector (KPD) to extract key points. The extracted points are sent to the point encoding module to be encoded into bitstream.

In decoder, the video bitstream and the key point bitstream are decoded to key frames and key points of non-key frames. When key points of non-key frames and the nearest key frame before and after them are available, they are sent to the Generator (GEN) to generate non-key frames.

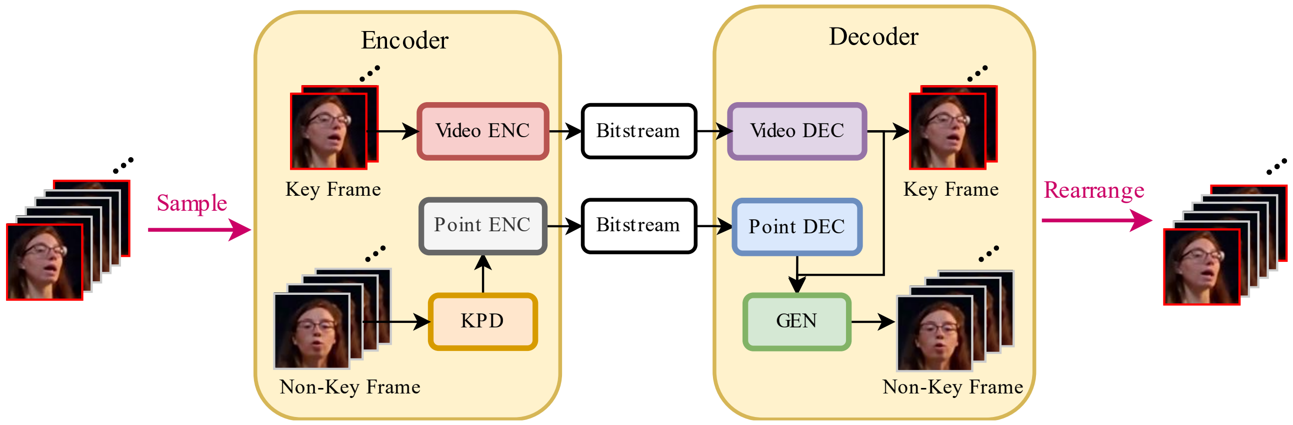


Figure 2 The procedure of [6]. In our system, the encoder is in the sender side while the decoder is in the receiver side, respectively.

The KPD is used to detect ten key points for each frame. Each key point is a two-dimensional normalized coordinate containing two float point numbers, representing sparse motion for talking faces.

The GEN firstly combines key points of both key frame and non-key frames to create sparse motion which are then used to warp key frame. The warped key frame and sparse motion are used to predict dense motion and occlusion map. Then, the original key frames are warped by dense motion and are masked by occlusion map. Finally, the warped features are decoded by a decoder network to generate reconstructed non-key frames.

In our system, the encoder structure and the decoder structure are located at the sender and receiver sides, respectively. On sender side, key frames are sent directly to the next encoding module. Non-key frames are sent to the KPD to extract key points first. The extracted key points are then fed to the encoding module. In receive side, when a new key frame is available, it is fed to the GEN with the previous key frame and the key points of the non-key frames between them to generate the images of non-key frames.

The audio data is sent directly to the next module without any processing.

Encoding and decoding

In encoding module, the key frames are encoded with low delay configured video codec without bi-directional interpolated prediction frames for the purpose of not introducing additional coding latency. Relatively high quality is needed for key frames to guarantee the quality of the generated non-key frames.

The key points of non-key frames are quantized to integers of eight or twelve bits, intra and inter predicted and coded with zero order Exponential-Golomb coding and adaptive arithmetic coding.

For audio, Lyra in [31], [33] is used in our system because it is open source and has been designed specifically for speech coding scenarios with ultra-low bit rates and acceptable quality.

In the decoding module, the corresponding decoders for key frames, key points and audio are used to get the reconstructed data.

Sending and receiving

Because our system needs flexible and customized application layer design, we directly use QUIC as underlying transport protocol. Quiche [34] is an implementation of the QUIC transport protocol and HTTP/3 as specified by the IETF, which implements QUIC kernel in Rust, and provides C/C++ APIs. Based on Quiche library, we designed send and receive module.

Send module fetches encoded data and package them into QUIC packets. Then packets are sent to peer in accordance with the order from control modules of QUIC such as flow control, congestion control and priority scheduling. Receive module receives packets and feedback states information to guide further packet sending. Besides, it unpacks packets and sends them into subsequent processing.

Virtual Avatar Workflow

In the virtual avatar workflow, the modules are similar to that in face encoding workflow, except the process module. Besides, because of the fact that instead of keyframes, only the key points of all frames are needed to drive the avatar, the key frame data flow is unused in this workflow. The audio flow is exactly the same as that in face encoding, while the key point flow is slightly different.

In our system, we use kalidokit [35] as a bridge between human face landmarks and virtual avatar driving parameters. Kalidokit is a blendshape and kinematics solver for Mediapipe [36] face and eyes tracking models, compatible with many SOTA face landmark detection methods. It takes 3D landmarks and calculates simple euler rotations and blendshape face values.

In the process module, a video frame is first fed into the face landmark detection sub-module to get the face landmarks. The landmarks are further processed by kalidokit to obtain the avatar drive parameters, which is sent to the net encoding module. The reason why drive parameters are passed to the next module instead of face landmarks is that their data volume is much smaller than face landmarks.

The drive parameters are some float numbers in different ranges, which are quantized according to the range. The quantized parameters are intra and inter predicted and encoded with zero order Exponential-Golomb coding and adaptive arithmetic coding to become bitstream.

In receive side, after the parameters of a frame are decoded and dequantized, the avatar model is driven with them.

Realistic Avatar Workflow

In the realistic avatar workflow, the modules are also similar to that in virtual avatar workflow, except the process module, which is described next.

Processing

Different from previous work, we utilize a 3D-based method to synthesize realistic talking face videos. Our system can take either audio or an additional driven image sequence as input, then generate novel talking face videos that is in synchronization with the audio track or the other image sequence.

To synthesize realistic talking face videos, we propose a comprehensive framework that can be driven by either audio input or another video faces. In brief, we first estimate facial shape and pose parameters from driving face images via [37], and combine the shape and pose parameters with 3D face expressions predicted from audio input using [38]. Then, we render face shapes with 3D Morphable Model [37] in GPU. Finally, we utilize a neural render to synthesize realistic images from input face shapes. Figure 3 shows the framework for virtual avatar and realistic avatar.

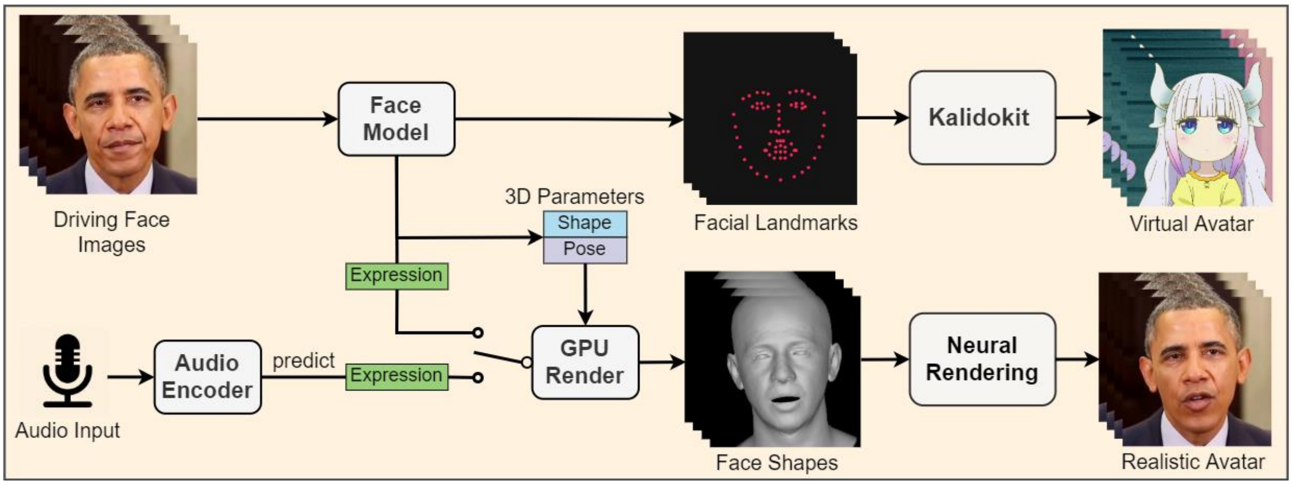


Figure 3 Our proposed framework for virtual avatar and realistic avatar.

To leverage the input of driving face images, we first estimate the facial landmarks or 3D facial expressions. The facial landmarks are used to analyse the motions of driving face and generate the videos of virtual avatar with kalidokit. The 3D facial parameters, including shape, pose and expression, are used for the generation of realistic avatar, where we first adopt a GPU render to synthesize face shapes, and then generate realistic avatar with a neural renderer. To use only the audio track as input to generate talking avatar, we extract the audio features to Mel-spectrum and predict the corresponding 3D face expressions. The estimated expressions are then combined with the shape and pose parameters (predefined if we only use audio) to render face shapes. Note that our method can use either the expressions from driving faces or from the audios. This allows the user to hide his face from the camera and generate talking face with only his speech.

This framework has two clear merits for video conferencing system. First, we can generate the talking face videos of both virtual avatar and realistic avatar in one system. Second, our framework can use different modalities of input to generate talking videos, which can better protect user privacy.

Synthesize summary

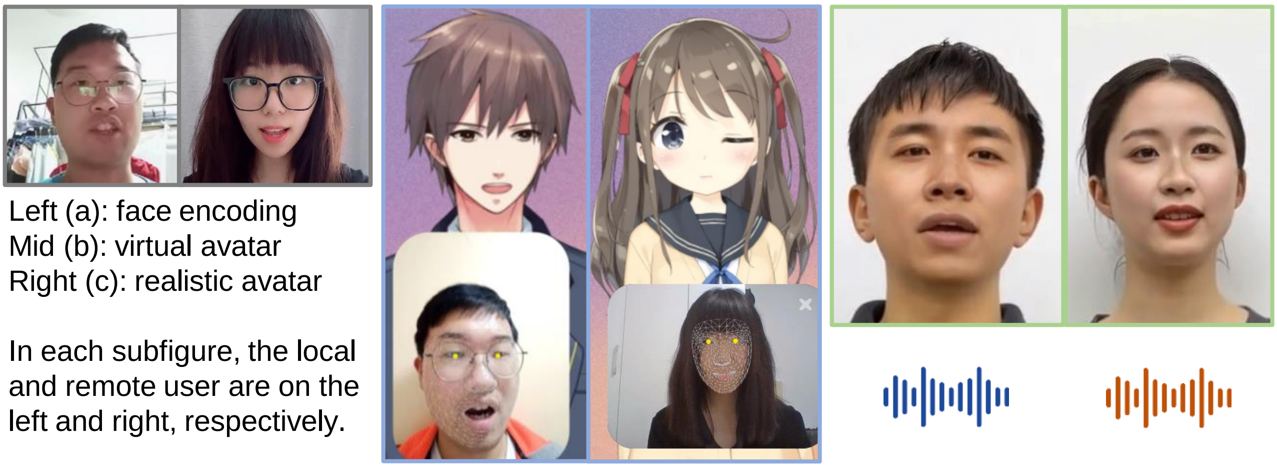
In the realistic avatar workflow, the process module in the sender side is responsible for extracting 3D facial parameters or just pass the audio data if we only use audio to render. The process module in the receiver side performs the whole framework with the acquired parameters (if have) and audio data to synthesize realistic images.

Besides audio encoding, the encode and decode module encodes and decodes the 3D facial parameters using the same way in virtual avatar workflow if we transfer them and use in the realistic avatar generation.

EXPERIMENT

Environment

In this section, we present the experimental results of the proposed system. The system is written in python with the operating system ubuntu 20.04 64bit. The whole system runs in a single computer with an i9-10900 CPU, 32-GB RAM, and a CUDA enabled NVIDIA GTX 1080Ti GPU.



Left (a): face encoding
Mid (b): virtual avatar
Right (c): realistic avatar

In each subfigure, the local and remote user are on the left and right, respectively.

Figure 4 The screenshots of the system in operation.

Figure 4 shows the screenshots of the system in operation, subfigure (a) in the grey boxes stands for face encoding, (b) in the blue boxes for virtual avatar and (c) in the green boxes for realistic avatar. In each subfigure, the local and remote user are on the left and right, respectively. In the subfigure (a), most frames of the remote user are generated locally. In (b), the net grid and yellow dot indicate the extracted key points. The points are transmitted to the other side and drive the virtual avatar. In (c), the realistic avatars are driven with audio. More demos can be seen on our GitHub repository.

To test the proposed system, we use two computers, one PC with the above devices and the other a regular laptop. During the experiment, the whole system is running on the PC while just a receive and a send module is running on the laptop, which means the laptop receives the data from the PC and send it back. With this approach, the experiment requires only one high-performance device.

Bitrate

For a video conferencing system, the most basic and critical metrics are bitrate and latency. Test video is 906 frames captured from a 720p 25fps camera. The frames are cropped and resized to 256x256 in face encoding and the key frame frequency is set to 5. The bitrate performance of the proposed system is shown in Table 1.

Table 1 The bitrate performance of the proposed system.

Workflow	Face encoding	Virtual avatar	Realistic avatar
Bitrate(kbps)	7.38(video)+3(audio)	0.77(video)+3(audio)	3(audio)

For audio encoding, lyra extracts features from speech every 40ms and are then compress them for transmission at a bitrate of 3kbps.

For face encoding, to encode key frames, [6] uses VVenC [39], which is an open source and optimized VVC encoder implementation, to obtain the optimal coding performance and relatively high coding efficiency. However, even VVenC and VVdeC are still quite far from practical use because of the extremely high coding complexity of VVC and its resulting coding latency. Therefore, we use x265 [40] as the video encoder for encoding key frames because of its stability, fast encoding speed and decent RD performance. For comparison, x265 uses the similar parameters to directly encode the original video at bitrate of 29.25kbps, while maintaining almost the same PSNR.

Latency

The RTT between the test devices is 2.48ms, tested with the Ping command. The latency performance of the proposed system is shown in Table 2. The audio flow latency is only in transfer (4.17ms) and decoding (60ms), thus is not shown in the table.

Table 2 The latency performance of the proposed system. K stands for key frame, and NK stands for non-key frame. The --- means nothing is processed here besides audio.

Workflow	Face encoding		Virtual avatar	Realistic avatar
	K	NK		
Process(sender) (ms)	<0.1	10.91	14.44	---
Encode (ms)	1.13	0.14	0.12	---
Transfer (ms)	4.31	3.69	4.09	---
Decode (ms)	0.14	<0.1	<0.1	---

Process(receiver) (ms)	<0.1	39.08	<0.1	79.61
End to end (ms)	205.57		60.10	139.61

The end-to-end latency for face encoding is relatively high because the generation of non-key frames requires the next key frame, which introduces latency of waiting other frames. When the next key frame is available, the last key frame can be displayed while the non-key frames are generated simultaneously without extra latency.

For virtual avatar workflow, the main latency bottleneck is audio decoding because the video related modules only consume a minimum amount of computing time. For realistic avatar workflow, audio decoding and avatar driving each causes half of the latency.

CONCLUSION

In this paper, we propose and implement an ultra-low bitrate video conferencing system with flexible virtual access patterns. Based on the SOTA developments in face encoding, virtual and realistic avatar driving, we achieve the goal of ultra-low bitrate by encoding and transmitting semantic and control information, partially or fully replacing video frames. Besides saving bandwidth usage, the virtual and realistic avatar access pattern protect the user privacy by not transmitting the captured video frames, but just some key points and audio data. The system, which will be open source soon, provides a good quality of experience and is easy to actually deploy.

In the future, we plan to investigate adaptive switching strategy for different access methods based on network conditions, which will further improve the flexibility and QoS of video conferencing system

ACKNOWLEDGEMENTS

This work was supported in part the National Key Research and Development Project of China under Grant 2019YFB1802701, MoE-China Mobile Research Fund Project under Grant MCM20180702, and Shanghai Key Laboratory of Digital Media Processing and Transmissions.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.

[2] B. Bross *et al.*, “Overview of the Versatile Video Coding (VVC) Standard and its Applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021, doi: 10.1109/TCSVT.2021.3101953.

[3] J. Han *et al.*, “A Technical Overview of AV1,” *Proc. IEEE*, pp. 1–28, 2021, doi: 10.1109/JPROC.2021.3058584.

[4] D. Feng, Y. Huang, Y. Zhang, J. Ling, A. Tang, and L. Song, “A Generative Compression Framework For Low Bandwidth Video Conference,” in *2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–6. doi: 10.1109/ICMEW53276.2021.9455985.

[5] M. Oquab *et al.*, “Low Bandwidth Video-Chat Compression using Deep Generative Models,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA, Jun. 2021, pp. 2388–2397. doi: 10.1109/CVPRW53098.2021.00271.

[6] A. Tang *et al.*, “Generative Compression for Face Video: A Hybrid Scheme,” *ArXiv220410055 Eess*, Apr. 2022, Accessed: Apr. 22, 2022. [Online]. Available: http://arxiv.org/abs/2204.10055

[7] T.-C. Wang, A. Mallya, and M.-Y. Liu, “One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 10034–10044. doi: 10.1109/CVPR46437.2021.00991.

[8] G. Konuko, G. Valenzise, and S. Lathuilière, “Ultra-low bitrate video conferencing using deep image animation,” *ArXiv201200346 Cs*, Dec. 2020, Accessed: Apr. 22, 2022. [Online]. Available: http://arxiv.org/abs/2012.00346

[9] “NVIDIA Maxine,” *NVIDIA Developer*, Oct. 01, 2020. https://developer.nvidia.com/maxine (accessed Jul. 22, 2022).

[10] “MPAI-MMC,” *MPAI community*. https://mpai.community/standards/mpai-mmc/ (accessed May 01, 2022).

[11] “Virtual Secretary for Videoconference,” *MPAI community*, Apr. 30, 2022. https://mpai.community/2022/04/30/virtual-secretary-for-videoconference/ (accessed May 01, 2022).

[12] H. Pratama, M. N. A. Azman, G. K. Kassymova, and S. S. Duisenbayeva, “The Trend in Using Online Meeting Applications for Learning During the Period of Pandemic COVID-19: A Literature Review,” *J. Innov. Educ. Cult. Res.*, vol. 1, no. 2, Art. no. 2, Dec. 2020, doi: 10.46843/jiecr.v1i2.15.

[13] B. Jansen, T. Goodwin, V. Gupta, F. Kuipers, and G. Zussman, “Performance Evaluation of WebRTC-based Video Conferencing,” *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 3, pp. 56–68, Mar. 2018, doi: 10.1145/3199524.3199534.

[14] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” Internet Engineering Task Force, Request for Comments RFC 3550, 2003. doi: 10.17487/RFC3550.

[15] K. Norrman, D. McGrew, M. Naslund, E. Carrara, and M. Baugher, “The Secure Real-time Transport Protocol (SRTP),” Internet Engineering Task Force, Request for Comments RFC 3711, 2004. doi: 10.17487/RFC3711.

[16] B. Marczak and J. Scott-Railton, “Move Fast and Roll Your Own Crypto: A Quick Look at the Confidentiality of Zoom Meetings,” University of Toronto, Citizen Lab Research Report No. 126, Apr. 2020. Accessed: Apr. 25, 2022. [Online]. Available: https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/

[17] A. Langley *et al.*, “The QUIC Transport Protocol: Design and Internet-Scale Deployment,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, Los Angeles CA USA, Aug. 2017, pp. 183–196. doi: 10.1145/3098822.3098842.

[18] C. Perkins and J. Ott, “Real-time Audio-Visual Media Transport over QUIC,” in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, Heraklion Greece, Dec. 2018, pp. 36–42. doi: 10.1145/3284850.3284856.

[19] M. Palmer, T. Krüger, B. Chandrasekaran, and A. Feldmann, “The QUIC Fix for Optimal Video Streaming,” in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, Heraklion Greece, Dec. 2018, pp. 43–49. doi: 10.1145/3284850.3284857.

[20] Y. Nirkin, Y. Keller, and T. Hassner, “FSGAN: Subject Agnostic Face Swapping and Reenactment,” 2019, pp. 7184–7193. Accessed: Apr. 24, 2022. [Online]. Available:

https://openaccess.thecvf.com/content_ICCV_2019/html/Nirkin_FSGAN_Subject_Agnostic_Face_Swapping_and_Reenactment_ICCV_2019_paper.html

[21] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First Order Motion Model for Image Animation," in *Advances in Neural Information Processing Systems*, 2019, vol. 32. Accessed: Apr. 24, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/31c0b36aef265d9221af80872ceb62f9-Abstract.html

[22] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, "AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis," 2021, pp. 5784–5794. Accessed: Apr. 30, 2022. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Guo_AD-NeRF_Audio_Driven_Neural_Radiance_Fields_for_Talking_Head_Synthesis_ICCV_2021_paper.html

[23] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural Voice Puppetry: Audio-Driven Facial Reenactment," in *Computer Vision – ECCV 2020*, Cham, 2020, pp. 716–731. doi: 10.1007/978-3-030-58517-4_42.

[24] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar, "A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild," in *Proceedings of the 28th ACM International Conference on Multimedia*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 484–492. Accessed: Apr. 29, 2022. [Online]. Available: https://doi.org/10.1145/3394171.3413532

[25] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical Cross-Modal Talking Face Generation With Dynamic Pixel-Wise Loss," 2019, pp. 7832–7841. Accessed: Apr. 30, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Hierarchical_Cross-Modal_Talking_Face_Generation_With_Dynamic_Pixel-Wise_Loss_CVPR_2019_paper.html

[26] H. Xue, J. Ling, L. Song, R. Xie, and W. Zhang, "Realistic Talking Face Synthesis With Geometry-Aware Feature Transformation," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1581–1585. doi: 10.1109/ICIP40778.2020.9190699.

[27] O. Fried *et al.*, "Text-based editing of talking-head video," *ACM Trans. Graph.*, vol. 38, no. 4, p. 68:1-68:14, Jul. 2019, doi: 10.1145/3306346.3323028.

[28] "hjp: doc: RFC 6716: Definition of the Opus Audio Codec." https://www.hjp.at/doc/rfc/rfc6716.html (accessed Apr. 22, 2022).

[29] M. Dietz *et al.*, "Overview of the EVS codec architecture," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 5698–5702. doi: 10.1109/ICASSP.2015.7179063.

[30] M. Neuendorf *et al.*, "The ISO/MPEG Unified Speech and Audio Coding Standard—Consistent High Quality for All Content Types and at All Bit Rates," *J. Audio Eng. Soc.*, vol. 61, no. 12, pp. 956–977, Dec. 2013.

[31] W. B. Kleijn *et al.*, "Generative Speech Coding with Predictive Variance Regularization," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 6478–6482. doi: 10.1109/ICASSP39728.2021.9415120.

[32] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 495–507, 2022, doi: 10.1109/TASLP.2021.3129994.

[33] *Lyra: a generative low bitrate speech codec*. Google, 2022. Accessed: Apr. 25, 2022. [Online]. Available: https://github.com/google/lyra

[34] *cloudflare/quiche*. Cloudflare, 2022. Accessed: Apr. 25, 2022. [Online]. Available: https://github.com/cloudflare/quiche

[35] Rich, *yeemachine/kalidokit*. 2022. Accessed: Apr. 25, 2022. [Online]. Available: https://github.com/yeemachine/kalidokit

[36] C. Lugaresi *et al.*, “MediaPipe: A Framework for Building Perception Pipelines,” *ArXiv190608172 Cs*, Jun. 2019, Accessed: Apr. 25, 2022. [Online]. Available: http://arxiv.org/abs/1906.08172

[37] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, “Learning an animatable detailed 3D face model from in-the-wild images,” *ACM Trans. Graph.*, vol. 40, no. 4, p. 88:1-88:13, Jul. 2021, doi: 10.1145/3450626.3459936.

[38] L. Chen, Z. Wu, J. Ling, R. Li, X. Tan, and S. Zhao, “Transformer-S2A: Robust and Efficient Speech-to-Animation,” *ArXiv211109771 Cs Eess*, Apr. 2022, Accessed: Apr. 30, 2022. [Online]. Available: http://arxiv.org/abs/2111.09771

[39] A. Wieckowski *et al.*, “Vvenc: An Open And Optimized Vvc Encoder Implementation,” in *2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–2. doi: 10.1109/ICMEW53276.2021.9455944.

[40] “x265, the free H.265/HEVC encoder - VideoLAN.” https://www.videolan.org/developers/x265.html (accessed Apr. 27, 2022).