Ultra-Lightweight Face Animation Method for Ultra-Low Bitrate Video Conferencing

LU Jianguo[1,2], ZHENG Qingfang[1,2]
(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202301008
https://kns.cnki.net/kcms/detail/34.1294.TN.20230224.1425.002.html, published online February 24, 2023
Manuscript received: 2022-08-25

Abstract: Video conferencing systems face the dilemma between smooth streaming and decent visual quality because traditional video compression algorithms fail to produce bitstreams low enough for bandwidth-constrained networks. An ultra-lightweight face-animation-based method that enables better video conferencing experience is proposed in this paper. The proposed method compresses high-quality upper-body videos with ultra-low bitrates and runs efficiently on mobile devices without high-end graphics processing units (GPU). Moreover, a visual quality evaluation algorithm is used to avoid image degradation caused by extreme face poses and/or expressions, and a full resolution image composition algorithm to reduce unnaturalness, which guarantees the user experience. Experiments show that the proposed method is efficient and can generate high-quality videos at ultra-low bitrates.

Keywords: talking heads; face animation; video conferencing; generative adversarial network

Citation (IEEE Format): J. G. Lu and Q. F. Zheng, "Ultra-lightweight face animation method for ultra-low bitrate video conferencing," *ZTE Communications*, vol. 21, no. 1, pp. 64 - 71, Mar. 2022. doi: 10.12142/ZTECOM.202301008.

1 Introduction

During the COVID-19 Pandemic, video conferencing systems have become indispensable tools for individuals to keep in touch with friends and for enterprises and organizations to connect with customers. Inside these systems, video compression technologies play critical roles in the efficient representation and transportation of video data. Great progress has been achieved in past years in representing high-fidelity videos with low bitrates; e.g., the high-efficiency video coding (HEVC)[1] was designed with the goal of allowing video content to have a data compression ratio up to 1 000:1. However, video conferencing systems still face the dilemma between smooth streaming and decent visual quality because current video compression technologies fail to produce bitstreams low enough for bandwidth-constrained networks due to a large number of concurrent users.

Recently, some novel talking-head video compression methods[2-5] based on face animation have been proposed, which can significantly cut down the bandwidth usage of video conferences. These face animation methods usually consist of two parts: encoder and decoder. The encoder is a motion extractor to derive a compact motion feature representation from the driving video frame, and the decoder is an image generator to synthesize photorealistic images according to the motion feature. Due to its extreme compactness, the extracted face feature can be used to reduce the bandwidth of video conferences and hence improve user experience in bandwidth-constrained networks. However, most of the talking-head video compression methods are too complicated to run in real time without the support of high-end graphics processing units (GPUs), let alone on mobile devices. For example, the model size of the First Order Motion Model (FOMM)[6] is 355 MB and the computation complexity is 121 G multiply-accumulate operations (MACs). Aiming at practical applications, we propose an ultra-lightweight motion extractor to obtain effective motion representations from the driving video and an animation generator to synthesize high-quality face videos accordingly.

We find out that the face animation method may sometimes fail, which is usually caused by extreme head poses and/or facial expressions. To tackle the problem, we propose an efficient visual quality evaluation method to reject the synthesized images that are visually unacceptable. We also notice that only displaying face without context regions looks unnatural and weird to users. To cope with it, we composite full-resolution images by stitching face regions with other body parts and backgrounds. These two mechanisms effectively prevent user experience degradation during a conference.

Our main contributions are as follows:

• An ultra-lightweight motion extraction algorithm is proposed to derive effective facial motion features from driving videos, which is efficient enough to run on mobile devices without high-end GPUs.

• An efficient visual quality evaluation algorithm is proposed to select visually acceptable generated images and an image composition algorithm to generate full-resolution videos, which ensures consistent and natural user experience during conferences.

• A practical video conferencing system is built to integrate the best parts of face-animation-based methods and traditional video-compression-based methods, which significantly reduces uplink bandwidth usage and ensures decent user experience even when the network bandwidth is constrained.

2 Related Work

Due to the space limitation, we only review previous works about face animation and deep video compression that are most related to ours.

2.1 Face Animation

Face animation is an image-to-image translation task, which transfers the talking-head motion of a person in an image to persons in other images. The former image is called the driving image, while the latter image is called the source image. Face animation has become a popular topic since the generative adversarial network (GAN) [7] was proposed by GOODFELLOW et al. Most recently published face animation methods can synthesize photo-realistic images with the help of GANs.

Some works[8 - 12] were proposed to solve the face animation task with the prior knowledge of the 3D Morphable Model (3DMM) [13]. However, the traditional 3D-based works[8 - 10] failed to render details of talking heads, such as hair, teeth and accessories. Ref. [11] allowed fine-scale manipulation of any facial input image into a new expression while preserving its identity with the help of a conditional GAN. To improve the realism of the rendering, Ref. [12] designed a novel space-time GAN to predict photorealistic video frames from the modified 3DMM directly.

Contrary to 3D-based models, 2D-based models synthesize talking heads directly without any prior knowledge of 3DMM. They can be classified into warping-based models and warping-free models.

Warping-free models[14 - 19] directly synthesize images without any warping. Few-shot vid2vid[16] learned to transform landmark positions into realistically looking personalized photographs with the help of meta-learning. Ref. [19] decomposed a person's appearance into a pose-dependent coarse image and a pose-independent texture image. LI-Net[20] decoupled the face landmark image into pose and expression features and reenacted those attributes separately to generate identity-preserving faces with accurate expressions and poses.

Warping-based methods[21 - 25] predicted dense motion fields to warp the feature maps extracted from the source images and inpaint the warped feature maps to generate photorealistic images. X2Face[22] used an encoder-decoder architecture to learn the latent embedding to encode pose and expression and recover the dense motion fields from it. Many works attempted to predict the dense motion field from sparse object keypoints. The key to those methods is how to represent motions with sparse object keypoints. Monkey-Net[23] was proposed to learn pure keypoints to describe motions in an unsupervised manner. Although it cannot describe subtle motions, Monkey-Net provided a strong baseline for further improvements. FOMM[6] represented sparse motion with some keypoints along with local affine transformations. Motion representations for articulated animation (MRAA)[24] defined the motion with regions using the motion estimation based on principal component analysis (PCA), rather than keypoints, to describe locations, shapes and pose. The thin-plate spline (TPS) motion model[25] estimated thin-plate spline motion to produce a more flexible optical flow. Ref. [5] extended the baseline to 3D optical flows to produce 3D deformations. The above mentioned methods extracted compact motion representations, which showed great potential in lowering the bitrate of video conferencing.

2.2 Deep Learning-Based Video Compression

For decades, researchers have made great efforts to transmit higher quality videos with lower bitrates. Recently several approaches based on deep learning were explored.

For general-purpose video compression, some works[26 - 27] attempted to reduce the bandwidth by making a balance between the cost of transferring the region of interest (ROI) and background. Compared to traditional codecs, such methods can achieve better visual quality with the same bitrate. Other works[28 - 29] focused on enhancing the visual quality of low bitrate videos by image super-resolution and image enhancement.

For the compression of talking-head videos, great progress has been achieved. In Ref. [30], the encoder detected and transmitted keypoints representing the body pose and the face mesh information, and the receiver displayed the motion in the form of puppets. However, this method failed to produce photorealistic images. Inspired by the promising results achieved by face animation models, many works demonstrated the effectiveness of video compression based on face animation. VSBNet[3] reconstructed original frames from face landmarks with a low bitrate of around 1 kB/s. Ref. [5] proposed a neural talking-head video synthesis model and set up a video conferencing system that achieves the same visual quality as the commercial H. 264 standard with only one-tenth of the bandwidth. Ref. [2] introduced an adaptive intra-refresh scheme to address the problem of reconstruction quality that might rapidly degrade due to the loss of temporal correlation as frames get farther away from the initial one. Ref. [4] evaluated the advantages and disadvantages of several deep generative adversarial approaches and designed a mobile-compatible architecture that can run at 19 f/s on iPhone 8. However, those methods can hardly run in real time without the support of high-end GPUs. What's more, they could only generate near-

frontal faces, looking unnatural and weird when faces were not near-frontal. In this paper, we specifically focus on improving the efficiency and visual quality of video compression based on face animation.

3 Proposed Ultra-Lightweight Face Animation Method

3.1 Overview

The overall pipeline of our video conference system is shown in Fig. 1. Each user provides an avatar image to the system and uses its animation during a conference for ensuring privacy and elegant presence. When the system starts running, videos of users are captured and the face region in each video frame is cropped out by the face detection algorithm. Face images are then encoded by the keypoint detector and represented as the keypoints described in Section 3.2. Before the encoded data are sent out, the visual quality of the face image that will be reconstructed by a decoder according to these keypoints is evaluated to prevent unnatural results. It is highlighted here that the visual quality evaluation method in Section 3.3 requires no actual reconstruction of the face image but executes on encoded data, for the sake of efficiency.

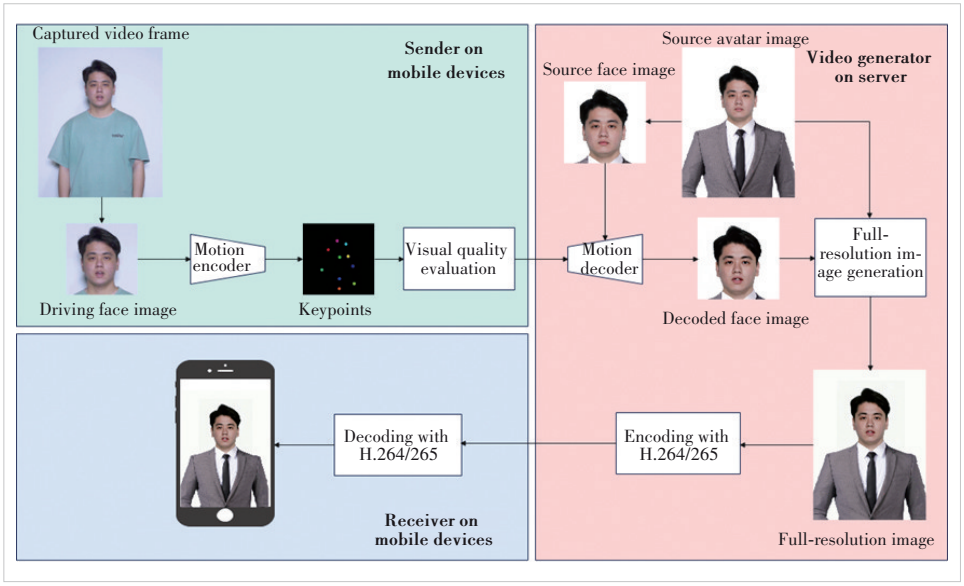
Upon receiving the encoded keypoint data from the sender, the conference server calls the image generator to synthesize the face image animated from the keypoints, as described in Section 3.2. The decoded face image replaces the face region in the avatar image by our method in Section 3.4 to create a full-resolution video frame, which is then encoded by H.264 or HEVC and sent to the receiver. The receiver simply decodes the video stream and displays it on the screen, which can usually take advantage of the hardware accelerator in the device's chip.

With the prevalence of mobile phones, the demand for running video conferencing on mobile devices is growing. In most commercial video conference systems, mobile devices account for a significant portion of all terminals. For better compatibility with existing commercial video conference systems, our system and algorithms here are intentionally designed to make the sender/receiver module deployable on mobile devices and to keep their computational burdens to a minimum, thus reducing power consumption and extending the working time of mobile devices.

3.2 Model Distillation

Giving a source image $\boldsymbol{S}$ of the target person, a driving video can be denoted as $\{\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \cdots, \boldsymbol{D}_N\}$, where $\boldsymbol{D}_i$ is the i-th frame in the sequence and N is the total number of frames in the video. The output images can be denoted as $\{\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \cdots, \boldsymbol{O}_N\}$, where $\boldsymbol{O}_i$ is the i-th frame of the output sequence. The output $\boldsymbol{O}_i$ shares the same identity with $\boldsymbol{S}$ and the same face motions with $\boldsymbol{D}_i$. We adopt the face animation model similar to FOMM, which consists of a keypoint detector $\boldsymbol{K}$ (encoder) and a generator $\boldsymbol{G}$ (decoder). First, face landmarks are estimated from $\boldsymbol{S}$ and $\boldsymbol{D}_i$ separately by $\boldsymbol{K}$, whose locations serve as the sparse motion information. Second, dense motion fields and occlusion maps are predicted by $\boldsymbol{G}$. Finally, $\boldsymbol{G}$ warps the feature map extracted from $\boldsymbol{S}$ with the dense motion fields and the warped feature map is masked by the occlusion maps to generate the output image $\boldsymbol{O}_i$. Following the idea of FOMM, we extract 10 keypoints and their corresponding Jacobian matrices from the face image.

We design our model to be lightweight and can generate an image with excellent visual quality. For the decoder, we adopt the same architecture as the generator model in FOMM but cut down the channels of the model by half. We denote the simplified generator as $\boldsymbol{G}_{\text{sim}}$. For the encoder, we replace the hourglass network in FOMM, which brings about high computational cost, with a greatly simplified version of MobileNetV2[31]. However, it is very difficult to train the proposed model from scratch since the training process often fails to converge. We come up with a training strategy described as fol-



▲ **Figure 1. Proposed video conference system consists of three parts: the sender on mobile devices, video generator on servers, and receiver on mobile devices. In the encoder part, the motion encoder extracts keypoints from the driving images. The feature-based image quality evaluation filters out unnatural images. The decoder synthesizes images from the keypoints and reconstructs full-resolution images, which are encoded by H.264 or H.265 and sent to the receiver. The receiver decodes the video stream and shows it on the phone screen**

lows to solve the problem.

1) Step 1: model distillation. We use the original encoder K_{fomm} in FOMM as the teacher model and our proposed encoder K_{pro} as the student model. The loss function consists of distillation loss L_{dis} and equivariance loss L_{eq}, which can be written as Eq. (1).

$$L_1 = L_{\text{dis}} + L_{\text{eq}} = \left| K_{\text{pro}}(I) - K_{\text{fomm}}(I) \right| + \left| K_{\text{pro}}(T(I)) - T(K_{\text{pro}}(I)) \right|, \quad (1)$$

where I is the training sample and T is a thin plane spline deformation. The distillation loss ensures that the student encoder extracts the same motion representation as the teacher encoder. And the equivariance loss ensures the consistency of the motion representation when random geometric transformations are applied to the images.

2) Step 2: iterative model pruning and distillation. Since the encoder has to extract motion representation from every video frame, it should be as lightweight as possible to reduce computational costs. In our attempt to further simplify the encoder, we find out most of the complexity comes from the last several convolutional layers. Therefore, we drop the last convolutional layer in the encoder model and retrain it following Step 1. This step can be repeated several times until we obtain K_{best} that strikes a balance between the model complexity and accuracy.

3) Step 3: generator fine-tuning. Due to the simplification made to the generator, we train the simplified generator G_{sim} along with the keypoint detector K_{fomm} of the original FOMM to make a good initialization of G_{sim}.

4) Step 4: overall fine-tuning. Once the encoder models K_{best} and G_{sim} are determined, we fine-tune K_{best} and G_{sim} accordingly in an unsupervised manner. Finally, K_{best} and G_{sim} act as the encoder and the decoder in our system respectively.

3.3 Visual Quality Evaluation

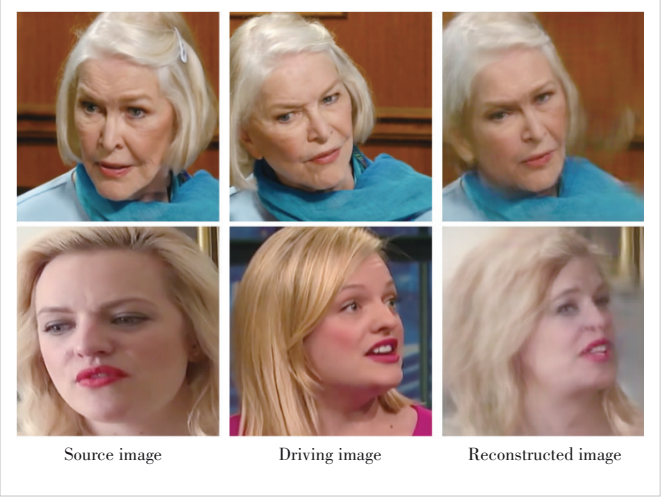
Although video conferences based on face animation can result in a very high video compression rate, the visual quality of a reconstructed image may sometimes degrade in the following two cases (Fig. 2). First, due to current algorithmic limitations, most of the face animation models may generate inaccurate expressions and visual artifacts on faces with large poses and/or extreme expressions. Second, with the increase of the frame distance, the temporal correlation weakens, and hence the quality of generated video deteriorates. This phenomenon becomes particularly obvious when faces are occluded. The degraded image brings inconsistent experience to users. In order to alleviate the problem, Ref. [2] introduced an adaptive intra-refresh scheme using multiple source frames. Before sending the features to the decoder, the sender reconstructs the image first and evaluates the generated image to avoid degraded images. However, this scheme not only incurs large computational costs which makes it impossible to run it on mobile devices, but also leads to significant time delay at the receiving end. What's more, frequent scene switching also requires the system's frequent sending of source frames, making the system lose its advantage of reducing video bandwidth.

We propose here an adaptive degraded frame filter method by an efficient image quality evaluation algorithm directly based on the extracted features. We find out that when a large head pose and/or extreme facial expression happens, most of the regions in the generated image are inpainted by the generator, which degrades the image quality. The difference between the driving image and the source image can be measured by analyzing the dense motion field, which is predicted from the sparse motion field in our setting. Therefore, instead of using the decoder to synthesize the generated image, we decide to evaluate image quality based on the relative motion. The loss L_2 in the algorithm can be formulated as follows.

$$L_2 = \alpha \sum_{i=0}^{10} \left\| v_{1i} - v_{2i} \right\| + \beta \sum_{i=0}^{10} \left\| J_{1i} J_{2i}^{-1} \right\|, \quad (2)$$

where v_{1i} is the value of the i-th keypoint in the first frame, v_{2i} is the value of the i-th keypoint in the second frame, J_{1i} is the Jacobian of the i-th keypoint in the first frame, J_{2i} is the Jacobian of the i-th keypoint in the second frame, and hyperparameters α and β control the weight of each part. In our experiments, we set the hyperparameters to 2 and 1 respectively.

In the proposed scheme, the balance between image quality and robustness is controlled by a threshold τ. Although the identity of the people in the driving images and the source image are the same, the two images may look different. For better visual quality, we adopt a relative motion transfer method, as described in Ref. [6]. We first find a driving image that has a



▲ **Figure 2. Examples of face animation failure. The first row shows a result caused by large-pose; the face area becomes blurred and there are some artifacts on the hair of the woman. The second row shows a degraded image caused by weak temporal correlation and the reconstructed image looks terrible and weird**

similar pose to the source image, which is called the initial image D_I. Then, we extract keypoints from the source image S and the initial image D_I, which can be denoted as Ks and K_I. The source keypoints are sent to the receiver. For every frame D_t, we estimate keypoints K_t from the frame, and compare the relative motion between K_t and K_s and that between K_I and K_s. If the former is smaller, we set this driving keypoint as an initial image. Finally, we compare the relative motion between K_t and K_I with the threshold τ. If the former is smaller, it means the relative motion is suitable for robust image generation. The relative motion is sent to the server. If the latter is smaller, the default motion is sent to avoid freezing in video streams. The default keypoints can be motions of some natural expressions, such as blinking and smiling. In this way, the degraded frames are replaced by frames of natural expressions. Compared to the method proposed in Ref. [2], our method can greatly reduce the computation cost at the sender and the delay at the receiver.

3.4 Full-Resolution Image Composition

The face animation described above cannot be directly used in video conferences due to two facts. Face animation cannot synthesize face images with a size up to video resolution (at least 1 280×720) because computational complexity grows exponentially with the image size. Also, only displaying the facial region on the screen without other body parts such as the neck and shoulder looks unnatural and weird. In order to make our face animation method applicable, instead of generating full-resolution images, we propose to generate a facial region with a size of no more than 384×384 and stitch it with other body parts and background regions in the source frame to form a full-resolution image. The problem is that there will be a sharp blocky artifact between the head region and body region because the head region moves while the body region may remain stationary. We find that the keypoints spread over the talking-head area and each keypoint is responsible for the local transformation of its neighborhood. To reduce the artifact, we fix the keypoints related to the shoulder part. As a result, the dense motion field predicted by the generator will stay stationary near the shoulder region and have a smooth transition from the head region to the shoulder region, which makes the composite image look more natural. We show the example images in Fig. 3 for comparison.

4 Experiments

4.1 Implementation Details

1) Datasets. We train and evaluate our face animation model on the VoxCeleb dataset and an in-house dataset. VoxCeleb[32] is a dataset of interview videos of different celebrities. We crop the videos and resize them to 256×256 for a fair comparison with the original FOMM and 384×384 for the generation of high-resolution images according to the bounding boxes of faces. The in-house dataset consists of 4 124 Chinese people videos collected from the Internet and is used to reduce bias towards Western people. We fine-tune our model on the in-house dataset to make better adaptations to Chinese.

2) Evaluation metrics. We evaluate the models using the L1 error, average keypoint distance (AKD) and average Euclidean distance (AED). The L1 error is the mean absolute difference between pixel values in the reconstructed images and the ground-truth images, which measures the reconstruction accuracy. AKD and AED stand for semantic consistency. AKD is the average distance between the face landmarks extracted from the ground-truth images and the reconstructed images respectively by the face landmark detector[33], which measures the pose difference between the two images. AED measures identity preservation, which is the L2 distance of the corresponding features extracted by a pre-trained re-identification network[34].

3) Hardware. In our video conference system, we implement a conferencing APP on a ZTE A30 Ultra mobile phone with Snapdragon 888 System on a Chip (SoC) and conferencing server software on a computer with Nvidia Tesla V100 GPU.

▲**Figure 3. Qualitative comparisons with state-of-the-art methods. The first three rows are images from the VoxCeleb dataset and the following four rows are images from our in-house dataset. Our method produces competitive results**

4.2 Comparisons with FOMM

1) Efficiency of the proposed face animation algorithm

First, we compare our encoder, i.e., the face motion extractor, with that of the original FOMM. We convert the encoder to the mobile neural network (MNN)[35] model and calculate the model size. As listed in Table 1, our encoder model is only 600 kB in size with theoretical computation complexity of 14.62 M MAC, both of which are about 1% of FOMM. Our encoder processes every frame in 3.5 ms on Snapdragon 888, which is 16.3 times faster than FOMM.

Second, we compare our decoder, i.e., the generator to synthesize a 384×384-resolution face image, also with FOMM. For the generator, we convert the model to TensorRT[36] model and calculate the model size. As listed in Table 1, our decoder model is 81.77 MB in size with theoretical computation complexity of 31.42 G MAC, and these two values are 26.0% and 27.3% of FOMM respectively. Our encoder runs in 5 ms on Tesla V100, which is 4 times faster than FOMM.

2) Effectiveness of the proposed face animation algorithm

We compare the visual quality of face images generated by our method with other face animation methods. For quantitative comparison, we evaluate our model with existing studies on the VoxCeleb dataset for an image generation task. For a fair comparison, we generate images with the resolution of 256×256. The first frame of each test video is set as the source image, while the subsequent frames are set as the driving images. Evaluation metrics are computed for every frame and our result is the mean value of all frames. The results are summarized in Table 2, which clearly shows the proposed method outperforms X2Face and Monkey-Net. Compared to FOMM, our method can generate competitive results, even though our model is much lighter than FOMM. For a qualitative comparison, we list some example images in Fig. 3 for visual comparisons.

▼ Table 1. Efficiency comparison between our face animation method and FOMM

Model		MAC	Parameters/M	Model size/MB	Inference time/ms
Encoder	FOMM	1 280 M	14.21	55.54	57
	Ours	14.62 M	0.16	0.60	3.5
Decoder	FOMM	120.70 G	45.56	299.10	20
	Ours	31.42 G	16.16	81.77	5

FOMM: First Order Motion Model MAC: multiply-accumulate operation

▼ Table 2. Visual quality comparison among different face animation methods on VoxCeleb dataset

	L1	AKD	AED
X2Face[22]	0.078	7.69	0.405
Monkey-Net[23]	0.049	1.89	0.199
FOMM[6]	0.041	1.27	0.134
Ours	0.043	1.37	0.147

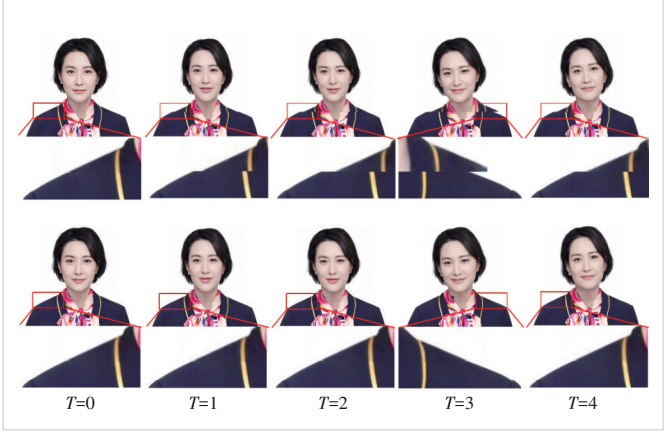
AED: average Euclidean distance FOMM: First Order Motion Model
AKD: average keypoint distance

4.3 Results of Full-Resolution Image Generation

The avatar images provided by a user are usually not face-only, but with other upper body parts. When head regions in the avatar images are cropped and animated by our method, they should be stitched back into original images to form new images with predefined resolutions, e.g., 1 280×720. Special treatment should be given to the point where the head region and body region connect because these regions move non-rigidly and disproportionately. As shown in the top two rows in Fig. 4, simply replacing the head region in an avatar image with a new animated head region will result in visual discontinuities. As comparisons, the bottom two rows show results of the proposed method described in Section 3.4. Our method successfully eliminates discontinuities and makes whole images visually natural.

4.4 Ultra-Low Bitrate Video Conference

As described in Section 3.1, our video conference system is comprised of server software running on the cloud server and application software, with the sender module and receiver module, running on the mobile phone. The most important difference between our sender module and those inside other video conference systems is we encode captured videos into compact keypoint motion information, rather than traditional H.264 or HEVC streams, which greatly cuts down the uplink bandwidth usage. For example, when encoded in H.264, 720 p conference videos are typical of bitrates between 1 Mbit/s and 2 Mbit/s. By comparison, each video frame is encoded by our sender module as 10 keypoint information, each of which includes a position (2 floating points) and a Jacobian matrix (4 floating points). We empirically determine the half precision floating point format (FP16) is enough for data representation and thus reaches the bitrate of 6×16×10×30=28.8 kbit/s, which is only less than 3% of H.264 encoding. We note the



▲ Figure 4. Results of full-resolution image generation. The first row shows images generated by simply replacing the head region in the source image with the new animated head region. The third row shows image results by our method in Section 3.4. In the second and fourth rows, connections between head regions and body regions are zoomed in for clearer comparison

keypoint information can be compressed by the entropy encoder for further bandwidth usage saving.

In our real-world user studies, reducing the uplink bitrate can greatly improve the conference user experience. For one thing, since wireless bandwidth is not evenly allocated for uplink and downlink data transportation, a smaller uplink bitrate can result in less congestion and faster upward transmission. For another thing, more aggressive schemes can be applied when Forward error correction (FEC) is used to tackle data loss in transmission, leading to less data retransmission, which brings about lower remote interaction latency and more real-time engagement.

The server software in our system runs on a cloud server with Nvidia GPUs because the image generator in face animation is much more computationally expensive than the keypoint extractor, as demonstrated in Section 4.1. Although our simplified image generator can be deployed on some flagship mobile phones with powerful GPUs, we choose server-side deployment to make our application software lightweight enough to run on most mobile phones and consume less power to extend working time, which is also critical to user experience.

5 Conclusions

In this paper, we propose a face-animation-based method to greatly reduce bandwidth usage in video conferences, compressing face video frames by using only 60 FP16 data to represent the face motion. We design an ultra-lightweight face motion extraction algorithm that runs on mobile devices, as well as an efficient visual quality evaluation algorithm and a full-resolution image composition algorithm to ensure consistent and natural user experience. We also build a practical system to enable user communication using animated avatars. Experimental results demonstrate the efficiency and effectiveness of our methods and their superiority over previous studies. However, one limitation of our current work is that our method is only applicable to upper-body videos. A full-body animation method should be our next work to cover more real-world scenarios. Another improvement to our system will be saving downlink bandwidth by reconstructing videos on mobile devices, which requires further research in GAN acceleration to meet real-time constraints on mobile devices.

References

[1] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. IEEE transactions on circuits and systems for video technology, 2012, 22(12): 1649 – 1668. DOI: 10.1109/TCSVT.2012.2221191

[2] KONUKO G, VALENZISE G, LATHUILIÈRE S. Ultra-low bitrate video conferencing using deep image animation [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 4210 – 4214. DOI: 10.1109/ICASSP39728.2021.9414731

[3] FENG D H, HUANG Y, ZHANG Y W, et al. A generative compression framework for low bandwidth video conference [C]//IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, 2021: 1 – 6. DOI: 10.1109/ICMEW53276.2021.9455985

[4] OQUAB M, STOCK P, GAFNI O, et al. Low bandwidth video-chat compression using deep generative models [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2021: 2388 – 2397. DOI: 10.1109/CVPRW53098.2021.00271

[5] WANG T C, MALLYA A, LIU M Y. One-shot free-view neural talking-head synthesis for video conferencing [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 10034 – 10044. DOI: 10.1109/CVPR46437.2021.00991

[6] SIAROHIN A, LATHUILIÈRE S, TULYAKOV S, et al. First order motion model for image animation [J]. Advances in neural information processing systems. 2019, 32: 7135 – 7145

[7] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets [J]. Advances in neural information processing systems. 2014, 27: 2672 – 2680

[8] VLASIC D, BRAND M, PFISTER H, et al. Face transfer with multilinear models [J]. ACM transactions on graphics, 2005, 24(3): 426 – 433. DOI: 10.1145/1073204.1073209

[9] DALE K, SUNKAVALLI K, JOHNSON M K, et al. Video face replacement [J]. ACM transactions on graphics, 2011, 30(6): 1 – 10. DOI: 10.1145/2070781.2024164

[10] THIES J, ZOLLHÖFER M, STAMMINGER M, et al. Face2Face: real-time face capture and reenactment of RGB videos [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 2387 – 2395. DOI: 10.1109/CVPR.2016.262

[11] NAGANO K, SEO J, XING J, et al. PaGAN: real-time avatars using dynamic textures [J]. ACM transactions on graphics, 2018, 37(6): 1 – 12. DOI: 10.1145/3272127.3275075

[12] KIM H, GARRIDO P, TEWARI A, et al. Deep video portraits [J]. ACM transactions on graphics (TOG), 2018, 37(4): 1 – 14. DOI: 10.1145/3197517.3201283

[13] BLANZ V, VETTER T. A morphable model for the synthesis of 3D faces [C]//26th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1999: 187 – 194. DOI: 10.1145/311535.311556

[14] BURKOV E, PASECHNIK I, GRIGOREV A, et al. Neural head reenactment with latent pose descriptors [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 13783 – 13792. DOI: 10.1109/CVPR42600.2020.01380

[15] OLSZEWSKI K, LI Z M, YANG C, et al. Realistic dynamic facial textures from a single image using GANs [C]//IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 5439 – 5448. DOI: 10.1109/ICCV.2017.580

[16] SONG Y, ZHU J W, LI D W, et al. Talking face generation by conditional recurrent adversarial network [C]//Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI, 2019: 919 – 925. DOI: 10.24963/ijcai.2019/129

[17] YU J H, LIN Z, YANG J M, et al. Generative image inpainting with contextual attention [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 5505 – 5514. DOI: 10.1109/CVPR.2018.00577

[18] ZAKHAROV E, SHYSHEYA A, BURKOV E, et al. Few-shot adversarial learning of realistic neural talking head models [C]//IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020: 9458 – 9467. DOI: 10.1109/ICCV.2019.00955

[19] ZAKHAROV E, IVAKHNENKO A, SHYSHEYA A, et al. Fast bi-layer neural synthesis of one-shot realistic head avatars [C]//European Conference on Computer Vision. Springer, 2020: 524 – 540. DOI: 10.1007/978-3-030-58610-2_31

[20] LIU J, CHEN P, LIANG T, et al. Li-Net: large-pose identity-preserving face reenactment network [C]//IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2021: 1 – 6. DOI: 10.1109/ICME51207.2021.9428233

[21] ZHAO R Q, WU T Y, GUO G D. Sparse to dense motion transfer for face image animation [C]//IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). IEEE, 2021: 1991 – 2000. DOI: 10.1109/ICCVW54120.2021.00226

[22] WILES O, KOEPKE A S, ZISSERMAN A. X2Face: a network for controlling face generation using images, audio, and pose codes [C]//European Conference on Computer Vision. Springer, 2018: 690 – 706. DOI: 10.1007/978-3-030-01261-8_41
[23] SIAROHIN A, LATHUILIÈRE S, TULYAKOV S, et al. Animating arbitrary objects via deep motion transfer [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 2372 – 2381. DOI: 10.1109/CVPR.2019.00248
[24] SIAROHIN A, WOODFORD O J, REN J, et al. Motion representations for articulated animation [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 13648 – 13657. DOI: 10.1109/CVPR46437.2021.01344
[25] ZHAO J, ZHANG H. Thin-plate spline motion model for image animation [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2022: 3647 – 3656. DOI: 10.1109/CVPR52688.2022.00364
[26] AGUSTSSON E, TSCHANNEN M, MENTZER F, et al. Generative adversarial networks for extreme learned image compression [C]//IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020: 221 – 231. DOI: 10.1109/ICCV.2019.00031
[27] KAPLANYAN A S, SOCHENOV A, LEIMKÜHLER T, et al. DeepFovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos [J]. ACM transactions on graphics, 2019, 38(6): 1 – 13. DOI: 10.1145/3355089.3356557
[28] LU G, OUYANG W L, XU D, et al. Deep kalman filtering network for video compression artifact reduction [C]//European Conference on Computer Vision. Springer, 2018: 591 – 608. DOI: 10.1007/978-3-030-01264-9_35
[29] GUO Y H, ZHANG X, WU X L. Deep multi-modality soft-decoding of very low bit-rate face videos [C]//28th ACM International Conference on Multimedia. ACM, 2020: 3947 – 3955. DOI: 10.1145/3394171.3413709
[30] PRABHAKAR R, CHANDAK S, CHIU C, et al. Reducing latency and bandwidth for video streaming using keypoint extraction and digital puppetry [C]//Data Compression Conference (DCC). IEEE, 2021: 360. DOI: 10.1109/DCC50243.2021.00057
[31] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 4510 – 4520. DOI: 10.1109/CVPR.2018.00474
[32] NAGRANI A, CHUNG J S, ZISSERMAN A. VoxCeleb: a large-scale speaker identification dataset [C]//18th Annual Conference of the International Speech Communication Association. ISCA, 2017: 2616 – 2620. DOI: 10.21437/interspeech.2017-950
[33] BULAT A, TZIMIROPOULOS G. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230 000 3D facial landmarks) [C]//IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 1021 – 1030. DOI: 10.1109/ICCV.2017.116
[34] AMOS B, LUDWICZUK B, SATYANARAYANAN M. Openface: a general-purpose face recognition library with mobile applications: CMU-CS-16-118 [R]. USA: School of Computer Science, Carnegie Mellon University, 2016. DOI:10.13140/RG.2.2.26719.07842
[35] JIANG X, WANG H, CHEN Y, et al. MNN: a universal and efficient inference engine [C]//Third Conference on Machine Learning and Systems. MLSys, 2020, 2: 1 – 13. DOI: 10.48550/arXiv.2002.12418
[36] NVIDIA. NVIDIA TensorRT [EB/OL]. [2022-02-22]. https://developer.nvidia.com/tensorrt

Biographies

LU Jianguo received his BS and MS degrees from Huazhong University of Science and Technology, China in 2017 and 2020 respectively. After graduation, he has been working at ZTE Corporation. His research interests include computer vision, artificial intelligence and augmented reality.

ZHENG Qingfang (zheng.qingfang@zte.com.cn) received his BS degree from Shanghai Jiao Tong University, China in 2002, and PhD degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences in 2008. He is now the chief scientist of cloud video product and deputy director of the Video Technology Committee at ZTE Corporation. His current research interests include video communication, computer vision and artificial intelligence.