Ultra Low Bit-rate Video Communication

Video Coding = Pattern Recognition

Hung-Son Lee, Ulrik Söderström and Haibo Li

Digital Media Lab,
Department of Applied Physics and Electronics, Umeå University,
901-87 Umeå, Sweden

Abstract. In this paper we introduce an ultra low bit-rate video communication scheme, which enables real-time video communication at a bitrate of as low as 100 bits/s. The magic behind this scheme is that the receiver will keep a personal facial mimic gallery and the transmitter tells which video frame the receiver should pick out from the gallery and display. Since only the index of video frames is transmitted, ultra-low communication bandwidth is needed. In the paper we describe how the video communication system works and how to handle the jitter problem in playback of reconstructed video with the locally linear embedding tool.

1 Introduction

Ultra low bitrate video coding means that video signals are compressed into bitrates around 100 bits/s. Technically, this is very challenging even today. It is believed that compressing videos into such low bitrate can only be achieved by 3D model-based coding. Truly, today realistic images can be created through computer graphics technology. A real-world image can be described with the scene, the objects, the light-sources and the camera. For each object in the scene, the shape, location, orientation, motion and optical properties of the object’s surface can be described by graphics data, called scene descriptors. The scene descriptors can be extracted through computer vision techniques that correspond to the image encoder. The computer graphic algorithm is used to render the image from the graphics data, which forms the image decoder. Such schemes are illustrated in Fig. 1.

Since only scene descriptors are to be transmitted compression into very low bitrate (around several hundred bits per second) can be achieved. The major problem with model-based coding is that this type of video compression couldn’t produce natural video since synthetic images would presumably differ from the real images. To render a more natural-looking image one has to either do a lot of manual work or employ expensive tracking equipment and talent, or to integrate waveform based coding, which will lead to a significant increase in bitrate.

To create natural videos it seems that we have to use and manipulate real videos. To transmit real videos at very low bitrate highly efficient compression techniques have to be used. In literature different technical solutions have been suggested, ranging from the scheme based on match pursuit [1], hybrid H.264 with face tracking [2, 3], PCA based [4, 5], active appearance model based solutions [6] etc. Since all of them have to handle real-videos, the bitrate is targeted to be around 10-20 kbits/s, which is normally called very low bitrate video coding.

2 Ultra Low Bitrate Video Communication System

In this paper we introduce a ultra low bitrate video communication system, which has the following three important features

– ultra low bitrate
– natural-looking video
– a practical communication system

The systematic scheme is shown in Fig. 2. It works as follows: the decoder maintains a personal video database called personal mimic gallery, which stores video of personal facial mimic. The video is individual for a user. These video frames in the gallery should in theory cover all facial expressions for a particular user. When the user is using the video communication system the face recognition module running at the encoder will check which video frame stored in the gallery that matches the input frame best. The index information is transmitted to the frame selection module in the decoder and the module will decide which video frame should be displayed.

The whole coding process can be specified as follows:
Suppose the gallery stores $N$ video frames from facial mimic videos: $J_i(x)$ where $i=1,2,...,N$ and $x = (x, y)^T$ are the pixel coordinates. Here each frame is viewed as an image of one person. The goal of the coding is to select a frame $J_k(x)$ from the gallery to represent the input frame $I_i(x)$ as accurately as possible.

Coding the input frame $I_i(x)$ is then a process of recognizing individual "faces" by performing the maximization of the cost function in equation 1

$$k = \arg \max_k \log p(I_i(x)|\lambda_k)$$ (1)

where $\lambda_k$ is the personal parameter associated to the user $k$.

Decoding an image is a simple pickup-table operation

$$\hat{I}_i(x) = J_k(x)$$ (2)

where $\hat{I}_i(x)$ is the reconstructed frame at the decoder.

Our system satisfies the three requirements above:

- A personal mimic gallery may contain several hundred video frames, however, they are highly correlated. Very few bits are required to index the stored frames. From our experiments we find on average that 4 bits are sufficient to specify a particular mimic. This makes it possible to transmit face video at 100 bits/s with a frame rate of 25 frames/s [7].
- Personal mimic videos are pre-installed and are not required to be transmitted in real-time, therefore, high quality video format can be used. Since the generated video frames are copies from the gallery, this will directly lead to very nature and high-quality video displayed at the decoder.
- The index is the output from the face recognition module. It is not necessary for the system to require a perfect face recognition module. This makes it possible for us to use existing, or commercially available automatic face recognition techniques. It is a fully automatic system since there is no any manual interaction with the system.

To our best knowledge there is no report on such ultra low bitrate video communication systems which satisfy these three requirements. The most related works include video rewrite system [8], voice puppetry [9], emotion synthesis [10], and manifold based facial expression synthesis [11]. However, this scheme is different in many aspects, particular in the principle.

3 Personal Facial Expression Space

Human facial emotion is important for the verbal communication between humans. Paul Ekman has stated that facial emotions can be modelled by six basic emotions [12]. All possible facial emotions can be represented by blending these six basic emotions in different ways.

In our system we create personal facial expression databases for individuals. Such a database stores a video sequence showing one person when he/she is displaying the six basic emotions.

To have a compact database, that is, with less number of video frames, a wearable camera is positioned in front of the face and a video sequence of a human face displaying the six basic emotions is recorded. The used setup ensures almost no global motion between the frames. The information conveyed in such a video sequence is only the local motion, that is, the facial mimic.

All the video frames $J_i(x)$ are stored in the database, here it is called personal mimic gallery. Unlike traditional facial expression databases where data can be in any order, the video frames in our databases are ordered according to the video sequences, which is very important when the video frames are organized with the manifold tool. Further discussion can be found in section 6.1.

A typical personal mimic gallery consists of several hundred frames. A natural question is how to store so many frames in an economic way. In our galleries all frames are coming from a front-view camera fixed to the head. Therefore, there is a strong correlation between
the frames. This can be seen from the distribution of the frames in a compact subspace as shown in Fig. 3. Therefore it can be represented very economically. In [7] we have previously shown that compression of facial video sequences can efficiently be performed with the use of principal component analysis. Results show that 10 to 15 frames are enough to represent the personal facial expression space (see Tab. 1).

Table 1. Average PSNR values for all facial mimic video sequences.

<table>
<thead>
<tr>
<th>Number of Eigen images</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR [dB]</td>
<td>34.56</td>
<td>36.82</td>
<td>38.13</td>
<td>39.07</td>
<td>39.75</td>
</tr>
</tbody>
</table>

Since spatial information is taken into account, the spatial misalignment, rotation and deformation can be well modelled.

4 Index from Face Recognition

One of the key ideas behind our video coding system is to treat the encoding process as a face recognition task (see Fig. 4). Although all video frames in a personal mimic gallery correspond to the same person, we view the frames as individual “persons”. Each person is labelled with an index $i$. For the input frame $I_i(x)$ from the new video, we assume that it is an image of the person who is already stored in the gallery. With the face recognition module, the person $k$ is identified since the frame $J_k(x)$ is the most similar to frame $I_i(x)$. The index $k$ is to be sent to the decoder.

In our system we simply employed the HMM based approach developed in our previous work [13]. In our HMM system, each “person” is assigned with a HMM, $\lambda_k$. To train the HMMs each video frame is decomposed into overlapping blocks by horizontally scanning from left to right and from top to bottom. Each extracted block is normalized independently so the system can handle correctly complex illumination changes. The normalized feature blocks are arranged column-wise to form the observation vectors. At the training step, feature blocks together with their positions are extracted and stored in a codebook. The observation sequence of the blocks is generated line by line in both vertical and horizontal direction. The spatial relation of the blocks in both vertical and horizontal directions are captured by HMMs. At the recognition step, feature blocks of the input frame $I_i(x)$ are extracted as above described. The scanning order of blocks is kept the same, however, the observation sequences are formed based on the matching of blocks of the query image to the blocks stored in the codebook under some constrains about their positions. The observation sequences, then, are put into the HMM classifier. The index $k$ corresponding to maximal likelihood function $\log p(I_i(x)|\lambda_k)$ will be selected.

5 Video Reordering

The new video is reconstructed by selecting frames from the gallery according to the index from the face recognition module. The problem is that there is no smooth transition between the frames and a jarring jump occurs often when the video is displayed. It is understandable since the frames are evaluated individually. The sequence of $k$ is obtained by $k \leftarrow \max \log p(I_i(x)|\lambda_k)$. There exists no sequential order at all! The consequence is that none will view the jerky video as a natural one!

To generate a natural-looking video we have to concern how smooth the transition between frames is. Our solution is to include a smoothness term into the cost function. We will modify the object function used in the optimization to

$$k \equiv \arg \max_k [\log p(I_i(x)|\lambda_k) + \alpha S(I_{i-1}(x), J_k(x))]$$

where $\alpha$ is a weight parameter and $S$ is a similarity measure.

The problem is now how to find a suitable similarity measure. Obviously, the L-norm based distance measures are not applicable here since they are all sensitive to minor spatial displacement and intensity changes.

In our system we employ a nonlinear dimension reduction tool, locally linear embedding (LLE) [11], to handle the difficulty of measuring the similarity between two frames. LLE is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs. LLE is able to learn the global structure of nonlinear manifolds. We are interested in embedding the facial deformations of a person in a very low dimensional space, which reflects the intrinsic structure of facial expressions.
The approach we used is to project two frames into locally linear embedding space and use the distance there to measure the similarity, that is

\[
S(\hat{I}_{i-1}(x), J_k(x)) = D(L(I_{i-1}(x)), L(J_k(x)))
\]

(4)

where \(L()\) is the LLE projection operation and \(D()\) is a distance measure on the locally linear embedding space.

With forcing smoothing transition, the jerk effect will be greatly reduced. Occasionally, there is an abrupt jumping from one frame to another frame. To avoid any jump, we can cross dissolve them. The side-effect is that this will introduce ghostlike blurry images in the transition. A better solution is to use image morph technology [14] to perform a smooth transform. In this way a natural-looking reconstruction of video can be expected.

6 Preliminary Results

6.1 Personal Facial Expression Gallery

For preliminary testing, we recorded two video sequences of one subject who was asked to perform the six basic facial expressions multiple times at two dates (a week interval). Each video sequence showed the person when he was displaying the six basic emotions proposed by Ekman. After each emotion the subject returned to a neutral state. The video sequences were approximately 30 seconds long, an emotion was displayed for 2-4 seconds and a new emotion was displayed approximately every 5 seconds. The framerate for both the video sequences was 15 fps and the resolution was 240x176 pixels.

To train our face recognition module, we have compacted the video stored in the gallery. In the video sequence, there were many short periods that showed no spatial motion changes, like those moments the subject kept the face appearance at the neutral state or at the peak of expressions. The frames that did not induce optical flows were removed from the sequence to reduce the computation cost for the recognition module. In the end the number of stored video frames is reduced from 400 into 200. The compact video sequence was fed into the recognition module to train a number of HMMs. Each HMM \(\lambda_k\) was trained with one video frame.

6.2 Frame Indexed by Face Recognition

Before fed into the face recognition module the input video frames are first processed to locate the face. With the state-of-the-art face detection techniques [15, 16], the system can locate a rectangular box containing the face. For real-time detection, the detector scans only in the neighborhood of the face location in the previous frame. After the face position is found, we approximately locate eyes and mouth area based on their relative ratio in the face box. Refined eye areas are located by a detector, which is in principle similar to the face detector [15]. The head, eye and mouth areas are preprocessed and then put into the face recognition module, which will find a video frame in the gallery that best matches the input frame.

To evaluate how good the result is, we performed a subjective evaluation of the quality of the reconstructed sequence. In the experiment, the subject was shown two frames, one from the original sequence and the other from the reconstructed sequence. The subject was asked to give his judgment on the match of these two frames. Matching quality was graded into 3 levels: ‘unacceptable, acceptable and good’. To avoid the effect of motion on this evaluation about the expression matching, the frames were taken from the video sequence in a random order. This result is shown in Tab. 2.

Despite the high performance that the HMM classifier can achieve, it is still very difficult to guarantee correctly and smoothly the appearance changes of a feature in the progress of deformation. For example, the motion of the mouth in the reconstructed video may be represented unnaturally due to the existence of couple frames with subtle difference in shape, however, be put and displayed in the inverse order to the onward showing expression.

6.3 Analysis in the LLE Space

![Fig. 5. Personal mimics face images mapped into the embedding space described by the first two coordinate of LLE](image-url)
To study the dynamic behavior of new generated video, we embed the video frames in the gallery into a two dimensional space. Fig. 5 shows the result of projecting gallery video frames into a two dimensional space using LLE embedding, where one can clearly see that there are three emotion classes. The joint point corresponds to the neutral face. Fig. 6 shows the sequence order of the sample frames in the embedding space. We can see that LLE represents the emotion expression in temporal very well.

Fig. 6. Sample video mapped into the embedding space described by the first two coordinate of LLE

When we display the reconstructed video sequence based on the indexes from the face recognition module in the LLE space in Fig. 7(a), one can clearly see that there are a lot of frame jumping across two emotion branches. This well explains why the result video looks so jerky! To generate a natural-looking video, we have to ensure the move from one frame to next frame as smooth as possible in the LLE space.

6.4 Smoothing in the LLE Space

To ensure that the transition from one frame to the next frame in the reconstructed videos is smooth, we apply dynamic programming (DP) to optimize the sequence of indices in the LLE space. When coming to the similarity measure, instead of using Euclidean distance directly to represent cost to move from one point to other point in the embedding space, we use the ordinal number computed based on the Euclidean distance between points for scoring. Fig. 8 shows three sequences of a mouth smiling in the video to be encoded and two coded versions. In the figure, the order of frames is from left to right then top to bottom. We can notice that the first frame on the second row of the reconstructed video without optimization in Fig. 8(b) looks quite similar to its counterpart in the video to be encoded in Fig. 8(a), but it does not match its neighbor frames in the reconstructed sequence. While in the reconstructed video with LLE optimization in Fig. 8(c), the selected frame does not match very well visually, however, it makes the whole sequence looks more natural. Thus with this optimization function, the number of jarring and jerky effects are reduced considerably. This is shown in Fig. 7(b)

We also performed a subjective evaluation of the quality of the reconstructed video sequence. From the result shown in Tab. 2, one can see that the number of "good" frames is slightly increased. However, when we playback the reconstructed video looks very natural. This is also proved from the distribution in the LLE space as shown in Fig. 7. Much fewer frames crosses show up.

7 Discussion

The preliminary results show very promising results. Of course to prove the concept we have a long way to go. There are a lot of open questions we have to answer

- how to measure the naturalness of a video?
- how to generate a generalized facial mimic space?
- how to ensure a smooth transition between frames?
- how to do optimization for real-time applications?
References


Fig. 8. Frames of a mouth smiling in (a) the video to be encoded (b) reconstructed video without optimization and (c) reconstructed video with frame indexing optimization