Face Distortion Recovery Based on Online Learning Database for Conversational Video

Xi Wang, *Member, IEEE*, Li Su, *Member, IEEE*, Honggang Qi, *Member, IEEE*, Qingming Huang, *Senior Member, IEEE*, and Guorong Li, *Member, IEEE*

***Abstract*—With the real-time requirement for video conversation, the coding system needs to adopt low delay and low complexity strategy to encode conversational videos, which may result in a significant decline of the video quality under the constrained bandwidth of network. In conversational videos, the face region attracts most human attentions. Therefore, recovering the distortion of face region will effectively improve the visual quality of conversational video. Actually, the participants in a conversation are usually unchanged in a relative long period, and similar facial expressions of the participants would be often repetitive. However, conventional video coding methods just consider the correlation of several neighboring frames while the long-range correlation of similar face regions in the whole conversational video has not been fully used. In this paper, we propose a face distortion recovery system to improve the visual quality of decoded conversational video by online learning an own face feature database for each user. First, at the sender side, the face feature database is established and online updated to include different facial expressions of the person. Then, at the receiver side, the low quality face regions in decoded video are recovered with the face patches in the database. Experimental results show that, under low bits rates the proposed method achieves average 5.22 dB gain with small burden to update the database.**

***Index Terms*—Conversational video coding, face alignment, face distortion recovery, online learning.**

I. Introduction

In THE recent years, with the rapid development of internet and mobile communication technology, video conversations have been widely applied in people's life. Meanwhile, with the enhancement of the accuracy of displaying devices, the requirements for higher video quality have been increased. However, with the real-time requirement of video conversation and limited computing ability of mobile devices, the high complexity coding system (such as the latest standard HEVC [1]) is difficult to be utilized. And the conventional coding system (such as H.261 [2]) may export lower quality conversational videos under the constrained bandwidth of wireless networks. Therefore, enhancing the visual quality of conversational videos is a meaningful and challenging research work.

Currently, most conversational video coding strategies are based on the Region-Of-Interest (ROI) detection methods [3]–[8], which take the human faces as areas of interest coded in higher quality. These strategies are usually composed of two parts: the encoder side processing [4], [5] and the decoder side processing [6]–[8]. At the encoder side, the quantization parameter and the Rate Distortion Optimization (RDO) Lagrange multiplier [9] are adapted to make the face region corresponds to higher distortion sensitivity. At the decoder side, the face region is taken as the most prominent area to perform error recovery or error resilient. Wang *et al.* [10] proposed a very low frame-rate video coding strategy for face-to-face teleconference. They reduce the frame rate and only encode the face regions of some selected frames with higher quality. However, these above methods do not fully take account of the long-range correlation of the conversational videos.

In the conversational videos, the human face is generally subjected to limited changes no matter the background changes or not. A video chat user, in most cases, talks with the same people. e.g., his family and good friends. Assuming there are some conversational videos and the people in these videos is the same person, we can infer that the face regions in different videos frames are highly correlated to each other. The existing video coding methods just utilize the correlation of a few adjacent frames without considering the global correlation of these videos. In this paper, exploiting the characteristics of conversational video, we propose to apply external data for the construction of higher quality face regions. We establish an optimal face feature database and online update the database for face distortion recovery. The proposed algorithm can be easily applied to the existing video coding systems without the restrictions of the coding standards. The main contributions of our work are summarized as follows:

1) We propose a novel framework for recovering the face distortion of conversational video with a personal face feature database.
2) We improve a face alignment algorithm to make it suitable for frame by frame face aligning requirements in our recovery system.

Manuscript received March 03, 2014; revised June 29, 2014; accepted August 26, 2014. Date of publication September 05, 2014; date of current version November 13, 2014. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, the National Natural Science Foundation of China under Grants 61025011, 61332016, 61303153, 61472388, and 61472389, and the China Postdoctoral Science Foundation under Grant 2014T70111. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Enrico Magli.

X. Wang and Q. Huang are with the Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: xi.wang@vipl.ict.ac.cn; qmhuang@ucas.ac.cn).

L. Su, H. Qi, and G. Li are with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: suli@ucas.ac.cn; hgqi@ucas.ac.cn; liguorong@ucas.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2355134

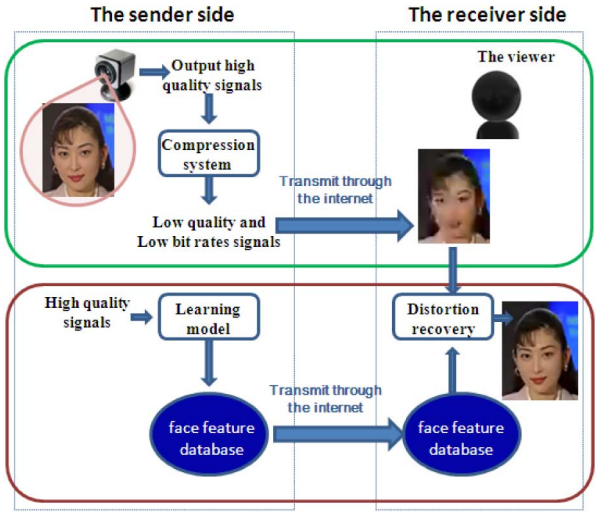


Fig. 1. Framework of our proposed face distortion recovery procedure for the video conversation system.

3) We develop an online learning face feature database to capture long-range correlation in the conversational videos of the same person at the sender side.
4) We establish a recovery model to improve the reconstructed video quality at the receiver side by using the face feature database.

The rest of this paper is organized as follows. An overview of our proposed system is given in Section II. In Section III, we present the face alignment process and online learning procedure of face feature database at the encoder side. Then, the face distortion recovery process is described in Section IV. The time requirements and bounds of recovery effects of our algorithm are analyzed in Section V. In Section VI, the experimental results are reported. Finally, the paper is concluded in Section VII.

II. The Face Distortion Recovery System

The proposed face distortion recovery system is shown in Fig. 1. The content in the green rectangle represents the traditional video conversation system. The part in the red rectangle is our proposed face distortion recovery system. First, a personal face feature database of the speaker is established and trained at the encoder side. Second, the database will be sent to the decoder and utilized to improve the quality of decoded conversational video. Third, the face database of the speaker will be gradually updated on both sides until it satisfies the given conditions. It should be noted that the size of the database is much smaller than the encoded conversational videos.

The proposed video coding algorithm can be divided into two parts: (1) The database training (2) The distortion recovery. The details of each part will be described in the following sections.

III. Face Database Online Learning

At the sender side, the high quality original videos can be accessed. The personal face database is established and updated based on the original videos. In our algorithm, all video sequences are transformed to the YUV format and we only focus on the recovery of luminance (Y) component. Thus, the following processes are only operated on luminance component. The procedure of database online learning is shown in Fig. 2. The red rectangles represent the input frames. Each consecutive n frames are collected for building a subdatabase. First, face alignment is applied to the input frames. Then, each n aligned face regions are divided into non-overlapping $n_0 \times n_0$ blocks. These blocks will be clustered into multiple groups by the classifier. The rest blocks that do not satisfy the given conditions will be put into the rest block collection, and will be re-clustered for creating a new subdatabase. The blocks in the rest groups which did not satisfy the given conditions will be put into the rest block collection, and need to be done blocks re-clustering for creating a new subdatabase. Finally, the face database is updated by fusing the subdatabases. This algorithm is divided into three parts to account for A) Face alignment method, B) Sub-database generation, C) Database online updating. The details are shown in the follows.

A. Face Alignment

Face alignment plays a crucial role in our system because face matching is very sensitive to the pose and angle. Yigang Peng *et al.* [12] proposed an effective Robust Alignment by Sparse and low-rank Decomposition for Linearly correlated images (RASL). But RASL is a kind of batch face alignment method which can not meet the requirements of our system. At the sender side, in order to obtain an optimized face feature database, the face alignment method should align all the frames but not just align a few adjacent frames. At the receiver side, with the real-time display requirement, the current frame need to be fast aligned. Therefore, we design a fast frame by frame face alignment method based on RASL [12] to satisfy our own requirement.

The basic idea of RASL is that for several given n grayscale face images of a person, if they are well-aligned, they should be linearly correlated. More precisely, suppose there are n well-aligned grayscale images $I_1^o, \ldots, I_n^o \in \mathbb{R}^{w \times h}$ of a same person. Let $vec() : \mathbb{R}^{w \times h} \rightarrow \mathbb{R}^m$ denotes the operator that selects a m-pixel region of interest from an image, and the region is stacked to a m-dimension column vector. Then, the matrix

$$A \doteq [vec(I_1^o)| \cdots |vec(I_n^o)] \in \mathbb{R}^{m \times n} \tag{1}$$

is approximately low-rank. Suppose that $I_1, I_2, \ldots, I_n$ represent n input images of a same person's face but misaligned to each other. Then, the well-aligned images can be obtained by doing a set of domain transformations on these misaligned images. It can be written as

$$D \circ \tau \doteq [vec(I_1^o)| \cdots |vec(I_n^o)] \in \mathbb{R}^{m \times n} \tag{2}$$

where $D = [vec(I_1)| \cdots |vec(I_n)]$, and τ represents the set of n domain transformations $\tau_1, \tau_2, \ldots, \tau_n$. $I_j^o = I_j \circ \tau_j$ for $j = 1, 2, \ldots, n$. A is the well-aligned matrix at pixel level. Therefore, the batch image alignment problem can be reduced to the following optimization problem:

$$\min_{A,\tau} rank(A) \quad s.t. \quad D \circ \tau = A \tag{3}$$

In addition to occlusions, real images typically contain some noise of small magnitude in each pixel. Let e_j represents the

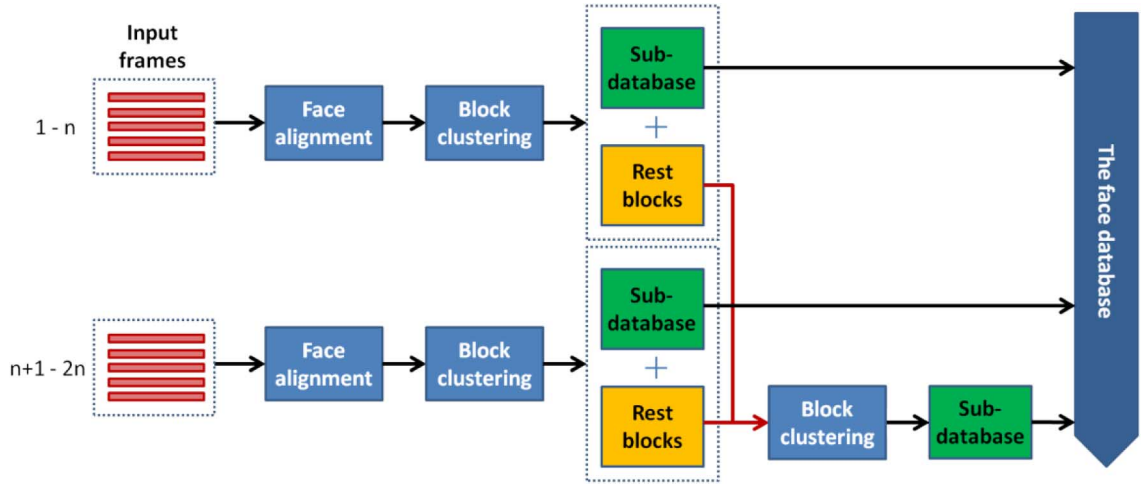


Fig. 2. Flowchart for online learning face feature database in the sender side.

Fig. 3. RASL alignment elements. (a) original images with different poses and expressions. The transformed images in (b) can be decomposed as the sum of images from a low-rank approximation in (c) and sparse large errors in (d). The original images are taken from the Labeled Faces in the Wild (LFW) [13] dataset (a) Original D (b) Aligned $D \circ \tau$ (c) Low-rank A (d) Errors E.

error corresponding to image I_j such that the images $\{I_j \circ \tau_j - e_j\}_{j=1}^{n}$ are well-aligned to each other, and free of any corruptions or occlusions. Therefore, the formulation (3) can be modified as follows:

$$\min_{A,\tau,E} \quad rank(A) \quad s.t. \quad D \circ \tau = A + E, \|E\|_0 \leq \varepsilon \tag{4}$$

where $E = [vec(e_1)|\cdots|vec(e_n)]$. The above is the main idea to solve a batch image alignment problem by RASL. A typical example of RASL is shown in Fig. 3.

Based on the RASL method, the proposed frame by frame face alignment method is that, suppose there is already a well-aligned matrix A_o, for a new input misaligned image I' which should be aligned with A_o, the Eq. (4) can be modified as:

$$\min_{A,\tau,E} rank([A \quad A_o]) \quad s.t. \quad D \circ \tau = A + E, \|E\|_0 \leq \varepsilon' \tag{5}$$

where $D = [vec(I')]$, all of D, A, E are $\mathbb{R}^m$ vectors. Then, the inner loop optimal solution in [12] is modified to

$$\begin{cases} A' = D \circ \tau + J\Delta\tau_k \epsilon\epsilon^T + \frac{1}{\mu_k} Y_k - E_k, \\ (U, \Sigma, V) = svd([A' \quad A_o]), \\ A_{k+1} = first(US_{\frac{1}{\mu_k}}[\Sigma]V^T), \\ E_{k+1} = S_{\frac{1}{\mu_k}}[D \circ \tau + J_\Delta \tau_k \epsilon\epsilon^T + \frac{1}{\mu_k} Y_k - A_{k+1}], \\ \Delta\tau_{k+1} = J^\dagger (A_{k+1} + E_{k+1} - D \circ \tau - \frac{1}{\mu_k} Y_k)\epsilon\epsilon^T \end{cases} \tag{6}$$

where $Y \in \mathbb{R}^m$ is a Lagrange multiplier matrix. J is the Jacobian of D with respect to the transformation parameters τ. $first(\cdot)$ denotes the first column of a matrix. $svd(\cdot)$ denotes the Singular Value Decomposition (SVD) operator, and $J^\dagger$ denotes the Moore-Penrose pseudoinverse of J. More details are described in [12]. In the procedure, the program spends most of the time doing the SVD operation, therefore, an incremental SVD method [14] is exploited to speed up the operation. Assuming that the SVD on the aligned $m \times n$ matrix A_o has been already obtained $A_o = U\Sigma V^T$. Then, for a new m-dimensional B, the goal is to efficiently compute the SVD of the concatenation of B and A_o: $[BA_o] = U'\Sigma' V'^T$. Letting $\tilde{B}$ be the component of B orthogonal to U, the concatenation of B and A_o in a partitioned form can be expressed as follows:

$$[B \quad A_o] = [\tilde{B} \quad U] \begin{bmatrix} \tilde{B}^T B & 0 \\ U^T B & \Sigma \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V^T \end{bmatrix} \tag{7}$$

Let $R = \begin{bmatrix} \tilde{B}^T B & 0 \\ U^T B & \Sigma \end{bmatrix}$, compute the SVD of R, $R = \tilde{U}\tilde{\Sigma}\tilde{V}^T$. Now the SVD of $[B \quad A_o]$ can be expressed as

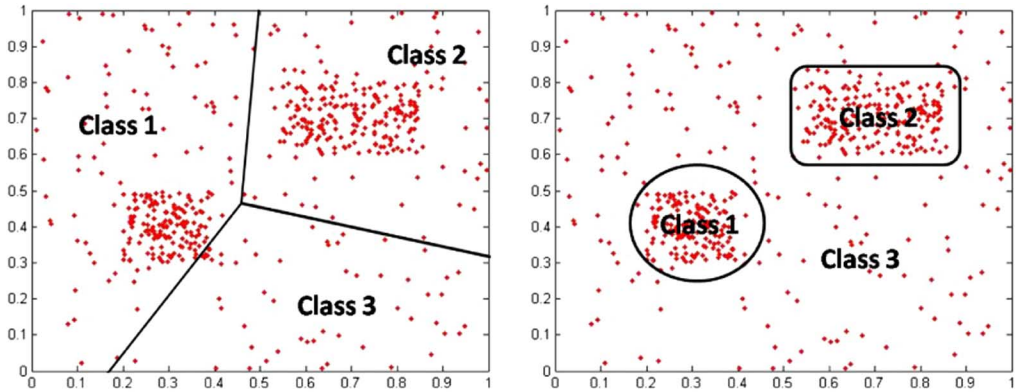
$$[B \quad A_o] = ([\tilde{B} \quad U]\tilde{U})\tilde{\Sigma} \left(\tilde{V}^T \begin{bmatrix} I & 0 \\ 0 & V^T \end{bmatrix} \right) \tag{8}$$

Finally, $U' = [\tilde{B} \quad U]\tilde{U}$, $\Sigma' = \tilde{\Sigma}$ and $V' = \tilde{V}^T \begin{bmatrix} I & 0 \\ 0 & V^T \end{bmatrix}$.

After that, the algorithm of frame by frame face alignment is terminated.

B. Sub-Database Generation

As shown in Fig. 2, consecutive n original frames are collected to generate a subdatabase. For a group of input n frames, after face alignment, we can obtain n well aligned face regions. These face regions are divided into non-overlapping blocks with size $n_0 \times n_0$, $B = \{b_1, b_2, \cdots, b_m\}$. The coordinates of the center of these blocks are recorded as $C = \{c_1, c_2, \cdots, c_m\}, c_i = (x, y)$. Assuming that the size of face region is $w \times h$, then, the number of blocks is $m = \frac{w \times h}{n_0^2} \times n$. First, block collection B is divided into multiple clusters by using the Shrinking and Expansion Algorithm



(a) (b)

Fig. 4. Clustering on data with uniform distributed background points. (a) clustering result of k-means (b) clustering result of SEA.

(SEA) [16], which is a unsupervised clustering method. The blocks in the same cluster should be similar to one another. In our previous work [11], the K-means method [15] is used as the classifier which needs to pre-specify the number of clusters. In the SEA method, the number of clusters does not need to be explicitly known, because it obtains a cluster by detecting a dense subgraph in the graph.

The schematic diagram of classification results of SEA is shown in Fig. 4(b). The subgraph with large density can be regarded as a cluster. Comparing with the classification results of K-means method as shown in Fig. 4(a), the blocks in a dense subgraph are much closer to each other. After classifying the blocks B using SEA, the clusters are further processed by using the coordinate information $C = \{c_1, c_2, \cdots, c_m\}$. A cluster is separated into several groups to make the blocks in a new group with the same coordinate.

After classification, the blocks belonging to a group have both similar structure and the same coordinates. For a group $g = \{b_1, b_2, \cdots, b_{n_1}\}$, the within class variance of group g is computed as

$$\sigma^2(g) = \frac{1}{n_1}\sum_{i=1}^{n_1}\frac{1}{n_0 \times n_0}(\|b_i - \overline{b}\|_2)^2 \tag{9}$$

where $\overline{b}$ is the mean value of blocks in g, $\sigma^2(g)$ measures the dispersion of the group g. We assume the following condition

$$\begin{cases} \sigma^2(g) & < \delta, \\ n_1 & > N. \end{cases} \tag{10}$$

where δ and N are given thresholds. Setting a smaller δ requires that the blocks in g tend to be closer to the mean $\overline{b}$. In our paper, we set $\delta = 10$. A larger N corresponds to a bigger size of g, N is set to be $\frac{n}{3}$, where n is the number of the input face regions.

If a group satisfies the condition (10), the mean value $\overline{b}$ and the coordinate $\overline{c}$ of the blocks in this group will be recorded as a unit $u = (\overline{b}, \overline{c})$. A subdatabase is composed by many units. If there are k groups satisfying the condition (10), the subdatabase U will have k units that are created by these groups, $U = \{u_1, u_2, \cdots, u_k\}$. The other groups that do not satisfy the condition (10) will be removed and blocks in these groups will be put into the rest block collection R. At the beginning, rest block collection R is empty. While the size of R is bigger than a pre-set threshold (we set it equal to m, the number of blocks of B), the blocks in R will be re-clustered to generate a new subdatabase and R will be set to empty again.

C. Database Online Updating

As video conversation goes on, new facial expressions of the speaker may appear. Therefore, the main database that is sent to the decoder side should be continually online updated.

Let dB denotes the database which also consists of units $dB = \{u_1^*, u_2^*, \cdots, u_i^*, \cdots\}$. At the beginning, the database dB is empty. During the online training processing, subdatabases will be used to update the database dB. The operation of using a new subdatabase U' to update dB can be described as $dB = fusion(dB, U')$. The $fusion(\cdot)$ procedure is presented in Algorithm 1. For a unit $u_i = (b_i, c_i)$ in the subdatabase U', if there exist a unit $u^* = (b^*, c^*)$ in dB that satisfies the following condition:

$$\|b^* - b_i\|_2 \leq \gamma, c^* = c_i \tag{11}$$

where γ is a given threshold, the database dB will not be changed; Else, u_i will be added into dB, and the length of dB will increase by one.

Algorithm 1 One Sub-database fuse to the Database

Input: Sub-database U', Database dB

Output: Database dB

$n = length(U'), m = length(dB)$
for $i = 1 : n$ **do**
  if $\exists u^* \in dB$ satisfy $(\|b^* - b_i\|_2 \leq \gamma \& c^* = c_i)$ **then**
    Continue
  else
    $u_{m+1}^* = u_i$
    $m = m + 1;$
  end if
end for

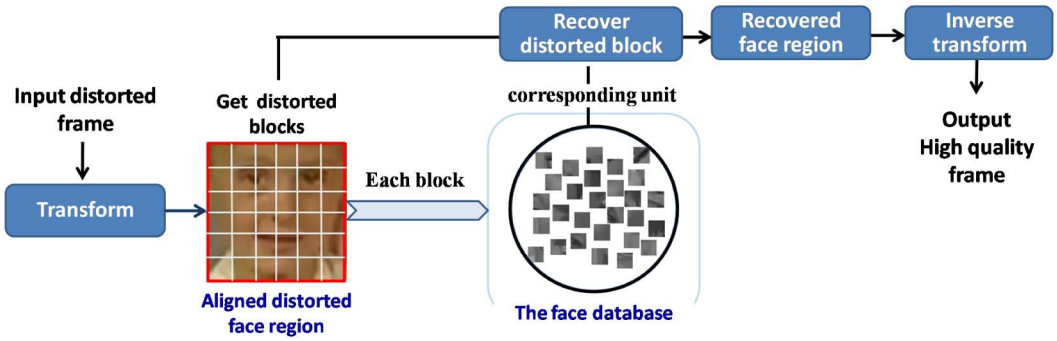


Fig. 5. Face distortion recovery at the decoder side.

In order to improve the speed, a link table is created to query units in the database by using the coordinates information. For the database $dB = \{u_1^*, u_2^*, \cdots, u_i^*\}$, the units in database are queried by the coordinate c. The size of aligned face regions is $w \times h$ and the divided block size is $n_0 \times n_0$. Therefore, the coordinates of blocks belong to $\frac{w \times h}{n_0^2}$ candidates. We create $\frac{w \times h}{n_0^2}$ lists to store the units. The jth unit $u_j^* = (b_j, c_j)$, $c_j = (x_j, y_j)$ in the database is saved in the $(\frac{w}{n_0} \times (\frac{y_j}{n_0} - 1) + \frac{x_j}{n_0})th$ list.

The size of a block is $n_0 \times n_0$. Thus, the time complexity of computing blocks distance $\|b^* - b_i\|_2$ is $O(n_0^2)$. Assuming there are $m1$ units in a subdatabase and $m2$ units in the database, the maximal cost for updating by using this subdatabase is $O(m1 \times m2 \times n_0^2)$. In fact, the units in the subdatabase only need to compare with the units which have the same coordinate in the database. Therefore, the time complexity for once updating database at the sender side is much smaller than $O(m1 \times m2 \times n_0^2)$. At the receiver side, new units are received. We only need to put the new $m3$ units into the corresponding list of the database. Therefore, the cost for this updating procedure is $O(m3)$.

IV. Face Distortion Recovery

At the receiver side, the face database and the encoded bitstreams are received. Before displaying, the distorted frames are recovered to make its visual quality higher by the proposed face distortion recovery algorithm. The procedure of the algorithm is shown in Fig. 5.

First, the aligned distorted face region is obtained by applying the domain transformation on the distorted frame. The domain transform parameter τ is obtained from the sender side by aligning the corresponding original frame. Second, the distorted face region is divided into non-overlapping blocks with size $n_0 \times n_0$. Each distorted block has a corresponding original block at sender side. For the distorted block, we recover it by using the unit that is most similar to the original block. More precisely, suppose there is a decoded block b_0^d with coordinate $c_0 = (x_0, y_0)$, and the corresponding original block at the sender side is b_0^o. At the sender side, we search the most similar unit of b_0^o in the face database by the following function:

$$u_m^*(b_m^*, c_m^*) = \min_{u_i^* \in dB} \|b_i^* - b_0^o\|_2^2 \quad constrain \quad \{c_i^* = c_0\} \tag{12}$$

where u_m^* is the searching result in the database. If u_m^* satisfies

$$\|b_i^* - b_0^o\|_2^2 \leq \xi \tag{13}$$

where ξ is a given threshold, the index of u_m^* in the face database will be transmitted to the decoder side, and the distorted block b_0^d will be recovered by b_m^*,

$$b_0^d = b_m^* \tag{14}$$

else the distorted block b_0^d will not be changed.

Finally, after finishing the block recovery, the inverse transforms are performed on the face region before displaying.

V. Algorithm Analysis

In this section, the algorithm's time requirement and the bounds of recovery effects are analyzed.

Let $\{U_1, U_2, \cdots, U_n\}$ be a set of subdatabases which corresponds to a set of status of the database $\{db_1, db_2, \cdots, db_n\}$. db_i is obtained by using U_i to update db_{i-1}. The database is independent of the encoder. Assuming that I is one of the original frames which are used to establish subdatabase U_i, and the corresponding distorted frame is I^d. Then, the statue of the database which is generated before db_i can be used to recovery I^d. Therefore, it has no real-time requirement for the subdatabase generation procedure. But the frame I^d needs to be real-time displayed, therefore, the face alignment and face distortion recovery algorithm must be low delay.

Each unit in the database is created as a cluster center of a group of blocks. The distance between a block and the cluster center reflects the quality of recovered images. As shown in Fig. 6, there have two groups of blocks within two circles. The two blue squares in the center of the circles denote the cluster center of the groups. Assuming that the cluster center of the left group corresponds to a unit u_a^* in the database, and the cluster center of the right group corresponds to a unit u_b in a subdatabase. According to the database updating procedure in Section III-C, if the distance between u_a^* and u_b ($distance1$ in Fig. 6) is smaller than the given threshold γ (described in Eq. (11)), u_b is similar to u_a^*, and for the block b_i in the right group, the corresponding distorted block in the receiver side should be recovered by u_a^*, the distance between b_i and u_a^* ($distance3$ in Fig. 6) is smaller than the sum of $distance1$ and $distance2$. $distance2$ is related to the within class variance σ^2 (described in Eq. (9)) of the right group, which means

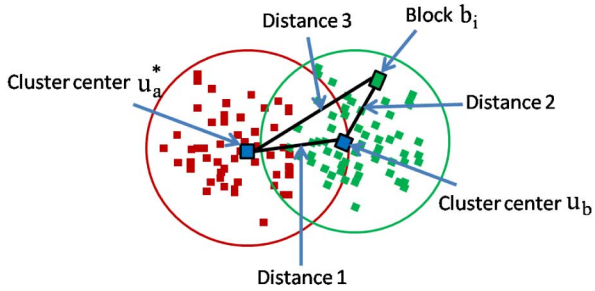


Fig. 6. Two groups of blocks. The center of left group corresponding a unit u_a^* in the face database, and the center of right group corresponding a unit u_b in a subdatabase.

that the within class variance σ^2 and the updating threshold γ will decide the quality of recovered image. A smaller σ^2 and γ will bring a higher quality of recovered images, meanwhile the number of units in the database will be larger.

VI. Experimental Verification

The proposed method is validated in a PC with 2 core 3.00 GHz CPU and 4 GB memory. All video sequences are in YCbCr 4:2:0 format and encoded by using the HEVC reference software HM12.0 [17] with $QP = 40$. The verification includes two parts: 1) The accuracy and time consuming of the proposed frame by frame face alignment method will be discussed. 2) 8 segments of standard YUV test sequences in CIF format are used to validate the performance of our face recovery method. Furthermore, since the length of standard test sequence is not long enough, we create our own test sequences from the news report videos. We prepare 8 video sequences for a news reporter, and establish her own face database to verify the performance on multiple long videos for the same person.

A. Frame by Frame Face Alignment Algorithm Validation

In Section III-A, we proposed a frame by frame face alignment method base on the RASL [12] algorithm. Different from the batch face alignment method, our goal is to align a new input face image with an existing well-aligned matrix. In this section, the performance of our method will be shown and compared with the batch face alignment method RASL [12]. The input un-aligned images are obtained from the standard YUV sequences with size 352×288. In order to make width and height be a multiple of 16, the target size of output face region is set to be 160×128. First, we discuss the align effect by using different sizes well-aligned matrices. Assuming there are 20 un-aligned images $\{I_0, \ldots, I_{20}\}$, and 4 well-aligned matrices $\{A_1 \in R^{20480\times 5}, A_2 \in R^{20480\times 10}, A_3 \in R^{20480\times 20}, A_4 \in R^{20480\times 40}\}$ contain $\{5, 10, 20, 40\}$ well-aligned face regions, respectively. The output aligned face regions are $\{vec(I_1^o), \cdots, vec(I_{20}^o)\}$. The experimental results of using different well-aligned matrices are shown in Table I. The effect of alignment is measured by the mean error which is computed as the SAD (Sum of Absolute Difference) between $vec(I_i^o)$ and $vec(\overline{I^o})$

$$MeanError = \overline{\sum_{i\in[1,20]} \|vec(I_i^o) - vec(\overline{I^o})\|} \qquad (15)$$

where $vec(\overline{I^o})$ is the mean value of $\{vec(I_1^o), \cdots, vec(I_{20}^o)\}$. We can observe that the mean error of RASL results are the smallest, because the object function (3) of RASL is directly aligning the total input frames. The 'Time(s)' in Table I presents the total time costs by aligning 20 frames. In our method, with the growing size of existing well-aligned matrix A_i, the cost time increases significantly, but the reduction of mean error gradually slows down. We can observe that the mean errors of using A_3 and A_4 are almost the same, but the cost time of using A_3 are much smaller than using A_4. A_3 contains 20 face regions, therefore, we take 20480×20 as the reference well-aligned matrix size in the following experiments. The experimental results show that the proposed method costs average 2.63 seconds to align 20 frames by using A_3, which is much smaller that the average cost time 30.62 seconds of RASL.

The subjective alignment results are shown in Fig. 7. The results of RASL are (a)(b)(c), and (d)(e)(f) are the results of proposed method by using A_3 as the reference well-aligned matrix. From these results, we can observe that our alignment effect is closed to RASL but the time cost is much smaller than RASL.

B. Database Analysis and Recovery Performance

Experiments on the Standard Test YUV Sequences: In this part, 8 standard YUV sequences are used. The number of frames of each YUV sequence is shown in Table II. Each sequence is divide into several segments with each segment contains 30 frames. As described in Section III-B, each 30 frames are aligned to establish a subdatabase.

First, we discuss the face database and the subdatabases which are trained in the sender side. As described in Section III-C, one subdatabase corresponds to once updating of the database. Therefore, each subdatabase U_i corresponds to a status of database db_i. The sizes of the database and subdatabases of each sequences are shown in Fig. 8. The x-axis is the index of subdatabases, and the y-axis denotes the units number. There are 3 lines in each figure. The blue line at the top describes the units number of subdatabase. The red line at the bottom shows the number of the increasing units of the database. Point (x, y^{red}) in the red line has the relationship $y^{red} = size(db_x) - size(db_{x-1})$. And the green line represents similarity units number between subdatabase U_i and db_{i-1}. The relationship is $y_x^{green} = y_x^{blue} - y_x^{red}$. The green line closer to the blue line means that there are more repetitive units between database and subdatabases. It can be observed that, as the index increases, the red line is getting closer to the x-axis, which means the database is tending towards stability. The accurate units numbers are shown in Table II. The 'Total blocks' represents the number of blocks which are used to generate subdatabases. The face regions with size 160×128 are divided into blocks with size 16×16, which means one face region contains 80 blocks. Therefore, the number of total blocks is computed as number of frames multiplied by 80. 'Database units' represents the number of units in the final database. We can observe that the number of the units in the database is much smaller than the number of original blocks. The average database units number is 0.72% of the original blocks number in the short periods.

TABLE I
THE PERFORMANCE OF THE PROPOSED FACE ALIGNMENT METHOD (COMPARED WITH THE BATCH FACE ALIGNMENT METHOD RASL: YIGANG PENG ET AL. [12], $A_1 \ldots A_4$ REPRESENTS THE LOW-RANK MATRICES WITH DIFFERENT SIZE)

Sequence	Mean error					Time(s)				
	RASL	Proposed				RASL	Proposed			
		use A_1	use A_2	use A_3	use A_4		use A_1	use A_2	use A_3	use A_4
akiyo	8.16	11.74	10.90	9.84	10.86	53.26	0.91	1.49	2.92	9.90
claire	4.50	5.75	5.13	4.65	4.62	20.55	1.18	1.51	3.01	6.18
deadline	2.18	3.27	2.11	2.09	2.25	39.27	1.03	1.90	3.03	6.13
grandma	4.19	4.61	4.42	4.34	4.40	20.44	0.70	1.09	1.98	6.31
missa	4.53	4.92	4.77	4.78	4.67	21.83	0.66	1.01	2.88	4.04
sign irene	6.30	6.47	7.14	8.07	8.62	27.17	0.99	1.04	1.86	3.87
silent	5.03	8.51	6.53	6.15	5.23	20.74	1.23	1.48	2.69	5.89
vtc	3.07	8.82	4.43	4.49	3.81	41.69	1.22	1.47	2.68	7.94
Average	**4.75**	**6.75**	**5.68**	**5.55**	**5.55**	**30.62**	**0.99**	**1.37**	**2.63**	**5.49**

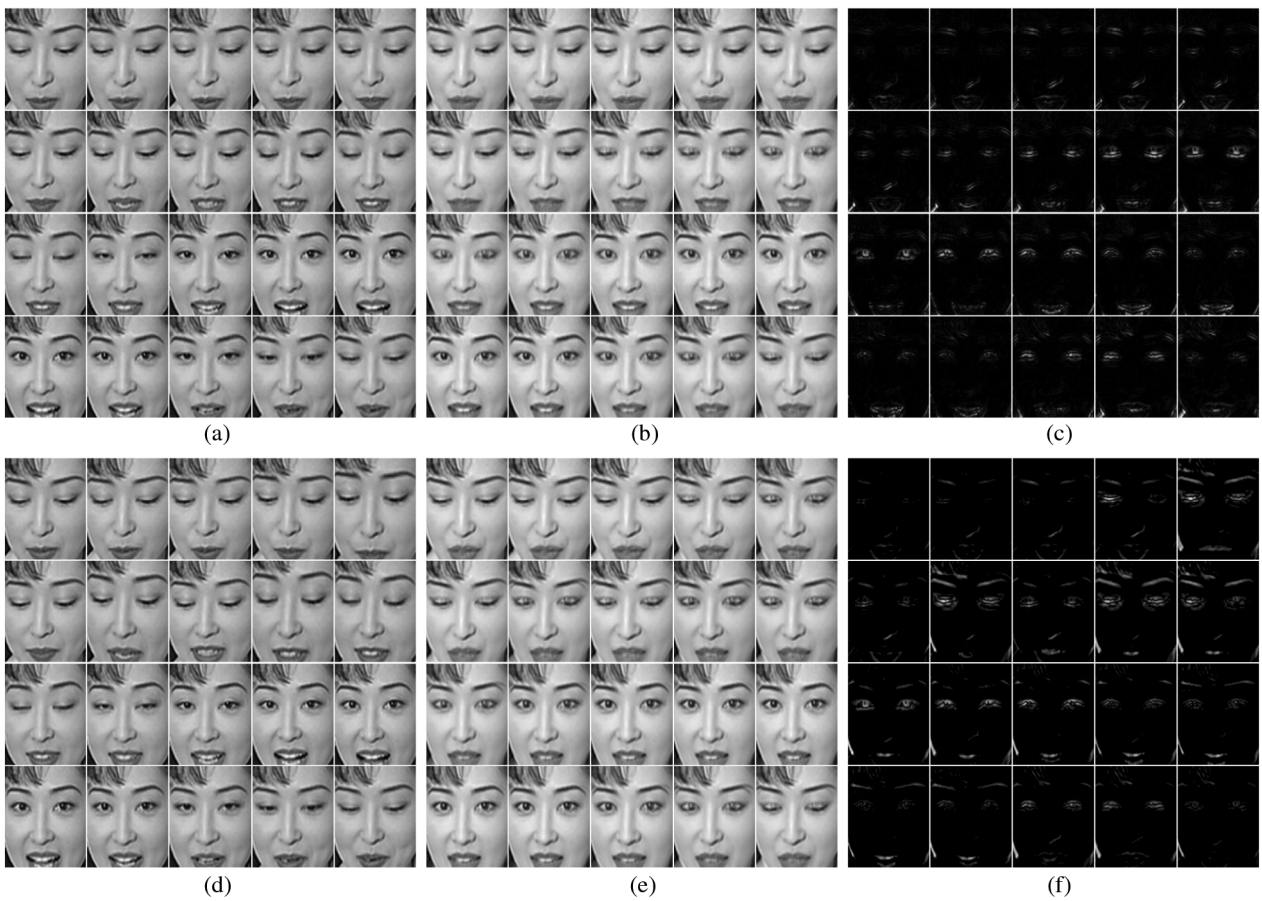


Fig. 7. Face alignment results. (a)(b)(c) are the results of RASL [12] by batch align 20 frames of Akiyo. (d)(e)(f) are the results of the proposed method. $D \circ \tau$ denote the aligned images, A denotes the low-rank matrix and E denote the sparse errors. The aligned images $D \circ \tau$ can be decomposed as the sum of low-rank matrix A and sparse errors E. Each frame is aligned by the frame by frame alignment method using an existing well-aligned matrix which contains 20 face regions. The size of each face region is 160×128 (a) RASL $D \circ \tau$ (b) RASL A (c) RASL E (d) Proposed $D \circ \tau$ (e) Proposed A (f) Proposed E.

The objective evaluation of face distortion recovery performance is shown in Table III. The PSNR of distorted face regions are shown in the second row, and the PSNR of recovered face regions are shown in the third row. On average, 1.12 PSNR gains by the proposed face distortion recovery method. Furthermore, as described in Section IV, for a distorted block in the face region, if we can find out such a unit in the database satisfies the given conditions, this distorted block will be recovered by the unit, else it will remain unchanged. The recovered ratio is computed and shown in the fifth row. The recovered ratio is defined as the ratio between the number of recovered blocks in the face regions and the total blocks number of face regions. The average recovered ratio is 48.04%. The PSNR of the recovered blocks is calculated and shown in the sixth row and seventh row. The average PSNR of these blocks before recovered is 30.60, and it achieves 35.82 after been recovered. On average, 5.22 dB gains by the proposed method. As described in Section III-B, a unit in subdatabase is computed as the cluster center of corresponding group of blocks, and the within class variance [described by (9)] of the group is related to the recovery effect. Therefore, we analyze the within class variance of groups. The probability distribution of within class variance is shown in Fig. 9. The mean value of within class variance is 6.09. And where the threshold γ in our experiment is set to be 10, then, the expected PSNR of

TABLE II
THE UNITS NUMBER COMPARISON

Sequence	Total frames	Total blocks	Total units of sub-databases	Database units	Database units/ Total blocks
Akiyo	300	24000	422	145	0.60%
Claire	150	12000	257	128	1.10%
Deadline	660	52800	651	94	0.18%
Grandma	300	24000	655	202	0.84%
Missa	150	12000	225	105	0.87%
Sign irene	540	43200	564	233	0.54%
Silent	300	24000	611	260	1.08%
Vtc	360	28800	587	433	1.50%
Average	**345**	**27600**	**497**	**200**	**0.72%**

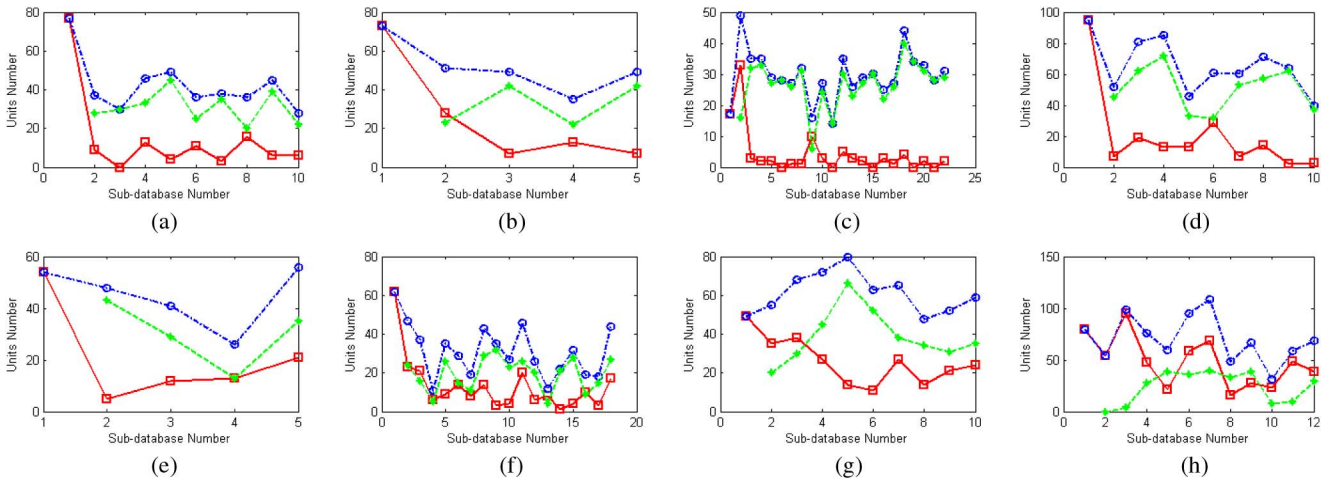


Fig. 8. The units number of subdatabase and database. The x-axis is the index of database and subdatabase, the y-axis denotes the units number. The blue line at the top describes the units number of subdatabase. The red line at the bottom shows the increase number of the units in the database. The green line is computed as blue line minus red line (a) Akiyo (b) Claire (c) Deadline (d) Grandma (e) Missa (f) Sign irene (g) Silent (h) Vtc.

TABLE III
THE OBJECTIVE PERFORMANCE (COMPARE WITH HEVC VIDEO CODING SOFTWARE HM [17])

Sequence	PSNR of face regions			Recovered	PSNR of recovered blocks			time
	HM	proposed	Δ	Ratio	HM	proposed	Δ	(s)
Akiyo	27.80	28.95	+1.15	55.76%	30.57	36.38	+5.81	0.032
Claire	28.62	29.17	+0.55	43.07%	32.18	36.05	+3.87	0.029
Deadline	28.09	28.76	+0.67	49.57%	31.76	36.28	+4.52	0.030
Grandma	30.30	32.71	+2.41	68.26%	30.90	35.92	+5.02	0.038
Missa	29.02	29.58	+0.56	35.77%	31.63	35.51	+3.88	0.022
Sign irene	26.77	27.27	+0.50	31.08%	29.96	35.13	+5.18	0.047
Silent	27.69	29.16	+1.47	50.06%	29.32	36.62	+6.31	0.038
Vtc	26.98	28.63	+1.65	50.74%	28.46	35.67	+7.21	0.041
Average	**28.16**	**29.28**	**+1.12**	**48.04%**	**30.60**	**35.82**	**+5.22**	**0.035**

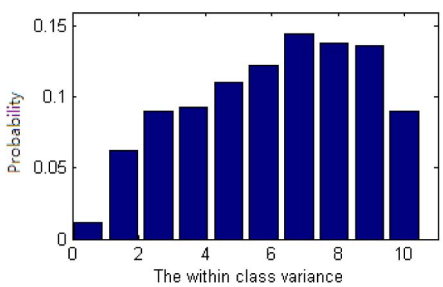


Fig. 9. Probability distribution of within class variance.

recovered blocks should nearly 36.06, $(10 \times log10(\frac{255\times255}{6.09+10}))$. We can observe that our experimental PSNR is nearly to the expected PSNR.

TABLE IV
OBJECTIVE PERFORMANCE OF NEWS REPORT SEQUENCES (COMPARED WITH HEVC VIDEO CODING SOFTWARE HM [17])

Sequence	HM PSNR	Proposed PSNR	Δ PSNR	Recovered Ratio
N5	31.13	36.51	+5.39	65.33%
N6	31.32	36.27	+4.95	66.65%
N7	31.37	36.16	+4.79	65.52%
N8	31.21	36.37	+5.16	64.71%
Average	**31.26**	**36.33**	**+5.07**	**65.55%**

The consuming time of the face distortion recovery procedure is shown in the last row. It contains the time of searching corresponding unit in database and using it to recover the distorted blocks. The average consuming time is 0.035 second per frame.

The subjective experimental results are shown in Fig. 10. For each sequence, we provide the comparison results of

Fig. 10. The subjective performance. For each test sequence, we give the comparison of consecutive 5 frames. (a)(b)(e)(f)(i)(j)(m)(n) are the decoded results of HM (The reference software of HEVC video coding system [17]). (c)(d)(g)(h)(k)(l)(o)(p) are the recovered results of our proposed method. The size of each face region is 160×128 (a) Akiyo HM (b) Claire HM (c) Akiyo Proposed (d) Claire Proposed (e) Deadline HM (f) Grandma HM (g) Deadline Proposed (h) Grandma Proposed (i) Missa HM (j) Sign irene HM (k) Missa Proposed (l) Sign irene Proposed (m) Silent HM (n) Vtc HM (o) Silent Proposed (p) Vtc Proposed.

consecutive 5 face regions. We can observe that the recovered face regions are obviously clearer.

Experimental Results on Long Sequences: In this part, the proposed algorithm is validated on multiple long sequences. 8

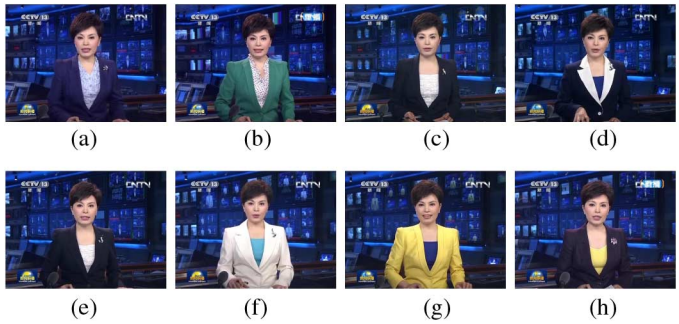


Fig. 11. Sequences of the reporter. The subheading in parenthesis is the broadcast date of the news.

news report videos are downloaded from the web site [18]. In our experiments, the algorithm is performed on the frames including the reporter. While the reporter appeared, the proposed algorithm begins to work. If the scene changes, we keep the decoded video unchanged and do not perform our procedure until the reporter appear again. The frames including the reporter are detected by a simple and effective method in our paper. For each news report video, we assign a frame of the reporter at the beginning. This frame can be written as $f_0 = (I_0, u_0, v_0)$, where I_0, u_0, v_0 represent the Y, U, V components of frame f_0, respectively. For the subsequent frames, the MSE between f_0 and the current frame f_c in Y component and V component are computed, respectively. If $mse(I_0 - I_c) < \alpha$ and $mse(v_0 - v_c) < \beta$, f_c is also a frame belonging to the reporter. In our paper, we set $\alpha = 46, \beta = 20$. To make the experiments more comparable, we keep each sequence containing 2100 reporter's frames. All of these videos belong to the same reporter are shown in Fig. 11.

Let the news reporter sequences be described as $News = \{N_1, N_2, \cdots, N_7, N_8\}$ which are divided into 2 groups. $\{N_1, N_2, N_3, N_4\}$ are in a group which is used as the training data to establish the face database, and the others $\{N_5, N_6, N_7, N_8\}$ to be anther group which is used as the test data to be recovered by using the face database. The objective experimental results are shown in Table IV.

The PSNR of recovered blocks are shown in Table IV. An average 5.07 PSNR is achieved by the proposed method. And the average recovered ratio of distorted blocks achieves 65.55%, which is higher than the recovered ratio of the standard YUV sequences in the previous section. We can observe that the face database for the same people can be used for the face distortion recovery in different scenes, which means that for a video conversation user, if there is already existing a face database, it can be used in the further conversational video face distortion recovery. And if it can achieve a target recovered ratio (e.g 60%) and PSNR (e.g. 36dB), the face database does not need to be updated anymore. In the experiment, the face database finally contains 1356 units. It is generated by total $6.72 \times 10^5 (2100 \times 4 \times 80)$ original blocks. The units number takes 0.20% of the number of total original blocks. And the units are used to recover $4.40 \times 10^5 (2100 \times 4 \times 80 \times 65.55\%)$ distorted blocks.

VII. Conclusions

In this paper, we propose a personal face database based distortion recovery scheme for conversational video coding. The long-range correlation of the same user's face in the conversation is employed. The face feature database is established and online updated at the sender side during the conversation. At the receiver side, the face database is received to recover the distorted face region. Additionally, since the face alignment is an important procedure in our method, we develop a fast frame by frame face alignment algorithm to align current face image with an existing well-aligned matrix. The experimental results prove that the proposed method improves the visual quality of the decoded videos significantly with a small burden on the face feature database. Furthermore, the personal face feature database can be reused for the same person in different conversational videos with seldom updating.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading and valuable comments on this paper.

References

[1] G. J. Sullivan, J. 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.

[2] *ITU-T recommendation H.261: Video codec for audiovisual services at px64 kbits*, , 1990.

[3] H. Liu, D. Xu, Q. Huang, W. Li, M. Xu, and S. Lin, "Semantically-based human scanpath estimation with HMMs," in *Proc. Int. Conf. Comput. Vision*, 2013, pp. 3232–3239.

[4] B. Xiong, X. Fan, and C. Zhu, "Face region based conversational video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 7, pp. 917–931, Jul. 2011.

[5] Y. Liu, Z. Li, and Y. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 134–139, Jan. 2008.

[6] G. Boccignone, A. Marcelli, P. Napoletano, G. D. Fiore, G. Iacovoni, and S. Morsa, "Bayesian integration of face and low-level cues for foveated video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1727–1740, Dec. 2008.

[7] F. Li and Y. Gao, "ROI-based error resilient coding of H.264 for conversational video communication," in *Proc. Int. Wireless Commun. Mobile Computing Conf.*, 2011, pp. 2033–2036.

[8] H. Yu, C. Wang, and S. Yu, "A novel error recovery scheme for H.264 video and Its application in conversational services," *IEEE Trans. Consumer Electron.*, vol. 50, no. 1, pp. 329–334, Jan. 2004.

[9] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Jun. 1998.

[10] J. Wang and M. E. Cohen, "Very low frame-rate video streaming for face-to-face teleconference," in *Proc. Data Compression Conf.*, 2005, pp. 309–318.

[11] X. Wang, L. Su, Q. Huang, G. Li, and H. Qi, "Online learning based face distortion recovery for conversational video coding," in *Proc. Data Compression Conf.*, 2013, p. 526.

[12] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," U. Mass. Amherst, Amherst, MA, USA, Tech. Rep., 2007, pp. 07–49.

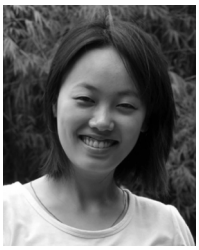
[14] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, pp. 125–141, 2007.

[15] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proc. Int. Conf. Mach. Learning*, 2003 [Online]. Available: http://www.vlfeat.org/

[16] H. Liu, L. Latecki, and S. Yan, "Fast detection of dense subgraphs with iterative shrinking and expansion," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2131–2142, Sep. 2013.

[17] HEVC reference software HM12.0. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/

[18] "The web site of news report videos," [Online]. Available: https://cctv.cntv.cn/lm/xinwenlianbo/



Xi Wang received the B.S. degree in computer science from the Communication University of China, Beijing, China, in 2007, and is currently working toward the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China.

Her research interests include video coding and image processing.

Li Su received the Ph.D. degree in computer science from the Graduate University of Chinese Academy of Sciences, Beijing, in 2009.

She is currently an Associate Professor of the School of Computer and Control, University of Chinese Academy of Sciences, Beijing, China. Her research interests include image processing and media computing.

Honggang Qi received the M.S. degree in computer science from Northeast University, Shenyang, China, in 2002 and the Ph D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Associate Professor of the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include video coding and VLSI design.

Qingming Huang (M'04–SM'08) received the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1994.

He was a Postdoctoral Fellow with the National University of Singapore from 1995 to 1996, and worked with the Institute for Infocomm Research, Singapore, as a Member of Research Staff from 1996 to 2002. He joined the Chinese Academy of Sciences under the Science100 Talent Plan in 2003, and is currently a Professor with the University of Chinese Academy of Sciences, Beijing, China. His current research areas are image and video analysis, video coding, pattern recognition, and computer vision.

Guorong Li received the B.S. degree in technology of computer application from the Renmin University of China, Beijing, China, in 2006, and the Ph.D. degree in technology of computer application from the University of the Chinese Academy of Sciences, Beijing, China, in 2012.

She is currently a Post-Doctoral Fellow with the University of the Chinese Academy of Sciences, Beijing, China. Her current research interests include cross-media analysis, object tracking, video analysis, pattern recognition, and computer vision.