Speech Communication 94 (2017) 50–61

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

A subspace based progressive coding method for speech compression

Serkan Keser[a,*], Ömer Nezih Gerek[b], Erol Seke[c], Mehmet Bilginer Gülmezoğlu[c]

[a] *Department of Electronic and Automation, Ahi Evran University, Kırşehir 40300, Turkey*
[b] *Department of Electrical & Electronics Engineering, Anadolu University, Eskişehir 26555, Turkey*
[c] *Department of Electrical & Electronics Engineering, Eskişehir Osmangazi University, Eskişehir 26480, Turkey*

ARTICLE INFO

Article history:
Received 15 February 2017
Revised 13 August 2017
Accepted 4 September 2017
Available online 13 September 2017

Keywords:
Independent Component Analysis (ICA)
Karhunen Loeve Transform (KLT)
Speech codecs
Subspace methods

ABSTRACT

In this study, two novel methods, which are based on Karhunen Loeve Transform (KLT) and Independent Component Analysis (ICA), are proposed for coding of speech signals. Instead of immediately dealing with eigenvalue magnitudes, the KLT- and ICA-based methods use eigenvectors of covariance matrices (or independent components for ICA) by geometrically grouping these vectors into fewer numbers of vectors. In this way, a data representation compaction is achieved. Further compression is achieved through discarding autocovariance eigenvectors corresponding to the small eigenvalues and applying vector quantization on the remaining eigenvectors. Additionally, this study proposes an iterative error refinement process, which uses the rest of the available bandwidth in order to transmit an efficient representation of the description error for better SNR. The overall process constitutes a new approach to efficient speech coding, with ICA being used in subspace speech coding for the first time. Constant bit rate (CBR) and variable bit rate (VBR) coding algorithms are employed with the proposed methods. TIMIT speech database is used in the experimental studies. Speech signals are synthesized at 2.4 kbps, 8 kbps, 12.2 kbps, 16 kbps, 16.4kbps and 19.85 kbps rates by using various frame lengths. The qualities of synthesized speech signals are compared to those of available speech codecs, i.e., LPC (2.4 kbps), G.728 (LD-CELP, 16 kbps), G.729A (CS-CELP, 8 kbps), EVS (16.4 kbps), AMR-NB (12.2 kbps) and AMR-WB (19.85 kbps).

1. Introduction

The goal of speech coding is to represent digital speech waveform with as few bits as possible while maintaining the intelligibility and quality that is required for the particular application (Gibson, 2005). In addition, most applications of speech coding require low coding delays, which is an undesirable property since long coding delays interfere with speech interaction (Chen et al., 1992). Major speech coders can be classified into two categories as waveform and parametric coders. The former includes speech coders such as PCM and ADPCM, and latter class (also known as vocoders) includes very low bit-rate synthetic speech coders (Kondoz, 2007). LPC-based coder is a parametric coder which is mostly used in audio signal processing and speech processing to represent the spectral envelope of speech waveform in a compressed form. This coder uses the information of a finite extent linear predictive model (Deng and O'Shaughnessy, 2003). Linear prediction based speech coding techniques (CELP, MELP, VSELP etc.) have been widely researched in the literature (Vasuki and Vanathi, 2006; Supplee et al., 1997; Gerson and Jasiuk, 1990; Chen et al., 1992). These types of speech coders are capable of synthesizing good quality speech at a reasonably low bit rate. An LPC variant, CELP, has evolved to become the dominant paradigm for real time speech compression (Devalapalli et al., 2003), which is capable of achieving high quality speech coding at rates from 16 kbps to 32 kbps. A further variant, namely the low delay-CELP (LD-CELP) algorithm was adopted by the International Telephone and Telegraph Consultative Committee (CCITT) for speech coding at 16 kbps with toll quality and became a standard as G.728 (Chen et al., 1992). Similarly, G.729A (CS-ACELP Annex A) is a high quality low bandwidth codec at 8 kbit/s with low complexity. ITU-T (International Telecommunication Union) has standardized G.729 as the standard speech coding algorithm for VoIP, DSVD (Digital Simultaneous Voice over Data) and multimedia applications (Rashed et al., 2013). The mixed excitation linear prediction (MELP) coder was chosen by the Digital Voice Processing Consortium to replace the existing 2400 bps Federal Standard FS-1015 (LPC-10). The MELP coder is based on the traditional LPC model, with additional features to improve its performance (Supplee et al., 1997). The vector sum excited linear prediction (VSELP) speech coder utilizes a codebook with a structure that allows for a very efficient search procedure (Gerson and Jasiuk, 1990). The coder uses two VSELP excitation

* Corresponding author.
E-mail addresses: skeser@ahievran.edu.tr (S. Keser), ongerek@anadolu.edu.tr (Ö.N. Gerek), eseke@ogu.edu.tr (E. Seke), bgulmez@ogu.edu.tr (M.B. Gülmezoğlu).

https://doi.org/10.1016/j.specom.2017.09.002
0167-6393/© 2017 Elsevier B.V. All rights reserved.

codebooks, a gain quantizer which is robust to channel errors, and a novel adaptive pre/postfilter arrangement.

Being the fundamental application medium of speech coders, GSM networks started with Full Rate (FR) speech codec and evolved into Enhanced Full Rate (EFR). The Adaptive Multi-Rate (AMR) codec was added to 3GPP (The 3rd Generation Partnership Project) Release 98 for GSM to enable codec rate adaptation to radio conditions (Holma and Toskala, 2011). AMR data rates range from 4.75 kbps to 12.2 kbps at a sampling rate of 8 kHz. The AMR-Wideband (AMR-WB) codec was added to 3GPP Release 5 (Holma and Toskala, 2011). AMR-WB uses a sampling rate of 16 kHz with data rates ranging from 6.6 kbps to 23.85 kbps, out of which, codec rates from 6.6 kbps to 19.85 kbps can be supported by GSM as well. AMR-WB offers clearly better voice quality than AMR (or AMR-NB) at the same data rate, so that it is also referred to as “wideband audio with narrowband radio transmission” (Holma and Toskala, 2011).

Enhanced Voice Services (EVS) has been developed in 3GPP and is described in 3GPP TS 26.441. EVS has been designed for high quality and efficient coding of speech and music. It has two operational modes which are primary and EVS AMR-WB Inter-Operable. EVS is generally used in narrow band and denoted as EVS (NB). It offers up to 20 kHz audio bandwidth and has high robustness to delay jitter and packet losses due to its channel aware coding (Atti et al., 2015) and improved packet loss concealment (Lecomte et al., 2015).

Among the vast literature regarding speech coding at various target bandwidths, subspace based methods attempt to reproduce a signal by using few coefficients in a transform domain. Classically applied transforms include Karhunen Loeve Transform (KLT) (Goyal, 2001), Independent Component Analysis (ICA) (Ferreira and Figueiredo, 2003), Discrete Cosine Transform (DCT) (Ahmed et al., 1974), Fast Fourier Transform (FFT) (Kumar and Kumar, 2012) and Wavelet Transform (WT) (Skodras et al., 2001) due to their known energy compaction properties. While DCT and WT are popular for image coding, signal specific methods such as KLT and ICA need further researches for practical speech coding applications.

KLT is the most statistically efficient orthonormal transform in terms of energy compaction and decorrelation. If a signal has Gaussian distribution with a certain temporal correlation (i.e. $R(\tau)\neq\delta(\tau)$), KLT is guaranteed to be more effective than the original signal domain (Kim and Kleijn, 2004; Lee and Kim, 2010; Ozerov and Kleijn, 2011). In addition, a Gaussian signal can be obtained by using the weighted sum of many independent components that have non-Gaussian distributions. Therefore, KLT- and ICA-based methods can be preferable to reconstruct a signal.

In this article, a novel eigen-representation grouping idea is employed together with iterated error improvement for speech coding. The performances of the proposed KLT- and ICA-based methods are evaluated for different bit rates. While similar subspace-based studies realize speech coding by considering basic structures of standard codecs (e.g. CELP, LPC, etc.) (Kim and Kleijn, 2004; Lee and Kim, 2010; Ozerov and Kleijn, 2011; Ju et al., 2014; Oger et al., 2006), the proposed speech codec does not utilize structures of standard codecs. In contrast to the studies that rely on standards, we applied vector quantization to the principal component vectors of matrices that are constructed by stacking KLT eigenvectors (we will call such matrices as “eigenvector matrices”). As a final step, this study proposes an iterative error refinement process, where an error signal is used to recursively improve SNR until the desired bit rate is achieved. The emphasis regarding the novelty of the proposed method is, therefore, two folds:

(i) Eigenvector clustering by alignment, and
(ii) Error data feedback to improve SNR at a target bit rate.

The proposed KLT-based coder uses eigenvalues and eigenvectors of the autocorrelation matrix that is estimated from a training speech data set. Once the codebooks of eigenvectors and transform coefficients are constructed, the codec is constructed and performances are measured on an isolated set of test speech signals. The proposed decoder synthesizes speech signals using the trained codebooks in a nearest neighbor sense. Several parameters need to be tested at the training and testing stages. The results are compared with the performances of conventional state-of-the-art speech coding methods. Particularly, the effects of codebook and code vector sizes, the effect of vector quantization (VQ) optimality, the effect of frame size and eventual data rate are thoroughly investigated. Additionally, independent component and mixing matrix codebooks are constructed for ICA-based coders by using the training speech data set. Again, the test speech signals are synthesized by using pre-trained codebooks. Constant bit rate (CBR) and variable bit rate (VBR) coding approaches are adopted with the proposed methods. Signal quality is determined by means of Perceptual Evaluation of Speech Quality (PESQ) (Kumar et al., 2014), Composite Measure (Cov) (Hu andLoizou, 2008; Krishnamoorthy, 2011) and Mean Opinion Score (MOS) (Osahenvemwen, 2015).

The rest of the study is organized as follows: The proposed KLT-based speech coder is described in Section 2.1. In Section 2.2, we present the vector quantization process, which is performed in order to reduce the computation time for KLT-based method. The proposed ICA-based speech coder is described in Section 2.3. We describe constant bit rate coding and variable bit rate coding in Section 3. Objective quality measures are given in Section 4. Experimental results, Discussion and Conclusion sections are presented in Sections 5–7, respectively.

2. Proposed subspace methods for speech coding

KLT- and ICA- based subspace methods are proposed for coding of speech signal frames. The codebooks are constructed by using speech frames in the training set of the TIMIT database (Zue et al., 1990). Test speech signals are synthesized by using these codebooks. In our experiments, codebook sizes are taken as 2^{10} to 2^{16}.

2.1. KLT–based subspace method

The first considered subspace strategy depends on the celebrated KLT, which is known to decorrelate signal samples. Basis functions of KLT are eigenvectors of the autocorrelation matrix of the input signal, rendered according to the eigenvalue magnitudes in a descending order. Since the signals of interest are normalized to a zero mean, KLT equivalently uses eigenvectors of the autocovariance matrix. We define the transform coefficient (associated with an eigenvector) as simply the projection of the input signal onto that eigenvector. If the decoder has knowledge of the eigenvectors and transform coefficients, the signal can be recovered as a linear combination of the orthogonal eigenvectors.

The proposed method starts by obtaining a covariance matrix from the frames of a phoneme. Length-N subframes, which correspond to 1-shifted versions of overlapping windows in each frame with the length of M ($M=2N-1$), are extracted in order to form $N\times N$ data matrices. Each shifted subframe occupies one row of data matrix. Fig. 1 illustrates how the mentioned data matrices are created.

The covariance matrix of pth data matrix ($\mathbf{X}_p$) in the training data set is obtained as

$$\mathbf{C}_p=\frac{1}{N-1}\sum_{i=1}^{N}\left[(\mathbf{x}_i-\mathbf{m}_x)(\mathbf{x}_i-\mathbf{m}_x)^{\mathrm{T}}\right],\ p=1,2,...,r, \tag{1}$$

where $\mathbf{x}_i$ ($\mathbf{x}_i\in\mathbf{R}^N$) is ith column vector which corresponds to the ith row vector in the pth data matrix and r is the number of covari-

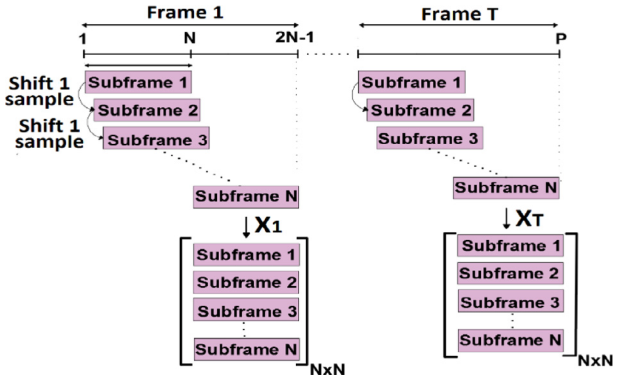

Fig. 1. The creation of $N \times N$ data matrices by subframes.

ance matrices (which may take values as 1024, 2048, 4096, 16384, 32768 and 65536 in the experiments). In Eq. (1), N is the number of rows and columns in a data matrix and $\mathbf{m}_x$ is the mean of the column vectors. Next, the eigenvector and coefficient codebooks are formed by using the eigenvectors of the covariance matrix. When the eigenvalues of the covariance matrix are sorted in descending order $\{\lambda_1 > \lambda_2 > \ldots \lambda_N\}$, a matrix, $\mathbf{\Phi}_p$, is formed by stacking the eigenvectors corresponding to the largest K eigenvalues of $\mathbf{C}_p$ as shown in Eq. (2)

$$\mathbf{\Phi}_p = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_K\}, \tag{2}$$

where $\boldsymbol{\phi}$'s are length-N eigenvectors and $\mathbf{\Phi}_p \in \mathbf{R}^{N \times K}$. The eigenvector codebook can be written as a stack of matrices $\mathbf{\Phi}_p$:

$$\mathbf{\Phi}^{cb} = \{\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_r\}. \tag{3}$$

The pth coefficient matrix $\mathbf{Y}_p$ is written as:

$$\mathbf{Y}_p = \mathbf{\Phi}_p^T \mathbf{X}_p, \tag{4}$$

where $\mathbf{X}_p$ is pth data matrix with size $N \times N$, and $\mathbf{Y}_p \in \mathbf{R}^{K \times N}$. Eventually, a coefficient codebook ($\mathbf{Y}^{cb}$) with a size of $K \times M$ $(M = N.r)$ is created by merging all columns of coefficient matrices. The number of coefficient vectors (M) is equal to one of 1024, 2048, 4096, 16384, 32768 and 65536, and is selected according to different frame lengths and bit rates.

The test speech signals are divided into non-overlapping frames ($\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^t$) in the testing phase. The qth test frame ($\mathbf{x}^q$) is compressed by projecting it onto the designed subspace, which is spanned by the eigenvector matrices. Euclidean distances are found for kth iteration mathematically by using each eigenvector matrix and coefficient vector for the qth test frame ($\mathbf{x}_k^q$) as follows;

$$f_p{}^q = \left\| \mathbf{x}_k^q - \left(\mathbf{\Phi}_p \mathbf{y}_p^q \right) \right\|, \quad p = 1, 2, \ldots, r, \tag{5}$$

where $\mathbf{\Phi}_p$ is qth eigenvector matrix that is taken from the eigenvector codebook and $\mathbf{y}_p^q = (\mathbf{\Phi}_p^T \mathbf{x}_k^q)$. The minimum distance is specified with index a_k for kth iteration and denoted as follows;

$$a_k = \operatorname{argmin}\left(f_p{}^q \right), \quad p = 1, 2, \ldots, r. \tag{6}$$

If the eigenvector matrix for which the minimum distance is obtained is notated as $\boldsymbol{\phi}_{a_k}$, the coefficient vector ($\mathbf{y}_k^q$) is found for kth iteration as follows;

$$\mathbf{y}_k^q = \mathbf{\Phi}_{a_k}^T \mathbf{x}_k^q. \tag{7}$$

The Euclidean distances (A_l^q) are obtained by using $\mathbf{y}_k^q$ as;

$$A_l^q = \left\| \mathbf{y}_k^q - \mathbf{Y}_l \right\|, \quad l = 1, 2, \ldots, M, \tag{8}$$

where $\mathbf{Y}_l$ is lth column vector in the coefficient codebook. The index of the coefficient vector, which gives the minimum Euclidean distance, can be specified with index h_k for kth iteration and expressed as follows;

$$h_k = \operatorname{argmin}\left(A_l^q \right), \quad l = 1, 2, \ldots, M. \tag{9}$$

The coefficient vector which gives the minimum distance is denoted by $\hat{\mathbf{y}}_k^q = \mathbf{Y}_{h_k}$ at the encoder side. Using the eigenvector matrix ($\boldsymbol{\phi}_{a_k}$) and the coefficient vector ($\hat{\mathbf{y}}_k^q$), approximate test frame ($\hat{\mathbf{x}}_k^q$) is computed as;

$$\hat{\mathbf{x}}_k^q = \mathbf{\Phi}_{a_k} \hat{\mathbf{y}}_k^q, \tag{10}$$

with an error signal which is expressed as;

$$\mathbf{e}_k^q = \mathbf{x}_k^q - \hat{\mathbf{x}}_k^q \;. \tag{11}$$

If the error reduction process is used for the desired bit rate, the error signal ($\mathbf{e}_k^q$) is substituted into Eq. (5) instead of $\mathbf{x}_k^q$. In other words, $\mathbf{e}_k^q$ is used as the test frame for the second iteration ($k = 2$) and $\hat{\mathbf{e}}_k^q$ is found by following the same procedure that is used to find $\hat{\mathbf{x}}_k^q$. The indices which are used to find $\hat{\mathbf{x}}_k^q$ and $\hat{\mathbf{e}}_k^q$ (belonging to the eigenvector matrix $\boldsymbol{\phi}_{a_k}$ and coefficient vector ($\hat{\mathbf{y}}_k^q$) are transmitted to the decoder. Using these indices, the decoder can synthesize the signal as;

$$\hat{\mathbf{x}}^q = \hat{\mathbf{x}}_1^q + \sum_{k=2}^{d} \hat{\mathbf{e}}_k^q. \tag{12}$$

Since the error is transmitted for further refinement, the error between the synthesized signal $\hat{\mathbf{x}}^q$ and $\mathbf{x}^q$ is reduced. In Fig. 2, the encoder and decoder parts of the proposed KLT-based subspace method are shown in parts (a) and (b), respectively. In this figure, k is the current iteration index which is initially set to 1. Here, d is maximum numbers of iterations that is used for CBR and VBR coding. A set of indices is found for all iterations and is sent to the decoder. This process is performed for all frames of the test signal.

2.2. Eigenvector quantization

Vector quantization is performed on eigenvectors to shorten the computational delay. K-means method (Jain, 2010), which uses the principal eigenvector of each eigenvector matrix, is used for the eigenvector quantization. Thus, the clusters are obtained from the eigenvector matrices including principal eigenvectors in similar directions. The new eigenvector codebook can be represented by $\boldsymbol{\theta}^{cb} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m\}$ after the quantization, where $m < r$. The new eigenvector matrix is found for jth cluster and defined as;

$$\boldsymbol{\theta}_j = \sum_{i=1}^{p} \mathbf{\Phi}_i, \quad j = 1, 2, \ldots, m, \tag{13}$$

where p is number of eigenvector matrices in the jth cluster. Each column of the eigenvector matrix is normalized by dividing it with its norm. The size of the normalized eigenvector matrix is eventually $N \times K$.

We construct a toy example for illustration purposes here. Let $\mathbf{A} = [\mathbf{a}_1 \; \mathbf{a}_2]$ and $\mathbf{B} = [\mathbf{b}_1 \; \mathbf{b}_2]$ be matrices of principal component vectors $\mathbf{a}_i$ and $\mathbf{b}_i$, and $\mathbf{u}_1$ and $\mathbf{u}_2$ are the sum of the first ($\mathbf{a}_1, \mathbf{b}_1$) and second ($\mathbf{a}_2, \mathbf{b}_2$) principle components of $\mathbf{A}$ and $\mathbf{B}$ matrices as $\mathbf{u}_1 = (\mathbf{a}_1 + \mathbf{b}_1)$ and $\mathbf{u}_2 = (\mathbf{a}_2 + \mathbf{b}_2)$, so that $\mathrm{U} = [\mathbf{u}_1 \; \mathbf{u}_2]$. The normalized matrix is $\mathbf{U}_{norm} = [\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \; \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|}]$. Two matrices $\mathbf{A}$ and $\mathbf{B}$ have the same cluster in the two dimensional vector space and the normalization process is shown in Fig. 3.

We used 3 different approaches for the search of optimal eigenvector matrix within the eigenvector codebook;

(i) In the first approach (we will denote as **A1**), speech signals are synthesized by searching all eigenvector matrices in the eigenvector codebook. (The computational delay is high).

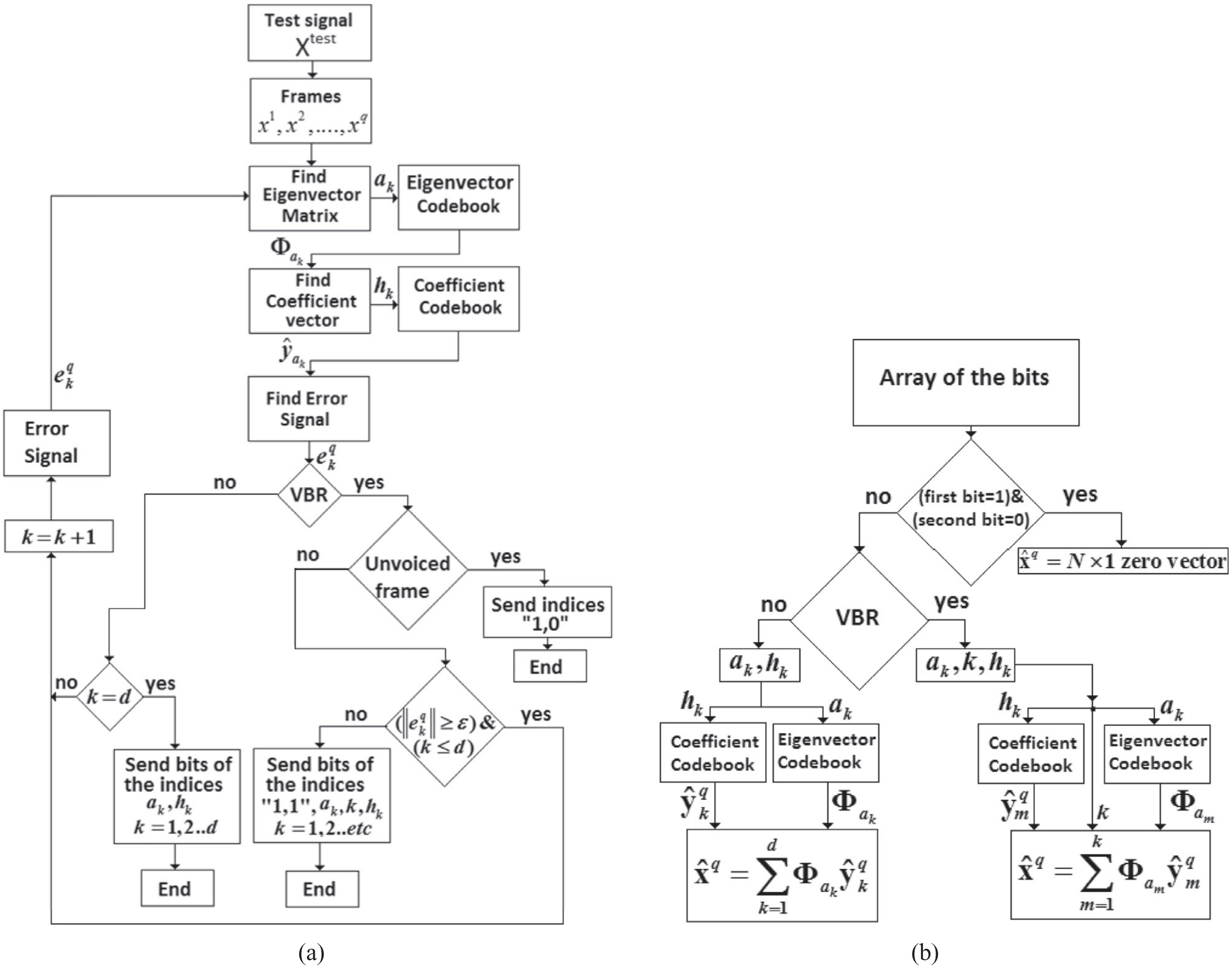

Fig. 2. Block diagrams of (a) the encoder and (b) the decoder for the proposed KLT-based method.

(ii) In the second approach (we will denote as **A2**), synthesis is done by searching only M normalized eigenvector matrices ($\mathbf{U}_{norm}$). (The computational delay is low). The *k*-means method is used in this approach.

(iii) In the third approach (we will denote as **A3**), in order to further reduce the computational delay and improve bit allocation efficiency, we have devised a new quantization technique, where the eigenvector matrices are quantized into clusters that contain equal number of eigenvector matrices inside their quantization regions. Due to the similarity and inspiration from the k-means method, we called the above method 'balanced *k*-means'.

The new codebook structure for the third approach (**A3**) is realized with the following algorithm. At the end of this algorithm, *M* clusters (each with *L* eigenvector matrices) is formed as a codebook structure.

Step 1) Set $t=1$.

Step 2) Find the Euclidean distances among the principal vectors of the first and the remaining eigenvector matrices;

$$\mathrm{D}_i = \|\boldsymbol{\phi}_1 - \boldsymbol{\phi}_i\|, \quad i = 2, 3, \dots, N$$

where *N* is the number of eigenvector matrices in the eigenvector codebook.

Step 3) Find *L*-1 eigenvector matrices which give the smallest *L*-1 Euclidean distances among *N*-1 eigenvector matrices. Combine the first eigenvector matrix and *L*-1 eigenvector matrices, and form *t*th cluster which consists of *L*-size eigenvector matrices (ψ_t). Here, *L* is the number of eigenvector matrices in *t*th cluster.

Step 4) Find *t*th normalized eigenvector matrix ($\mathbf{U}_{norm}^t$) by using the eigenvector matrices (ψ_t) in the *t*th cluster as explained in Fig. 3. These eigenvector matrices in the *t*th cluster are removed from the eigenvector codebook.

Step 5) Increase *t* by 1. If $t=M$ terminate the algorithm. Otherwise, go to **Step 2**. Here *M* is the number of clusters.

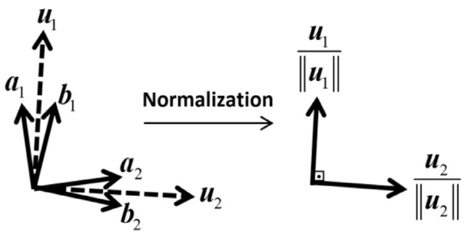

Fig. 3. Representation of the vector quantization in the two dimensional vector space.

For a test frame, the most appropriate cluster is selected by using a normalized eigenvector matrix which gives the smallest Euclidean distance in Eq. (5) ($\mathbf{U}_{norm}^t$, $t=1,2,\dots,M$). Then, the most suitable eigenvector matrix, which has the smallest Euclidean distance, is found from the L-size eigenvector matrices belonging to the selected cluster. Since $(L+M) << N$, the computational delay is considerably reduced. In addition, the number of used bits is equal to the new codebook structure (bit number $=\log_2 M + \log_2 L$) which is automatically equal to the number of bits for the non-quantized codebook (bit number $=\log_2 N$). In other words, the number of allocated bits is equal for the codebooks in the first approach and the third approach. However, the encoding delay in the third approach is less than that of the first approach.

Let us explain the situation with a numerical case example. Suppose that we have 64 eigenvector matrices and we want to create 16 clusters, the third approach produces a codebook with 16 sets, each of which has 4 eigenvector matrices (making the total number of bits $\log_2 4 + \log_2 16 = 6$, which is naturally equal to $\log_2 64$. According to the process strategy of the third approach (**A**3), these 4 eigenvector matrices in each cluster are normalized as in Fig. 3 to construct an eigenvector matrix. Continuing with the same example, 16 normalized eigenvector matrices are generated. For the test frame, the most appropriate normalized eigenvector matrix or cluster which gives the smallest Euclidean distance in Eq. (5) is selected from these 16 normalized eigenvector matrices. Then, the most appropriate eigenvector matrix is selected (i.e. among the four eigenvector matrices corresponding to the selected cluster). Therefore, a total of only 20 (i.e. $16+4$) eigenvector matrices are searched instead of the total set of 64 eigenvector matrices.

A similar algorithm is performed for the coefficient codebooks, so the computational delay of the coefficient vectors is considerably shortened for the third approach. For the second approach (**A2**), the coefficients and eigenvectors in the clusters are not searched as in the third approach. Instead, the quantized coefficient and eigenvector codebooks are searched. It must be noted that the number of quantized coefficients and eigenvectors in the coefficient and eigenvector codebooks is M for this approach, meaning that the search time for the second approach is also low.

2.3. ICA-based subspace method

ICA is quite similar to KLT in view of their properties. However, unlike KLT, the basis vectors of ICA are not orthogonal to each other. In this work, we investigate the performance differences of these two subspace methods.

Similar to the autocorrelation concept (which yields the KLT), mutual information of random variables is a measure of the mutual dependence among the variables. ICA of a random vector consists of finding a linear transformation that minimizes the statistical dependence among its components (Comon, 1992). Applications of ICA include data compression, detection and localization of sources or blind identification and deconvolution (Comon, 1992). In this study, the ICA method is implemented through the FastICA algorithm (Hyvärinen and Oja, 2000) in MATLAB. In our case, frames of training speech signals are divided into non-overlapping subframes. FastICA algorithm is applied to the data matrix that is obtained from subframes of each frame. Then, mixing matrix (**A**) and independent component matrix (**S**) are found for every data matrix, which are used for constructing the codebook. It must be noted that this codebook generation process uses the training data. In the test phase, test signals are synthesized by using these codebooks. The process details can be explained as follows.

Let $\mathbf{X}_p \in \mathbf{R}^{M \times N}$ denote pth data matrix, which is defined as

$$\mathbf{X}_p = \mathbf{A}_p \mathbf{S}_p, \ \ p = 1, 2, \ldots, r, \tag{14}$$

where $\mathbf{A}_p$ is an $M \times M$ mixing matrix, $\mathbf{S}_p$ is an $M \times N$ matrix which includes independent components with $M < N$, and r is the number of data matrices, which are obtained from the training set. The independent component codebook is created by a set of independent component matrices as:

$$\mathbf{S}^{cb} = \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_r\}. \tag{15}$$

The mixing codebook ($\mathbf{U}^{cb}$) is constructed by concatenating all rows of mixing matrices. Here, $\mathbf{U}^{cb} \in \mathbf{R}^{Z \times M}$ and Z is equal to $M \times r$. In the testing phase, test speech signals are divided into non-overlapping frames ($\mathbf{x}^1$, $\mathbf{x}^2, \ldots,$ $\mathbf{x}^t$) and pth signal for kth iteration is computed as

$$\mathbf{x}^q_{p_k} = \left(\mathbf{x}^q_k \mathbf{S}^T_p \left(\mathbf{S}_P \mathbf{S}^T_p \ \right)^{-1}\right)\mathbf{S}_p, \ \ p = 1, 2, \ldots, r, \tag{16}$$

where $\mathbf{x}^q_k$ is qth test frame for kth iteration, and $\mathbf{S}_p$ is pth independent component matrix in the independent component codebook. The index of kth iteration (a_k) is found from the minimum Euclidean distance between $\mathbf{x}^q_k$ and $\mathbf{x}^q_{p_k}$ as follows;

$$a_k = \text{argmin}\left(\left\|\mathbf{x}^q_k - \mathbf{x}^q_{p_k}\right\|\right), \ \ p = 1, 2, \ldots, r. \tag{17}$$

By taking $\mathbf{S}_{a_k}$ from the independent component codebook, an approximate mixing vector $\mathbf{u}^q_{a_k}$ is written as

$$\mathbf{u}^q_{a_k} = \mathbf{x}^q_k \mathbf{S}^t_{a_k}\left(\mathbf{S}_{a_k}\mathbf{S}^t_{a_k}\right)^{-1}. \tag{18}$$

Then, by searching all rows of the mixing codebook, the index h_k is found by using

$$h_k = \text{argmin}\left\|\mathbf{u}^q_{a_k} - \mathbf{U}^{cb}_l\right\|, \ \ l = 1, 2, \ldots, Z, \tag{19}$$

where $\mathbf{U}^{cb}_l$ is lth row vector in the mixing codebook. When $\mathbf{u}_{h_k} = \hat{\mathbf{u}}^q_k$ is chosen from $\mathbf{U}^{cb}$, $\hat{\mathbf{x}}^q_k$ is computed as,

$$\hat{\mathbf{x}}^q_k = \mathbf{u}_{h_k}\mathbf{S}_{a_k}, \tag{20}$$

and the approximate error signal is expressed as

$$\mathbf{e}^q_k = \mathrm{x}^q_k - \hat{\mathbf{x}}^q_k \tag{21}$$

In this work, the same error reduction processes are used for KLT- and ICA- based methods. In Fig. 4, the encoder and decoder parts of the proposed ICA-based method are shown in parts (a) and (b), respectively.

3. Constant and variable bit rate coding

In this study, we used both CBR and VBR based coding with KLT-, and ICA-based methods. CBR coding is realized with the same number of bit allocation for voiced and unvoiced frames of the speech signal. Conversely, the number of bits for each frame is allowed to vary in the VBR coding.

With VBR coding, it is possible to synthesize a higher quality speech signal than CBR by assigning less bits for the unvoiced frames and more bits for the voiced frames. The main idea of this study is to start with autocovariance eigenvectors for an approximate speech representation and then to iteratively reduce the error between the actual and synthesized signal. The number of iterations is adjusted according to target bit rates. Different error reduction algorithms are used for CBR and VBR cases.

In CBR coding, the encoder naturally aims at the bit rate of the output samples. If $\boldsymbol{M}^e$ (or $\boldsymbol{M}^i$) and $\boldsymbol{M}^c$ (or $\boldsymbol{M}^m$) are the size of eigenvector (or independent component) and coefficient (or mixing) codebooks respectively, we need $\log_2\boldsymbol{M}^e + \log_2\boldsymbol{M}^c$ (or $\log_2\boldsymbol{M}^i + \log_2\boldsymbol{M}^m$) bits. Higher values of $\boldsymbol{M}^e$ and $\boldsymbol{M}^c$ correspond to better quality with a lower compression ratio. If FL is the length of frame, d is maximum number of iterations and FS is the sampling frequency, then the desired bit rate (kbps) for CBR coding is defined as

$$b_{CBR} = \frac{[tb \ \times \ FS] \times \ d}{FL}, \tag{22}$$

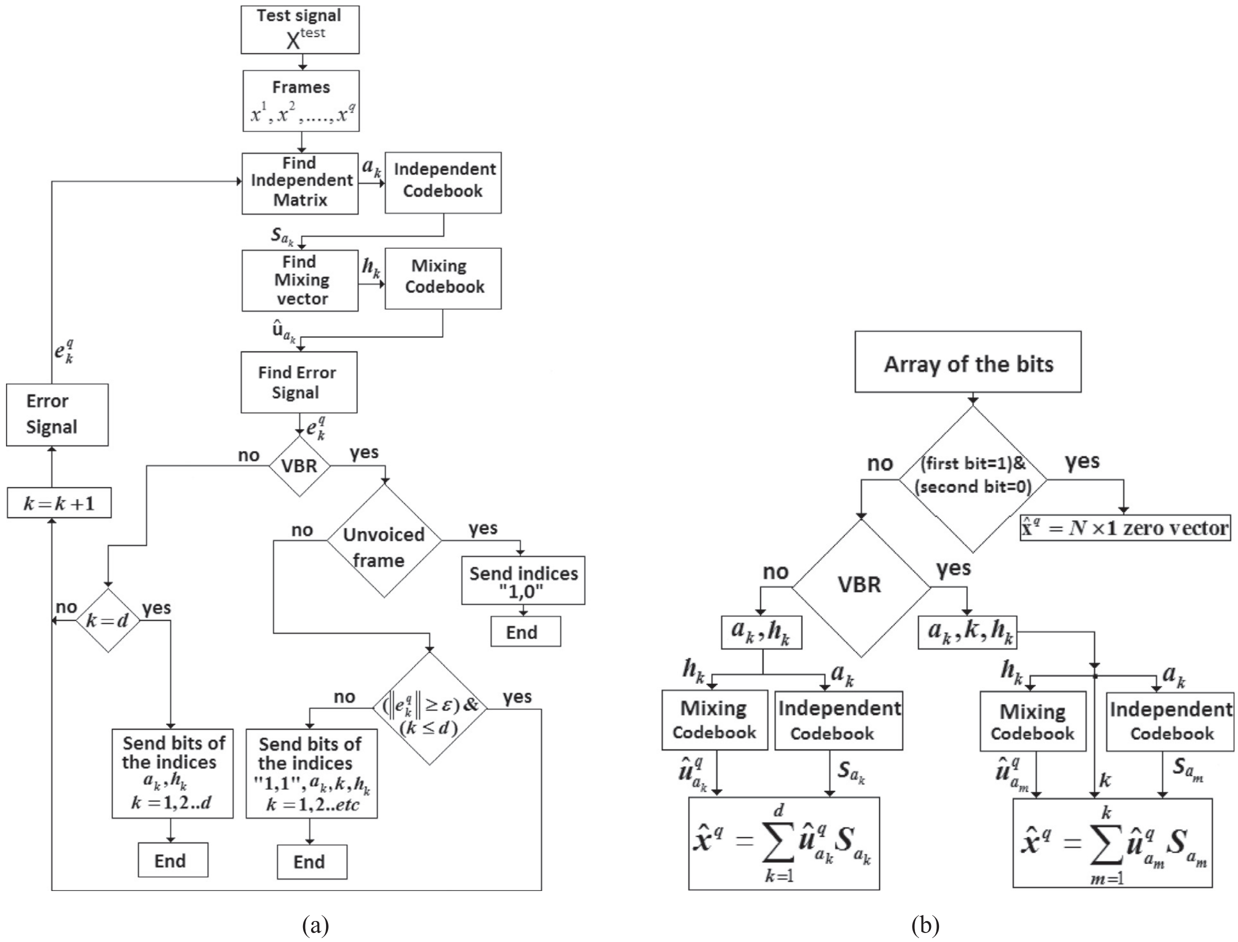

Fig. 4. The block diagrams of (a) the encoder and (b) the decoder for the proposed ICA-based method.

Table 1
Bit allocations for KLT_CBR and ICA_CBR methods.

Frame length (Samples)	NI	Eigenvector codebook's bits	Coefficient codebook's bits	Total bits	Kbps
80	1	12 bits	12 bits	$1 \times (12+12) = 24$ bits	2.4
32	1	16 bits	16 bits	$1 \times (16+16) = 32$ bits	8
64	3	16 bits	17 bits	$3 \times (16+17) = 99$ bits	12.2
32	2	16 bits	16 bits	$2 \times (16+16) = 64$ bits	16
48	3	16 bits	17 bits	$3 \times (16+17) = 99$ bits	16.4
96	4	15 bits	15 bits	$4 \times (15+15) = 120$ bits	19.85

where *tb* is the number of bits required to send indices of codebooks and is equal to ($\log_2\boldsymbol{M}^{\mathbf{e}}$+$\log_2\boldsymbol{M}^{\mathbf{c}}$) or ($\log_2\boldsymbol{M}^{i} + \log_2\boldsymbol{M}^{m}$). The parameter *d* is a constant to represent the maximum number of iterations. This parameter is known a-priori by both the encoder and the decoder. The bit allocations for KLT_CBR and ICA_CBR methods are given in Table 1 in which NI is the number of iteration.

At the encoder side, in the first iteration ($k=1$), an error signal between the test and the synthesized frames is found as $\mathbf{e}_1^q = \mathbf{x}^q - \hat{\mathbf{x}}_1^q$. In the second iteration ($k=2$), the error signal (e$_1^q$) is used as the test frame and $\hat{\mathbf{e}}_2^q$ is found by following the same steps that are used to find $\hat{\mathbf{x}}_1^q$. In the third iteration ($k=3$), the error signal ($\mathbf{e}_2^q = \mathbf{x}^q - \hat{\mathbf{x}}_1^q - \hat{\mathbf{e}}_2^q$) is used as the test frame and $\hat{\mathbf{e}}_3^q$ is found. When the desired bit rate is reached, the process is stopped ($k=d$). The synthesis of a test frame at decoder side and bit allocations for the proposed KLT_CBR and ICA_CBR methods are illustrated in Fig. 5.

In Fig. 5, bit(a_1), ..., bit(a_k) and bit(h_1), ..., bit(h_k) indicate bit allocations corresponding to the eigenvector (or independent component) codebook and coefficient (or mixing) codebook respectively.

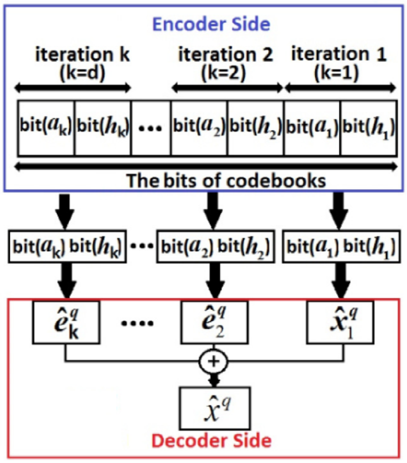

Fig. 5. Bit allocations for the proposed KLT_CBR and ICA_CBR methods.

Table 2
Bit allocations for KLT_VBR and ICA_VBR methods.

Frame length (Samples)	Eigenvector codebook's bits	Coefficient codebook's bits	Iteration numbers	Iteration bits	Voiced frame bits	Unvoiced frame bits	Average bits per frame	Kbps
80	12	12	$1 < K < 2$	1	K × (12 + 12)	2	24	2.4
32	16	16	$1 < K < 2$	1	K × (16 + 16)	2	32	8
64	16	16	$1 < K < 4$	2	K × (16 + 16)	2	99	12.2
32	16	16	$1 < K < 3$	2	K × (16 + 16)	2	64	16
48	16	16	$1 < K < 4$	2	K × (16 + 16)	2	99	16.4
96	15	15	$1 < K < 5$	3	K × (15 + 15)	2	119	19.85

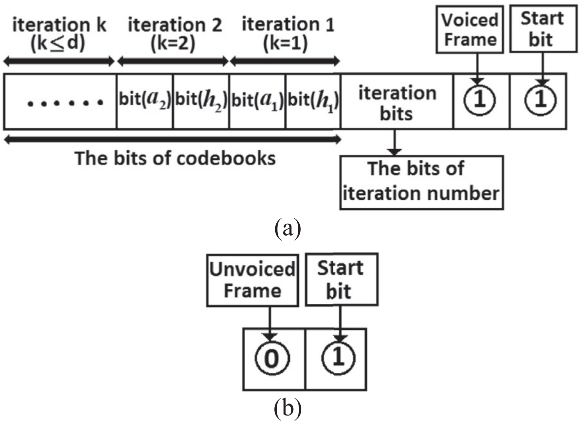

Fig. 6. The structures of bit array for (a) voiced and (b) unvoiced frame for the proposed KLT_VBR and ICA_VBR.

For CBR coding, the codebook indices are only transmitted to the decoder side for all frames of test signal and test signals are synthesized by using the same codebooks at the decoder side.

In VBR coding, a predefined error threshold (ε) and a maximum allowed number of iterations (d) are dynamically determined in order to achieve the desired bit rates. The process given in Fig. 5 is also used in VBR coding, but the bit allocation for VBR coding differs according to whether a frame can be voiced or unvoiced. Then the algorithm continues until the error norm is less than the predefined threshold (ε) or the index of current iteration (k) is set equal to the maximum iteration value (d):

$$\left\| \mathbf{x}_k{}^{test} - \mathbf{x}_k{}^{syn} \right\| < \ \varepsilon \text{ or } k = d, \tag{23}$$

where $\mathbf{x}_k{}^{\text{test}}$ and $\mathbf{x}_k{}^{\text{syn}}$ are test and synthesized frames for kth iteration.

An additional bit is used to mark whether the current frame is voiced or unvoiced, in the VBR coding. As shown in Fig. 6, first bit of the total bit array is always the start bit. If the frame is voiced, then the second bit of the array is '1', otherwise it is '0', indicating that the frame is unvoiced. If norm value of error signal of the qth frame is less than ε_{un} in the encoder side for first iteration, then this frame is evaluated as unvoiced frame. Otherwise, qth frame is evaluated as voiced. The parameter ε_{un} is a predefined threshold value for unvoiced frames $\varepsilon_{un} < \varepsilon < 1$. For unvoiced frames, the total bit array only consists of "0 1" and the decoder consequently generates a zero vector at the whole size of the frame. Different number of iterations can be used at encoder side for each voiced frame. Therefore, the number of iterations must be known at decoder side. Fig. 6(a) shows the bit structure of voiced frames with bits corresponding to the number of iterations, whereas Fig. 6(b) indicates bits in case of unvoiced frames.

Table 3
The scores for quality of the speech and ranges of MOS values.

Quality of the speech	Score	MOS indicator	MOS values
excellent	5	Very satisfied	4.3–5
good	4	Satisfied	4.0–4.3
fair	3	Some user satisfied	3.6–4.0
poor	2	Many user dissatisfied	2.6–3.6
bad	1	Not recommended	1.0–2.6

The desired bit rate (kbps) for VBR coding is defined as,

$$B_{VBR} = \frac{2f_{uv} + \sum_{y=1}^{f_v} (tb+2) + f_v \cdot \ in}{(\ f_{uv} + f_v\ )} \times \frac{FS}{FL} \tag{24}$$

where k_y is the number of iterations used for yth voiced frame ($1 \leq \ k_y \ \leq \ d, \ y = 1, \ 2, \ldots, \ f_v$). In Eq. (24), f_v and f_{uv} are the number of voiced and unvoiced frames, respectively and in is the number of bits corresponding to the number of iterations used for a voiced frame. There is no bit allocation for ε, ε_{un} and d parameters in the VBR, as their values are previously determined and they are known by the encoder and decoder. In Table 2, bit allocations are given for different bit rates using KLT_VBR and ICA_VBR.

As can be seen from Table 2, VBR coding has a different bit allocation structure according to CBR coding. The training set in the TIMIT database is used to determine the sizes of the eigenvector codebooks. The same codebooks are utilized for CBR and VBR codings. Naturally, VBR coding involves a few more iterations than CBR coding for the voiced frames.

4. The quality measurements of speech coders

The qualities of speech coders are evaluated using MOS (Mean Opinion Score), as well as PESQ (Perceptual Evaluation of Speech Quality), WSS (Weighted Slope Spectral distance), and LLR (Log Likelihood Ratio) objective measures (Hu and Loizou, 2008; Krishnamoorthy, 2011). PESQ is a test methodology for objective prediction of perceived speech quality and has been widely used in telecommunications and IP networks. It is asserted to have the highest correlation with the subjective measurements (Goudarziand Sun, 2009).

4.1. MOS (Mean opinion score)

The Mean Opinion Score (MOS) provides a numerical measure of the voice quality in telephony networks. MOS is obtained from subjective tests by using human listeners. The ratings depend on each listener's perception (Osahenvemwen, 2015). MOS is defined as the arithmetic mean of subjective evaluations in listening or conversational tests, with score values of 1 to 5, corresponding to verbal explanations as given in Table 3.

Table 4
PESQ values for 16 kbps (KLT_VBR(**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
4	3,07	3,32	3,52	3,68	3,74	3,79	3,74	3,71	3,66	3,61
6	3,11	3,37	3,53	3,65	3,70	3,88	3,97	3,75	3,73	3,65
8	2,96	3,21	3,38	3,44	3,52	3,73	3,62	3,57	3,53	3,51
10	2,83	3,00	3,14	3,27	3,30	3,34	3,35	3,30	3,28	3,26

Table 5
Cov values for 16 kbps (KLT_VBR(**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
4	3.80	4.06	4.26	4.41	4.47	4.51	4.46	4.44	4.39	4.35
6	3.82	4.13	4.28	4.35	4.46	4.58	4.62	4.47	4.45	4.39
8	3.69	3.93	4.11	4.17	4.29	4.46	4.40	4.37	4.32	4.30
10	3.57	3.74	3.87	4.01	4.09	4.12	4.15	4.05	4.02	4.00

4.2. PESQ (Perceptual evaluation of speech quality)

This evaluation describes an objective method to predict the subjective quality of 3.1 kHz (narrow-band) handset telephony and narrow-band speech codecs. PESQ is used to calculate a distance between the original and degraded speech signal (PESQ score). The PESQ score is mapped to a MOS-like scale, which is a single number in the range of 1 to 4.5 (Kumar et al., 2014).

4.3. WSS (Weighted spectral slope)

The WSS measure is a frequency domain expression, based on an auditory model. The WSS measure is defined as (Krishnamoorthy, 2011);

$$WSS = \frac{1}{M}\sum_{m=0}^{M-1} \frac{\sum_{j=1}^{K} WSS(j,m)(S_o(j,m)-S_s(j,m))^2}{\sum_{j=1}^{K} WSS(j,m)}, \qquad (25)$$

where *WSS(j,m)* are the weights computed as described in (Krishnamoorthy, 2011). In Eq. (25), K is taken as 25, M is the number of data segments, $S_o(j,m)$ and $S_s(j,m)$ are the spectral slopes for the jth frequency band of the original and processed speech signals, respectively.

4.4. LLR (Log likelihood ratio)

LLR measure is one of the LPC-based objective measures. It mainly concerns with the similarity of spectral envelopes. The LLR for each 20ms frame is defined as (Hu and Loizou, 2008);

$$\text{LLR}(\mathbf{a}_o,\mathbf{a}_s) = \log\left(\frac{\mathbf{a}_s\mathbf{R}_o\mathbf{a}_s}{\mathbf{a}_o\mathbf{R}_o\mathbf{a}_o}\right), \qquad (26)$$

where a_o and a_s are the LPC vectors of the original and the processed speech frames, respectively, and R_o is the autocorrelation matrix of the original speech frame.

4.5. Composite measure

Since conventional objective measures are not sufficient to provide high correlations in terms of speech/noise distortion and overall speech quality, it is necessary to combine different objective measures in order to create a Composite measure (Hu and Loizou, 2008). In this study, we have used a composite measure (Cov) for overall speech quality. The Cov measure is an overall planning and combination of the evaluation measures in time domain, frequency domain and perceptual field, and is defined as follows (Hu and Loizou, 2008):

$$Cov = 1,594 + 0,805 \cdot PESQ - 0,512 \cdot LLR - 0,007 \cdot WSS. \qquad (27)$$

5. Experimental study

5.1. Database

In the experimental studies, the speech material of the TIMIT database has been divided into training and testing sets. The training and test sets contain 4620 and 1344 utterances, respectively. The speech signal in the database has a sampling frequency of 16 kHz (16 bit, PCM format). The sampling frequency was converted to 8 kHz with downsampling process and 45 phonemes were obtained by merging similar utterances from 61 phonemes in the speech database. Only the sampling frequency of 16 kHz at 19.85 kbps is used for the experiments.

In the training phase, equal number of phonemes (varying between 15 and 100) is used for each phoneme class. The length of each phoneme is equal to 960 and 1920 samples at 8 kHz and 16 kHz in the training phase, respectively. Then, 960 samples or 1920 samples of phonemes are divided into frames and data matrices are constructed by using these frames for ICA- and KLT-based methods. 30 utterances of sentences randomly selected from the test set of the TIMIT database are assigned to be used in the testing phase. MOS, PESQ and Cov values are found for each utterance. Then, average MOS, PESQ and Cov values are computed for all sentences.

5.2. Results

Multiple tests are performed in order to analyze the efficiency of the proposed KLT and ICA based methods. In experimental studies, KLT_CBR and KLT_VBR naturally correspond to CBR and VBR coding for KLT, respectively. Similarly, ICA_CBR and ICA_VBR correspond to CBR and VBR coding for ICA, respectively. The average values of PESQ and Cov are shown in Tables 4–11 for the first searching approach (**A1**). Frame length of 2ms was not used for KLT_VBR with 16 kbps and KLT_CBR with 8 kbps, because 8 kbps and 16 kbps bitrates are exceeded when the number of iterations is more than one. Similarly, frame lengths of 2 ms and 4 ms are not used for KLT_VBR method with 8 kbps, because 8 kbps is exceeded when the number of iterations is more than one. Speech signals are reproduced by using eigenvectors corresponding to the largest K eigenvalues for all different frame lengths (K = 1,2,3,…,10).

Average PESQ and Cov values of 30 test utterances are given in Tables 12–18. These tables also include MOS values, which were obtained by subjective listening tests of synthesized and original speech waveforms by 10 students and 10 academicians. Listening tests were realized in a quiet environment using high quality headphones.

The average MOS values of 20 users are given in Tables 12 and 14–18. In Tables 12–18, **A1, A2** and **A3** indicate the first, second

Table 6
PESQ values for 16 kbps (KLT_CBR(**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
2	2,90	3,22	3,53	3,57	3,37	3,11	2,89	2,7	2,62	2,48
4	2,85	3,08	3,26	3,52	3,50	3,60	3,65	3,48	3,41	3,27
6	2,84	3,03	3,21	3,59	3,66	3,76	3,82	3,70	3,63	3,61
8	2,77	2,98	3,16	3,23	3,49	3,32	3,27	3,25	3,23	3,19
10	2,74	2,90	3,00	3,13	3,27	3,22	3,16	3,14	3,12	3,08

Table 7
Cov values for 16 kbps (KLT_CBR(**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
2	3,65	3,98	4,27	4,29	4,14	3,90	3,70	3,51	3,41	3,26
4	3,59	3,81	4,02	4,25	4,24	4,33	4,37	4,21	4,15	4,02
6	3,57	3,78	3,96	4,27	4,33	4,41	4,49	4,38	4,29	4,26
8	3,52	3,76	3,93	4,01	4,19	4,09	4,03	4,01	3,98	3,93
10	3,50	3,67	3,76	3,91	3,98	3,99	3,94	3,92	3,90	3,80

Table 8
PESQ values for 8 kbps (KLT_VBR (**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
6	2,66	2,86	2,95	3,11	3,23	3,37	3,50	3,39	3,27	3,26
8	2,65	2,84	3,00	3,05	3,08	3,13	3,13	3,15	3,13	3,12
10	2,68	2,81	2,89	2,98	2,99	3,04	3,01	3,03	3,01	2,98

Table 9
Cov values for 8 kbps (KLT_VBR (**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
6	3,22	3,55	3,68	3,79	3,92	4,02	4,10	4,03	3,87	3,86
8	3,32	3,52	3,67	3,74	3,78	3,83	3,83	3,85	3,85	3,83
10	3,34	3,49	3,58	3,67	3,69	3,74	3,71	3,72	3,71	3,70

Table 10
PESQ values for 8 kbps (KLT_CBR (**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
4	2,64	2,82	2,98	3,12	3,20	3,09	2,99	2,87	2,78	2,73
6	2,6	2,78	2,92	3,01	3,03	3,08	3,10	3,10	3,08	3,06
8	2,51	2,67	2,82	2,9	2,88	2,86	2,79	2,68	2,66	2,61
10	2,22	2,32	2,41	2,39	2,39	2,36	2,38	2,26	2,26	2,21

Table 11
Cov values for 8 kbps (KLT_CBR (**A1**)).

FL (ms)	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
4	3,21	3,41	3,6	3,82	3,87	3,78	3,68	3,59	3,5	3,46
6	3,37	3,58	3,66	3,72	3,75	3,78	3,80	3,81	3,8	3,78
8	3,15	3,36	3,51	3,6	3,61	3,59	3,47	3,39	3,31	3,27
10	2,88	3,02	3,12	3,11	3,11	3,1	3,09	2,98	2,95	2,89

Table 12
Average MOS, PESQ and Cov values at 2.4 Kbps.

METHOD	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	2,5	2,38	2,94
ICA_CBR	ER (+)	2,47	2,32	2,85
KLT_VBR (A1)	ER (+)	2,52	2,55	3,02
KLT_CBR (A1)	ER (+)	2,47	2,48	2,98
KLT_VBR (A3)	ER (+)	2,42	2,43	2,81
KLT_CBR (A3)	ER (+)	2,33	2,29	2,58
LPC	–	2,31	2,35	2,61

and third searching approaches respectively. In these tables, ER (+) indicates that the error reduction procedure is additionally applied.

The experimental results for different codebook sizes are shown in Table 13. It is observed that computational time increases for large size codebooks in high bit rates. Therefore, the size of eigenvector codebook (number of eigenvector matrices) is decreased from 65,536 to 16,384 and 1024 by using vector quantization in the codebook, as described in Section 2.2. In Table 13, K indicates the length of the coefficients or number of largest eigenvalues which result in best quality.

In Table 13, the number of the clusters (each of the clusters has a normalized eigenvector matrix) is $M = 1024$. The number of eigenvector matrices in each cluster for the third approach (**A3**) is $L = 64$. A total of $M + L$ $(1024 + 64)$ eigenvector matrices are searched to find the most suitable eigenvector matrix. As seen from the Table 13, the quality values decrease when the quantization method is used for the second approach, but the third approach keeps the quality close to that of full search for 16 kbps and 19.85 kbps. In Tables 14–18, we compared well-known speech codecs G729A, G728, EVS, AMR-NB and AMR-WB with the best

Table 13
Average PESQ and Cov values based on quantization method at 16 Kbps and 19.85 Kbps.

			KLT_CBR				KLT_VBR			
Searching Approaches	Eigenvector codebook size	Kbps	*K*	ER	PESQ	Cov	*K*	ER	PESQ	Cov
A2	1024	16	5	ER (−)	2,86	3,54	6	ER (+)	3,01	3,70
A2	16384	16	3	ER (−)	3,41	4,01	6	ER (+)	3,54	4,12
A1	65536	16	7	ER (−)	3,82	4,49	7	ER (+)	3,97	4,62
A1	65536	19.85	7	ER (+)	4,01	4,68	7	ER (+)	4,11	4,74
A3	1024 + 64	19.85	5	ER (+)	3,85	4,53	5	ER (+)	4,01	4,67
A3	1024 + 64	16	7	ER (+)	3,65	4,30	5	ER (+)	3,81	4,47

Table 14
The average MOS, PESQ and Cov values at 8 Kbps.

Methods	ER	MOS	PESQ	Cov
ICA_VBR	ER(+)	3,38	2.98	3.77
ICA_CBR	ER(−)	3,28	2.83	3.56
KLT_VBR (A1)	ER(+)	3,61	3.50	4.10
KLT_CBR (A1)	ER(−)	3,42	3,20	3,82
KLT_VBR (A3)	ER(+)	3,40	3.25	3.85
KLT_CBR (A3)	ER(−)	3,30	3,03	3,68
G729A	–	3,58	3,38	3,91

Table 15
The average MOS, PESQ and Cov values at 12.2 Kbps.

Methods	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	3,94	3,48	4,11
ICA_CBR	ER (+)	3,80	3,36	3,95
KLT_VBR (A1)	ER (+)	4,05	3,78	4,38
KLT_CBR (A1)	ER (+)	4,00	3,65	4,22
KLT_VBR (A3)	ER (+)	3,95	3,64	4,23
KLT_CBR (A3)	ER (+)	3,82	3,38	4,04
AMR-NB	–	4,11	3,74	4,32

Table 16
The average MOS, PESQ and Cov values at 16 Kbps.

Methods	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,05	3.63	4.29
ICA_CBR	ER (−)	3,98	3.50	4.05
KLT_VBR (A1)	ER (+)	4,15	3,97	4,62
KLT_CBR (A1)	ER (+)	4,08	3,82	4,49
KLT_VBR (A3)	ER (+)	4,06	3,81	4,47
KLT_CBR (A3)	ER (+)	4,00	3,65	4,30
G728	–	4,11	3,68	4,43

Table 17
The average MOS, PESQ and Cov values at 16.4 Kbps.

Methods	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,13	3,67	4,32
ICA_CBR	ER (+)	4,03	3,56	4,11
KLT_VBR (A1)	ER (+)	4,19	4,00	4,65
KLT_CBR (A1)	ER (+)	4,10	3,84	4,51
KLT_VBR (A3)	ER (+)	4,08	3,83	4,48
KLT_CBR (A3)	ER (+)	4,04	3,68	4,32
EVS (NB)	–	4,43	4,16	4,80

Table 18
The average MOS, PESQ and Cov values at 19.85 Kbps.

Methods	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,23	3,84	4,60
ICA_CBR	ER (+)	4,17	3,73	4,51
KLT_VBR (A1)	ER (+)	4,36	4,11	4,74
KLT_CBR (A1)	ER (+)	4,29	4,01	4,68
KLT_VBR (A3)	ER (+)	4,25	4,01	4,67
KLT_CBR (A3)	ER (+)	4,18	3,85	4,53
AMR-WB	–	4,50	4,19	4,66

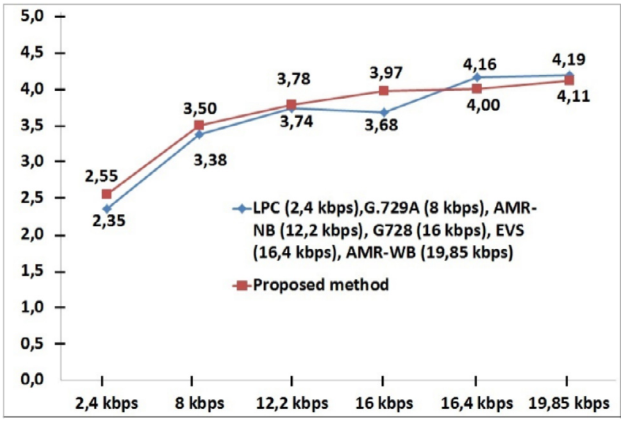

Fig. 7. PESQ values obtained for the KLT_VBR (**A1**) and other codecs.

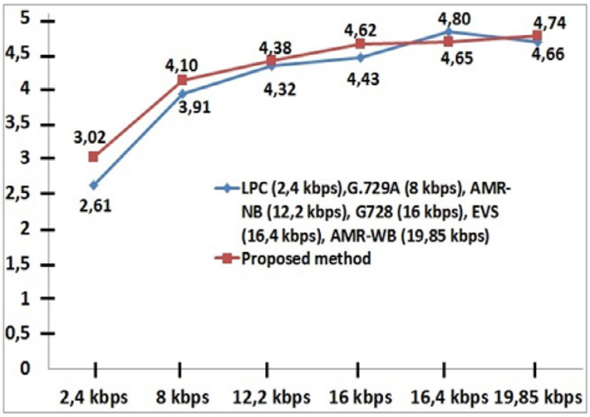

Fig. 8. Cov values obtained for the KLT_VBR (**A1**) and other codecs.

operation points of the proposed KLT- and ICA-based methods at the same available bit rates. The bit rates of standard speech codecs are 8 kbps, 12.2 kbps, 16 kbps, 16.4 kbps and 19.85 kbps for G729A, AMR-NB, G728, EVS (Narrow-Band) and AMR-WB respectively. In Tables 14–17, the best results are found by using mixing matrices of size 4 × 4 and independent component matrices of size 4x*N* for the ICA-based method, where *N* corresponds to the length of frames. In Tables 12 and 18, mixing matrices of size 2 × 2 and independent component matrices of size 2x*N* are used for ICA-based method.

The PESQ and Cov values that are obtained for the proposed method (KLT_VBR (**A1**)) and other speech codecs are comparatively shown in Figs. 7 and 8, respectively. A sampling rate of 16 kHz is used for the results in Table 18.

For KLT- and ICA-based methods, encoder and decoder delays per frame are given in Tables 19 and 20 respectively. In these tables, FL is the frame length (milliseconds), ED and DD indicate the encoder and decoder delays (milliseconds), respectively.

Table 19
Encoder delays per frame for KLT- and ICA-based methods.

	2,4 kbps		8 kbps		12,2 kbps		16 kbps		16,4 kbps		19,85 kbps	
	ED	FL	ED	FL	ED	FL	ED	FL	ED	FL	ED	FL
KLT_CBR (A1)	382	10	1260	4	1520	6	2190	6	2210	6	2420	6
KLT_CBR (A3)	14	10	29	4	43	6	59	6	63	6	74	6
KLT_VBR (A1)	302	10	1182	6	1440	6	2082	6	2098	6	2280	6
KLT_VBR (A3)	12	10	26	6	37	6	54	6	59	6	71	6
ICA_CBR	266	20	965	4	1380	4	1918	2	1981	2	2195	6
ICA_VBR	204	20	742	4	1208	4	1650	2	1705	2	2026	6

Table 20
Decoder delays per frame for KLT- and ICA-based methods.

	2,4 kbps		8 kbps		12,2 kbps		16 kbps		16,4 kbps		19,85 kbps	
	DD	FL	DD	FL	DD	FL	DD	FL	DD	FL	DD	FL
KLT_CBR (A1)	0,6	10	0,8	4	0,9	6	0,98	6	1,01	6	1,1	6
KLT_CBR (A3)	0,18	10	0,32	4	0,45	6	0,50	6	0,51	6	0,57	6
KLT_VBR (A1)	0,57	10	0,78	6	0,87	6	0,96	6	0,98	6	1,07	6
KLT_VBR (A3)	0,12	10	0,28	6	0,4	6	0,48	6	0,50	6	0,55	6
ICA_CBR	0,51	20	0,67	4	0,83	4	0,92	2	0,94	2	1,01	6
ICA_VBR	0,46	20	0,63	4	0,79	4	0,91	2	0,92	2	0,98	6

Realization of subjective listening tests is a difficult and time consuming task. In this work, we had conducted subjective tests over 10 young students (age: 20~26, 6 male, 4 female) and 10 relatively older academicians (age: 35~50, 5 male, 5 female). Since it was not possible to reach to a larger set of subjects, we also put synthesized speech waveforms to accessible web storages for self assessment: https://github.com/serkankeser/speech.

6. Discussion

KLT is known to provide maximum energy compaction in the average sense among orthogonal transforms. If the signal is well correlated, the energy compaction results in packing most of the signal energy into the few transform coefficients, rest of which can be discarded in encoding.

While KLT uses only second order statistics to find the most important signal components, ICA depicts higher order statistics. Due to this difference, both of these transforms are applied in a novel speech coding approach that incorporates CBR or VBR. The subspace representation was used for rough approximation and error refinement was iteratively performed until the desired bit rate was met.

In the experimental studies, four different English speech utterances which have been generated as test signals of ITU-T for telecommunication systems are tested with the proposed methods. The average PESQ and Cov values of these tests are observed to be very close, especially for the frame lengths of 4 ms and 6 ms using TIMIT dataset. Besides, the results stand well against newest standard coders, indicating that the proposed methods must be considered as plausible alternatives for speech coding. The algorithms are implemented in MATLAB, running on a simple desktop PC with intel core i5 processor and 4GB RAM. The computational times were observed to vary according to desired bit rate and codebook size. The first approach (**A1**) naturally gives the best quality (due to its extensive search). Consequently, its calculation delay was high. The third approach (**A3**, where the optimal eigenvetor matrices are searched) had a synthesis quality close to the first approach for 16 kbps, 16.4 kbps and 19.85 kbps, whereas its computation delay was visibly low. We observed that quality of the synthesized speech signals was the worst in the second approach (**A2**) as compared to **A1** and **A3**.

The quality values of speech signals that are synthesized by using the KLT_VBR(**A1**) method are higher when compared to those of LPC, G729A, G.728, AMR-NB (12.2 kbps) and AMR-WB (19.85 kbps) coding methods. PESQ and Cov values of the KLT_VBR(A1) and KLT_VBR(**A3**) methods are somewhat lower than that of EVS (16.4 kbps). Only the PESQ value of the KLT_VBR(**A1**) method stands lower than that of AMR-WB (19.85 kbps). However, Cov value of KLT_VBR(**A1**) is found higher than that of AMR-WB. The KLT_CBR(**A1**) method gives higher PESQ and Cov values than those of G728 and LPC methods. In addition, the KLT_CBR(**A1**) method gives a higher Cov value than that of AMR-WB. The ICA_VBR method gives higher PESQ and Cov values than those of the LPC method. In addition, the ICA_CBR method gives a higher Cov value than that of the LPC method. However, the performances of ICA_VBR and ICA_CBR are lower than those of G729A, AMR-NB and AMR-WB methods. The KLT_VBR(**A3**) has lower the computational delay than KLT_VBR(**A1**). However, the KLT_VBR(A3) has slightly lower PESQ and Cov values than KLT_VBR(A1). Furthermore, KLT_VBR(**A3**) has higher PESQ and Cov values than LPC, G728 and has higher Cov value than AMR-WB. PESQ and Cov value of KLT_VBR(**A3**) is found slightly lower than those of AMR-NB and G729A.

For lower bit rates (such as 2.4 kbps), the proposed methods give higher MOS values than those of LPC at the same bit rate. For 8 kbps, the MOS value of speech signal that are synthesized by using the KLT_VBR(**A1**) is higher than that of G729A. For 12.2 kbps, MOS value of AMR-NB is higher than MOS values of the proposed methods. For 16 kbps, only MOS value of KLT_VBR(**A1**) is higher than that of G728. For 16.4 kbps and 19.85 kbps, MOS values of EVS (NB) and AMR-NB are higher than MOS values of the proposed methods. Although the PESQ values of ICA-based methods are visibly lower than the PESQ values of KLT-based methods, the MOS values of ICA-based methods are obtained closer to the MOS values of KLT-based methods.

7. Conclusion

A contribution of this study is to apply vector quantization to the principal component vectors of the eigenvector matrices (or independent components) which are obtained from KLT (or ICA). This method is applied to ICA for the first time in the literature. Another contribution is to process an iterative error refinement, where the error signal is used to recursively improve SNR until the desired bit rate is achieved. These two subspace methods (KLT and ICA) are applied for speech compression under CBR and VBR conditions. In all these combinations (KLT_CBR, KLT_VBR, ICA_CBR and ICA_VBR), a novel covariance eigenvector grouping strategy is proposed. Adopting a strategy of vector alignment is believed to be an insightful alternative to directly vector quantizing subspace bases. The comparison of these two subspace methods in various target bit rates for both CBR and VBR is believed to constitute a thorough justification for the usefulness of these subspace methods.

Apart from the above-mentioned alignment strategy in vector quantization, a new method for feeding the error description back to the signal representation is proposed to improve SNR at a given target bit rate. The KLT-based methods, especially KLT_VBR, gave experimentally more satisfactory results than the ICA-based and other CBR methods. Utilization of KLT at VBR was observed to provide plausible performance (quality and decoding delay) as compared to several state-of-the-art speech coding standards at analogous bit rates.

For the synthesized signals, high quality speech sounds are obtained for 12.2 kbps, 16 kbps, 16.4 kbps and 19.85 kbps bit rates. However, the computation time is observed to become a concern at these bit rates by using the first approach (**A1**). The computation delay increases together with the codebook size in high bit rates. As a remedy of this problem the computational delay is reduced by applying a novel quantization method (**A3**) which is observed to keep the original high quality. The proposed basis alignment (quantization) and error refinement processes are expected to provide a new insight to the speech coding problem.

References

Ahmed, N., Natarajan, T., Rao, K.R., 1974. Discrete Cosine Transform. IEEE Trans. Comput. 100 (1), 90–93.

Atti, V., Sinder, D.J., Subasingha, S., Rajendran, V., Dewasurendra, D., Chebiyyam, V., Zhang, X., 2015. Improved error resilience for VOLTE and VOIP with 3GPP EVS channel aware coding. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, pp. 5713–5717.

Chen, J.H., Cox, R.V., Lin, Y.C., Jayant, N., Melchner, M.J., 1992. A low- delay CELP coder for the CCITT 16kb/s speech coding standard. IEEE J. Selected Areas Commun. 10 (5), 830–849. doi:10.1109/49.138988.

Comon, P., 1992. Independent component analysis. In: Lacoume, J.-L. (Ed.), Higher-Order Statistics. Elsevier, Amsterdam, London, pp. 29–38.

Deng, L., O'Shaughnessy, D., 2003. Speech processing. In: Adynamic and Optimization-Oriented Approach. CRC Press, New York, pp. 41–50.

Devalapalli, S.K., Rangarajan, R., Venkataramanan, R., 2003. Design of a CELP Speech Coder and Study of Complexity vs Quality Trade-offs for Different Codebooks. EECS 651-Source Coding Theory.

Ferreira, A.J., Figueiredo, M.A., 2003. Class-adapted image compression using independent component analysis. In: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, 1. IEEE I-625.

Gerson, I.A., Jasiuk, M.A., 1990. Vector sum excited linear prediction (VSELP) speech coding at 8 kbps. In: Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on, 1, pp. 461–464. doi:10.1109/ICASSP.1990.115749.

Gibson, J.D., 2005. Speech coding methods, standards, and applications. IEEE Circuits Syst. Mag. 5 (4), 30–49. doi:10.1109/MCAS.2005.1550167.

Goudarzi, M., Sun, L., 2009. Performance analysis and comparison of PESQ and 3SQM in live 3G mobile networks. In: Advances in Communications, Computing, Networks and Security: Proceedings of the MSc/MResprogrammes from the School of Computing, Communications and Electronics, 2007-2008, 6, p. 48.

Goyal, V.K., 2001. Theoretical foundations of transform coding. IEEE Signal Process. Mag. 18 (5), 9–21.

Holma, H., Toskala, A., 2011. LTE For UMTS: Evolution to LTE-advanced. John Wiley & Sons, New York, pp. 351–352.

Hu, Y., Loizou, P.C., 2008. Evaluation of objective quality measures for speech enhancement. IEEE Trans. Audio, Speech, Lang. Process. 16 (1), 229–238.

Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications. Neural Netw. 13 (4), 411–430.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognit. Lett. 31 (8), 651–666.

Ju, H., Lee, S., Kim, M.Y., 2014. Complexity reduction in Karhunen-Loeve transform based speech coder for voice transmission. IEEE Trans. Consum. Electron. 60 (1), 130–136.

Kim, M.Y., Kleijn, W.B., 2004. KLT-based adaptive classified VQ of the speech signal. IEEE Trans. Speech Audio Process. 12 (3), 277–289.

Kondoz, A.M., 2007. Digital speech. In: Coding For Low Bit Rate Communication Systems. John Wiley & Sons, p. 6.

Krishnamoorthy, P., 2011. An overview of subjective and objective quality measures for noisy speech enhancement algorithms. IETE Tech. Rev. 28 (4), 292–301.

Kumar, P., Kumar, P., 2012. Performance evaluation of DFT-spread OFDM and DCT-spread OFDM for underwater acoustic communication. In: Vehicular Technology Conference (VTC Fall). IEEE, pp. 1–5.

Kumar, S., Bhattacharya, S., Patel, P., 2014. A new pitch detection scheme based on ACF and AMDF. In: Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on. IEEE, pp. 1235–1240. doi:10.1109/ICACCCT.2014.7019296.

Lecomte, J., Vaillancourt, T., Bruhn, S., Sung, H., Peng, K., Kikuiri, K., Faure, J., 2015. Packet-loss concealment technology advances in EVS. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, pp. 5708–5712.

Lee, Y., Kim, M.Y., 2010. KLT-based adaptive entropy-constrained quantization with universal arithmetic coding. IEEE Trans. Consum. Electron. 56 (4), 2601–2605.

Oger, M., Ragot, S., Antonini, M., 2006. Low-complexity wideband LSF quantization by predictive KLT coding and generalized Gaussian modeling. In: Signal Processing Conference, 2006 14th European. IEEE, pp. 1–5.

Ozerov, A., Kleijn, W.B., 2011. Asymptotically optimal model estimation for quantization. IEEE Trans. Commun. 59 (4), 1031–1042.

Osahenvemwen, O.A., 2015. Subjective Speech Evaluation on Mobile Communication Networks.

Rashed, M.A., El-Garf, T.A., Tarrad, I.F., Almotaafy, H.A., 2013. The Effect of Weight Factor on the Performance of G. 729A Speech Coder.

Skodras, A., Christopoulos, C., Ebrahimi, T., 2001. The JPEG 2000 still image compression standard. IEEE Signal Process. Mag. 18 (5), 36–58.

Supplee, L.M., Cohn, R.P., Collura, J.S., McCree, A.V., 1997. MELP: the new federal standard at 2400 bps. In: Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, 2. IEEE, pp. 1591–1594.

Vasuki, A., Vanathi, P.T., 2006. A review of vector quantization techniques. IEEE Potentials 25 (4), 39–47.

Zue, V., Seneff, S., Glass, J., 1990. Speech database development at MIT: TIMIT and beyond. Speech Commun. 9 (4), 351–356.