Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Computer Speech & Language xxx (2018) xxx-xxx

www.elsevier.com/locate/csl

Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation☆

Q1 Pulkit Sharma*, Vinayak Abrol, Nivedita, Anil Kumar Sao

School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi, India

Received 28 August 2017; received in revised form 4 April 2018; accepted 4 May 2018
Available online xxx

Abstract

In this paper, we have explored the framework of compressed sensing (CS) and sparse representation (SR) to reduce the footprint of unit selection based speech synthesis (USS) system. In the CS based framework, footprint reduction is achieved by storing either CS measurements or signs of CS measurements, instead of storing the raw speech waveforms. For efficient reconstruction using CS measurements, the speech signal should have a sparse representation over a predefined basis/dictionary. Hence, in this work, we have also studied the effectiveness of sparse representation for compressing the speech waveform. The experimental results are demonstrated using an analytical dictionary (DCT matrix), and several learned dictionaries, derived using K-singular value decomposition (KSVD), method of optimal directions (MOD), greedy adaptive dictionary (GAD) and principal component analysis (PCA) algorithms. To further increase compression in SR based framework of footprint reduction, the significant coefficients of sparse vector are selected adaptively, based on the type of speech segment (e.g., voiced, unvoiced etc.). Experimental studies on two different Indian languages suggest that CS/SR based footprint reduction methods can be used as an alternative to existing compression methods employed in USS system.

Keywords: Sparse representation; Speech synthesis; Dictionary learning; Compressed sensing

1. Introduction

Speech synthesis systems generates acoustic waveform from the given text, and are built to process text either for limited or unlimited domain. Limited domain systems are built specifically for an application with a finite vocabulary (set of words and/or sentences) to synthesize, such as travel information systems, weather forecasts etc. (Black and Lenzo, 2000). On the contrary, speech synthesis system, for an unlimited domain, is a generic system which can synthesize speech for any text corresponding to the language under consideration.

The available approaches for speech synthesis can be categorized into three classes: unit selection based (Hunt and Black, 1996; Sagisaka, 1988; Vepa and King, 2006), statistical parametric based (Black et al., 2007; Zen et al., 2009; Tokuda et al., 2013) and articulatory modeling based speech synthesis (Toutios and Narayanan, 2013; Sondhi and

☆ This paper has been recommended for acceptance by Roger Moore.

* Corresponding author.

E-mail address: pulkit_s@students.iitmandi.ac.in (P. Sharma), vinayak_abrol@students.iitmandi.ac.in (V. Abrol), niveditathakur16@gmail.com (V. Abrol), anil@iitmandi.ac.in (A.K. Sao).

https://doi.org/10.1016/j.csl.2018.05.003
0885-2308/ 2018 Elsevier Ltd. All rights reserved.

Schroeter, 1987; Greenwood, 1997). In the unit selection based speech synthesis (USS) system, appropriate speech units from a pre-recorded database are selected and concatenated on the basis of the target and the concatenation cost, to synthesize a speech waveform (Hunt and Black, 1996). In statistical parametric speech synthesis (SPS), speech is synthesized using parametric models derived from similar sounding speech units (Zen et al., 2009). Here, generative models e.g., hidden Markov model (HMM) (Zen et al., 2009) and deep neural networks (DNN) (Zen et al., 2013) are used to model spectral and excitation parameters of speech units (on a frame by frame basis). During synthesis, these models are used to estimate the parameters corresponding to different speech units in the text, which in turn are used to synthesize speech. In articulatory speech synthesis (ASS), position of the speech articulators e.g., lips, jaw, tongue etc. are used to approximate the vocal tract (VT) shape, and air flow through VT representation is simulated to synthesize speech (Toutios and Narayanan, 2013; Sondhi and Schroeter, 1987; Greenwood, 1997; Qinsheng et al., 2011).

It has been shown that USS systems produce better quality of the synthesized speech as compared to its contemporary approaches (Sagisaka, 1988; Vepa and King, 2006). The quality of the synthesized speech in USS system improves as the number of speech units (under different contexts) in the pre-recorded database increases (King and Karaiskos, 2011). However, the improvement in speech quality comes at the expense of large storage requirements, hence it is not possible to use such system in low resource platforms e.g., mobile phone (Gruber et al., 2014). This paper focus on reducing the footprint of speech corpus stored in USS system using the compressed sensing (CS) and sparse representation (SR) based signal processing, which have been explored extensively for image compression (Bryt and Elad, 2008; Skretting and Engan, 2011).

Existing works in the literature had applied CS and SR based signal processing for various speech related tasks. For instance, work in Sharma et al. (2015a) used SR based on PCA based dictionaries for tasks in speech recognition, Giacobello et al. (2014) used SR derived from LP based dictionary for speech coding, Jafari and Plumbley (2011) proposed a greedy dictionary for efficient speech signal reconstruction using SR, Abrol et al. (2015) proposed to use CS with LP based dictionary for voiced/nonvoiced detection. Leveraging the recent advancements in CS and SR based signal processing, we propose to use (i) compressed measurements obtained from speech signal (denoted as FRCS), (ii) 1-bit measurements obtained from sparse vector corresponding to speech signal (denoted as FRCS1), and (iii) significant coefficients of sparse vector (denoted as FRSV), for compression of speech corpus in USS system. In FRCS, compressed measurements obtained using a measurement matrix corresponding to a speech frame are stored. Each measurement obtained in FRCS is quantized using a finite number of bits. For further compression, we explore the extreme quantization where only one bit is used to store sign corresponding to each measurement, known as 1-bit CS (Boufounos and Baraniuk, 2008). In both FRCS and FRCS1, SR is initially estimated (given a dictionary), which is then used to synthesize speech. On the contrary, in FRSV, only the significant coefficients of the estimated sparse vector are stored.

In order to capture the subtle variations present in the speech data, multiple dictionaries are preferred over a single overcomplete dictionary (Tosic and Frossard, 2011; Sharma et al., 2017). The number of learned dictionaries is governed by the choice of unit e.g., words, syllables, phonemes etc. Choosing a bigger unit such as word will require a large number of dictionaries to be learned and stored. Therefore, we have chosen the smallest sound units i.e., phoneme which generally ranges from 30 to 60 in a given language (Rabiner and Schafer, 2010). Several dictionaries learning approaches proposed in the literature, namely greedy adaptive dictionary (GAD) (Jafari and Plumbley, 2011), K-singular value decomposition (KSVD) (Aharon et al., 2006), method of optimal directions (MOD) (Engan et al., 1999) and principal component analysis (PCA) (Dong et al., 2011) are explored to demonstrate the experimental results. Since, the learned dictionary requires extra storage space, the compression performance is also studied using an analytical dictionary (discrete cosine transform (DCT) matrix), which does not require storage space and can be generated during synthesis. In addition, our approach exploits the fact that there is a significant glottal activity (i.e., the vibration of vocal folds) during the production of voiced speech, while the same is not true for unvoiced speech (Abrol et al., 2015; Ananthapadmanabha and Yegnanarayana, 1979). Hence, in FRSV, different number of significant coefficients of the sparse vector are stored for different speech regions (e.g., voiced, unvoiced etc.), which lead to further reduction in required storage space.

1.1. Related work & contributions

Existing methods to reduce the size of speech corpus can be categorized into two categories: (i) compression based approaches (Black and Lenzo, 2001; Chazan et al., 2005; Strecha et al., 2007; Nurminen et al., 2013), and (ii)

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

pruning based approaches (Gruber et al., 2014; Tsiakoulis et al., 2008; Hanzlíček et al., 2013). In compression based approaches, the speech corpus is stored in an encoded form, and during synthesis a decoder is used to synthesize speech signal from its coded form. On the contrary, in pruning based approaches, the size of speech corpus is reduced by removing specific instances of different linguistic segments (based on appropriate criteria) from the speech corpus. Such methods are implemented either using a top-down or a bottom-up approach. In top-down approaches, rarely selected units are removed after analyzing the speech units usage in synthesized speech (Gruber et al., 2014). However, in bottom-up approaches the training speech corpus is analyzed, using a speech units similarity measure (Black and Taylor, 1997). In such approaches, various occurrences of same speech units are clustered according to an acoustic similarity measure. Spurious and overly common units in the database are removed by using the distance of the units from the cluster center (Black and Taylor, 1997). In pruning based methods, higher compression can be achieved by removing more number of units, however it is not a good choice in unlimited domain speech synthesis. Thus, the compression based approaches for reducing the size of speech corpus become more suitable for unlimited domain speech synthesis, and in this work CS/SR concepts are explored for the same. Moreover, compression based methods can be employed over the pruned database to further reduce the footprint of USS systems.

Various compression based methods are proposed in the literature to compress the size of acoustic inventory in USS system (Black and Lenzo, 2001; Chazan et al., 2005; Strecha et al., 2007). A method based on a harmonic sinusoidal speech representation comprising of different amplitude and phase parameterization is proposed in Chazan et al. (2005). Another popular low footprint USS system, Flite, stores linear prediction (LP) coefficients and residual error for every speech frame of all the utterances in the speech corpus (Black and Lenzo, 2001). Alternatively, line cepstral quefrencies (LCQ) derived after applying the line spectral frequencies transformation to the cepstrum are also used for USS system footprint reduction (Strecha et al., 2007). Most of these compression based approaches are motivated by LP based speech coding methods. In recent years, SR based signal processing is also proposed for LP analysis (Giacobello et al., 2014; 2010). However, these methods may suffer from the problem of unstable LP filters which leads to degradation in speech quality (Giacobello et al., 2014). Moreover, code book based methods to encode LP coefficients also lead to degradation in speech quality (Sreenivas and Kleijn, 2009). Further residual or excitation is not exactly sparse and discarding least significant coefficients affects the quality. The works in Giacobello et al. (2014, 2010) are proposed for speech coding, and are not specifically used to compress speech database in USS system. On the contrary, this work is not focused on speech coding, instead it focuses on compressing the speech corpus to be stored in USS systems. In general, speech coding methods are proposed to efficiently compress unseen speech data. However, in USS systems, the speech data to be compressed is available beforehand, which can be analyzed for efficient compression, and in this work, CS/SR based methods are explored for the same. These methods use dictionaries learned for a set of speech frames (corresponding to a phoneme), which can capture structures and patterns (in the training data) more efficiently. Hence, sparsest representations can be estimated for training data, which in turn results in lesser footprint of speech corpus in USS systems. In addition, the behavior and characteristics of the sparse vector are also exploited in demonstrating its effectiveness to compress the speech database in USS system.

This paper is an extension of our existing work published in Sharma et al. (2015b), and the highlights of this work are: (i) compression using CS measurements and 1-bit CS (extreme quantization) to reduce the speech corpus footprint, (ii) results are demonstrated using learned dictionaries for individual speech units (phonemes), (iii) exploiting the behavior of SR corresponding to different speech regions for efficient compression, and (iv) extensive experimentation to demonstrate the effectiveness of CS/SR based footprint reduction methods.

The rest of the paper is organized as follows: Section 2 gives an overview of compressed sensing and sparse representation for speech signals. Section 3 provides the detailed description of the proposed methods i.e., FRCS, FRCS1 and FRSV for footprint reduction of USS system. Experimental observations in terms of memory requirements and quality of synthesized speech for the proposed methods along with their comparison to existing methods is provided in Section 4. Finally, the paper is summarized in Section 5.

2. Overview of compressed sensing and sparse representation for speech signals

The data acquired using artificial sensors (e.g., microphone for speech) resides in a high dimensional signal space. However, the captured data is highly redundant as the relevant information about the underlying processes is generally of reduced dimensionality as compared to the recorded data sets. This phenomenon can be exploited to estimate efficient representations for natural signals (Tosic and Frossard, 2011). CS/SR based signal processing provides an effective way to estimate such representations, where the observations can be described by a subset of atoms from a dictionary (Bryt and Elad, 2008).

2.1. CS/SR based signal processing

The CS framework provides an efficient way of signal reconstruction via recovery of its SR from few measurements (Donoho, 2006). Here, a frame of speech signal $\mathbf{s}^i \in \mathbb{R}^n$ (sparse in domain $\Psi \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{l}}$, and $l = n$ for complete dictionary) is expressed as:

$$\mathbf{y}^i = \Phi\mathbf{s}^i = \Phi\Psi\boldsymbol{\alpha}^i = \mathbf{A}\boldsymbol{\alpha}^i, \tag{1}$$

where, $\Phi \in \mathbb{R}^{m\times n}$ $(m \ll n \le l)$ is sensing/measurement matrix and $\mathbf{y}^i \in \mathbb{R}^m$ denotes measurement vector corresponding to speech frame $\mathbf{s}^i$. Assuming that the speech frame $\mathbf{s}^i$ is k sparse in a dictionary $\boldsymbol{\Psi}$ (i.e., it is well approximated using k $(k \ll n)$ atoms of Ψ), $\mathbf{y}^i$ can be used to estimate the sparse vector α^i by solving the following optimization problem:

$$\boldsymbol{\alpha}^i_{cs} = \text{argmin}_{\boldsymbol{\alpha}^i} \ \| \mathbf{y}^i - \mathbf{A}\boldsymbol{\alpha}^i \|_2^2 \quad \text{s.t.} \quad \| \boldsymbol{\alpha}^i \|_0 \le k, \tag{2}$$

where $\| \ \|_0$ is ℓ_0-norm, a sparsity promoting function (Elad, 2010). Eq. (2) can be solved using a greedy approach such as orthogonal matching pursuit (OMP) (Rubinstein et al., CS Technion, 2008). Under appropriate constraints e.g., sparsity and restricted isometry prop erty (RIP), the reconstructed signal $\mathbf{s}^i_{cs} = \Psi\boldsymbol{\alpha}^i_{cs}$ will be good estimate of original signal $\mathbf{s}^i$ (Candes, 2008).

Since $m \ll n$, measurement vector $\mathbf{y}^i$ can be used to compress the speech frame $\mathbf{s}^i$. However, it requires $\mathbf{s}^i$ to be k sparse in the dictionary Ψ. Thus, alternatively $\mathbf{s}^i$ can also be reconstructed (with tolerable error) using k significant coefficients of sparse vector α. This is known as SR based signal processing, such that the signal can be reconstructed as $\mathbf{s}^i_s = \Psi\boldsymbol{\alpha}^i_s$, where, $\boldsymbol{\alpha}^i_s$ is estimated as:

$$\boldsymbol{\alpha}^i_s = \underset{\boldsymbol{\alpha}^i}{\text{argmin}} \ \| \mathbf{s}^i - \Psi\boldsymbol{\alpha}^i \|_2^2 \quad \text{s.t.} \quad \| \boldsymbol{\alpha}^i \|_0 \le k. \tag{3}$$

The sparse vector can also be estimated using an alternate optimization function defined as:

$$\boldsymbol{\alpha}^i_s = \underset{\boldsymbol{\alpha}^i}{\text{argmin}} \ \| \boldsymbol{\alpha}^i \|_0 \quad \text{s.t.} \quad \| \mathbf{s}^i - \Psi\boldsymbol{\alpha}^i \|_2^2 \le \epsilon, \tag{4}$$

where ϵ is a small error term. Both CS and SR based processing of signals are very much influenced by the choice of dictionary, which can either be analytical or learned. Analytical dictionaries feature a fast implementation but cannot model all the complexities in the natural signals such as speech. On the contrary, the learned dictionaries are derived from the data itself, thus better fit the training data and can effectively model the variations present in the data (Tosic and Frossard, 2011).

2.2. 1-bit CS based signal processing

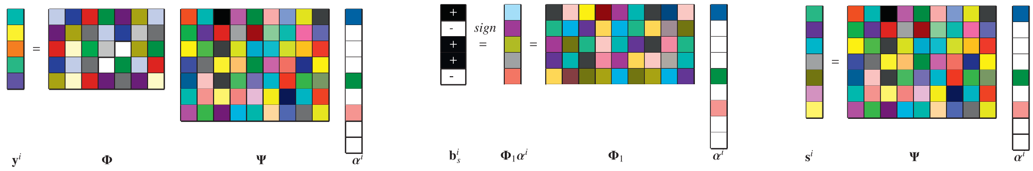
Depending on the floating point precision, each non-zero value of CS measurement and sparse vector require multiple bits to store. However, in recently proposed 1-bit CS concept, only the signs of the measurements are retained using 1-bit per measurement (Boufounos and Baraniuk, 2008; Jacques et al., 2013). It enables good reconstruction provided the vector acquired using the linear measurement system is highly sparse (Boufounos and Baraniuk, 2008). To this aim, the measurements corresponding to the sparse vector $\boldsymbol{\alpha}^i_s \in \mathbb{R}^l$ are acquired as:

$$\mathbf{b}^i_s \ = \ C(\boldsymbol{\alpha}^i_s) \ := \ sign(\Phi_1\boldsymbol{\alpha}^i_s), \tag{5}$$

where $\Phi_1 \in \mathbb{R}^{m_1\times l}$ represents measurement matrix, and $m_1 \ll l$[1]. $C(.)$ is a measurement operator and represents the mapping from original $\mathbb{R}^l$ space to the Boolean cube $\mathbb{B}^{m_1} := \{-1, 1\}^{m_1}$ (Boufounos and Baraniuk, 2008; Jacques et al., 2013). During quantization, the scale (magnitude) of the original signal is not preserved. Thus, one

[1] Measurement matrix used in 1-bit CS ($\Phi_1 \in \mathbb{R}^{m_1\times l}$) is different from the measurement matrix used to derive CS measurements ($\Phi \in \mathbb{R}^{m\times n}$).

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003



(a) FRCS (b) FRCS1 (c) FRSV

Fig. 1. Illustration of proposed footprint reduction methods. Compression is achieved by storing (a) reduced number of measurements (FRCS), (b) sign of measurements corresponding to estimated sparse vector (FRCS1), and (c) significant coefficients of estimated sparse vector (FRSV).

normalization constraint is applied by limiting the search space for solution of sparse signals on unit hyper-sphere (to make optimization tractable) (Laska and Baraniuk, 2012). Mathematically, this can be written as $\boldsymbol{\alpha}_s^i \in \sum_k * := \{\boldsymbol{\alpha}_s^i \in H^{l-1} : \parallel \boldsymbol{\alpha}_s^i \parallel_0 \leq k\}$, where $H^{l-1} := \{\boldsymbol{\alpha}_s^i \in \mathbb{R}^l : \parallel \boldsymbol{\alpha}_s^i \parallel_2 = 1\}$ is the unit hyper-sphere of dimension l (Boufounos and Baraniuk, 2008).

A non-linear decoder can be used to estimate of sparse vector from signs of CS measurements such that the solution $\boldsymbol{\alpha}_{1cs}^i$ is consistent. It means signs of the estimated sparse vector's ($\boldsymbol{\alpha}_{1cs}^i$) measurements are similar to that of original sparse vector's ($\boldsymbol{\alpha}_s^i$) measurements. Thus, the desired sparsest signal, consistent with the measurements, and lying on the unit hyper-sphere can be expressed as a solution to the following optimization problem (Jacques et al., 2013):

$$\boldsymbol{\alpha}_{1cs}^i = \underset{\boldsymbol{\alpha}_s^i \in H^{l-1}}{\operatorname{argmin}} \parallel \boldsymbol{\alpha}_s^i \parallel_0 \ \text{s.t.} \ \mathbf{b}_s^i := sign(\Phi_1 \boldsymbol{\alpha}_s^i). \tag{6}$$

The optimization function in Eq. (6) is computationally intractable, however, approximate solution can be obtained using a binary iterative hard thresholding (BIHT) based greedy method as proposed in Jacques et al. (2013).

3. Proposed approach to reduce the footprint of speech database in USS systems

In this work, footprint reduction in USS system is achieved using three methods namely, FRCS, FRCS1 and FRSV, as illustrated in Fig. 1 (a), (b) and (c), respectively. In FRCS, a low dimensional measurement vector ($\mathbf{y}^i$) corresponding to each speech frame is stored. On the contrary, only one bit per measurement is used to store sign of measurements in FRCS1. In both FRCS and FRCS1, the speech is synthesized using the sparse vector estimated from the measurements and signs of measurements, respectively. However, the significant coefficients of the estimated sparse vector (corresponding to each speech frame) are used for footprint reduction in FRSV. The sparsity of speech signal is generally unknown and varies for different speech frames (e.g., voiced (V), unvoiced (U) and transition (T)[2]). Hence, footprint reduction is achieved by storing varying number of significant coefficients (along with index locations) for different speech frames.

In this work, USS systems are built using phoneme as a unit, and thus a dictionary is learned for each phoneme. Pre-recorded speech data is labeled at phoneme level, and matrix $\mathbf{S}^{ji} = [\mathbf{s}^{j1}, \mathbf{s}^{j2}, \ldots, \mathbf{s}^{jn_j}]$ contains all the speech frames of jth, $j = 1, 2, \ldots, n_p$ phoneme in the pre-recorded speech corpus as its columns; n_p is the total number of phonemes in the database; i, $i = 1, 2, \ldots, n_j$ represents the frame index; n_j correspond to total number of speech frames for jth speech unit. Overcomplete dictionaries for individual phonemes are learned using speech data labeled at phoneme level, using the following optimization problem:

$$\operatorname{argmin}_{\Psi^j,\ \boldsymbol{\alpha}_s^{ji}} \sum_{i=1}^{n_j} \parallel \mathbf{s}^{ji} - \Psi^j \boldsymbol{\alpha}_s^{ji} \parallel_2^2 \ \text{s.t.} \ \parallel \boldsymbol{\alpha}_{\mathbf{s}}^{\mathbf{ji}} \parallel_0 \leq k. \tag{7}$$

It can also be solved using a relaxed form optimization defined as:

$$\operatorname{argmin}_{\Psi^j,\ \boldsymbol{\alpha}_s^{ji}} \sum_{i=1}^{n_j} \parallel \mathbf{s}^{ji} - \Psi^j \boldsymbol{\alpha}_s^{ji} \parallel_2^2 \ + \ \lambda \parallel \boldsymbol{\alpha}_{\mathbf{s}}^{\mathbf{ji}} \parallel_1, \tag{8}$$

[2] Transition frames represent frames consisting a transition form silence to speech and vice versa.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

where $\| \|_1$ is ℓ_1-norm (Elad, 2010) and λ is a small constant. These optimization functions can be solved using various existing dictionary learning (DL) algorithms such as greedy adaptive dictionary (GAD) (Jafari and Plumbley, 2011), K-singular value decomposition (KSVD) (Aharon et al., 2006), principal component analysis (PCA) (Dong et al., 2011) and method of optimal directions (MOD) (Engan et al., 1999).

3.1. FRCS

In this approach, measurement vectors obtained using a measurement matrix, for all the speech frames (in a speech utterance) are stored. During synthesis, speech is synthesized using the sparse vectors estimated from the measurements corresponding to frames of individual speech units. Number of measurements required to be stored are less than the number of speech samples required per frame and thus we have also investigated the performance of speech synthesis with varying the number of measurements.

Since the number of measurements is less than the dimension of sparse vector, the estimation of sparse vector from measurements is ill-conditioned. However, if support set i.e., the non-zero entries in the sparse vector $\boldsymbol{\alpha}_{cs}^{ji}$ is known, then for $m \geq k$ this ill-conditioned problem can be well conditioned with the following necessary and sufficient condition (Baraniuk, 2007)

$$1-\tau\leq\frac{\| \mathbf{A}^j\vartheta \|_2}{\| \vartheta \|_2}\leq 1+\tau, \tag{9}$$

where vector ϑ has the same support set as that of $\boldsymbol{\alpha}_{cs}^{ji}$ and τ is a small positive constant. In other words, the matrix $\mathbf{A}^j = \Phi\Psi^j$ must preserve the distances among these particular k-sparse vectors, but the location of these k significant coefficients (in $\boldsymbol{\alpha}_{cs}^{ji}$) is generally not known. However, there exist a sufficient condition to obtain a stable solution for both exact k-sparse and compressible signals, known as the RIP (Candes, 2008). Although certain probabilistic methods are used to generate matrices satisfying RIP with high probability, verifying RIP for a given sensing matrix is NP hard (Calderbank et al., 2010). Thus, a related condition, proxy to RIP known as incoherence can be used, which in gross sense means that the rows of sensing matrix $\boldsymbol{\Phi}$ cannot sparsely represents the columns of dictionary $\boldsymbol{\Psi}^j$ and vice versa (Baraniuk, 2007). Different methods to generate measurement matrices satisfying such properties exist in the literature (Malloy and Nowak, 2014; Sun et al., 2013), however in this work, we use Grassmannian frames (Elad, 2010) to construct the measurement matrix. Grassmannian matrices are equiangular tight frames such that each and every pair of columns in it have the smallest possible same angle. Matrices closer to Grassmannian matrix are shown to be best for sensing purposes, and hence the same is used in this work. Pseudo code of the FRCS method is given in Algorithm 1.

Algorithm 1. Proposed FRCS method to reduce footprint of USS system.

```
Training Phase:
    Input:
    Speech database labeled at phoneme level
    $\mathbf{S}^{ji} = [\mathbf{s}^{j1}, \mathbf{s}^{j2}, \ldots, \mathbf{s}^{jn_j}]$; $j = 1, 2, \ldots, n_p$; $i = 1, 2, \ldots, n_j$
    $\mathbf{\Phi} \in \mathbb{R}^{m\times n}$;   $k$
    $\mathbf{\Psi}^j \leftarrow \mathrm{DL}(\mathbf{S}^{ji}, k)$
    $\mathbf{y}^{ji} \leftarrow \mathbf{\Phi}\mathbf{s}^{ji}$.
    $\forall j, i$ Store $\mathbf{\Phi}$, $\mathbf{\Psi}^j$, $\mathbf{y}^{ji}$
Synthesis Phase:
    Input: Text to be synthesized
    Select appropriate units.
    $\forall$ selected $j$, $i$
    $\mathbf{A}^j \leftarrow \mathbf{\Phi}\mathbf{\Psi}^j$
    $\boldsymbol{\alpha}_{cs}^{ji} \leftarrow \mathrm{OMP}(\mathbf{y}^{ji}, \mathbf{A}^j, k)$
    $\mathbf{s}_{cs}^{ji} \leftarrow \mathbf{\Psi}^j\boldsymbol{\alpha}_{cs}^{ji}$.
        DL can either be KSVD, MOD, GAD or PCA
```

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

Algorithm 2. Proposed FRCS1 method to reduce footprint of USS system.

```
Training Phase:
    Input: Speech database labeled at phoneme level
    $\mathbf{S}^{ji} = [\mathbf{s}^{j1}, \mathbf{s}^{j2}, \ldots, \mathbf{s}^{jn_j}]$, $j = 1, 2, \ldots, n_p$; $i = 1, 2, \ldots, n_j$
    $\mathbf{\Phi}_1 \in \mathbb{R}^{m_1 \times l}$,   $k$
$\mathbf{\Psi}^j \leftarrow$ DL$(\mathbf{S}^{ji}, k)$
$\boldsymbol{\alpha}_s^{ji} \leftarrow$ OMP$(\mathbf{s}^{ji}, \mathbf{\Psi}^j, k)$
$\mathbf{b}_s^{ji} \leftarrow$ sign$(\mathbf{\Phi}_1 \boldsymbol{\alpha}_s^{ji})$
$\forall j,\ i$ Store $\mathbf{b}_s^{ji}, \mathbf{\Phi}_1, \mathbf{\Psi}^j$.

Synthesis Phase:
    Input: Text to be synthesized
Select appropriate units
$\forall$ selected $j,\ i$.
$\boldsymbol{\alpha}_{1cs}^{ji} \leftarrow$ BIHT$(\mathbf{b}_s^{ji}, \mathbf{\Phi}_1, k)$
$\mathbf{s}_{1cs}^{ji} \leftarrow \mathbf{\Psi}^j \boldsymbol{\alpha}_{1cs}^{ji}$.
        DL can either be KSVD, MOD, GAD or PCA
```

3.2. FRCS1

Depending on the bit-budget, multiple bits are required to store each CS measurement on the device. Hence, the total number of bits required to store CS measurements for each frame are constrained by the bit-budget. However, it has been shown in the literature that a signal can be approximately recovered from the sign of its measurements (known as 1-bit CS) (Boufounos and Baraniuk, 2008). In FRCS1, only the signs of the measurements are stored instead of storing raw samples of measurement vector. For effective reconstruction in 1-bit CS, the acquired vector should be highly sparse, and hence in the FRCS1 approach measurements are obtained from SR ($\boldsymbol{\alpha}_s^{ji}$). During synthesis, the sparse vector $\boldsymbol{\alpha}_{1cs}^{ji}$ is initially estimated from the signs of the measurements. The estimated sparse vector corresponding to speech frames of individual speech units is used to synthesize speech as $\mathbf{s}_{1cs}^{ji} = \Psi^j \boldsymbol{\alpha}_{1cs}^{ji}$. Pseudo code of the FRCS1 method is given in Algorithm 2.

3.3. FRSV

In both FRCS and FRCS1, speech is synthesized using the sparse vector estimated from measurements and measurement's signs, respectively. Alternatively sparse vector obtained for each speech frame using Eq. (3) can also be used to compress the footprint. Hence, in FRSV, sparse coding techniques are used to estimate sparse vectors ($\boldsymbol{\alpha}_s^{ji}$) for all the speech waveforms in the speech database (on a frame by frame basis). Due to sparse nature of $\boldsymbol{\alpha}_s^{ji}$, only a few significant coefficients can be used to reconstruct speech waveform, with tolerable error. However, in FRSV, location of k significant coefficients of SR is also needed, hence a total of $2k$ coefficients are stored. Thus, significant coefficients of sparse vector (obtained after solving a sparse solver) along with their index locations are stored for each speech waveform. During synthesis, individual speech units and thus the whole speech waveform is reconstructed using these significant coefficients of the sparse vector. Pseudo code of the FRSV method is given in Algorithm 3.

It is observed that the support set of sparse vector is not fixed for different frames but depends on the dictionary, and the type of speech frame (V, UV & T).

The sparse vectors estimated for three different regions of speech signal i.e., V, UV and T, are shown in Fig. 2.[3] It can be observed that there are huge variations in the magnitude of significant coefficients of sparse vector. The range of amplitude values (variance) of significant coefficients of the sparse vector is very high

[3] The speech signal used in this experiment is sampled at 16 kHz, and is processed at a frame rate of 25 ms. For more details about experimental settings see Section 4.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

Algorithm 3. Proposed FRSV method to reduce footprint of USS system.

```
Training Phase:
  Input: Speech database labeled at phoneme level
  $\mathbf{S}^{ji} = [\mathbf{s}^{j1}, \mathbf{s}^{j2}, \ldots, \mathbf{s}^{jn_j}]$. $j = 1, 2, \ldots, n_p$; $i = 1, 2, \ldots, n_j$, $k$
  $\boldsymbol{\Psi}^j \leftarrow \mathrm{DL}(\mathbf{S}^{ji}, k)$
  $\boldsymbol{\alpha}_s^{ji} \leftarrow \mathrm{OMP}(\mathbf{s}^{ji}, \boldsymbol{\Psi}^j, k)$
  $\forall j,\ i$ Store $\boldsymbol{\Psi}^j,\ \boldsymbol{\alpha}_s^{ji}$.

Synthesis Phase:
  Input: Text to be synthesized
  Select appropriate units
  $\forall$ selected $j,\ i$
  $\mathbf{s}_s^{ji} = \boldsymbol{\Psi}^j \boldsymbol{\alpha}_s^{ji}$
      DL can either be KSVD, MOD, GAD or PCA
```

for the voiced frame (Fig. 2 (d)). One possible reason for this may be the vibration of vocal folds while producing voiced speech (Rabiner and Schafer, 2010). The same does not happen during production of unvoiced speech, hence the range of amplitude values of the corresponding sparse vector is very low (see Fig. 2 (f)). This property of sparse vector is exploited recently in Abrol et al. (2015) for efficient voiced/non-voiced detection. The nature of the sparse vector corresponding to the transition speech frame lies between the voiced and the unvoiced speech frames. As an illustration, Fig. 2 shows that the speech frames reconstructed using sparse vector are similar to the original speech frames. The behavior of the sparse vectors obtained for different regions of the speech signal is consistent and can be observed in Fig. 3, which shows histogram of the sparse coefficients for 500 examples of each region (V, T and UV) obtained using both a learned KSVD (complete) and an analytical (DCT matrix) dictionary.

In SR based signal processing, the goal is to find a decomposition such that the obtained weights are sparse, meaning they should have probability densities highly peaked at zero with heavy tails (Hoyer, 2002). Fig. 3 shows that the SR obtained using learned dictionary (KSVD) have significant heavy tails as compared to the analytical dictionary (DCT matrix). Thus, KSVD dictionary results in better SR as compared to the DCT dictionary. In order to further verify the efficiency of KSVD dictionary, we reconstructed 500 different speech utterances with fixed reconstruction error (for each speech frame in a speech utterance) using the sparse vectors corresponding to different complete dictionaries.[4] The average sparsity for this fixed

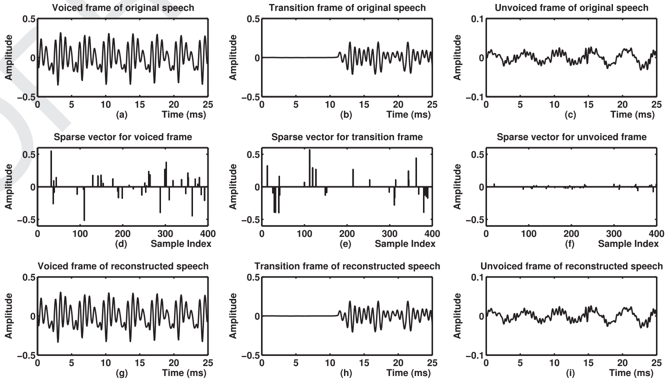


Fig. 2. (d), (e) and (f) show sparse vectors corresponding to voiced, transition and unvoiced frames of original speech shown in (a), (b) and (c), respectively. (g), (h) and (i) show voiced, transition and unvoiced frames reconstructed using SR. Dictionary used is complete KSVD.

[4] For this experiment optimization function in Eq. (4) is solved, with $\epsilon = 10^{-3}$.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

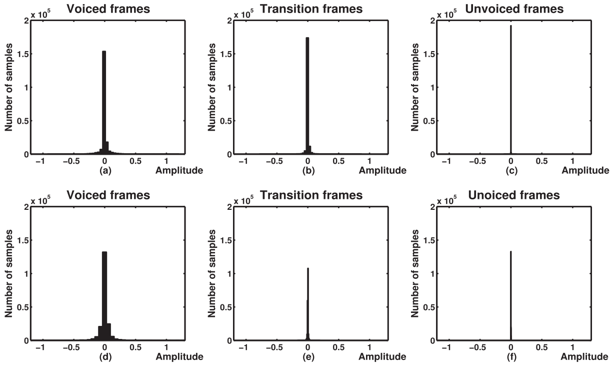


Fig. 3. Histogram of sparse coefficients: (a), (b) & (c) shows histogram of SR for V, T and UV frames, using DCT matrix as dictionary. (d), (e) & (f) shows histogram of SR for V, T and UV frames using complete KSVD dictionary. Total number of speech frames used for each histogram are 500.

Table 1
The average sparsity (in % of number of coefficients in 25 ms speech frame sampled at 16 kHz, rounded off to nearest integer) for reconstruction error of 10^{-3} averaged over all the frames of 500 speech utterances using different dictionaries.

Dictionary	Sparsity (%)
DCT	15
KSVD	9
GAD	13
PCA	11
MOD	12

reconstruction error (averaged over all the frames of 500 speech utterances) is shown in Table 1, which shows that KSVD dictionary results in efficient SR and hence can give maximum reduction in storage space. According to the variance of the amplitude of estimated sparse vector coefficients, the descending order arrangement of three different regions of speech signal is: V, T and UV. This behavior implies that varying number of significant coefficients can be stored for efficient compression of speech signals.

4. Experiments

This section describes the experimental studies which are divided into four parts: (i) synthesized speech quality using proposed footprint reduction methods is analyzed with varying number of measurements and sparsity, (ii) comparison of proposed footprint reduction methods with low footprint Flite USS system in terms of synthesized speech quality and memory requirements, (iii) comparison with standard speech coders in terms of mean opinion scores (MOS) and bit rates, and (iv) analysis of computational complexity of the proposed footprint reduction methods.

Speech database used in this work is studio recorded by professional female and male speaker, sampled at 16 kHz and is processed on short time frame basis where framing has been achieved by applying a hamming window of 25 ms with 50% overlap. During synthesis, the speech is synthesized using a standard overlap add method. Various DL algorithms based on KSVD, MOD, GAD and PCA are used to derive learned dictionaries, while DCT matrix is used as an analytical dictionary. For KSVD and MOD an overcomplete dictionary (of size $n \times 2n$ i.e., $l = 2n$) is learned, while other learned dictionaries are complete. For 1-bit CS, the reconstruction error scales as $O\left(\frac{k}{m_1}\log\left(\frac{m_1 l}{k}\right)\right)$ (Jacques et al., 2013), and hence for efficient reconstruction the signal under consideration should be highly sparse. Thus, in case of FRCS1, we have learned a highly overcomplete (of size $n \times 5\,n$ i.e., $l = 5\,n$) GAD, KSVD and MOD dictionary, while other dictionaries used are complete. Sensing matrix $\boldsymbol{\Phi}$ is constructed from

Grassmannian frames (Elad, 2010) as explained in Abrol et al. (2013), while a random Gaussian matrix is used as $\boldsymbol{\Phi}_1$. Orthogonal matching pursuit (OMP) (Rubinstein et al., CS Technion, 2008) is used to solve $\| \|_0$ with a fixed value of sparsity (k), except FRCS1, where binary iterative hard thresholding (BIHT) is used to estimate the sparse vector (Jacques et al., 2013). The percentage of significant coefficients (or sparsity k) used in this work is calculated with respect to the dimension of a speech frame.

In our experiments, both monolingual (Indian language) and bilingual (Indian language + Indian English) USS systems are built for Hindi and Rajasthani language. Data used for Hindi is: (i) male Hindi (Speaker 1) − 7 h (ii) male Hindi accented English (Speaker 1) − 6 h, (iii) female Hindi (Speaker 2) − 7 h (iv) female Hindi accented English (Speaker 2) − 5 h . The data used for Rajasthani is: (i) female Rajasthani (Speaker 3) − 9 h (ii) female Rajasthani accented English (Speaker 3) − 9 h, (iii) male Rajasthani (Speaker 4) − 9 h (iv) male Rajasthani accented English (Speaker 4) − h Both male and female data are used to build monolingual and bilingual (Hindi/Rajasthani + Hindi/Rajasthani accented English) USS system. Bilingual systems can synthesize text corresponding to both languages e.g., system built using Hindi and Hindi accented English, can synthesize both Hindi and English text. All the training data labeled at phoneme level is used to learn individual dictionaries for each phoneme. Thus, for bilingual systems, dictionaries are learned for all the phonemes of two languages. Total number of phonemes for Hindi, Rajasthani and Indian English are 59, 55 and 43, respectively. In our experiments, all USS systems are built using Festival speech synthesis engine with phoneme as a unit (Black and Taylor, 2010). Segmentation of speech corpus is done using a hybrid approach with group delay processing as described in Shanmugam and Murthy (2014). The actual footprint of the speech data used in this work is: male Hindi − 806.4 MB, male Hindi accented English − 689.04 MB, female Hindi − 810.6 MB, female Hindi accented English − 576.6 MB, male Rajasthani − 1048.1 MB, male Rajasthani accented English − 1047.2 MB, female Rajasthani − 1093.06 MB, female Rajasthani accented English − 1044.7 MB.

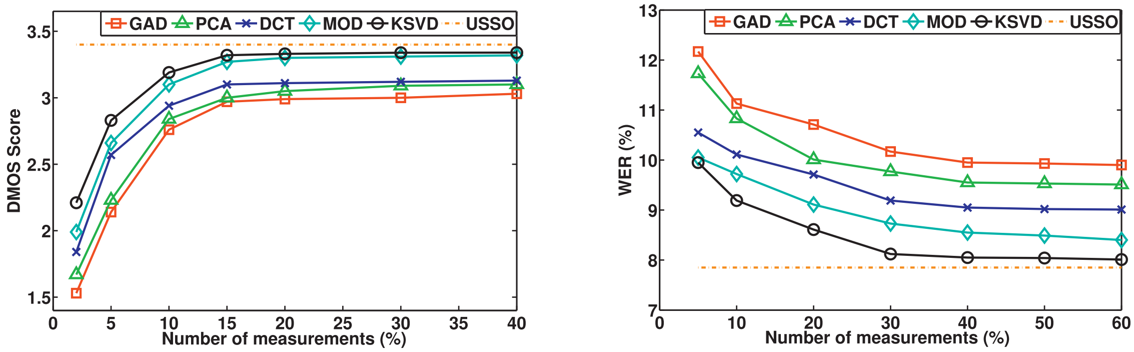
We have used degradation mean opinion scores (DMOS) (ITU-T, 2013; Goldberg and Riek, 2000) and word error rates (WER) (in %) (Pellegrini et al., 2012) to measure the quality of the synthesized speech for all USS systems. DMOS and WER, for each of USS system is calculated using 400 different speech utterances i.e., 20 different speech files are played in random order to 20 different native listeners.[5] In case of a DMOS test, each test utterance is preceded by the original reference utterance and the listener is asked to rate the degradation of the test utterance as compared to the reference utterance. The listener rates the reduction in quality on a scale from 1 to 5, with rate as: 1 - very annoying, 2 - annoying, 3 - slightly annoying, 4 - audible but not annoying and 5 - inaudible. The studio recorded speech utterances used to build the USS systems are used as reference utterances in the DMOS experiments. All the listening tests are performed with headphones in silent environment, and the speech files used for listening tests are not included in the training corpus. In addition, preference test (A/B test) on the synthesized speech using proposed footprint reduction methods is also conducted (Zen and Senior, 2014). In preference test, native speakers listen to two speech samples and indicate which one is more natural and the choices are system 1 (A), system 2 (B) or no preference (N/P). p-value in the preference test is the probability that the difference observed in the current study is due to chance. For preference test, pair of 100 different speech waveform pairs, synthesized using two USS systems in consideration are presented to 20 native listeners.

4.1. Synthesized speech quality

In this experiment, the quality of the synthesized speech is analyzed for proposed compression methods. In order to measure the usefulness of these methods, a baseline system (labeled as USSO) i.e., a USS system with no compression is built for male Hindi voice. DMOS and WER for this system are 3.4 and 7.85%, respectively and these metrics are considered as benchmark for measuring the naturalness and intelligibility of the synthesized speech using other approaches. DMOS and WER obtained for synthesized speech using FRCS method, with different number of measurements are shown in Fig. 4 (a) and (b), respectively. It indicates that 15% measurements of the speech signal can preserve the naturalness and intelligibility in the synthesized speech.

[5] All the 20 files played for a listener are distinct from all the files played to all other listeners.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003



(a)

(b)

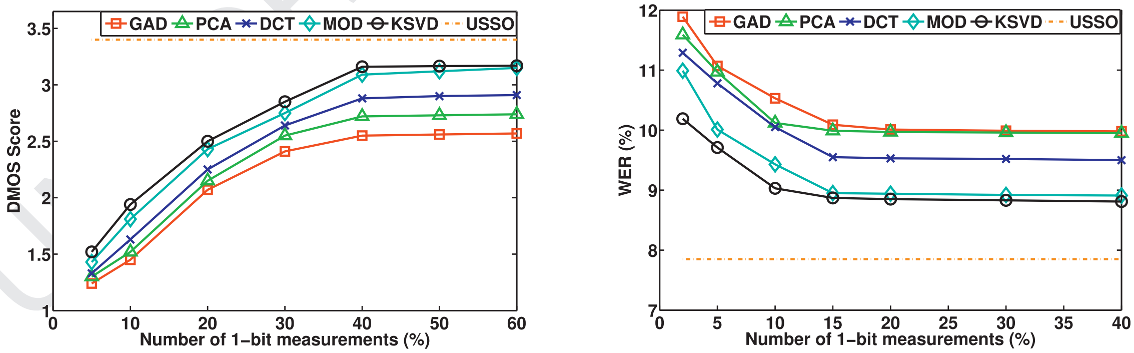
Fig. 4. DMOS and WER scores for reduced footprint male Hindi USS system using the FRCS method with varying number of measurements ($\mathbf{y}^i$) in (a) and (b), respectively.

DMOS and WER obtained for the synthesized speech from signs of the measurements FRCS1 is shown in Fig. 5 (a) and (b), respectively. Fig. 5 (a) indicates that the synthesized speech is not natural as compared to synthesized speech using FRCS.

DMOS and WER for the synthesized speech with varying sparsity k in the FRSV method are shown in Fig. 6 (a) and (b), respectively. It can be observed that only 10% significant coefficients of the sparse vector obtained using KSVD dictionary results in synthesized speech having quality similar to the original USS system (USSO). The improvement in the results is not significant for further increase in the number of significant coefficients of the sparse vector.

The nature of sparse vector estimated for different frames (V, UV and T) is not same (as evident from Figs. 2 and 3), hence we propose to use 10%, 5% and 2% significant coefficients of sparse vector for V, T and UV frames, respectively. The percentage of significant coefficients is computed experimentally and the resulting system is labeled as Var. The quality of the synthesized speech using 10% significant coefficient for all the frames and varying number of coefficients for different frames (Var) is shown in Table 2. It is evident from this table that the method employing varying number of coefficients for different frames performs similar to the method using 10% significant coefficients for each frame, however the former requires less storage space.

It has been observed that KSVD and MOD dictionaries performed better than GAD, possibly because GAD learns sparse atoms as a result of which obtaining SR is not possible. Similarly, being complete, the performance in case of PCA and DCT dictionaries is not at par with KSVD and MOD dictionaries. In all the proposed compression methods, the KSVD dictionary performs best among all the dictionaries used, hence the results now onwards are shown with KSVD dictionary only, unless and otherwise explicitly stated. The reason for best performance of KSVD is possibly because the weight update step is performed with a fixed sparsity (while learning KSVD dictionary), and thus

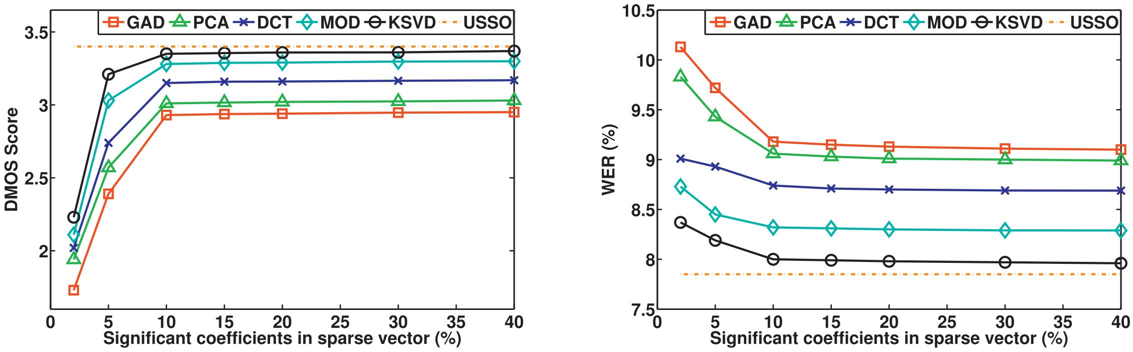


(a)

(b)

Fig. 5. DMOS and WER scores for reduced footprint male Hindi USS system using the FRCS1 method with varying number of 1-bit measurements ($\mathbf{b}_s$) in (a) and (b), respectively.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003



(a)

(b)

Fig. 6. DMOS and WER scores for reduced footprint male Hindi USS system using the FRSV method with varying sparsity (*k*) in (a) and (b), respectively.

sparsest representations can be obtained for the training data. Henceforth, the low footprint USS systems are built with these settings: (i) in FRCS 15% of measurements, (ii) in FRCS1 40% of measurements, and (iii) in FRSV varying number of significant coefficients (Var) are stored. Different percentage of measurements/coefficients for different methods are selected experimentally, which result in the best quality of the synthesized speech (see Figs. 4–6), and also results in similar storage (as discussed in Section 4.3). Preference scores for the proposed compression methods in terms of overall speech quality are shown in Table 3. Preference test indicates that the FRSV method is most preferred system followed by FRCS and FRCS1, respectively.

4.2. Comparison with Flite

The proposed footprint reduction methods are compared with the existing low footprint USS system Flite. DMOS and WER scores for proposed footprint reduction methods along with their comparison to Flite for male and female Hindi USS systems are shown in Tables 4 and 5, respectively. It can be observed that the synthesized speech quality using the FRSV footprint reduction method is better than the Flite. The preference scores for proposed reduced footprint USS systems along with their comparison to Flite are shown in Table 6. Results of preference tests support the claim that the synthesized speech using FRSV preserve naturalness efficiently as compared to Flite.

Table 2
DMOS and WER for USS systems build with the FRSV compression scheme. *A* and *B* represents compression achieved using 10% and varying number of coefficients (Var) of sparse vector corresponding to KSVD dictionary.

	Male				Female			
	Hindi		Bilingual		Hindi		Bilingual	
	A	B	A	B	A	B	A	B
DMOS	3.35	3.34	3.32	3.3	3.29	3.24	3.25	3.2
WER (%)	8.0	8.01	8.31	8.35	8.91	8.93	9.8	9.92

Table 3
Preference scores (in %) for proposed footprint reduction methods.

FRSV	FRCS	FRCS1	N/P	*p*-value
41.5	19.3	–	39.2	$< 10^{-3}$
–	**20.7**	15.2	64.1	$< 10^{-2}$
48.7	–	13.7	37.6	$< 10^{-3}$

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

Table 4
DMOS and WER scores for proposed footprint reduction methods along with their comparison to original USS system (USSO) and Flite for male Hindi and bilingual USS systems.

	Male Hindi					Male bilingual				
	USSO	FRCS	FRCS1	FRSV	Flite	USSO	FRCS	FRCS1	FRSV	Flite
DMOS	3.4	3.32	3.16	3.34	3.29	3.38	3.22	3.14	3.3	3.24
WER(%)	7.85	8.05	8.87	8.01	8.27	7.67	8.47	9.58	8.35	8.93

Table 5
DMOS and WER scores for proposed footprint reduction methods along with their comparison to original USS system (USSO) and Flite for female Hindi and bilingual USS systems.

	Female Hindi					Female bilingual				
	USSO	FRCS	FRCS1	FRSV	Flite	USSO	FRCS	FRCS1	FRSV	Flite
DMOS	3.32	3.18	3.1	3.24	3.17	3.29	3.16	3.09	3.2	3.17
WER(%)	8.17	9.71	8.97	8.93	9.13	9.37	10.73	10.13	9.92	10.85

Table 6
Preference scores (in %) for FRSV and Flite in male Hindi (M-H), male bilingual (M-B), female Hindi (F-H) and female bilingual (F-B) USS systems.

USS system	FRSV	Flite	N/P	*p*-value
M-H	**39.9**	20.7	39.4	$< 10^{-3}$
M-B	**40.6**	21.3	38.1	$< 10^{-3}$
F-H	**40.7**	21.5	37.8	$< 10^{-2}$
F-B	**39.2**	22.6	38.2	$< 10^{-2}$

The performance of the proposed footprint reduction methods is also analyzed for Rajasthani language. Total number of measurements and significant coefficients of sparse vector stored per frame for low footprint Rajasthani USS system are same as for the Hindi USS system. The comparison of the proposed low footprint male and female Rajasthani USS systems with USS system without compression (USSO) and Flite is shown in Tables 7 and 8, respectively. Similarly, preference scores for the proposed low footprint Rajasthani USS system and Flite are shown in Table 9. These results indicate that the synthesized speech using the FRSV method results in DMOS and WER better than the state-of-the-art low footprint Flite system.

4.3. Analysis on memory requirements

The performance in terms of requirement of memory, for proposed footprint reduction methods is compared with the method used in Flite. For comparison, we have used the number of coefficients required to store measurements

Table 7
DMOS and WER scores for proposed footprint reduction methods along with their comparison to original USS system (USSO) and Flite for male Rajasthani and bilingual USS systems.

	Male Rajasthani					Male bilingual				
	USSO	FRCS	FRCS1	FRSV	Flite	USSO	FRCS	FRCS1	FRSV	Flite
DMOS	3.25	3.11	2.73	3.19	3.12	3.19	3.01	2.71	3.11	3.05
WER(%)	8.53	9.14	9.98	9.05	9.39	8.78	9.5	10.03	9.25	9.74

Table 8
DMOS and WER scores for proposed footprint reduction methods along with their comparison to original USS system (USSO) and Flite for female Rajasthani and bilingual USS systems.

	Female Rajasthani					Female bilingual				
	USSO	FRCS	FRCS1	FRSV	Flite	USSO	FRCS	FRCS1	FRSV	Flite
DMOS	3.15	3.01	2.69	3.09	3.03	3.1	2.97	2.68	3.04	2.99
WER(%)	8.73	9.26	9.94	9.15	9.41	9.14	10.13	10.58	9.61	10.05

Table 9
Preference scores (in %) for FRSV and Flite in male Rajasthani (M-R), male bilingual (M-B), female Rajasthani (F-R) and female bilingual (F-B) USS systems.

USS system	FRSV	Flite	N/P	*p*-value
M-R	**40.4**	21.9	37.7	$< 10^{-3}$
M-B	**39.6**	22.3	38.1	$< 10^{-2}$
F-R	**39.8**	22.7	37.5	$< 10^{-1}$
F-B	**40.6**	21.2	38.2	$< 10^{-3}$

in FRCS, sign of measurements in FRCS1, sparse vector in FRSV, and residual in Flite. The storage space required for the measurement matrix and/or dictionary in proposed methods and LP coefficients in case of Flite are not taken into consideration, as memory requirements for them will be similar. In this experiment, speech sampled at 16 kHz is processed with a frame size of 25 ms with 50% overlap, and hence raw speech requires 400 coefficients to be stored for each frame. However, for comparison purposes only the number of bits required to store each frame are reported in this section. Compression in FRCS is achieved by storing 15% of measurements ($\mathbf{y}^{ji}$), and hence $\left(\frac{400\times15}{100}\right) = 60$ coefficients are required to be stored for each frame. In case of FRCS1, signs of 40% measurements ($\mathbf{b}_s^{ji}$) are stored, and hence total number of signs required are $\left(\frac{400\times40}{100}\right) = 160$. Similarly, in FRSV, variable number of coefficients ($\boldsymbol{\alpha}_s^{ji}$) are used for each frame (labeled as Var). Hence, for a frame of 25 ms duration sampled at 16 kHz, proposed approach requires (on an average) $\frac{1}{3}\left[\left(\frac{400\times10}{100} + \frac{400\times5}{100} + \frac{400\times2}{100}\right)\right] = 22.66$ coefficients.[6] The significant coefficients of the sparse vector have varying location and hence index locations are also needed to be stored, therefore total coefficients required to be stored are $22.66 \times 2 = 45.3$. On the contrary, footprint reduction in Flite is achieved by storing 50 coefficients of residual per frame.

In this work, the sparse coefficient matrix is first represented in column compressed form. The values of the coefficients are encoded via linear quantization followed by entropy coding, and the locations of these coefficients via difference coding of the row indices followed by entropy coding (Horev et al., 2012). Similarly, the coefficients in FRCS are also encoded via linear quantization followed by entropy coding. It has been observed that very few bits (4−6) for the quantized values results in similar performance. Thus, in this work, 4 bits are used to store each coefficient on a computer, hence total number of bits required to store each frame are 240, 200 and 182 in FRCS, Flite and FRSV, respectively. On the contrary, in case of FRCS1 only signs of 160 coefficients are needed to be stored, thus total number of bits required are 160. In terms of percentage, the proposed FRCS, FRCS1 and FRSV requires 10%, 15% and 11.5% of the size required to store the uncompressed speech corpus, while Flite requires 12.5%. Thus, the FRCS1 and FRSV methods results in higher compression than Flite, with FRCS1 resulting in maximum compression. However, the FRCS method requires more memory as compared to Flite. These results suggest that the FRSV method can be used as an alternative to Flite.

4.4. Comparison with existing speech coding techniques

The proposed speech compression methods are also compared with the existing speech coders available in terms of bit rate and MOS score (ITU-T, 2013; Goldberg and Riek, 2000). The MOS is expressed as a single number

[6] Number of V/UV/T frames in a speech waveform may not be exactly equal, but in this example, for comparison we are assuming them to be equal.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

Table 10
Averaged MOS and bit rates for proposed methods and standard speech coders.

	Proposed methods			Standard speech coders (Goldberg and Riek, 2000)				
	FRSV	FRCS	FRCS1	LPC-10 (FS-1015)	MELP	CELP (FS-1016)	ACELP (G.723.1)	LD-CELP (G.728)
MOS	3.16	3.02	2.78	2.24	3.2	3.2	3.6	4
Bit rate (kbps)	4.7	4.8	3.2	2.4	4	4.8	5.3	16

ranging between 1 and 5, where 1- bad, 2- poor, 3 - fair, 4- good and 5- excellent. Since most of the speech coders are evaluated at 8 kHz, for a fair comparison with speech coders, speech signals are re-sampled at 8 kHz, and are processed at a frame rate of 25 ms without any overlap using a rectangular window. In addition, the speech waveform unseen in training data are used to evaluate the speech coding performance. Here, SR or CS vector is obtained for an unseen waveform with respect to a dictionary learned on the training data. For the case of FRSV, variable number of significant coefficients are used for each frame. The number of significant coefficients chosen are 10%, 7% and 5% for voiced, transition and unvoiced frames.[7] Thus, for a frame size of 25 ms sampled at 8 kHz average number of coefficients needed for each frame are $\frac{1}{3}\left(\frac{200\times10}{100}+\frac{200\times7}{100}+\frac{200\times5}{100}\right)=14.67$. Since, $\boldsymbol{\alpha}_s^{ji}$ is sparse in nature, the location (index) of the coefficients is also required to be stored, and hence total number of coefficients needed to be stored are $14.67\times2=29.34$. In this work, 4 bits are used to store the significant coefficients as discussed in Section 4.3. Hence, the bit rate for FRSV is $29.34\times4\times40=4693=4.7$ kbps (number of frames in a second are 40). On the similar lines we calculated bit rates for other compression methods proposed in this paper with 15% measurements $(\mathbf{y}^{ji})$ for FRCS and sign of 40% measurements $(\mathbf{b}_s^{ji})$ for FRCS1. Hence, the bit rates for FRCS and FRCS1 are $\left(\frac{200\times15}{100}\right)\times4\times40=4.8$ kbps and $\left(\frac{200\times40}{100}\right)\times40=3.2$ kbps, respectively. The comparison of proposed methods with existing speech coding methods in terms of MOS and bit rates is provided in Table 10.

The speech coding performance of the proposed methods is comparable to existing speech coders. In addition, our experiments reveal that the MOS for speech data not available during training, using FRSV is 3.16 (Table 10), while the same for speech data available during training (seen data) is 3.31. On the contrary, the average MOS for any speech signal is 3.2 using the MELP. The speech coders are designed for efficient compression of any speech signal. However, this is not the case for the proposed FRSV approach which give better compression for the training data. In case of USS systems, the speech units from the pre-recorded speech corpus are concatenated to synthesize speech. In other words, all the speech data to be compressed in USS systems is available during training (for dictionary learning in case of FRSV). Thus, use of proposed methods for footprint reduction is more suitable in the case of USS systems, where the speech data is available apriori.

4.5. Computational complexity

In the proposed approach, both dictionary and sparse vector are obtained during the system building phase and hence their computational complexity is not considered here. Computational complexity of selecting a unit is also same across all the systems under consideration. Hence, the bottleneck during synthesis is estimating speech signal from compressed representations i.e., measurements, measurement's sign or significant coefficients of sparse vector. Here, computational complexity (during the speech synthesis step), of the proposed methods can be arranged in order: FRCS1 > FRCS > FRSV. For FRSV, it involves multiplying the sparse vector corresponding to each frame with the respective dictionary and hence is least expensive, computationally. In FRCS, the computational complexity is more as optimization problem in Eq. (2) is required to be solved for each frame (during synthesis). On the contrary, FRCS1 is most expensive among the proposed approaches. The computational complexity (for each speech frame) of the proposed methods is summarized as:

(i) FRSV require matrix multiplication of the sparse vector $\boldsymbol{\alpha}_s^{ji}\in\mathbb{R}^l$ with the corresponding dictionary $\Psi^j\in\mathbb{R}^{n\times l}$. The sparse vector has only k non-zero coefficients, and hence the computational complexity required for this matrix multiplication is $O(nk)$.

[7] Number of significant coefficients chosen for speech sampled at 8 kHz are different than those for speech sampled at 16 kHz, and are obtained empirically.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

(ii) FRCS require the estimation of sparse vector $\boldsymbol{\alpha}_{cs}^{ji} \in \mathbb{R}^l$ from measurements $\mathbf{y} \in \mathbf{R}^m$. Orthogonal matching pursuit (OMP) is used to obtain this sparse vector and its computational complexity scales as $O(\delta m^2)$ per frame (Mailhe et al., 2009), where $\delta = \frac{l}{m}$ is the overcompleteness factor. After this step, speech is synthesized by multiplying the sparse vector with the respective dictionary, which has computational complexity of $O(nk)$, where k is sparsity of sparse vector. Thus, total computational complexity of FRCS scales as $O(\delta m^2) + O(nk)$.

(iii) FRCS1 uses binary iterative hard thresholding (BIHT) algorithm to obtain the sparse vector from 1-bit measurements, and its computational complexity scales as $O(m_1 l)$. Final step in synthesis involves multiplication of the sparse vector with the respective dictionary with computational complexity of $O(nk)$. Hence, total computational complexity of FRCS1 scales as $O(m_1 l) + O(nk)$. Since $m < m_1 < n < l$, the computational complexity for FRCS1 is the highest.

5. Summary

This work is focused on the feasibility of CS and SR to reduce the footprint of USS system. For efficient CS/SR based speech processing, the dictionary should be able to model different variations in the signal under various contexts. Hence, the proposed method employs dictionaries learned for individual phonemes and compression is achieved by storing either CS measurements, sign of CS measurements or significant coefficients of SR. The behavior of the SR obtained for different speech regions (voiced, unvoiced and transition) is also exploited further in reducing the memory requirements, without degradation in the perceivable speech quality. Experimental results demonstrate the effectiveness of these methods and confirms that the performance in terms of speech quality and storage requirements is comparable or better than the existing state-of-the-art low footprint Flite USS system. However, the proposed FRCS and FRCS1 methods are computationally expensive and hence efficient algorithms are needed to address this drawback. Furthermore, efficient DL algorithms are needed to obtain sparsest representation, such that the quality of the synthesized speech is enhanced along with reduction in storage requirements.

References

Abrol, V., Sharma, P., Sao, A.K., 2013. Speech enhancement using compressed sensing. In: Proceedings of the Interspeech, pp. 3274–3278.

Abrol, V., Sharma, P., Sao, A.K., 2015. Voiced/nonvoiced detection in compressively sensed speech signals. Speech Commun. 72, 194–207.

Aharon, M., Elad, M., Bruckstein, A., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. 54 (11), 4311–4322.

Ananthapadmanabha, T., Yegnanarayana, B., 1979. Epoch extraction from linear prediction residual for identification of closed glottis interval. IEEE Trans. Acoust. Speech Signal Process. 27 (4), 309–319.

Baraniuk, R.G., 2007. Compressive sensing lecture notes. IEEE Signal Process. Mag. 24 (4), 118–121.

Black, A.W., Lenzo, K.A., 2000. Limited domain synthesis. In: Proceedings of the International Conference of Spoken Language Processing (ICSLP), pp. 411–414.

Black, A.W., Lenzo, K.A., 2001. Flite: a small fast run-time synthesis engine. In: Proceedings of the Fourth ISCA Tutorial and Research Workshop on Speech Synthesis.

Black, A.W., Taylor, P., 1997. Automatically clustering similar units for unit selection in speech synthesis. In: Proceedings of the Eurospeech, pp. 601–604.

Black, A.W., Taylor, P., 2010. The Festival Speech Synthesis System, Version 2.1. The University of Edinburgh.

Black, A.W., Zen, H., Tokuda, K., 2007. Statistical parametric speech synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1229–1232.

Boufounos, P.T., Baraniuk, R.G., 2008. 1-bit compressive sensing. In: Proceedings of the Conference Information Science and Systems (CISS), pp. 16–21.

Bryt, O., Elad, M., 2008. Compression of facial images using the K-SVD algorithm. J. Vis. Commun. Image Represent. 19 (4), 270–282.

Calderbank, R., Howard, S., Jafarpour, S., 2010. Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property. IEEE J. Sel. Topics Signal Process. 4 (2), 358–374.

Candes, E.J., 2008. The restricted isometry property and its implications for compressed sensing. Comput. Rendus Math. 346 (9), 589–592.

Chazan, D., Hoory, R., Kons, Z., Sagi, A., Shechtman, S., Sorin, A., 2005. Small footprint concatenative text-to-speech synthesis system using complex spectral envelope modeling. In: Proceedings of the Interspeech, pp. 2569–2572.

Dong, W., Zhang, D., Shi, G., Wu, X., 2011. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. IEEE Trans. Image Process. 20 (7), 1838–1857.

Donoho, D.L., 2006. Compressed sensing. IEEE Trans. Inf. Theory 52 (4), 1289–1306.

Elad, M., 2010. Sparse and Redundant Representations − From Theory to Applications in Signal and Image Processing. Springer.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003

Engan, K., Aase, S.O., Husoy, J.H., 1999. Method of optimal directions for frame design. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2443–2446.

Giacobello, D., Christensen, M.G., Jensen, T.L., Murthi, M.N., Jensen, S.H., Moonen, M., 2014. Stable 1-norm error minimization based linear predictors for speech modeling. IEEE/ACM Trans. Audio Speech Lang. Process. 22 (5), 912–922.

Giacobello, D., Christensen, M.G., Murthi, M.N., Jensen, S.H., Moonen, M., 2010. Retrieving sparse patterns using a compressed sensing framework: applications to speech coding based on sparse linear prediction. IEEE Signal Process. Lett. 17 (1), 103–106.

Goldberg, R., Riek, L., 2000. A Practical Handbook of Speech Coders. CRC Press.

Greenwood, A.R., 1997. Articulatory speech synthesis using diphone units. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1635–1638.

Gruber, M., Matoušek, J., Tihelka, D., Hanzlíček, Z., 2014. Reducing footprint of unit selection TTS system by removing linguistic segments with rarely selected units. In: Proceedings of the International Conference on Signal Processing, pp. 494–499.

Hanzlíček, Z., Matoušek, J., Tihelka, D., 2013. Experiments on reducing footprint of unit selection TTS system. In: Proceedings of the International Conference on Text, Speech, and Dialogue. Springer, pp. 249–256.

Horev, I., Bryt, O., Rubinstein, R., 2012. Adaptive image compression using sparse dictionaries. International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 592–595.

Hoyer, P.O., 2002. Non-negative sparse coding. In: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, pp. 557–565.

Hunt, A.J., Black, A.W., 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 373–376.

ITU-T, 2013. Methods for objective and subjective assessment of speech quality: Mean opinion score interpretation and reporting. ITU-T, Series P: Terminals and Subjective and Objective Assessment Methods, ITU-T P.800.2, ITU

Jacques, L., Laska, J.N., Boufounos, P.T., Baraniuk, R.G., 2013. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory 59 (4), 2082–2102.

Jafari, M.G., Plumbley, M.D., 2011. Fast dictionary learning for sparse representations of speech signals. IEEE J. Sel. Topics in Signal Process. 5 (5), 1025–1031.

Q3 King, S., Karaiskos, V., 2011. The blizzard challenge. In: Proceedings of the Blizzard Challenge Workshop.

Laska, J.N., Baraniuk, R.G., 2012. Regime change: bit-depth versus measurement-rate in compressive sensing. IEEE Trans. Signal Process. 60 (7), 3496–3505.

Mailhe, B., Gribonval, R., Bimbot, F., Vandergheynst, P., 2009. A low complexity orthogonal matching pursuit for sparse signal approximation with shift-invariant dictionaries. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3445–3448.

Malloy, M.L., Nowak, R.D., 2014. Near-optimal adaptive compressed sensing. IEEE Trans. Inf. Theory 60 (7), 4001–4012.

Nurminen, J., SilĘn, H., Gabbouj, M., 2013. Speaker-specific retraining for enhanced compression of unit selection text-to-speech. In: Proceedings of the Interspeech, pp. 388–391.

Pellegrini, T., Costa, A., Trancoso, I., 2012. Less errors with TTS? A dictation experiment with foreign language learners. In: Proceedings of the Interspeech, pp. 1291–1294.

Qinsheng, D., Jian, Z., Lirong, W., Lijuan, S., 2011. Articulatory speech synthesis: a survey. In: Proceedings of the IEEE International Conference on Computational Science and Engineering, pp. 539–542.

Rabiner, L.R., Schafer, R.W., 2010. Theory and Applications of Digital Speech Processing. Pearson Education Limited.

Rubinstein, R., Zibulevsky, M., Elad, M.,2010. Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. CS Technion.

Sagisaka, Y., 1988. Speech synthesis by rule using an optimal selection of non-uniform synthesis units. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 679–682.

Shanmugam, A., Murthy, H., 2014. A hybrid approach to segmentation of speech using group delay processing and HMM based embedded reestimation. In: Proceedings of the Interspeech, pp. 1648–1652.

Sharma, P., Abrol, V., Dileep, A.D., Sao, A.K., 2015a. Sparse coding based features for speech units classification. In: Proceedings of the Interspeech, pp. 712–715.

Sharma, P., Abrol, V., Sao, A.K., 2015b. Compressed sensing for unit selection based speech synthesis. In: Proceedings of the European Signal Processing Conference (EUSIPCO), pp. 1731–1735.

Sharma, P., Abrol, V., Sao, A.K., 2017. Deep sparse representation based features for speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 25 (11), 2162–2175.

Skretting, K., Engan, K., 2011. Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. In: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, pp. 1517–1520.

Sondhi, M.M., Schroeter, J., 1987. A hybrid time-frequency domain articulatory speech synthesizer. IEEE Trans. Acoust. Speech and Signal Process. 35 (7), 955–967.

Sreenivas, T.V., Kleijn, W.B., 2009. Compressive sensing for sparsely excited speech signals. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4125–4128.

Strecha, G., Eichner, M., Hoffmann, R., 2007. Line cepstral quefrencies and their use for acoustic inventory coding. In: Proceedings of the Interspeech, pp. 2873–2876.

Sun, J., Wang, S., Dong, Y., 2013. Sparse block circulant matrices for compressed sensing. IET Commun. 7 (13), 1412–1418.

Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K., 2013. Speech synthesis based on hidden Markov models. Proc. IEEE 101 (5), 1234–1252.

Tosic, I., Frossard, P., 2011. Dictionary learning. IEEE Signal Proc. Mag. 28 (2), 27–38.

Toutios, A., Narayanan, S., 2013. Articulatory synthesis of French connected speech from EMA data. In: Proceedings of the Fourteenth Interspeech, pp. 2738–2742.

Tsiakoulis, P., Chalamandaris, A., Karabetsos, S., Raptis, S., 2008. A statistical method for database reduction for embedded unit selection speech synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4601–4604.

Vepa, J., King, S., 2006. Subjective evaluation of join cost and smoothing methods for unit selection speech synthesis. IEEE Trans. Audio Speech Lang. Process. 14 (5), 1763–1771.

Zen, H., Senior, A., 2014. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3872–3876.

Zen, H., Senior, A., Schuster, M., 2013. Statistical parametric speech synthesis using deep neural networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 7962–7966.

Zen, H., Tokuda, K., Black, A.W., 2009. Statistical parametric speech synthesis. Speech Commun. 51 (11), 1039–1064.

Please cite this article as: P. Sharma et al., Reducing footprint of unit selection based text-to-speech system using compressed sensing and sparse representation, Computer Speech & Language (2018), https://doi.org/10.1016/j.csl.2018.05.003