PAPER

Distributed Source Coding for Real-Time ECG Signal Monitoring*

Hung-Tsai WU[†], **Wei-Ying TSAI**[††], ***Nonmembers***, ***and*** **Wen-Whei CHANG**[†a)], ***Member***

SUMMARY Wireless patient monitoring is an active research area with the goal of ubiquitous health care services. This study presents a novel means of exploiting the distributed source coding (DSC) in low-complexity compression of ECG signals. We first convert the ECG data compression to an equivalent channel coding problem and exploit a linear channel code for the DSC construction. Performance is further enhanced by the use of a correlation channel that more precisely characterizes the statistical dependencies of ECG signals. Also proposed is a modified BCJR algorithm which performs symbol decoding of binary convolutional codes to better exploit the source's *a priori* information. Finally, a complete setup system for online ambulatory ECG monitoring via mobile cellular networks is presented. Experiments on the MIT-BIH arrhythmia database and real-time acquired ECG signals demonstrate that the proposed system outperforms other schemes in terms of encoder complexity and coding efficiency.

key words: *wireless patient monitoring, distributed source coding, ECG data compression, mobile cellular networks*

1. Introduction

Electrocardiogram (ECG) signal is a recording of the electrical activity of the heart and is a clinical diagnostic tool for cardiac diseases. Uncompressed raw ECG signals require high data rates and thus consume excessive bandwidth and energy. This may limit its practical applicability in ubiquitous health care systems. To realize such systems and reduce the constraints on the patient, wireless transmission of signals from ECG sensors to a remote medical server is essential. However, wireless transmission is known to be one of the largest contributors to power consumption in ambulatory sensors [1]. Thus, compression of ECG signals prior to transmission should be used to reduce the data redundancy. Such redundancy typically appears in intra-beat correlation between adjacent samples and inter-beat correlation between adjacent heartbeats. With this in mind, ECG data compression aims to achieve maximum data volume reduction while preserving the clinical information content upon reconstruction. As for online ambulatory monitoring, the sensor calls for real-time compression with the added restriction of low power operation.

Many ECG data compression methods have been proposed, including direct time-domain methods and transformation methods [2], [3]. Most of the methods adopt one-dimensional (1-D) representations for ECG signals and focus on the utilization of the intra-beat correlation between adjacent samples. Since the ECG signals have both intra- and inter-beat correlation, better algorithms have been proposed to get the most benefit from both types of correlation [4], [5]. These methods generally start with a preprocess which converts 1-D ECG signals to 2-D data arrays through the combined use of QRS detection and period normalization. The constructed 2-D ECG data arrays are then ready to be further compressed by the beat-based gain-shape vector quantization (GSVQ) [4] or the wavelet-based JPEG2000 image compression standard [5]. When applied directly to online patient monitoring, however, these methods are expected to show some limitations for use on the sensors due to power and delay constraints. First, JPEG2000 incurs significant encoding delay because it operates on an entire 2-D data array and must buffer an array-size's worth of ECG samples before it can start encoding. This could severely hamper the real-time feasibility of the compression scheme as power and memory are scarce at the sensors. Second, the method in [4] takes an entire heartbeat segment as the input for GSVQ, which differs from other VQ-based schemes [3]. As the vector dimension is increased the reconstructed signal will suffer higher levels of distortion unless the codebook size is also increased. With such a combination, the computational burden sometimes limits its practical applicability to the online ECG monitoring. Finally, most of the current methods have in common that they reduce the data redundancy by exploiting source statistics at the encoder. Implementation of such systems require much more computation for the encoder than for the decoder. This asymmetry is well-suited for the traditional broadcasting paradigm, where source signals are compressed once at the encoder and decoded many times. In contrast, online monitoring applications may require the dual system, where the bulk of the computation is shifted from the encoder to the decoder.

It has recently been shown [6] that the traditional balance of complex encoders and simple decoders can be reversed within the framework of distributed source coding (DSC). This new paradigm is the consequence of Slepian-Wolf theorem [7], which stated that separate encoding and joint decoding of two correlated sources are as efficient as their joint encoding and decoding. Slepian-Wolf codes are generally derived based on binning of linear channel

Manuscript received October 24, 2013.
Manuscript revised March 7, 2014.

[†]The authors are with the Institute of Communications Engineering, National Chiao-Tung University, Hsinchu, Taiwan.
[††]The author is with the Engineering Department, Garmin Corporation, Taipei, Taiwan.

*This research was supported by the National Science Council of Taiwan, under Grant NSC 102-2221-E-009-030-MY3.

a) E-mail: wwchang@cc.nctu.edu.tw

DOI: 10.1587/transinf.E97.D.2157

codes, but fundamentally harder for practical applications due to the general statistics of the correlation channel between source sequence and side information [8]. In this study, we propose a low-complexity encoding algorithm that uses inter-beat correlation of ECG signals to reduce the data redundancy and provide a joint decoding algorithm that uses the correlation as side information in a maximum *a posteriori* probability (MAP) algorithm. With DSC, low-complexity compression of ECG signals can be achieved by leveraging knowledge of the source statistics at the decoder only. Finally, we develop a complete setup system for field testing of the proposed ECG data compression technique. The patient unit comprises a wireless sensor for ECG signal acquisition and an embedded system development board for data compression. The compressed ECG data are fed to a Bluetooth-enabled mobile telephone and then transmitted to the remote server via mobile cellular networks. Upon receiving the compressed data, the server reconstructs ECG signals and displays them on a screen for health monitoring.

The rest of this paper is organized as follows. Section 2 describes the ECG fundamentals and presents a preprocess which converts 1-D ECG signals to 2-D data arrays. Details of the algorithms for low-complexity encoding and joint decoding are provided in Sects. 3 and 4, respectively. Section 5 presents the simulation results of various data compression schemes using the MIT-BIH arrhythmia database [9]. Both the hardware architecture and software implementation required for wireless ECG signal monitoring are provided in Sect. 6. Finally, Sect. 7 gives our conclusions.

2. Correlation Channel Modelling

DSC refers to the problem of lossless compression of two correlated binary sources with coders that do not communicate with each other. Constructing a proper correlation model is the first step in applying DSC to data compression and one that conditions all subsequent steps of the implementation. We generally assume that the statistical dependence between the two sources is modelled by a virtual channel analogous to a binary symmetric channel (BSC) defined by a bit error rate (BER). The input of the correlation channel is the first source X and its output is the second source $Y = X \oplus Z$ referred to as side information, where Z represents the bit error sequence. The lower the BER, the better DSC is at exploiting the correlation and achieving lower data rates with the same average distortion. Thus, there is a need to develop a correlation channel with low BER that can be corrected by conventional channel codes. To begin, we apply a preprocessor which can be viewed as a cascade of two stages. In the first stage, the QRS complex in each heartbeat is firstly detected for segmenting and aligning a 1-D ECG signal to a 2-D data array and in the second stage, the constructed 2-D ECG data array is compressed by GSVQ. Figure 1 shows the block diagram of the proposed ECG data compression scheme.

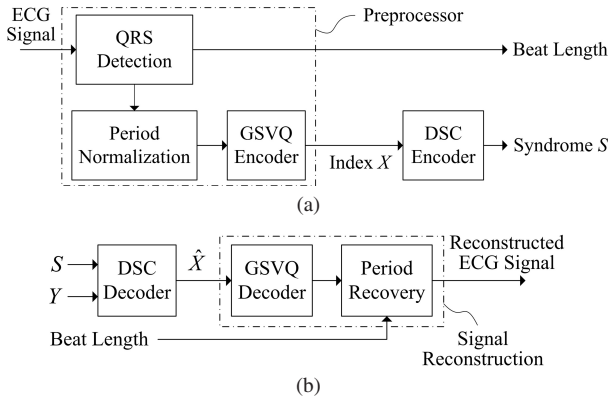


Fig. 1 Block diagrams of the proposed ECG data compression scheme. (a) Encoder and (b) decoder.

2.1 ECG Data Sources

Two types of ECG sources are provided: the acquisition from the MIT-BIH arrhythmia database [9] and the real-time acquisition through an ambulatory sensor. For the online testing, we use the wearable BtECG sensor [10] with ECG electrodes placed on the chest of the patient. The real-time acquired ECG data are sampled at 250 Hz with a resolution of $n_{res} = 12$ bits/sample. On the other hand, ECG records taken from the MIT-BIH arrhythmia database are primarily used in computer simulations to evaluate the performance of the proposed scheme. The database is composed of 48 ECG records and each record is about half-hour in duration. The ECG data were sampled at 360 Hz and each sample has a resolution of $n_{res} = 11$ bits/sample. In the experiments, we use two datasets formed by taking eight records from the MIT-BIH arrhythmia database. For each ECG record, the dataset with the first 1500 beats is used for training, and the dataset with the other 100 beats is used for testing.

The ECG waveform of a normal heartbeat consists of a P wave, a QRS complex, and a T wave. The P wave corresponds to the sequential depolarization of the right and left atria. The QRS complex is produced when the ventricles depolarize and squeeze the blood from the right ventricle to the aorta. The T wave occurs due to ventricular repolarization. An ECG signal tends to exhibit considerable similarity between adjacent heartbeats, along with short-term correlation between adjacent samples. Thus, by dividing an ECG waveform into heartbeat segments with length equal to the beats, there should be a large correlation between individual segments.

2.2 2-D Data Array Construction

ECG itself is 1-D in the time domain, but can be viewed as a 2-D signal in terms of its implicit periodicity. The QRS complex is the most characteristic wave in an ECG waveform and hence, its peak can be used to identify each beat. In consideration of speed and accuracy, the first-derivative approach presented in [11] was adopted to detect the QRS

complexes. The QRS complexes have the maximum variation in the slopes. Based on this property and an adaptive threshold was applied, the method evaluated the slope in the time domain and use it to locate the onset of the QRS complex. Then, an ECG signal is divided into heartbeat segments and each segment is stored as one row of a 2-D data array. Having constructed the array as such, the intra-beat correlation is in the horizontal direction of the array and the inter-beat correlation is in the vertical direction. Since the heartbeat segments may have different lengths, each row of the array is period normalized to a fixed length of $N_p = 288$ samples via linear interpolation. This choice was based on the observation that the average value of the detected heartbeat lengths is about 0.8 second, which corresponds to 288 samples for a 360-Hz sampling frequency. Accordingly, the original heartbeat length was represented with $n_l = 9$ bits and transmitted as side information. Afterwards each row of N_p samples is segmented uniformly into vectors with length M for a further application of the GSVQ.

GSVQ [12] is a product code technique which organizes the set of reproduction vectors as the Cartesian product of a vector codebook describing the shape of input vectors and a scalar codebook describing the gain. To begin, an input vector of length M is scaled in amplitude by a gain to produce the shape vector. Afterwards the GSVQ encoder searches through the shape codebook for the codevector that best matches the gain-normalized input vector, and then sends the corresponding index to the DSC encoder. Here the shape codebook consisting of $N = 2^n$ codevectors is trained using the well-known LBG algorithm [13]. By increasing the vector dimension and codebook size, the data redundancy could be reduced, but the resulting improvement in compression performance will be offset by the accompanying increase in computational complexity and encoding delay. With this in mind, values of $M = 4$ and $N = 64$ were empirically determined and used throughout the experiments. As a consequence, each heartbeat segment is represented by a total of $n_v = N_p/M = 72$ GSVQ indexes of bit-length $n = 6$. In the testing dataset each record contains 100 beats and among them, the GSVQ indexes of the first beat are transmitted without compression and used as side information Y, and those of the remaining $n_c = 99$ beats are used as the source sequences X for a further application of the proposed DSC-based compression.

A preliminary experiment was first conducted on two ECG records to examine the BER of the correlation channel for the case of GSVQ with $N = 64$ and $M = 4$. The results show that the calculated BER for records 102 and 220 of the MIT-BIH arrhythmia database are 20.47% and 18.40%, respectively. It is apparent that these BERs are too high to be corrected by conventional channel codes and hence, further process is needed toward DSC encoding. Recognizing this, we propose using the GSVQ in conjunction with a multiple-choice index assignment scheme. The first step is to use the GSVQ indexes of the training dataset to construct a reference model consisting of N_p/M indexes, each of which occurs the most frequent in the corresponding shape codevector. Upon quantizing an input vector, the indexes associated with the top-three closest shape codevectors are compared with the reference model to search for the one that has the minimal Hamming distance. Proceeding in this way, the calculated BER of the correlation channel for records 102 and 220 can be reduced to 7.35% and 5.59%, respectively.

3. Low-Complexity Encoding Algorithm

Computation and delay limitations have made real-time compression an important feature for online ambulatory monitoring systems. For this investigation we formulate the ECG data compression as an asymmetric DSC problem, where X is the source to be compressed and Y is the side information available at the decoder. Within the DSC framework, the encoding can be thought of as a technique which generates syndromes of the linear channel code to correct the errors introduced by the correlation channel. Viewing from this perspective, designing a DSC encoder becomes a well-defined two-step process: finding a good channel code and constructing a syndrome former (SF) for this code.

3.1 Distributed Source Coding

According to Shannon's source coding theory [14], a rate given by the joint entropy $H(X, Y)$ is sufficient for joint encoding of two sources X and Y. As for separate encoding, the Slepian-Wolf theorem [7] states that X and Y can be losslessly recovered if and only if coding rates R_X and R_Y satisfy the following entropy bounds: $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X, Y)$. Here we consider an asymmetric DSC problem, where the side information Y is assumed losslessly available at the decoder and the source X is compressed as much as possible. Assume Y can be compressed at its entropy $R_Y = H(Y)$, asymmetric DSC corresponds to $R_X \geq H(X|Y)$, regardless of the encoder's access to side information Y. It is clear that the compression performance of DSC crucially depends on the correlation between X and Y, since the descending correlation rises the conditional entropy $H(X|Y)$.

A practical construction of DSC can be transformed to an equivalent channel coding problem with decoder side information [6], [15]. Toward this end, the binning approach of linear channel codes provides a general framework for constructing DSC encoder/decoder pairs. For an (n, k) linear channel code, the $k \times n$ generator matrix G specifies the code space containing all the valid codewords, and the $(n-k)\times n$ parity-check matrix H specifies its null space such that $GH^T = 0_{k\times(n-k)}$. To apply the code binning, a total of 2^n source sequences $\boldsymbol{x}$ of length n are viewed as the virtual codewords of the linear channel code. Then, the codeword space of source sequence $\boldsymbol{x}$ is divided into 2^{n-k} bins such that the same distance property is preserved in each bin. Each bin consists of 2^k codewords and can be indexed by a length-$(n-k)$ syndrome, defined as $\boldsymbol{s} = \boldsymbol{x}H^T$. This constraint makes sure that all source sequences in a bin share the same syndrome and that all valid codewords of a linear

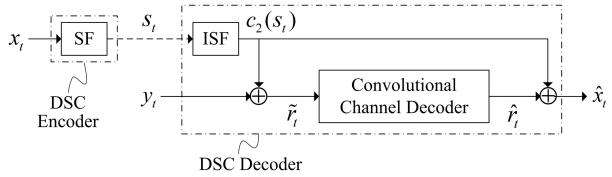


Fig. 2 Block diagram of the DSC encoder/decoder pair.

channel code belong to the bin with syndrome $\boldsymbol{s} = \boldsymbol{0}$.

3.2 Syndrome Former

A block diagram for the proposed DSC encoder/decoder pair is shown in Fig. 2, where the GSVQ index sequence is treated as the source sequence to compress. The DSC encoder behaves like a SF that maps a source sequence $\boldsymbol{x}$ of length n to its syndrome $\boldsymbol{s}$ of length $n-k$. By grouping source sequences into bins and transmitting the bin index instead of the source sequence, a compression ratio of $n:n-k$ is achieved. The design of SF usually consists of selecting a channel code with a certain rate and correction capability matched to the fidelity requirement of data compression. In this work, we apply a rate-compatible punctured convolutional (RCPC) code that is obtained from puncturing the parity bits of a recursive systematic convolutional (RSC) mother code [16]. To begin, let us first consider an (n,k) RSC code with generator matrix $G = [I_k, P_{k\times(n-k)}]$, where I_k is a $k \times k$ identity matrix and $P_{k\times(n-k)}$ denotes a submatrix containing $k \times (n-k)$ generator polynomials. It has been shown [15] that the SF for RSC codes can be implemented using an $n \times (n-k)$ transfer matrix $H^T = [P^T_{k\times(n-k)}, I_{n-k}]^T$. Proceeding in this way, we first select a $(2,1)$ RSC code and use it as the mother code for puncturing to produce a $(6,4)$ RCPC code. More precisely, we start with the 1×2 generator matrix of the mother code and replace each element in the matrix by its 4-th polycyclic pseudocirculant [16], leading to an expanded matrix with dimension 4×8. Afterwards, two of the columns corresponding to parity bits are punctured to produce a 4×6 generator matrix of the $(6,4)$ RCPC code. As indicated in [16], following the above procedures will enable us to construct a recursive systematic RCPC code and hence, its SF implementation proceeds in a way similar to the RSC code.

4. Joint Decoding Algorithm

Goal of the joint decoder is to use the received syndrome sequence together with the side information to identify the original source sequence. As shown in Fig. 2, the DSC decoder consists of an inverse syndrome former (ISF) and a convolutional decoder matched to the underlying correlation channel. The decoder begins with the ISF and adds its output to the side information, whose result is then fed into the channel decoder to perform the error-correcting decoding. If the RCPC channel code is sufficiently powerful, adding the ISF output to the output of the channel decoder will recover the original source sequence.

4.1 Inverse Syndrome Former

SF and ISF are functions that map the codeword space of source sequence to the syndrome space and vice versa. The transfer matrix H^T fulfills the function of a SF and, by taking the left inverse of the SF, a matching ISF can be implemented using the matrix $(H^{-1})^T$. Suppose at time t, the source sequence $\boldsymbol{x}_t = \boldsymbol{c}_1(\boldsymbol{s}_t)$ is a codeword $\boldsymbol{c}_1$ taken from the bin with syndrome $\boldsymbol{s}_t$. The SF will map the sequence $\boldsymbol{x}_t$ to $\boldsymbol{s}_t$ and sends the syndrome to the decoder. Upon receiving $\boldsymbol{s}_t$, the ISF will find an arbitrary sequence $\boldsymbol{c}_2(\boldsymbol{s}_t) = \boldsymbol{s}_t(H^{-1})^T$ from the bin indexed by the syndrome $\boldsymbol{s}_t$. Now let the side information be denoted by $\boldsymbol{y}_t = \boldsymbol{x}_t \oplus \boldsymbol{z}_t$, where $\boldsymbol{z}_t$ is the bit error sequence introduced by the correlation channel. By adding $\boldsymbol{c}_2(\boldsymbol{s}_t)$ to the side information $\boldsymbol{y}_t$, we obtain

$$\boldsymbol{y}_t \oplus \boldsymbol{c}_2(\boldsymbol{s}_t) = \boldsymbol{c}_1(\boldsymbol{s}_t) \oplus \boldsymbol{c}_2(\boldsymbol{s}_t) \oplus \boldsymbol{z}_t = \boldsymbol{c}_3(\boldsymbol{0}_t) \oplus \boldsymbol{z}_t, \tag{1}$$

where $\boldsymbol{c}_3(\boldsymbol{0}_t)$ denotes a valid codeword $\boldsymbol{c}_3$ with all-zero syndrome $\boldsymbol{0}_t$. Implicit in this equation is the fact that the sum of any two codewords in the same bin leads to a valid codeword with all-zero syndrome [17]. Hence, if the channel code is sufficiently powerful, the channel decoder can recover the valid codeword $\boldsymbol{c}_3(\boldsymbol{0}_t)$. Since $\boldsymbol{c}_3(\boldsymbol{0}_t) = \boldsymbol{x}_t \oplus \boldsymbol{c}_2(\boldsymbol{s}_t)$, adding back the ISF output sequence $\boldsymbol{c}_2(\boldsymbol{s}_t)$ yields the original source sequence $\boldsymbol{x}_t$.

4.2 Modified BCJR Algorithm

In the proposed system RSC codes are used to construct the SF-ISF pair, so the soft-output channel decoding can be implemented efficiently by the BCJR algorithm [18]. The commonly used BCJR algorithm is derived on the basis of a bit-level trellis diagram and hence, it can only exploit bitwise source *a priori* knowledge. Although the GSVQ has been applied to reduce the data redundancy, but due to vector length constraints, the GSVQ indexes will still exhibit considerable redundancy. Such residual redundancy appears on index-level, either in terms of a non-uniform distribution or in terms of time-correlation between consecutive indexes. In order to better exploit the residual redundancy, we propose a modified BCJR algorithm which uses the GSVQ indexes rather than single index-bits as the bases for the trellis-based decoding of a binary convolutional code. Assume that an (n,k) RSC encoder maps an input symbol $\boldsymbol{u}_t$ of length k to an n-length codeword $\boldsymbol{r}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t)$, where $\boldsymbol{u}_t$ and $\boldsymbol{v}_t$ denote the systematic and parity symbol, respectively. For convenience, we say that the trellis diagram of RSC code forms a finite-state machine defined by its state transition function $f_s(\boldsymbol{u}_t, \sigma_{t-1})$ and output function $f_o(\boldsymbol{u}_t, \sigma_{t-1})$. More precisely, the channel codeword associated with the branch from state σ_{t-1} to state $\sigma_t = f_s(\boldsymbol{u}_t, \sigma_{t-1})$ can be written as $\boldsymbol{r}_t = f_o(\boldsymbol{u}_t, \sigma_{t-1})$. With respect to an implementation of DSC it has to be emphasized that $\boldsymbol{r}_t = \boldsymbol{c}_3(\boldsymbol{0}_t)$ refers to a valid codeword with all-zero syndrome, and $\tilde{\boldsymbol{r}}_t$ refers to its noise-corrupted version introduced by the correlation channel, i.e., $\tilde{\boldsymbol{r}}_t = \boldsymbol{r}_t \oplus \boldsymbol{z}_t$. We denote the sequence consisting of T noisy

codewords by $\tilde{\boldsymbol{r}}_1^T = \{\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \ldots, \tilde{\boldsymbol{r}}_T\}$. Viewing from this perspective, the channel decoder takes the sequence $\tilde{\boldsymbol{r}}_1^T$ as input and produces as its output the *a posteriori* probability (APP) for each systematic symbol $\boldsymbol{u}_t$, which is denoted by $P(\boldsymbol{u}_t|\tilde{\boldsymbol{r}}_1^T)$. These APPs are used to obtain an estimate $\hat{\boldsymbol{u}}_t$ according to the MAP criterion. Then, the corresponding channel codeword $\hat{\boldsymbol{r}}_t = \hat{\boldsymbol{c}}_3(\boldsymbol{0}_t)$ is added with the ISF output $\boldsymbol{c}_2(\boldsymbol{s}_t)$ to obtain the decoded GSVQ index $\hat{\boldsymbol{x}}_t$. Finally, the shape codevector corresponding to $\hat{\boldsymbol{x}}_t$ is multiplied by the gain and then fed into the period recovery module for signal reconstruction.

We next derive a modified BCJR algorithm which computes the APP of a systematic symbol $\boldsymbol{u}_t = i$ given the sequence $\tilde{\boldsymbol{r}}_1^T$. Taking the trellis state σ_t into consideration, we first rewrite the APP as follows:

$$P(\boldsymbol{u}_t = i|\tilde{\boldsymbol{r}}_1^T) = C \sum_{\sigma_t} \alpha_t^i(\sigma_t)\beta_t^i(\sigma_t), \tag{2}$$

where $\alpha_t^i(\sigma_t) = P(\boldsymbol{u}_t = i, \sigma_t, \tilde{\boldsymbol{r}}_1^t)$, $\beta_t^i(\sigma_t) = P(\tilde{\boldsymbol{r}}_{t+1}^T|\boldsymbol{u}_t = i, \sigma_t, \tilde{\boldsymbol{r}}_1^t)$, and $C = 1/P(\tilde{\boldsymbol{r}}_1^T)$ is a normalizing factor. For the recursive implementation, the forward and backward recursions are to compute the following metrics:

$$\begin{aligned}\alpha_t^i(\sigma_t) &= \sum_{\sigma_{t-1}} \sum_j P(\boldsymbol{u}_t = i, \boldsymbol{u}_{t-1} = j, \sigma_t, \sigma_{t-1}, \tilde{\boldsymbol{r}}_1^t) \qquad (3)\\ &= \sum_{\sigma_{t-1}} \sum_j \alpha_{t-1}^j(\sigma_{t-1})\gamma_{i,j}(\tilde{\boldsymbol{r}}_t, \sigma_t, \sigma_{t-1}),\\ \beta_t^i(\sigma_t) &= \sum_{\sigma_{t+1}} \sum_j P(\boldsymbol{u}_{t+1} = j, \sigma_{t+1}, \tilde{\boldsymbol{r}}_{t+1}^T|\boldsymbol{u}_t = i, \sigma_t, \tilde{\boldsymbol{r}}_1^t) \qquad (4)\\ &= \sum_{\sigma_{t+1}} \sum_j \beta_{t+1}^j(\sigma_{t+1})\gamma_{j,i}(\tilde{\boldsymbol{r}}_{t+1}, \sigma_{t+1}, \sigma_t),\end{aligned}$$

and in (3)

$$\begin{aligned}&\gamma_{i,j}(\tilde{\boldsymbol{r}}_t, \sigma_t, \sigma_{t-1}) \qquad (5)\\ &= P(\boldsymbol{u}_t = i, \sigma_t, \tilde{\boldsymbol{r}}_t|\boldsymbol{u}_{t-1} = j, \sigma_{t-1}, \tilde{\boldsymbol{r}}_1^{t-1})\\ &= P(\sigma_t|\boldsymbol{u}_t = i, \sigma_{t-1})P(\tilde{\boldsymbol{r}}_t|\boldsymbol{u}_t = i, \sigma_{t-1})P(\boldsymbol{u}_t = i|\sigma_{t-1}).\end{aligned}$$

Having a proper representation of the branch metric $\gamma_{i,j}(\tilde{\boldsymbol{r}}_t, \sigma_t, \sigma_{t-1})$ is a critical step in applying symbol decoding to error mitigation. As a practical manner, several additional factors must be considered to take advantage of the trellis structure and BSC assumption of the correlation channel. First, making use of the trellis structure, the value of $P(\sigma_t|\boldsymbol{u}_t = i, \sigma_{t-1})$ is either one or zero depending on whether symbol i is associated with the transition from state σ_{t-1} to state $\sigma_t = f_s(\boldsymbol{u}_t, \sigma_{t-1})$. For BSC channels, the second term in (5) is reduced to

$$\begin{aligned}P(\tilde{\boldsymbol{r}}_t|\boldsymbol{u}_t = i, \sigma_{t-1}) &= P(\tilde{\boldsymbol{r}}_t|\boldsymbol{r}_t = f_o(\boldsymbol{u}_t, \sigma_{t-1})) \qquad (6)\\ &= P(\boldsymbol{c}_3(\boldsymbol{0}_t) \oplus \boldsymbol{z}_t|\boldsymbol{c}_3(\boldsymbol{0}_t))\\ &= (1-\epsilon)^{n-d} \cdot \epsilon^d,\end{aligned}$$

where ϵ is the BER of the correlation channel, and d denotes the Hamming distance between $\boldsymbol{r}_t$ and $\tilde{\boldsymbol{r}}_t$. To exploit the source *a priori* information on index-level, we rewrite the last term in (5) as

$$\begin{aligned}P(\boldsymbol{u}_t = i|\sigma_{t-1}) &= P(\boldsymbol{c}_3(\boldsymbol{0}_t) = f_o(\boldsymbol{u}_t, \sigma_{t-1})) \qquad (7)\\ &= P(\boldsymbol{x}_t = f_o(\boldsymbol{u}_t, \sigma_{t-1}) \oplus \boldsymbol{c}_2(\boldsymbol{s}_t)).\end{aligned}$$

The first equality follows the fact that if σ_{t-1} is known, there is a one-to-one mapping between the systematic symbol and the output codeword for RSC codes. As for the second equality, it demonstrates that we can integrate the probability distribution of GSVQ indexes which was trained in advance into the branch metric. Clearly, the accuracy of the APP computation can be enhanced by the use of the index-level source *a priori* information.

5. Simulation Results

Computer simulations were conducted to compare the performance of various compression schemes for ECG signals. Two ECG compression approaches based on DSC, denoted by DSC1 and DSC2, are presented and investigated. They both applied a $(6, 4)$ RCPC code for construction of the SF-ISF pair and performed trellis-based MAP decoding to correct the errors introduced by the correlation channel. Unlike the DSC1 which uses a conventional BCJR decoder, the channel decoder in the DSC2 uses the modified BCJR algorithm which performs symbol decoding to incorporate the index-level source *a priori* knowledge. The ECG sources include eight records taken from the MIT-BIH arrhythmia database, numbered by 101, 102, 109, 112, 118, 122, 214, and 220. For each ECG record, we perform QRS detection and period normalization so that all the heartbeat segments have a fixed length of $N_p = 288$ samples. Let n_{sam} denote the total number of original samples in the record, n_{res} bits/sample the original resolution, n_c the number of testing beats, n_b the length of the bin index, $n_v = N_p/M$ the number of vectors per beat, and n_l the number of bits used to code the original heartbeat length. The parameter values used in the experiment for n_{res}, n_c, n_b, n_v and n_l were 11, 99, 2, 72, and 9, respectively.

The system performance is evaluated in terms of the percent root mean square difference (PRD) and the compression ratio (CR). The CR is defined as

$$CR = \frac{n_{sam} \cdot n_{res}}{n_c \cdot (n_v \cdot n_b + n_l)}. \tag{8}$$

The PRD is used to evaluate the reconstruction distortion and is defined by

$$PRD(\%) = \sqrt{\frac{\sum_{i=1}^{n_{sam}} [x_{ori}(i) - x_{rec}(i)]^2}{\sum_{i=1}^{n_{sam}} x_{ori}(i)^2}} \times 100, \tag{9}$$

where x_{ori} and x_{rec} represent the original and reconstructed ECG signals, respectively. The training and testing dataset are organized as mentioned in Sect. 2.1. The 2-D ECG data array of the proposed scheme is constructed using the GSVQ with a vector dimension of $M = 4$ and a codebook size of $N = 2^n = 64$. Starting with a $(2, 1)$ RSC mother code with generator matrix $[1, 37/21]_{oct}$, we constructed the $(6, 4)$ binary RCPC code by replacing each element in the matrix

Table 1 Encoder complexity analysis of different compression methods.

	VQ [3]	FSVQ [3]	DSC2
Avg. PRD	2.99%	2.89%	2.65%
(M, n)	$(24, 12)$	$(18, 9)$	$(4, 6)$
Memory Space	$2^n \cdot M$	$(2^n + 1) \cdot K \cdot M$	$2^n \cdot (M + 2 \cdot 16) + (N_p/M)$
Add./Beat	$(N_p/M) \cdot 2^n \cdot (2M - 1)$	$(N_p/M) \cdot (2^n + K) \cdot (2M - 1)$	$(N_p/M) \cdot [2^n \cdot (2M - 1) + M - 1 + 3 \cdot (2n - 1)]$
Multi./Beat	$(N_p/M) \cdot 2^n \cdot M$	$(N_p/M) \cdot (2^n + K) \cdot M$	$(N_p/M) \cdot (2^n + 2) \cdot M$

Table 2 PRD comparison between the proposed method and other methods.

Record	CR	VQ [3]	FSVQ [3]	JPEG2000 [5]	DSC1	DSC2
101	25.86	7.07%	5.54%	4.95%	5.79%	3.53%
102	21.45	8.02%	6.10%	5.65%	9.40%	3.87%
109	18.46	6.23%	4.23%	3.21%	4.08%	2.46%
112	18.64	3.04%	2.38%	1.39%	4.29%	1.39%
118	21.62	5.44%	4.48%	3.27%	3.28%	2.07%
122	19.52	3.26%	2.45%	1.96%	5.90%	1.31%
214	20.82	10.65%	8.50%	5.96%	7.04%	4.22%
220	22.72	6.55%	5.29%	3.00%	3.84%	2.32%
Avg.	21.14	6.28%	4.87%	3.67%	5.45%	2.65%

by its 4-th polycyclic pseudocirculant, and by puncturing the 6-th and 8-th column.

We first compare the computational complexity of the proposed method with other methods that are based on VQ and finite-state VQ (FSVQ) [3]. A FSVQ can be viewed as a finite collection of ordinary VQ, where each successive input vectors is encoded using a VQ determined by the current encoder state [12]. Let K denote the number of encoder states used in the FSVQ. The computational complexity is measured in terms of the arithmetic capabilities and memory requirements it demands to implement the encoding process. The arithmetic capabilities are expressed as number of addition/multiplication operations per beat. For purposes of fair comparison, parameter values of each method were empirically determined to achieve comparable performances in PRD and CR. If 3% PRD is taken as an upper limit in distortion, then our preliminary results suggest that CRs in the region of 21 can be achieved by using $(M, n) = (24, 12)$ for VQ, $(18, 9)$ for FSVQ with $K = 8$, and $(4, 6)$ for DSC1 and DSC2. The encoder complexity analysis results for different schemes are given in Table 1. Notice that these three methods have in common the 2-D representation for ECG signals and hence, the computational complexity of the preprocessor is not included. In accordance with (M, n) values listed in the table, the arithmetic capabilities required for the proposed method is eight (or sixty) times more effective than FSVQ (or VQ). In addition, the memory space needed for the proposed method is reduced thirty (or forty) times relative to FSVQ (or VQ).

The next step is to evaluate the PRD performances of various system setups which yield comparable performances with respect to CR and encoding complexity. To achieve an average CR of 21.14 with similar encoding complexity, (M, n) values were empirically determined to be $(4, 6)$ for the proposed method, and $(12, 6)$ for VQ and four-state FSVQ. In addition to the above-mentioned schemes, we also simulated the JPEG2000 image codec for compression of ECG signals [5] with comparable CRs. Table 2 presents the PRD performances of various schemes for eight ECG records. The results clearly demonstrate the improved performance achievable using the proposed method in comparison to other methods. Compared with DSC1, the better performance of DSC2 can be attributed to its ability to compute the APP taking the index-level source *a priori* information into consideration. Notice that the JPEG2000 codec operates on an entire 2-D data array and must buffer a total of $n_c * N_p$ samples before it can start encoding. Thus, although the JPEG2000 performs well in all the tests, the long delay may limit its practical applicability to online monitoring systems. Next, we compare the original and reconstructed ECG signals to illustrate further from a subjective point of view. Although the proposed method shows good results for normal ECG signals, it may suffer from irregular rhythms mainly due to the QRS detection stage and the common drawback of VQ schemes. First, because of the limited size of GSVQ codebook, not all types of QRS complex morphology can be stored therein. Second, to utilize the inter-beat correlation, the performance of the proposed algorithm depends heavily on the accuracy of the QRS detection scheme. For the simple detection algorithm based on the first-derivatives [11], the number of QRS false detections may increase significantly in the presence of noise and varying QRS morphology. The following test cases demonstrate how the proposed algorithm performs with and without changes in QRS complex morphology. Figure 3 shows a 10-sec segment of a normal sinus rhythm trace from record 220. The experimental results for record 112 with severe baseline drift and record 101 with varying QRS morphology are shown in Figs. 4 and 5, respectively. The results indicate that characteristic features are well preserved and hence, the proposed method DSC2 is suitable for various morphologies of ECG data. Finally, we examine how the PRD performance of DSC2 changes as a function of its training dataset size. The results are indicated in Fig. 6 for different ECG records. As should be expected, the PRD decreases as the number of heartbeats from each ECG record used for train-

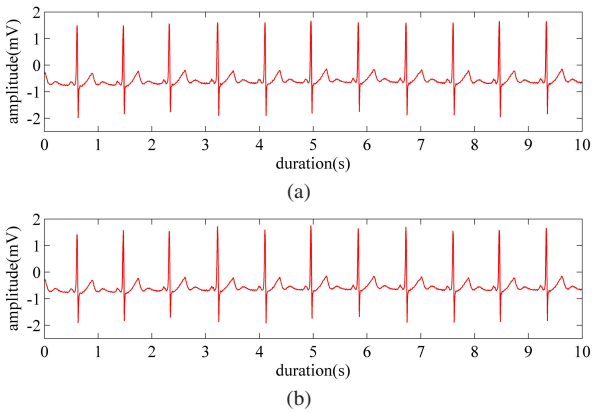


Fig. 3 (a) Original and (b) reconstructed normal sinus rhythm traces from record 220.

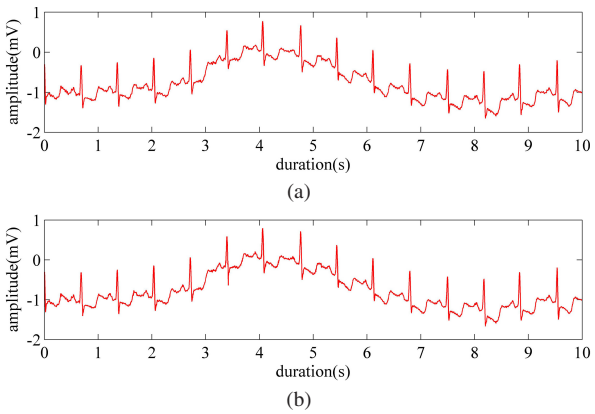


Fig. 4 (a) Original and (b) reconstructed ECG traces from record 112 that exhibits baseline drift.

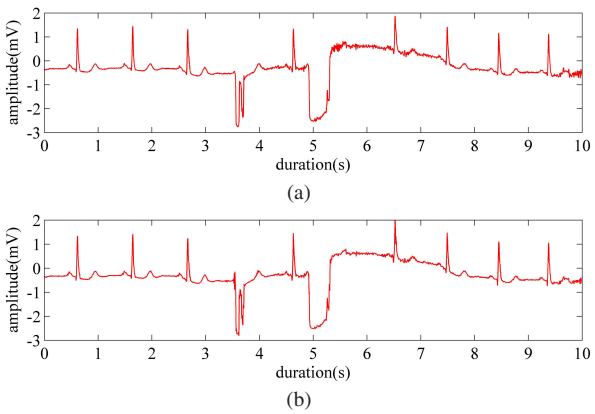


Fig. 5 (a) Original and (b) reconstructed ECG traces from record 101 that exhibits varying QRS morphology.

ing increases. In these experiments good performance was reached using a training dataset with more than 400 beats per ECG record.

6. Real-Time Implementation

In this section, we describe the main design features of a

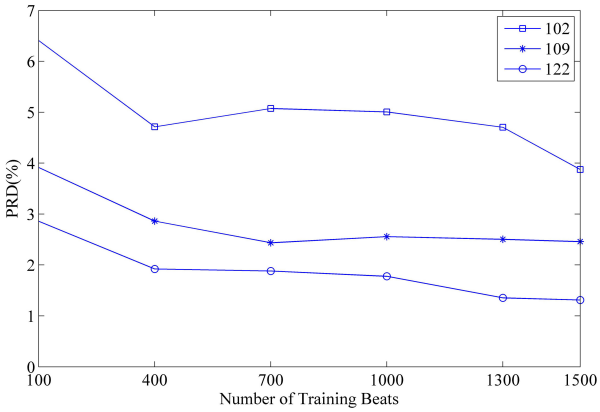


Fig. 6 PRD performance for different training dataset sizes.

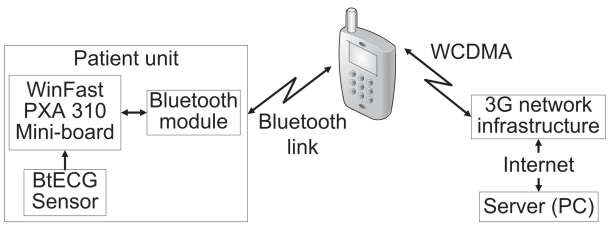


Fig. 7 Mobile ECG signal monitoring system.

complete setup system for field testing of real-time acquired ECG signals. As shown in Fig. 7, the proposed system comprises a BtECG sensor [10] for ECG data acquisition, an embedded system development board WinFast PXA310 [19] for signal compression, and a PC as the back-end server for signal reconstruction and monitoring. In addition, wireless communication techniques such as Bluetooth and mobile cellular networks are involved in the real-time implementation.

6.1 Hardware Architecture

The hardware architecture starts with a patient unit consists of two main components, whose specifications are shown in Table 3. For the acquisition of ECG signals, the patient unit uses the wearable BtECG sensor with ECG electrodes placed on the chest of the patient. A built-in class II Bluetooth module enables the sensor's wireless transmission of ECG data in a range of 10 meters. The low-complexity ECG encoding algorithm is implemented on the WinFast PXA310 development board, which is connected to a Bluetooth adapter for communication with the BtECG sensor. The board is equipped with a Marvell PXA310 processor and runs the Windows Embedded CE 6.0 operating system. The compressed ECG data are transmitted through a wireless link to the back-end server for signal reconstruction, which includes a Bluetooth link between the board and a mobile telephone, WCDMA to a base station, and Internet to the server. In the proposed system we use a commercially available Bluetooth-enabled 3G mobile telephone SAMSUNG S3370. Before building up a Bluetooth link, the

Table 3 The specification of the patient unit.

BtECG sensor	Size: 5.5 cm × 3.5 cm × 1.6 cm Channel: one ECG channel ADC resolution: 12 bits Sampling rate: 250 Hz Band-pass filter: 0.1 Hz and 40 Hz
WinFast PXA310 mini-board	Size: 11 cm × 9.5 cm × 3.5 cm Marvell PXA 310 processor: -Intel XScale core with 624 MHz -2 × 256 kB SRAM -USB 1.1 and other peripheral OS: WinCE 6.0
Bluetooth module	Version: v2.1+EDR Frequencies: 2.402–2.48 GHz RF power coverage: 10 meters (Class II) Theoretical data rate: 3 Mbps

board uses Service Discovery Protocol (SDP) to discover nearby devices in terms of their addresses and Radio Frequency Communication (RFCOMM) channel numbers. Serial Port Profile (SPP) and Dial-Up Networking (DUN) profile are used by the board for communication with the sensor and the mobile telephone, respectively. In addition, the board creates a virtual COM port to receive ECG data from the sensor. The board also creates another port to establish the link to the mobile telephone, and uses it as a modem to access the Internet via mobile cellular networks. Afterwards ECG data are transmitted from the board to the server using TCP protocol.

6.2 Software Implementation

The software implementation of the proposed system can be divided into two phases. The first phase runs on the WinFast PXA310 development board and is responsible for compression of real-time acquired ECG signals. To this end, we apply the development tools provided by Windows Embedded CE 6.0 platform builder. On the other hand, the second phase is realized on the server for signal reconstruction and is developed using MATLAB 7.6. The software implementation on the board typically comprises three stages:

1) System initialization: First, raw ECG data coming from the BtECG sensor are stored in *recvBuf* with a buffer size of 4 KB. A COM port and a network socket are created for communication with the mobile telephone and the server, respectively. To transmit data to the server, the IP address and the port of the server will be specified.

2) Signal processing: The PXA 310 processor is notified to read the data in *recvBuf* and perform QRS detection when the buffer is full. Once the peak of R wave within each beat has been detected, the board will proceed with period normalization and ECG data compression. Afterwards the compressed ECG data will be stored in the *sendBuf* prior to transmission.

3) Wireless communication: After each complete beat has been compressed, the board will retrieve and transmit the data from the *sendBuf* to the server based on TCP protocol. To this end, we use the Windows Sockets (Winsock) application programming interface provided by Microsoft.

To make the applications on the board more responsive to the input/output, multithreading programming technique is used for software development on the board. More precisely, while one thread is used to perform ECG data compression, two other threads are responsible for receiving data from the sensor and for transmitting data to the server, respectively. Finally, upon receiving the compressed data, the server will reconstruct and display the ECG signals on a screen for patient monitoring.

6.3 Experimental Results

In the experiment, we first collect a dataset of 1500 beats and use it for off-line training of the GSVQ codebook. Other system parameters are the same as those used in computer simulations. Experimental results on real-time acquired ECG signals demonstrate that an average PRD of 0.66% can be achieved with a compression ratio of 16.76. Compared with off-line computer simulations, the lower PRD can be attributed to the fact that real-time acquired ECG signals are measured on a healthy person. On the other hand, computer simulations were conducted on the MIT-BIH arrhythmia database and hence, the GSVQ codebook may not be large enough to characterize various irregular heart functions. As for the lower CR, it is a consequential result of differences in sampling rate and resolution.

With respect to meeting real-time requirements, some experiments were conducted to evaluate the system setup in terms of execution time and consumed memory size. First, we applied multiple application programming interfaces (APIs) to measure the elapsed time required by compressing a single heartbeat segment. The average elapsed time was empirically determined to be about 54.5 ms per beat. The maximum amount of memory usage reported here is 464 KB, including the buffers *recvBuf* and *sendBuf* with the size of 4 KB each and the large-size overheads introduced by the Windows Embedded CE 6.0 operating system. The analysis results indicates that the system setup meets the real-time requirements and can be used even with poor processing hardware and under low communication channel capacity. Moreover, the proposed system setup represents the basis for more experiments which will be focused on deployment of new algorithms in wireless telecardiology applications. For example, we now investigate new biometric techniques which read the compressed ECG instead of raw ECG to obtain unique features for human identification. As biometric templates created from the compressed ECG are substantially reduced in size, it is expected to achieve faster biometric authentication and treatment of emergency cardiac patients.

7. Conclusions

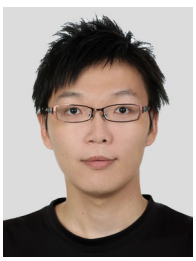
This study presents a novel means of exploiting the temporal correlation of ECG signals in the design of low-complexity compression algorithm within the DSC framework. We first emphasize the importance of matching the statistical depen-

dency of ECG signals to the correlation channel on which the DSC design is based. This task was done by converting a 1-D ECG signal to 2-D data array through the combined use of QRS detection, period normalization, and GSVQ. Performance is further enhanced by the use of a modified BCJR algorithm which performs symbol decoding of binary convolutional codes by exploiting the source *a priori* information on index-level. The results of off-line computer simulations demonstrated that the proposed scheme outperforms other codecs in terms of encoder complexity and coding efficiency. Finally, a complete setup system for online ECG signal monitoring via mobile cellular networks is presented. The system comprises a patient unit for signal acquisition and compression, and a remote server for signal reconstruction. Experiments on real-time acquired ECG signals demonstrated that the proposed system provides a powerful aid to wireless patient monitoring applications. Still, there are drawbacks in the proposed method when sudden changes appear in QRS amplitudes. We speculate that further performance improvements are possible by the application of computationally more expensive QRS detection algorithms. Continuing this research, we will address ourselves to the study of designing an ECG compression algorithm utilizing the inter-beat correlation without miscellaneous preprocessing.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Comput. Netw., vol.38, no.4, pp.393–422, March 2002.
[2] S.M.S. Jalaleddine, C.G. Hutchens, R.D. Strattan, and W.A. Coberly, "ECG data compression techniques — A unified approach," IEEE Trans. Biomed. Eng., vol.37, no.4, pp.329–343, April 1990.
[3] B. Wang and G. Yuan, "Compression of ECG data by vector quantization," IEEE Eng. Med. Biol. Mag., vol.16, no.4, pp.23–26, July/Aug. 1997.
[4] C.C. Sun and S.C. Tai, "Beat-based ECG compression using gain-shape vector quantization," IEEE Trans. Biomed. Eng., vol.52, no.11, pp.1882–1888, Nov. 2005.
[5] A. Bilgin, M.W. Marcellin, and M.I. Altbach, "Compression of electrocardiogram signals using JPEG2000," IEEE Trans. Consum. Electron., vol.49, no.4, pp.833–840, Nov. 2003.
[6] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," IEEE Signal Process. Mag., vol.21, no.5, pp.80–94, Sept. 2004.
[7] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. Inf. Theory, vol.19, no.4, pp.471–480, July 1973.
[8] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," IEEE Trans. Inf. Theory, vol.49, no.3, pp.626–643, March 2003.
[9] MIT-BIH Arrhythmia Database, http://physionet.org/physiobank/database/mitdb/, accessed Sept. 16. 2012.
[10] Yang Ying Inc., "Bluetooth ECG monitor for research purpose," http://www.yangying.com.tw/en-products02-4.htm, accessed Oct. 15. 2013.
[11] H.H. So and K.L. Chan, "Development of QRS detection method for real-time ambulatory cardiac monitor," Proc. 19th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society, pp.289–292, Chicago, IL, 1997.
[12] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Massachusetts, 1991.
[13] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol.28, no.1, pp.84–95, Jan. 1980.
[14] C.E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol.27, pp.379–423, July 1948.
[15] Z. Tu, J. Li, and R.S. Blum, "An efficient SF-ISF approach for the Slepian-Wolf source coding problem," Eurasip J. Appl. Signal Process., vol.2005, no.6, pp.961–971, June 2005.
[16] W.C. Huffman, R.A. Brualdi, and V.S. Pless, Handbook of Coding Theory, Volume 1, Part 1: Algebraic Coding, Elsevier Science, New York, 1998.
[17] S. Lin and D.J. Costello, Error Control Coding, 2nd ed., Prentice-Hall, New Jersey, 2004.
[18] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inf. Theory, vol.20, no.2, pp.284–287, March 1974.
[19] Leadtek Research Inc., "Product Description: PXA 310 Mini (VGA)," http://www.leadtek.com.tw/pxa310mini/, accessed Oct. 15. 2013.

Hung-Tsai Wu was born in 1985. He received the B.S. degree in Communications Engineering from National Chiao-Tung University, Taiwan, in 2009. Currently, he is working toward the Ph.D. degree under the Institute of Communications Engineering, National Chiao-Tung University, Taiwan. His research interests include digital signal processing, biomedical engineering, and wireless communication.



Wei-Ying Tsai was born in 1987. He received the B.S. degree in Computer Science and Information Engineering from National Cheng-Kung University, Taiwan, in 2009, and the M.Eng. degree in Communications Engineering from National Chiao-Tung University, Taiwan, in 2012. Currently, he is a software engineer with the Engineering Department, Garmin Corporation, Taiwan.

Wen-Whei Chang was born in 1958. He received the B.S. degree in Communications Engineering from National Chiao-Tung University, Taiwan, in 1980, and the M.Eng. and Ph.D. degrees in Electrical Engineering from Texas A&M University, College Station, TX, USA, in 1985 and 1989, respectively. Since August 2001, he has been a professor with the Institute of Communications Engineering, National Chiao-Tung University, Taiwan. His research interests include biomedical signal processing, speech processing, joint source-channel coding, and wireless communication.