Compressive Wireless Pulse Sensing

Hsieh-Chung Chen, Harnek Gulati, H. T. Kung and Surat Teerapittayanon

Abstract— **Enabled by recent sensing and wireless chips, small wireless sensors can now be attached to various wearable devices such as finger rings and wrist bands. In this paper, we describe a compressive sensing based approach to such wireless devices that exploits sparsity of signal to reduce power consumption for both sensing and transmission. Using subsampling methods, we can lower sensor wake-up frequency and data transmission rate. We use a trained dictionary to recover the signal from the subsampled measurements. The same dictionary can also help recover from possible outliers in sensor measurements. In addition, to protect against burst packet loss over wireless channels, we rearrange packets and randomize their send orders. In order to demonstrate these concepts, we have built a prototype wearable system and report performance results from this experimental system.**

I. INTRODUCTION

With the emergence of the Internet of Things (IoT), wireless sensors are becoming more common. They are used to monitor and control things on our body, in our house, in the office and in public places like shopping malls. One restriction imposed on these wireless sensors is power. These sensing devices need to last a reasonable amount of time to avoid frequent battery replacements.

In this paper, we propose a compressive sensing based approach that exploits sparsity of signal to reduce power consumption during sensing and transmission in applications such as wireless wearable pulse sensing. We use subsampling to lower the sensor wake-up frequency and transmission rate and a dictionary to reconstruct the full signal. As opposed to traditional compressive sensing using data-oblivious orthonormal bases, our approach uses data-driven overcomplete dictionaries. This allows us to reconstruct signals in high fidelity and also shun outliers in sensor measurements. The dictionary based recovery is applicable to a variety of sensing problems as long as sufficient sensing examples are available for the dictionary training. In addition, we provide improved protection against lossy wireless channels by rearranging packets and randomizing their send orders. We demonstrate the approach with a prototype wireless pulse sensor.

Our approach is generally applicable to wearable sensor, where minimizing power consumption is essential. Applications can benefit from this approach, as long as the target signal is sparse in some domain spanned by atoms in a trained dictionary.

This paper is organized as follows. In the next section, we introduce background and related work. Then, we describe our proposed approach, followed by experiments. Finally, we discuss our results and conclude.

II. BACKGROUND AND RELATED WORKS

Traditionally, to reduce data transmission for sensors, the measurement data is first compressed and then sent. Compressing data with conventional methods is computationally intensive and can significantly reduce the battery life of wearable sensors. Recent studies have shown that we can avoid computationally expensive on-device compression by using a framework called compressive sensing.

By applying compressive sensing theories, researchers have shown that random sensing matrices preserve necessary information for signal recovery if the original signal can be sparsely representable in some orthogonal basis. In this case, the sensing matrices can provide low-cost encoding for compression [1], [2], [3]. In this paper, we extend this result to easily-implementable uniform subsampling via the use of learned dictionaries as overcomplete sparsifying bases in signal recovery.

To obtain a learned dictionary, we use dictionary learning algorithms such as K-SVD [4] which learns overcomplete sparsifying bases that can compactly represent the signals. These compact representations can then be recovered via sparse coding algorithms such as OMP [5].

For the pulse sensing application of this paper, a traditional platform derives certain information from the wearable device, such as heart rate, and only transmits the derived information, e.g., heart rate, to reduce the transmission power [6]. In contrast, by applying compressive sensing, dictionary learning and sparse coding theories, we can reconstruct the entire pulse wave, containing much more information than the derived information such as heart rate with a lower or comparable amount of power. This recovered pulse wave can in turn be used to derive other interesting information such as detecting blood pressure (BP) from multiple pulse wave forms [7] or detecting certain disease from analyzing the pulse wave form [8], [9].

III. PROPOSED APPROACH

This section lays out our proposed approach. We first describe the uniform subsampling sensing matrix that will be applied to sensor values under the compressive sensing framework. Then, we describe packet interleaving scheme and packet randomization to provide increased protection against burst packet loss over wireless channel. Finally, we describe how to recover the full signal values from our compressed measurements with a data-driven overcomplete dictionary.

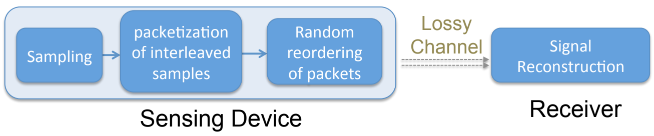


Fig. 1: Pulse Sensing Pipeline.

A. Signal Acquisition with Subsampling Using Compressive Sensing

Compressive sensing theory stipulates that if a signal $\mathbf{x}$ is sparse, that is, $\mathbf{x}$ can be approximated by $\mathbf{Dz}$ for some $\mathbf{z}$ with $\|\mathbf{z}\|_0 < k$ where k is a small sparsity constant[1] and $\mathbf{D}$ is an orthogonal sparsifying basis matrix, then the signal can be reconstructed from a compressing projection $\mathbf{y} = \Phi\mathbf{x}$ where Φ is a $m \times n$ random projection matrix with $m = O(k \log n) \ll n$ [2]. We demonstrate that pulse signal is indeed sparse in Section IV-B.1.

To handle a signal which arrives over time, we break it into segments of length n and denote them as $\mathbf{x}^{(i)}$. For compressive signal acquisition, we take uniformly spaced subsampled measurements:

$$\mathbf{y}_u^{(i)} = \Phi_u \mathbf{x}^{(i)} \tag{1}$$

where Φ_u consists of m uniformly spaced selected rows from the identity matrix $\mathbb{I}_n$. We call Φ_u the subsampling matrix. In Section V, we show that with our reconstruction method based on a learned overcomplete dictionary, the subsampling rate can be much lower than the Nyquist rate. This means that we can subsample the signal uniformly with little reconstruction error, allowing the sensor to wake up less often and send less data.

B. Packet Interleaving and Transmission Randomization

A conventional way of dealing with packet losses or corrupted packets is by using re-transmission scheme like Automatic Repeat Request (ARQ) [10]. However, this can increase power consumption and transmission delay, especially over lossy and congested wireless channels.

By exploiting correlation among signal values in a dictionary atom, we can recover lost or corrupted signal values without re-transmissions as long as enough signal samples are received to match up the atom.

To make our scheme more tolerant to bursty packet loss, a loss behavior usually observed over a wireless channel [11], we propose a sending scheme with sample interleaving (Fig. 2) and packet send order randomization (Fig. 3).

The sensor collects p subsampled measurements of $\mathbf{x}^{(i)}$ using Eq.(1) to form a $p \times m$ matrix $\mathbf{Y} = [\mathbf{y}_u^{(1)}, \mathbf{y}_u^{(2)}, \ldots, \mathbf{y}_u^{(m)}]^T$, where each row corresponds to a measurement vector $\mathbf{y}_u^{(i)}$. Packets are formed out of columns of $\mathbf{Y}$ as shown in Fig. 2. Then, these packets are sent out in random order to provide better protection against burst packet losses (Fig. 3). Note that with this scheme, a single packet loss is equivalent to losing just one (randomized) measurement within each $\mathbf{y}_u^{(i)}$.

[1] $\|\cdot\|_p$ denotes the ℓ_p norm.

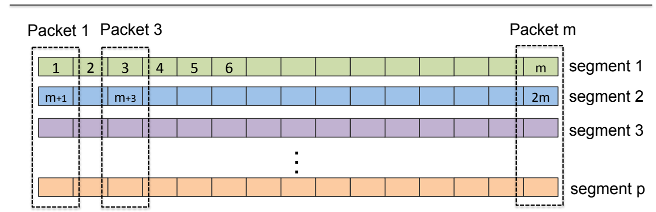


Fig. 2: Packetization of signal by interleaving signal subsamples over time, indexed with consecutive integers.

Thus, this scheme can provide better recovery against burst packet losses.

Through a lossy wireless channel, the receiver will receive measurements that passed through the channel $\mathbf{y}_r^{(i)} = \Phi_r \mathbf{y}_u^{(i)}$, where Φ_r is a channel loss matrix consisting of rows of $\mathbb{I}_m$ that correspond to the indices of the received packets. For example, if the first two packets are received successfully, Φ_r will contain the first two rows of $\mathbb{I}_m$. Note that when there is no packet loss, $\Phi_r = \mathbb{I}_m$.

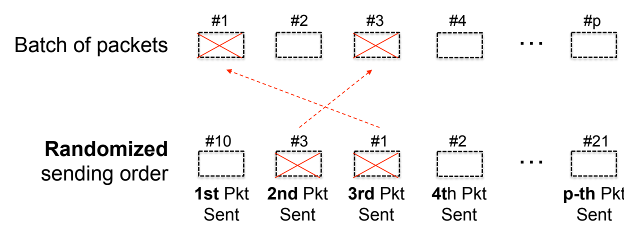


Fig. 3: Distributing burst packet loss over signal samples in different measurement vectors by sending packets in randomized order. Lost packets are marked with a red cross.

Under these mechanics, the receiver will receive a compressed signal

$$\mathbf{y}^{(i)} = \Phi \mathbf{x}^{(i)} \tag{2}$$

where $\Phi = \Phi_r \Phi_u$ with Φ_u and Φ_r being the subsampling and channel loss matrices, respectively, and $\mathbf{x}^{(i)}$ is the original signal segment that we want to recover. With this scheme, consecutive packets losses in transmission corresponds to random sample losses which can be recovered with compressive sensing.

C. Dictionary Learning and Signal Reconstruction

In this section, we describe how to recover $\mathbf{x}^{(i)}$ from $\mathbf{y}^{(i)}$ using a trained dictionary. We deviate from the standard compressive sensing setting with orthogonal bases, and train data-driven overcomplete dictionaries instead. Because trained dictionaries learn correlation patterns in the data, they allow for easy and aggressive compression schemes such as uniform subsampling. We will demonstrate this in the following experiment.

Dictionary $\mathbf{D}$ is trained to express signal samples $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}]$ with linear combinations of at most k atoms in $\mathbf{D}$. This $\mathbf{D}$ is computed by solving the following problem with K-SVD [4]:

$$\mathbf{D} = \underset{\hat{\mathbf{D}}}{\operatorname{argmin}} \|\mathbf{X} - \hat{\mathbf{D}}\mathbf{Z}\|_2 \text{ s.t. } \|\mathbf{z}^{(i)}\|_0 < k, \tag{3}$$

for some small sparsity constant k, where $\mathbf{Z} = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(t)}]$ and t is the number of samples. Note that since this problem is NP-hard in general, all efficient algorithms are based on heuristics. However, K-SVD has been shown to perform well for many problems and tends to converge faster than methods like MOD [12].

Given a trained dictionary $\mathbf{D}$, upon receiving $\mathbf{y}^{(i)}$ in Eq.(2), we can reconstruct an approximation $\hat{\mathbf{x}}^{(i)}$ for $\mathbf{x}^{(i)}$ with the following steps:

$$\hat{\mathbf{z}}^{(i)} = \underset{\mathbf{z}^{(i)}}{\operatorname{argmin}} \|\mathbf{y}^{(i)} - \Phi\mathbf{D}\mathbf{z}^{(i)}\|_2 \text{ s.t. } \|\mathbf{z}^{(i)}\|_0 < k \tag{4}$$

$$\hat{\mathbf{x}}^{(i)} = \mathbf{D}\hat{\mathbf{z}}^{(i)}. \tag{5}$$

There are many algorithms that can solve the sparse coding problem in (4), including IHT [13], CoSaMP [14] and OMP [5]. In this paper, we use OMP unless otherwise mentioned.

An alternative to solving (4) is to enforce sparseness of $\mathbf{z}^{(i)}$ using ℓ_1-norm penalty term:

$$\hat{\mathbf{z}}^{(i)} = \underset{\mathbf{z}^{(i)}}{\operatorname{argmin}} \|\mathbf{y}^{(i)} - \Phi\mathbf{D}\mathbf{z}^{(i)}\|_2 + \lambda\|\mathbf{z}^{(i)}\|_1 \tag{6}$$

where λ is a spareness penalty parameter. This can be solved with LARS [15]. The ℓ_1 formulation in (6) tends to yield better results than the ℓ_0 formulation in (4), but has a higher computation cost.

In the next section, we present an extension to this recovery scheme that deals with corrupted samples.

D. Extension to Handle Corrupted Samples

Sometimes, the sensor may have a few corrupted sensor readings that result in spike noises (outliers). For example, such noises can be caused by unintentional movements of a fingertip pulse sensor. We can handle this problem by expanding the dictionary to include bases for spike noise. More specifically, we concatenate $\mathbf{D}$ with the identity matrix $\mathbb{I}_m$, where m is the dimension of $\mathbf{y}^{(i)}$. The problem in Eq. (4) now becomes:

$$\tilde{\mathbf{z}}^{(i)}, \tilde{\mathbf{g}}^{(i)} = \underset{\mathbf{z}^{(i)}, \mathbf{g}^{(i)}}{\operatorname{argmin}} \left\| \mathbf{y}^{(i)} - \Phi \begin{bmatrix} \mathbf{D} & \mathbb{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{z}^{(i)} \\ \mathbf{g}^{(i)} \end{bmatrix} \right\|_2 \tag{7}$$
$$\text{s.t. } \|\mathbf{z}^{(i)}\|_0 < k \text{ and } \|\mathbf{g}^{(i)}\|_0 < e,$$

where e is the number of corrupted samples, $\begin{bmatrix} \mathbf{D} & \mathbb{I}_m \end{bmatrix}$ is the concatenated matrix of $\mathbf{D}$ and $\mathbb{I}_m$ and $\begin{bmatrix} \mathbf{z}^{(i)} & \mathbf{g}^{(i)} \end{bmatrix}^T$ is the concatenated matrix of $\mathbf{z}^{(i)}$ and $\mathbf{g}^{(i)}$. We use the same constant k that we use to find $\mathbf{D}$ from Eq. 3. In this equation, $\mathbf{g}^{(i)}$ is used to identify the outlier values that don't accurately represent the signal and isolate them from $\mathbf{z}^{(i)}$. After $\tilde{\mathbf{z}}^{(i)}$ and $\tilde{\mathbf{g}}^{(i)}$ are obtained, we discard $\tilde{\mathbf{g}}^{(i)}$ and use $\tilde{\mathbf{z}}^{(i)}$ to reconstruct $\hat{\mathbf{x}}^{(i)}$:

$$\hat{\mathbf{x}}^{(i)} = \mathbf{D}\tilde{\mathbf{z}}^{(i)}. \tag{8}$$

This process separates out the corrupted samples from the rest of the signal. Fig. 4 depicts the reconstruction process.

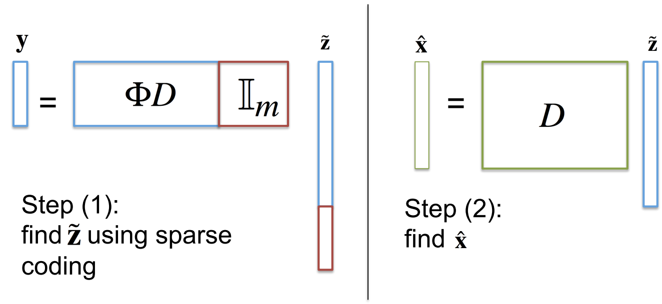


Fig. 4: Reconstruction with corrupted samples.

IV. Experiments and Analysis

In this section, we evaluate the proposed approach as described in Section III. First, we describe our experimental setup. This includes the description of our prototype sensing hardware and software system and how we collect data to analyze. Then we analyze our system in terms of 1) signal sparseness, 2) effects due to different settings in sensing matrix and compression rate, 3) reconstruction error as a function of channel loss rates, and 4) tolerance against outliers or corrupted samples.

A. Setup

We utilize off-the-shelf pulse sensing hardware and software components to capture and transmit the pulse wave signal. We use the LightBlue Bean available at `https://punchthrough.com/bean/` as our hardware platform. It is a Bluetooth Low Energy Arduino, making it easier for us to prototype our system. As for the pulse sensor, we use open pulse sensor hardware available at `http://pulsesensor.com/`.

Our pulse sensor hardware and software system is depicted in Fig. 5. The pulse sensor hardware is idle most of the time. The process begins with a smart phone sending commands to the pulse sensor hardware to collect measurement at what interval and how long. This allows us to specify the rate of uniform subsampling as described in Eq. 1. The sensor then collects the samples, interleaves the data values, forms multiple packets and sends those packets to the phone through a lossy wireless channel (as shown in Fig 5). The phone then reorganizes these packets in order and sends data to a remote server through HTTP protocol. Finally, the server reconstructs the full pulse wave signal based on the received packets. Fig. 6 shows our prototype hardware.

B. Analysis

In this section, we provide analysis on the following aspects of the system: signal sparseness; effects due to different settings in the sensing matrix and compression rate; interplay between subsampling rates and channel loss conditions; heart rate estimation; and tolerance to corrupted sensor measurement readings.

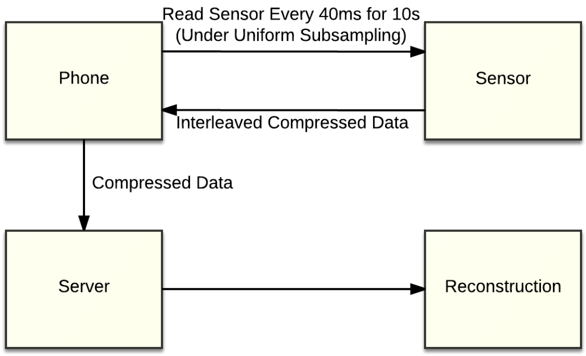


Fig. 5: Pulse sensing system overview. Phone sends a command to our wireless pulse sensing hardware. The sensor sends back the interleaved measurement through a lossy wireless channel. Phone reorganizes the interleaved measurement and uploads that to the server, where reconstruction occurs.

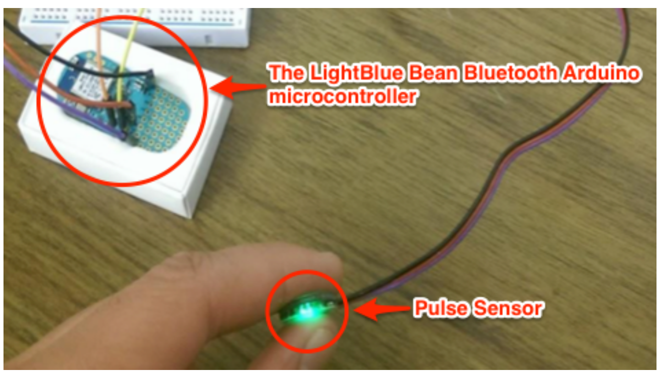


Fig. 6: Prototype wireless pulse sensing hardware showing the LightBlue Bean Bluetooth Arduino microcontroller and the pulse sensor.

1) Sparseness of pulse signal: We compare the reconstructed signal obtained from our system with the original signal obtained directly from the pulse sensor at 60Hz. We split the signal into t segments of 128 samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}$, and use sparse coding to find their sparse representations $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(t)}$, and compute $\tilde{\mathbf{x}}^{(i)} = \mathbf{D}\mathbf{z}^{(i)}$ for reconstruction. The dictionary $\mathbf{D}$ is learned from data collected earlier[2]. We then concatenate the reconstructed segments $\tilde{\mathbf{x}}^{(i)}$ to form the reconstruction of the entire signal. The reconstruction error is measured in the ℓ_2-norm.

Fig. 7 shows that the signal can be represented with small error while the number k of atoms required is small. When the dictionary $\mathbf{D}$ has 50 atoms and k is 15, the error is about .15. The error drops to about .10 when $\mathbf{D}$ has 250 atoms. These empirical results suggest that the pulse sensing signal is sparse in the domain spanned by trained dictionary atoms.

[2]The data could be collected from the same user to train a user-specific dictionary. In this paper we use data from multiple users to train a generic dictionary.

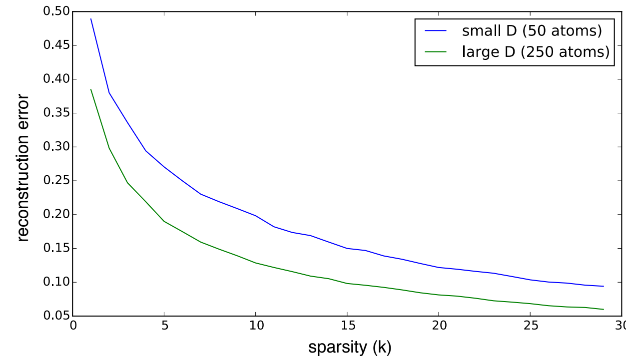


Fig. 7: The signal can be expressed as a linear combination of a small number k of dictionary atoms. This figure shows the reconstruction error (in the ℓ_2-norm) of recovered 128-sample signal segments when approximated by such linear combinations. Using larger size for dictionary $\mathbf{D}$ reduces the reconstruction error.

2) Comparison of sensing matrices: In this section, we compare uniform subsampling sensing matrix (Φ_u) with other sensing matrices: random subsampling sensing matrix (Φ_r), random linear combination sensing matrix (Φ_{RLC}) and a sensing matrix (Φ_{PCA}) computed by performing PCA on $\mathbf{D}$.

The random linear combination sensing matrix (Φ_{RLC}) has random entries drawn from a Gaussian distribution and has been proposed in compressive sensing literature to work with arbitrary orthogonal sparsifying bases. However, given $\mathbf{D}$ of a large number of rows, the optimal projection onto m vectors in terms of preserving distances between atoms is the projection onto the largest m eigenvectors of $\mathbf{D}\mathbf{D}^T$ [16]. We call this projection Φ_{PCA}. However, using a dense sensing matrix like Φ_{PCA} requires the sensor to perform linear combinations of samples. In contrast, the uniform subsampling matrix Φ_u as described in Section III-A is much simpler and cheaper to implement; it does not need any computation beyond a uniform selection of a subset of samples.

Fig. 8 shows that PCA on $\mathbf{D}$ sensing matrix performs best among these sensing matrices as expected, followed closely by the uniform subsampling sensing matrix. These empirical results show that we can use Φ_u instead of Φ_{PCA} without losing much reconstruction quality. The reason why there is only a small performance gap between Φ_{PCA} and Φ_u is discussed in Section V. Note that reconstruction with LARS results in less error, as discussed in Section III-C.

3) Interplay between subsampling rates and channel loss conditions: To understand the relationship between uniform subsampling and channel loss, we simulate reconstruction error under different subsampling and channel loss rates. As shown in Fig. 9, we can be quite aggressive with subsampling when channel loss is low. This suggests that our approach supports an adaptive sensing scheme where the sampling rate changes according to channel quality. This results from a nice property of compressive sensing recovery that it

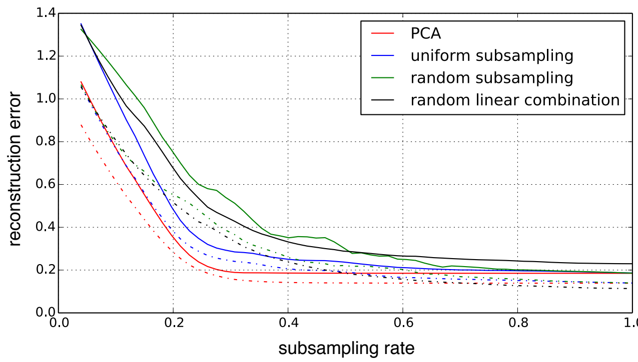


Fig. 8: Uniform subsampling is inferior to PCA by a small amount in terms of compression rate. However, uniform subsampling requires extremely little computation and is thus selected for our scheme. A dictionary of 50 atoms is used to generate this figure. Solid lines and dotted lines indicates reconstruction using OMP and LARS, respectively.

can reconstruct the signal as long as enough packets are successfully received over the channel. There is no need to re-transmit any particular lost packet.

4) Heart rate estimation based on recovered pulse signal: We consider the impact of reconstruction error on applications such as heart rate estimation. We use a simple local peak detector [17] to find beats in the signal (see Fig. 10 for an example), based on which we estimate heart rate by calculating beats per minute (bpm). For the area covered in blue in Fig. 9 where reconstruction error is less than 0.15, we can derive heart rate from the reconstructed signal reliably [3].

5) Tolerance against corrupted samples: Here, we evaluate our proposed scheme in terms of tolerance against corrupted samples (outliers). One of such examples based on real sensor data from our experimental system is shown in Fig. 10 where a dip is caused by a sudden unintentional movement of the sensor on a fingertip. The figure shows that this outlier is smoothed out in the reconstructed signal using the reconstruction method depicted in Fig 4. As illustrated in the figure, this means the heart rate can still be accurately estimated from the recovered signal in spite of the presence of the corrupted sensor data sample.

We further simulate scenarios with extreme signal corruption. We create a corrupted signal by randomly modifying a few of its samples to the highest or lowest observed value. Fig. 11 shows that our reconstruction can tolerate multiple corrupted samples without losing much accuracy, even at a high compression rate, like 0.3.

This protection against corrupted samples exploits the correlation between local signal values captured in the trained dictionary atoms to automatically recover one or more corrupted signal values. The correct signal values usually have stronger correlation with the values trained in the dictionary atoms than the corrupted signal values.

[3] Less than 1% error in terms of bpm on average.

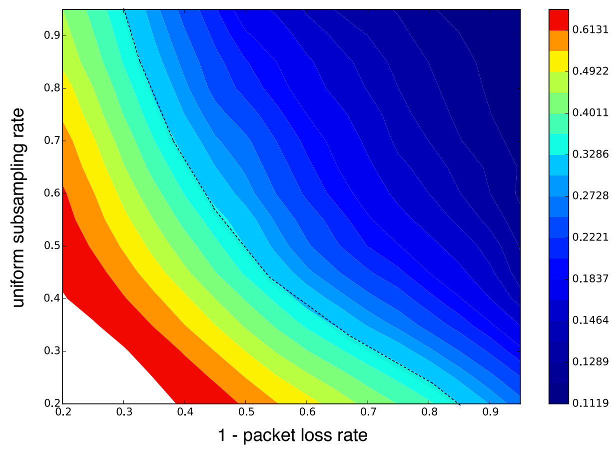


Fig. 9: Reconstruction error at different subsampling rates and channel loss conditions, where colors depict reconstruction errors of increasing magnitude. The blue region above the dashed line is of sufficient reconstruction quality (reconstruction error < 0.15) for heart rate estimation. For a given channel loss rate, we can increase the subsampling rate to ensure the reconstruction error is below a target threshold.

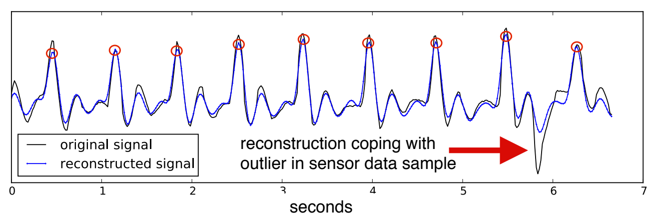


Fig. 10: A segment of pulse sensing signal overlaid with the reconstructed signal from sparse codes (number of of dictionary atoms = 250, $k = 10$). The reconstructed signal is nearly identical to the original signal. The red circle marks peaks detected from the reconstructed pulse waveform. Heart rate can be estimated by counting the number of peaks over a period of time and converting it to beats per minute (bpm). The reconstruction has a dip around the 5.8 seconds mark, which reflects a corrupted sample. Heart rate can still be estimated with about the same accuracy from this reconstructed signal.

This allows us to derive the correct sparse codes and use the codes to reconstruct close approximations to the correct signal values to replace the corrupted ones.

V. DISCUSSION

In this section, we elaborate on design decisions and justifications for our proposed approach.

A. *Justification for uniform subsampling*

We choose uniform subsampling over other sensing matrices for two main reasons:

1) Uniform subsampling requires less computation and thus makes it easier to implement on wearable sensing hardware,

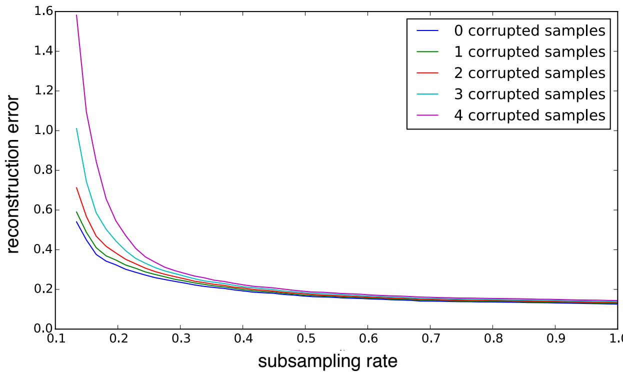


Fig. 11: Tolerance against corrupted samples. Reconstruction errors of the pulse wave signals when there are a number of corrupted samples under various subsampling rates.

2) Uniform subsampling works well with a trained dictionary.

The first point is quite clear. Uniform subsampling only needs to read certain values once in a while, and thus the sensor can stay off most of the time.

In regards to the second point, we provide a mathematical proof in Section V-B below. We argue that uniform subsampling works well with a trained dictionary because a subsampled signal and a subsampled dictionary atom lie in similar low frequency bands. Uniform subsampling of a signal input removes certain high-frequency components. Similarly, dictionary atoms are obtained by averaging a number of signal inputs via clustering algorithms such as K-means or K-SVD. This averaging operator removes certain high-frequency components.

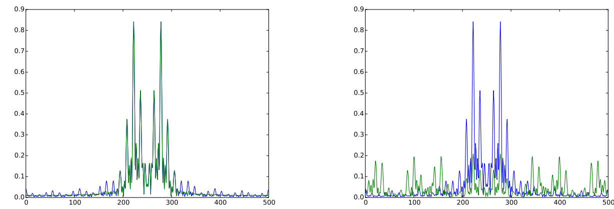
By comparing a subsampled signal input with subsampled dictionary atoms, the input signal has a better chance of matching with the right atom since they lie in similar frequency bands (low-frequency components). In contrast, other sensing matrices, such as one which leads to a randomly subsampled input signal, may have frequency components that are less useful for differentiating dictionary atoms. This frequency discrepancy creates noises when comparing the input signal with dictionary atoms.

Nonetheless, the choices of sensing matrices have to be decided based on the needs. For example, if we want to minimize the reconstruction errors and are willing to spend some power to do the calculation, we may choose PCA as our sensing matrix based on the result of Fig. 8.

B. Proof for uniform subsampling

We first show that $\Phi_{\text{PCA}}\mathbf{x}$ has approximately the same sparse code as $\mathbf{x}$. Then, we apply the same argument to the uniform subsampling sensing matrix.

We note that 1) since dictionary atoms $\mathbf{d}$ are derived by averaging, they have essentially no high-frequency components and 2) input $\mathbf{x}$ has essentially no high-frequency components either, since biological processes that affect the pulse occur at a comparatively low frequency. Since Φ_{PCA} is derived from



(a) Φ_{PCA} (b) Φ_u

Fig. 12: Frequency spectrum of $\mathbf{y}$ from two sensing matrices Φ_{PCA} and Φ_u. The optimal projection Φ_{PCA} preserves low frequency components, where inter-atom variations lie; Φ_u preserves the same frequency band of interest.

$\mathbf{D}$, it also tends to preserve low-frequency components, as shown in Fig. 12.

Consider any dictionary atom $\mathbf{d}$. Expressing $\mathbf{x}-\mathbf{d}$ in terms of orthonormal frequency basis vectors $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \ldots$ with increasing frequencies, we have that

$$\mathbf{x}-\mathbf{d} = c_1\mathbf{f}_1 + c_2\mathbf{f}_2 + c_3\mathbf{f}_3 + \ldots,$$

where coefficients $c_i \approx 0$ for $i > \tau$ for some small τ. In the rest of the proof we assume $\tau = 3$ to simplify the description. It is easy to see that the proof works for other values of τ. Since Φ_{PCA} has essentially no high-frequency components, say, its top principle components drop c_i, for $i > 3$ also, we have

$$\Phi_{\text{PCA}}(\mathbf{x}-\mathbf{d}) = c_1\mathbf{f}_1 + c_2\mathbf{f}_2 + c_3\mathbf{f}_3,$$

and

$$\|\Phi_{\text{PCA}}(\mathbf{x}-\mathbf{d})\|_2 = c_1^2 + c_2^2 + c_3^2.$$

Now, note that

$$\|\mathbf{x}-\mathbf{d}\|_2 = c_1^2 + c_2^2 + c_3^2 + \ldots,$$

and $c_i \approx 0$ for $i > 3$. Thus, $\|(\mathbf{x}-\mathbf{d})\|_2$ is approximately equal to $\|\Phi_{\text{PCA}}(\mathbf{x}-\mathbf{d})\|_2$ That is, Φ preserves distance. This means that $\Phi_{\text{PCA}}\mathbf{x}$ has approximately the same sparse code as $\mathbf{x}$, when the subsampled dictionary $\Phi_{\text{PCA}}\mathbf{D}$ is used.

We now apply the proof to uniform subsampling sensing matrix. First, we note that the uniform subsampling sensing matrix Φ_u can be regarded as a low-pass filter that captures all relevant information (see Fig. 12)[4]. When we multiply Φ_u to $\mathbf{x}-\mathbf{d}$, we have

$$\|\Phi_u(\mathbf{x}-\mathbf{d})\|_2 \sim c_1^2 + c_2^2 + c_3^2.$$

Thus, $\|(\mathbf{x}-\mathbf{d})\|_2$ is approximately equal to $\|\Phi_u(\mathbf{x}-\mathbf{d})\|_2$ That is, Φ_u approximately preserves distance. This means that $\Phi_u\mathbf{x}$ has approximately the same sparse code as $\mathbf{x}$. This result about uniform subsampling sensing matrix is somewhat surprising and should be noted. This would allow us to create a simple and efficient sensing scheme for next-generation wearable devices.

[4]This would hold even if $\mathbf{x}$ had high-frequency components.

VI. CONCLUSION

We have proposed a novel approach to sensing and transmitting pulse signals based on compressive sensing, dictionary learning and sparse coding theories. Our method saves battery power of wearable sensors by lowering sensor wake-up frequency and reducing measurement data to transmit.

The contribution of this paper is the integration of the following elements for wireless sensors: 1) sub-Nyquist uniform sampling for signal acquisition, 2) signal reconstruction with overcomplete dictionaries, 3) interleaving and randomized transmission to enhance signal reconstruction under packet loss, and 4) error correction for corrupted samples. Note that all of these exploits the sparseness of signal, and can be explained using the same sparse coding framework.

We have demonstrated the feasibility of this proposed approach by building a working prototype for high-fidelity pulse signal reconstruction. Future work includes applying our approach to other sensors such as wireless wearable blood pressure devices. With the emergence of IoT, there will be many scenarios where our approach can be applied.

ACKNOWLEDGMENTS

This research is supported in part by gifts from the Intel Corporation.

REFERENCES

[1] R. Baraniuk, "Compressive sensing," *IEEE signal processing magazine*, vol. 24, no. 4, 2007.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[4] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: Design of dictionaries for sparse representation," *Proceedings of SPARS*, vol. 5, pp. 9–12, 2005.

[5] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[6] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, "Wearable photoplethysmographic sensorspast and present," *Electronics*, vol. 3, no. 2, pp. 282–302, 2014.

[7] G. Fortino and V. Giampa, "Ppg-based methods for non invasive and continuous blood pressure measurement: an overview and development issues in body sensor networks," in *IEEE International Workshop on Medical Measurements and Applications Proceedings (MeMeA)*, 2010, pp. 10–13.

[8] S. Sarkar, A. K. Bhoi, and G. Savita, "Fingertip pulse wave (ppg signal) analysis and heart rate detection," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 9, pp. 404–408, 2012.

[9] P. Pradhapan, M. Swaminathan, H. K. S. V. Mohan, and N. Sriraam, "A novel detection approach for cardio-respiratory disorders using ppg signals," *International Journal of Biomedical and Clinical Engineering (IJBCE)*, vol. 1, no. 2, pp. 13–23, 2012.

[10] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Communications Magazine*, pp. 5–17, 1985.

[11] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.

[12] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 1999, pp. 2443–2446.

[13] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[14] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[15] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[16] I. Gkioulekas and T. Zickler, "Dimensionality reduction using the sparse linear model," *Advances in Neural Information Processing Systems*, pp. 1–9, 2011.

[17] G. Palshikar *et al.*, "Simple algorithms for peak detection in time-series," in *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*, 2009.