Simultaneous Greedy Analysis Pursuit for Compressive Sensing of Multi-Channel ECG Signals

Yurrit Avonds, Yipeng Liu*, *Member, IEEE*, and Sabine Van Huffel, *Fellow, IEEE*

***Abstract*— This paper addresses compressive sensing for multi-channel ECG. Compared to the traditional sparse signal recovery approach which decomposes the signal into the product of a dictionary and a sparse vector, the recently developed cosparse approach exploits sparsity of the product of an analysis matrix and the original signal. We apply the cosparse Greedy Analysis Pursuit (GAP) algorithm for compressive sensing of ECG signals. Moreover, to reduce processing time, classical signal-channel GAP is generalized to the multi-channel GAP algorithm, which simultaneously reconstructs multiple signals with similar support. Numerical experiments show that the proposed method outperforms the classical sparse multi-channel greedy algorithms in terms of accuracy and the single-channel cosparse approach in terms of processing speed.**

***Index Terms*— compressive sensing, cosparsity, greedy analysis pursuit, multi-channel ECG.**

I. Introduction

Mobile ECG monitors allow patients to be continuously monitored, even when they are not in a hospital. They have limited available energy, which limits the amount of data they can transmit. Compression can be used to reduce the amount of transmitted data over these energy-hungry wireless body sensor network links. Currently, the signal is sampled at a considerably high sampling rate and then compressed, which requires temporary storage of a large amount of uncompressed data before reducing it.

Compressive sensing (CS) provides a solution by compressing while sampling, avoiding the high sampling rate and storage requirements imposed by the Nyquist rate. For illustration convenience, the sampling model is formulated in the discrete setting as $\mathbf{y} = \boldsymbol{\phi}\mathbf{x}$, where $\boldsymbol{\phi} \in \mathbb{R}^{n\times N}$ and $n < N$, making measurement $\mathbf{y}$ a compressed form of signal $\mathbf{x}$.

To exploit sparsity for signal recovery, a signal $\mathbf{x}$ can be represented as $\mathbf{x} = \boldsymbol{\psi}\mathbf{s}$, i.e. a linear combination of the columns from a dictionary or basis $\boldsymbol{\psi} \in \mathbb{R}^{n\times M}$. If $\mathbf{s}$ contains a small number of nonzero elements, it is said to be a sparse vector. To estimate the sparse vector $\mathbf{s}$ from $\mathbf{y}$, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, we can formulate the synthesis-based ℓ_0-minimization problem:

$$\min_{\mathbf{s}} ||\mathbf{s}||_0, \quad s.\,t. \quad \mathbf{y} = \boldsymbol{\phi}\boldsymbol{\psi}\mathbf{s} \tag{1}$$

where $||\mathbf{s}||_0$ counts the number of nonzero elements in $\mathbf{s}$.

Recently, a cosparse approach was introduced, stating that an analysis matrix $\boldsymbol{\Omega} \in \mathbb{R}^{p\times N}$ can provide a sparse representation $\boldsymbol{\Omega x}$. [1] In order to recover $\mathbf{x}$, an analysis-based ℓ_0-minimization problem can be formulated:

$$\min_{\mathbf{x}} ||\boldsymbol{\Omega}\mathbf{x}||_0, \quad s.\,t. \quad \mathbf{y} = \boldsymbol{\phi}\mathbf{x} \tag{2}$$

To solve the cosparse approach based nonconvex programming problem (2), there are two popular ways. The Analysis-by-Synthesis (ABS) method is a convex relaxation that rewrites the problem in the form of a synthesis-based problem [2]. Greedy Analysis Pursuit (GAP) is a greedy algorithm that is the analysis duality of Orthogonal Matching Pursuit (OMP) [1]. Other algorithms include Analysis IHT (AIHT), analysis HTP (AHTP), analysis CoSaMP (ACoSaMP) and Analysis SP (ASP). The four greedy-like algorithms are developed corresponding to their synthesis counterparts in an analysis setting [3], [4].

Compared to previous works on CS for single-channel (SC) ECG [5], [6], the processed data is large-scale, since multiple channels are processed simultaneously. Assuming channels share a similar support [7], multi-channel (MC) algorithms can be used to recover the MC ECG signals with reduced computational complexity. To the best of our knowledge, all current MC methods, such as simultaneous orthogonal matching pursuit (SOMP) [8], the multiple response extension of the standard Sparse Bayesian Learning (M-SBL) [9], and Reduce MMV and Boost (ReMBO) [10], are based on a sparse representation of the synthesis form. However, our research shows that the cosparsity based convex relaxation gives higher reconstruction accuracy for ECG signals than the classical sparse convex relaxation methods [5].

In this paper, the SC GAP is generalized to process MC signals to reduce the total processing time for all channels. The algorithm is used for simultaneous reconstruction of MC ECG signals. The choice of a greedy algorithm is motivated by its lower computational complexity which is an important advantage for wireless ECG monitoring. Numerical experiments show that the proposed algorithm improves reconstruction accuracy without increasing computational complexity compared to sparse MC ECG signal reconstruction and reconstructs multiple ECG channels faster than the SC GAP algorithm without reducing the reconstruction accuracy.

This work was supported by Research Council KUL GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC); FWO project G.0108.11 (Compressed Sensing);iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAP P7/19/ (DYSCO, 'Dynamical systems, control and optimization', 2012-2017); EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013), ERC Advanced Grant: BIOTENSORS (n 339804). *Asterisk indicates corresponding author.*

All the authors are with ESAT-STADIUS / iMinds Future Health Department, Dept. of Electrical Engineering, KU Leuven, Kasteelpark Arenberg 10, box 2446, 3001 Leuven, Belgium. email: yurrit.avonds@student.kuleuven.be; {yipeng.liu;sabine.vanhuffel}@esat.kuleuven.be

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

The remaining part of the paper is organized as follows. In Section II the new reconstruction algorithm is presented. In Section III the performed experiments are described. In Section IV the results of these experiments are presented. Finally, in Section V a conclusion about the results is drawn.

II. Methods

In this section, we first improve GAP by introducing some precalculations. Next, the Simultaneous Greedy Analysis Pursuit (SGAP) method is introduced.

A. GAP

While OMP searches the support of the sparse vector $\mathbf{s}$ to give a sparse estimate [11], the GAP algorithm finds the co-support (collection of zero elements) of $\mathbf{\Omega x}$. This is done by first initialising a full co-support $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$. In each iteration, the elements in the co-support that correspond to large values in the product of $\mathbf{\Omega}$ and the current estimate of $\hat{\mathbf{x}}$ are removed from the co-support estimate. Ideally, this leads to the removal of all nonzero elements from the co-support estimate [1].

During the co-support update, only the index of the largest element in $\boldsymbol{\alpha} = \Omega\hat{\mathbf{x}}_{k-1}$ is selected and removed from the co-support. It is also possible to select the indices corresponding to the t largest values in $\boldsymbol{\alpha}$. In this paper, t is set to 10. The maximum number of iterations K_{max} is set as $\lfloor (p-t)/t \rfloor$.

The second stopping criterion checks whether the norm of the solution estimate relative to the previous estimate is larger than in the previous iteration, since this indicates a decrease in reconstruction accuracy.

In [1], it is stated that

$$\hat{\mathbf{x}_k} = \begin{bmatrix} \boldsymbol{\phi} \\ \sqrt{\lambda}\mathbf{\Omega}_{\hat{\Lambda}_k} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} = (\boldsymbol{\phi}^T\boldsymbol{\phi} + \lambda\mathbf{\Omega}_{\hat{\Lambda}_k}^T\mathbf{\Omega}_{\hat{\Lambda}_k})^{-1}\boldsymbol{\phi}^T\mathbf{y} \quad (3)$$

with λ a small positive constant, set to $\lambda = 0.05$. Since neither $\boldsymbol{\phi}$ nor $\mathbf{y}$ change their value during the reconstruction process, the result of two matrix multiplications ($\boldsymbol{\phi}^T\boldsymbol{\phi}$ and $\boldsymbol{\phi}^T\mathbf{y}$) can be precalculated. This results in a shorter processing time, especially when $\boldsymbol{\phi}$ is large.

B. SGAP

SGAP extends GAP for MC measurements, assuming that the ECG signals in all channels share the same co-support, since each channel can be seen as a projection of the cardiac activity towards the electrode associated with the channel [12]. All MC signals $\mathbf{x}$ and measurements $\mathbf{y}$ are now represented by capital letters ($\mathbf{X}$, $\mathbf{Y}$) to indicate that they are matrices instead of vectors.

To generalize the GAP, several steps should be adapted. The first difference is the addition of precalculations for speed improvement, as discussed in section II-A. Secondly, since $\boldsymbol{\alpha} = \Omega\hat{\mathbf{X}}$ is now a matrix, a different approach to find positions of nonzero elements in the co-support is required. A row-wise summation of $\boldsymbol{\alpha}$ creates a vector with large values at positions that correspond to elements that are not part of the co-support of some or all of the channels [8]. Finally, the second stopping criterion is adapted. Using a column-wise norm calculation, a vector with one value representing the change in norm compared to the previous solution estimate for each channel can be obtained. The algorithm iterates until the residual for at least one of the current channel approximations becomes larger than in the previous iteration.

Like the GAP algorithm, 10 elements are removed from the co-support in each iteration. K_{max} is set like in the GAP algorithm as well. The complete SGAP algorithm is presented as Algorithm 1.

Algorithm 1: Simultaneous Greedy Analysis Pursuit

- **In:** $\mathbf{Y} \in \mathbb{R}^{n\times c}$, $\boldsymbol{\phi} \in \mathbb{R}^{n\times N}$, $\mathbf{\Omega} \in \mathbb{R}^{p\times N}$
- **Out:** $\hat{\mathbf{X}} \in \mathbb{R}^{N\times c}$
- Initial Co-Support: $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$
- Initial Solution: $\hat{\mathbf{X}}_k = (\boldsymbol{\phi}_\phi + \lambda\mathbf{\Omega}^T\mathbf{\Omega})^{-1}\boldsymbol{\phi}_Y$
- Precalculations $\boldsymbol{\phi}_\phi = \boldsymbol{\phi}^T\boldsymbol{\phi}$; $\boldsymbol{\phi}_Y = \boldsymbol{\phi}^T\mathbf{Y}$

repeat

- $k := k + 1$
- Analysis Representation: $\boldsymbol{\alpha} = \Omega\hat{\mathbf{X}}_{k-1}$
- Row-Wise Summation: $\boldsymbol{\alpha}_R = \boldsymbol{\alpha}.\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^T$
- Update Co-Support: $\hat{\Lambda}_k = \hat{\Lambda}_{k-1} \backslash \{\underset{i\in\hat{\Lambda}_{k-1}}{argmax}\ |\boldsymbol{\alpha}_{Ri}|\}$
- Update Solution: $\hat{\mathbf{X}}_k = (\boldsymbol{\phi}_\phi + \lambda\mathbf{\Omega}_{\hat{\Lambda}_k}^T\mathbf{\Omega}_{\hat{\Lambda}_k})^{-1}\boldsymbol{\phi}_Y$, where $\mathbf{\Omega}_{\hat{\Lambda}_k}$ is $\mathbf{\Omega}$ with rows not in $\hat{\Lambda}_k$ set to 0

until $k \geq K_{max}$ *or* $\mathbf{r}_k(i) > \mathbf{r}_{k-1}(i)$ *for at least one* $i \in \{1, ..., c\}$;

where $\mathbf{r}_k(i) = 1 - \frac{||\hat{\mathbf{X}}_k(i)||_2}{||\hat{\mathbf{X}}_{k-1}(i)||_2}$ for $i \in \{1, ..., c\}$ and $\mathbf{X}_k(i)$ is the i-th column in the solution $\mathbf{X}$ of the k-th iteration.

III. Numerical Experiments

In the experiments, the MIT-BIH Arrhythmia Database [13], [14], a commonly used ECG database, is used. It consists of 2-channel ECG recordings from 48 patients, sampled at 360 Hz. A total of 943 2-second segments of the first 23 patients in the database (the first 41 segments of each patient) are compressed and subsequently reconstructed. Segments of only 2 seconds are used to keep the processing time per segment relatively low.

In order to compare the performance of SGAP to a corresponding sparse algorithm, SOMMP - a faster version of SOMP where multiple dictionary elements are selected in each iteration - with a Daubechies wavelet dictionary is used. Moreover, separate channel reconstructions by OMMP and GAP are used to demonstrate the shorter processing times of MC algorithms. Both SOMMP and OMMP select 4 support elements per iteration, since it was empirically found that this renders the most accurate reconstructions for the dataset.

Different from the one in [5], the analysis matrix used in these experiments is a second order derivative matrix, obtained by multiplying 2 first order derivative matrices. This type of matrix renders faster and more qualitative results than

a first order derivative matrix.

$$\mathbf{\Omega}_2 = \mathbf{\Omega}_1^2 = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & 1 & -1 \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix}^2 \tag{4}$$

The amount of applied compression is expressed as the compression ratio $CR = n/N$. Values of CR ranging from 0.9 to 0.2 in steps of 0.1 are used in the experiments. In order to quantify the reconstruction accuracy, the percentage root-mean-square difference (PRD) was used [6].

$$PRD = \frac{||x - \hat{x}||_2}{||x||_2} \tag{5}$$

To quantify the speed of the algorithm, the processing time until convergence is measured. For OMMP and GAP, these are calculated as the respective sums of the processing times from the separate channel reconstructions.

All experiments were performed in MATLAB®2013a, on a quadcore 3.10GHz Intel®Core™i5-3450 system with 8GB RAM, running CentOS 6.4 with Linux kernel version 2.6.32.

IV. Results

All of the results are presented as boxplots with five characteristic values. These are, in increasing order of magnitude: lower whisker (LOW), 25th percentile ($P25$), median (MED), 75th percentile ($P75$) and higher whisker ($HIGH$), where $LOW = P25 - w.(P75 - P25)$ and $HIGH = P75 + w.(P75 - P25)$, with $w = 1.5$. Outliers (red crosses) are values outside the $[LOW, HIGH]$ range.

A. Reconstruction Speed

The processing times and number of iterations until convergence are presented in Fig. 1. The (S)GAP requires only a few iterations to converge compared to (S)OMMP. This is because the co-support estimate in (S)GAP is merely used as a condition for the inversion problem and therefor does not have to be very close to the actual co-support. The support estimate in (S)OMMP, on the other hand, actually determines which elements in the dictionary are used for the signal estimate and therefor needs to be close to the actual support for the algorithm to converge to a solution.

Note that during each iteration 4 elements are added to the support in (S)OMMP, while in (S)GAP 10 elements are removed from the co-support. These numbers should be equal for a comparison of the number of iterations of both algorithms, though the goal of this paper is to compare their optimal reconstruction accuracy.

Despite the large difference in the number of iterations, SOMMP is only slightly slower than SGAP, though SGAP processing times vary less than those of SOMMP. This is because SOMMP solves a smaller inversion problem in each iteration, since it slowly builds a support starting from an empty one and only estimates coefficients corresponding to the support. SGAP, on the other hand, removes elements from an initally full co-support and uses this knowledge to solve a large inversion problem to find the best possible estimate of the original signal.

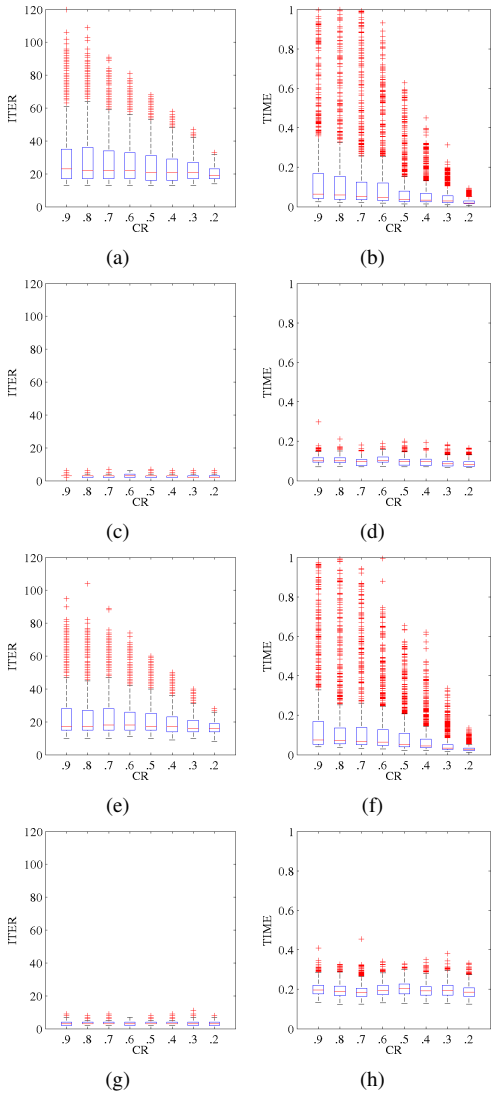

Fig. 1: Number of iterations (left) and processing times (right) until convergence of ((a),(b)) SOMMP, ((c),(d)) S-GAP, ((e),(f)) OMMP and ((g),(h)) GAP as function of CR.

With decreasing CR, the total number of iterations of SOMMP decreases due to the fact that its stopping criterion depends on the norm of the residual, which reduces with each iteration. This residual, which is based on the original measurement, will be smaller to begin with and reduces to a sufficiently small value faster at higher CR. In SGAP, the number of iterations remains similar for each CR value.

The total processing time required to process both channels by GAP or OMMP is significantly higher than the time required by SGAP or SOMMP to process both channels at once. This indicates that it is indeed beneficial in terms of processing times to use an MC algorithm instead of an SC algorithm to process multiple channels. The reduction in time when using SGAP instead of GAP is also larger than the reduction obtained by using SOMMP instead of OMMP.

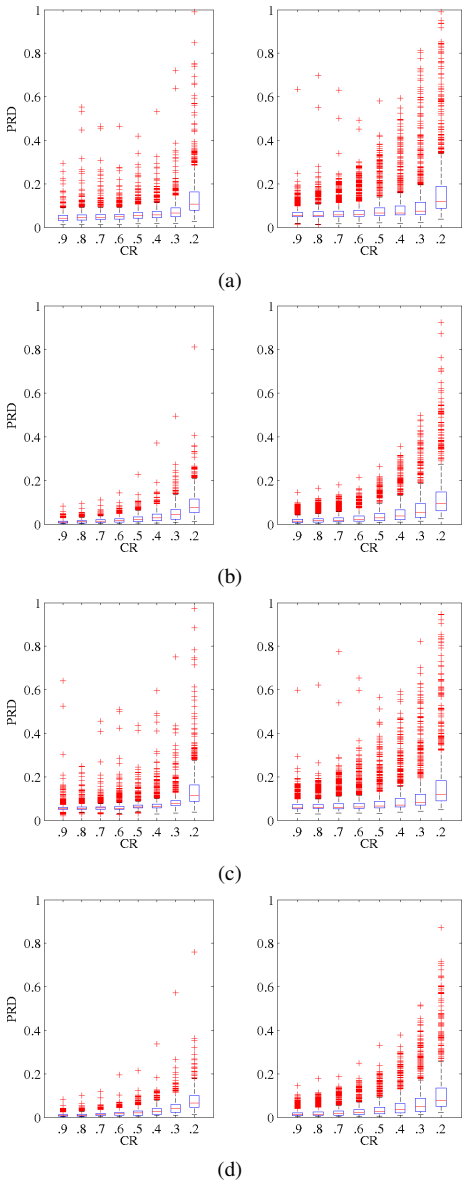

Fig. 2: PRD values of (a) SOMMP, (b) SGAP, (c) OMMP and (d) GAP as a function of CR for the 1st channel (left) and 2nd channel (right).

B. Reconstruction Accuracy

PRD values of all algorithms are presented for both channels in Fig. 2. The PRD values of the cosparse algorithms are far below those of the sparse algorithms at all CR values. As expected, the quality decreases with increasing CR values for all algorithms. The sparse algorithms also show more outliers compared to their cosparse counterparts.

Both MC algorithms show little difference in reconstruction accuracy, compared to their SC equivalents, although joint sparsity is expected to increase the accuracy. This is because it is assumed that all channels share the same support. In ECG, the support of the projections on each lead may differ because of the different electrodes positions. Still, because each projection originates from a common source (the heart), the support will be similar enough to apply CS, but the benefits of using joint sparsity will be lost.

These results indicate that MC algorithms could be an interesting alternative to their SC versions, as long as the (co-)support of the different channels is similar. They also indicate that using cosparse algorithms in ECG applications is in fact advantageous to the use of their sparse counterparts.

V. Conclusion

In this paper, a cosparse algorithm for simultaneous reconstruction of MC ECG from CS measurements, was presented. It is shown that the algorithm reduces processing times, compared to the equivalent SC algorithm. Despite the fact that the mean processing time of SGAP is slightly higher than that of SOMMP, SGAP processing times vary less.

Simultaneous reconstruction of an MC ECG signal could also be achieved by running multiple GAP algorithms in parallel. However, it would require significantly more processing power, which is not always available.

References

[1] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[2] N. Cleju, M. G. Jafari, and M. D. Plumbley, "Analysis-based sparse reconstruction with synthesis-based solvers," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, March 25–30, 2012, pp. 5401–5404.

[3] R. Giryes, S. Nam, M. Elad, R. Gribonval, and M. E. Davies, "Greedy-like algorithms for the cosparse analysis model," *Linear Algebra and its Applications*, 2013.

[4] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "Cosparse analysis modeling-uniqueness and algorithms," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5804–5807.

[5] Y. Liu, M. De Vos, I. Gligorijevic, V. Matic, Y. Li, and S. Van Huffel, "Multi-structural signal recovery for biomedical compressive sensing," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2794–2805, 2013.

[6] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 9, pp. 2456–2466, 2011.

[7] J. M. Kim, O. K. Lee, and J. C. Ye, "Compressive music: revisiting the link between compressive sensing and array signal processing," *Information Theory, IEEE Transactions on*, vol. 58, no. 1, pp. 278–301, 2012.

[8] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[9] D. P. Wipf, "Bayesian methods for finding sparse representations," Ph.D. dissertation, UC San Diego, 2006.

[10] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4692–4702, 2008.

[11] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[12] R. Sameni, M. Shamsollahi, and C. Jutten, "Multi-channel electrocardiogram denoising using a bayesian filtering framework," in *Computers in Cardiology*. Valencia, Spain: IEEE, September 17–20, 2006, pp. 185–188.

[13] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 3, pp. 45–50, 2001.

[14] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13).