34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

High Performance Biomedical Time Series Indexes Using Salient Segmentation*

Jonathan Woodbridge[1], Bobak Mortazavi[1], Alex A.T. Bui[2], and Majid Sarrafzadeh[1]

Abstract— **The advent of remote and wearable medical sensing has created a dire need for efficient medical time series databases. Wearable medical sensing devices provide continuous patient monitoring by various types of sensors and have the potential to create massive amounts of data. Therefore, time series databases must utilize highly optimized indexes in order to efficiently search and analyze stored data. This paper presents a highly efficient technique for indexing medical time series signals using Locality Sensitive Hashing (LSH). Unlike previous work, only salient (or interesting) segments are inserted into the index. This technique reduces search times by up to 95% while yielding near identical search results.**

I. INTRODUCTION

The advent of remote and wearable medical sensing has created a dire need for efficient medical time series databases. Wearable medical sensing devices provide continuous patient monitoring by various types of sensors, such as accelerometers for activity monitoring; electrocardiogram (ECG) for heart monitoring; and pulse oximeters for blood oxygen saturation monitoring. These devices have the potential to create massive amounts of data. For example, there are currently over 3 million people worldwide implanted with a pacemaker [1]. If these systems had the ability to gather, store, and transmit a continuous ECG signal, we could expect to receive over 560 terabytes of data per day, just from such individuals (assuming a three channel ECG, sampling rate of 360 Hz, and 2 byte ADC). Therefore, time series databases must utilize highly optimized indexes in order to search and mine stored data.

Locality Sensitive hashing (LSH) [2] is one technique for indexing high dimensional objects (such as medical time series signals). Searches on LSH indexes have a provable sub-linear computational complexity with respect to the size of the database. This is unlike spatial indexes that have been shown both theoretically and experimentally to perform worse than linear scan with a sufficient number of dimensions ($D > 10$) [3]. LSH is a probabilistic hashing method with the property that similar objects have a higher probability of collision. Searches using LSH return both matches and non-matches and filters the non-matches through pruning. Authors in [4] showed that the search complexity of LSH is largely dominated by this pruning, and therefore, a reduction in pruning can significantly improve the overall run times of an LSH index.

This paper proposes the use of Salient Segmentation [5] to intelligently reduce the size of an LSH index. Salient Segmentation is the process of extracting unlikely (or interesting) segments from a time series signal. Segmentation is accomplished through the exploitation of the stationary and cyclical properties of medical time series signals and ensures the following two properties: 1) All salient patterns are segmented; and 2) All salient patterns are segmented consistently (i.e., alignment). Populating the index with only salient segments decreases the number of LSH non-matches due to mis-alignments, thereby improving the overall run times with minimal degradation to the quality of the respective search results.

There are two main contributions to this paper. First, this paper presents an improved Salient Segmentation technique. The original technique presented in [5] relies heavily on filtering techniques to extract salient segments. The parametrization of this filtering is domain specific and often arbitrary. The proposed Salient Segmentation algorithm in this paper requires no filtering, thereby eliminating subjectivity. Second, this paper presents the performance improvements of LSH indexes populated with only salient segments over LSH indexes populated with all segments (both salient and non-salient).

The proposed method is evaluated on three publicly available datasets consisting of real physiological data. Search results of this method are compared to using LSH alone (e.g., populating the index with all segments including both salient and non-salient segments). In all three datasets, the salient index returned near identical results to the complete index and reduced the number of computations by 80% or more.

II. BACKGROUND

LSH was introduced as an alternative to spatial indexing schemes [2]. Spatial indexing, such as those in [6][7], are not a feasible solutions for indexing medical time series signals as these methods have been shown both theoretically and experimentally to perform worse than sequential search for data with as little as ten dimensions [3].

LSH is based on a family of hashing functions H that are (r_1, r_2, p_1, p_2)-sensitive meaning that for any $v, q \in S$:

- if $v \in B(q, r_1)$ then $\Pr_H[h(q) = h(v)] \geq p_1$
- if $v \notin B(q, r_2)$ then $\Pr_H[h(q) = h(v)] \leq p_1$,

*This publication was supported by Grant Number T15 LM07356 from the NIH/National Library of Medicine Medical Informatics Training Program.

[1]Jonathan Woodbridge (jwoodbri@cs.ucla.edu), Bobak Mortazavi (bobakm@cs.ucla.edu), and Majid Sarrafzadeh (majid@cs.ucla.edu) are with the Computer Science Department, University of California, Los Angeles, CA, 90095, USA.

[2] Alex A.T. Bui (buia@mii.ucla.edu) is with the Medical Imaging Informatics Group, Department of Radiological Sciences, University of California, Los Angeles, CA, 90024, USA.

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

where v and q are high dimensional objects within search space S, $B(q,r)$ represents the set of objects within distance r to q, $p_1 > p_2$, and $r_2 > r_1$. The gap between p_1 and p_2 is increased by combining several functions from the same (r_1, r_2, p_1, p_2)-sensitive family. For the purpose of this paper, $r_1 = R$ and $r_2 = cR$ where c is a constant.

More simply, an LSH scheme guarantees (within some probability) that all objects within distance R to the query object are returned. In addition, all objects that fall at a distance greater than cR are not returned with some probability. The result sets of LSH are pruned such that all objects greater than distance R are suppressed. [2] show that the computational complexity of a search is sub-linear and dominated by $O(n^\rho)$ distance computations (pruning) where $\rho = \frac{\ln 1/p_1}{\ln 1/p_2}$.

Pruning is extremely costly even with a sub-linear number of distance computations. One method to improve pruning times is to reduce the number of segments indexed within a databases. However, this must be done intelligently as arbitrarily removing objects will severely reduce the quality of search results. Salient Segmentation [5] is a generic approach to reducing the size of a time series index without degrading search performance. Salient segmentation extracts the most interesting (or salient) segments from a time series signal. A time series is defined as an ordered set of points of length n within the time domain. More formally:

$$T_n = t_1, t_2, ..., t_n, \tag{1}$$

A segment is defined as an ordered set of points of a fixed size m within a time series where $m \leq n$. More formally:

$$s_i = t_{i-\frac{m}{2}}, ..., t_i, ..., t_{i+\frac{m}{2}-1}, \tag{2}$$

Salient Segmentation comprises of a function Φ that transforms a time series T to a time series saliency function (TSF). The TSF is formally defined as:

$$TSF = \Phi(T). \tag{3}$$

Each point in the original time series that corresponds to a local maximum in the TSF is determined to be locally salient. Each respective segment s_i centered at a salient point i is extracted. All other segments are ignored.

The saliency transformation proposed by [5] modelled the time series as a Markov chain where each point in the TSF was calculated as:

$$TSF_i = -\log \prod_{j=i-\lfloor m/2 \rfloor}^{i+\lfloor m/2 \rfloor} \Pr(T_j | T_{j-1} = t_{j-1}) \tag{4}$$

Equation (4) has two issues. First, Markov chains are not inherently good at localizing the exact location of a salient point. Fig. 1 shows three alignments of the same pattern with the most salient point denoted. The TSF defined in (4) labels all three alignments with the same saliency resulting in poor localization of the salient point. Second, (4) results in an extremely noisy TSF making it difficult to label local maximums. Both these issues were addressed in [5] by filtering the TSF. However, the parametrization of the filtering is extremely subjective and domain specific. An improper parametrization can lead to over segmentation or under segmentation.

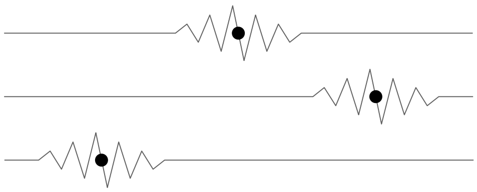

Fig. 1. Displays the same pattern at three different alignments. The most salient point is marked for each alignment. The TSF defined in (4) would label all three alignments with the same saliency resulting in poor localization of the salient point.

This paper address both issues in the Salient Segmentation algorithm proposed in [5]. First, saliency is calculated by using a range of segment lengths unlike one static length in [5]. The average of each segment width's saliency centered around point i defines the saliency of i (TSF_i). Second, the saliency function is updated to use entropy. The updated saliency function provides a far smoother TSF resulting in the ability to localize salient points without filtering.

III. Method

There are two main components of the experimental implementation: index structure and index population. Index structure utilizes LSH and is the process of indexing segments. Index population is the process of inserting salient segments into the index structure.

A. Index Structure

The index structure utilizes the LSH scheme based on p-stable distributions defined in [4]. The authors propose the following hash function:

$$h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{w} \rfloor, \tag{5}$$

where a is a randomized vector following a Gaussian distribution, b is a uniformly randomized vector, and w is a predefined constant. Using the properties of the p-stable distribution, the authors show that the probability of collision is calculated as:

$$p(c) = \int_0^r \frac{1}{c} f_p(\frac{t}{c})(1 - \frac{t}{r}) \mathrm{d}x, \tag{6}$$

with c being the distance between two vectors. As can be seen by (6), the probability of collision decreases monotonically as c increases.

As stated earlier, the gap between p_1 and p_2 is enlarged by combining several functions together. This is accomplished in two ways. First, k hashes are combined to form one parent hash. Objects x and y are matches if all k hashes match between x and y. Second, L parent hashes are created such

TABLE I

LSH PARAMETRIZATION

Parameter	Value
$\|v\|$	512 (ECG, GAIT) 64 (WALK)
R	3
w	4
c	4

that objects x and y are matches if at least one of the L parent hashes match between x and y. Therefore, a total of kL hash functions (defined by (5)) are created such that each a and b are drawn independently.

The parametrization of the LSH hashing scheme is shown in Table I. A detailed explanation of the parametrization of LSH is given in [4].

B. Index Population

The index is populated using an improved Salient Segmentation method. This method models a time series as a Markov model such that the probability of each point is defined as follows:

$$p(t_i) = \Pr(T_i | T_{i-1} = t_{i-1}) \tag{7}$$

The saliency of a time point t_i is calculated using the entropy of different window sizes centered at t_i:

$$TSF_i = -\frac{1}{|W|} \sum_{w \in W} \sum_{j=i-\frac{w}{2}}^{i+\frac{w}{2}} p(t_i) \log p(t_i), \tag{8}$$

where W is the set of window sizes. For best results, window sizes should range from the smallest pattern expected to the largest pattern expected. The experimental window sizes are given in Table II.

Each point t_i that corresponds to a local maximum at TSF_i is extracted as a salient segment defined in the range $[i - \frac{m}{2}, i + \frac{m}{2}]$, where m is the size of the extracted (indexed) segment.

IV. RESULTS

Two databases were created for the experimental section. Both databases use the LSH indexing structure presented by [4]. The first database uses a *salient index* and populates the index with only salient segments. The second database uses a *full index* and populates the index with all segments (i.e., both salient and non-salient segments). As shown in [5], Salient Segmentation yields similar alignments for similar patterns. This means that two similar salient patterns may differ slightly in their alignment. Non-elastic distance measures, such as Euclidean distance, can be largely effected by small misalignments. Therefore, recall can be improved dramatically by including a small number of neighboring segments to each salient segment. For example, assuming a salient segment centered at t_i, all segments centered between $t_{i-\frac{p}{2}}$ and $t_{i+\frac{p}{2}}$ are also inserted into the index, where p is defined as the added buffer. During experimentation, p was varied to test its effects.

TABLE II

SALIENT SEGMENTATION PARAMETRIZATION

Dataset	Min	Max	Increment
ECG	25	250	25
GAIT	100	200	10
WALK	20	40	2

This paper leveraged three datasets in its assessment of the proposed method. These datasets include:

1) MIT-BIH Arrhythmia Database [8] (ECG). This dataset contains several 30-minute segments of two-channel ambulatory ECG recordings. These sample included arrhythmias of varying significance.
2) Gait Dynamics in Neuro-Degenerative Disease Database [9] (GAIT). This dataset contains data gathered from force sensors placed under the foot. Healthy subjects as well as those with Parkinson's disease, Huntington's disease, and amyotrophic lateral sclerosis (ALS) were asked to walk while the data was recorded. Data includes 5-minute segments for each subject.
3) WALK [5]. This dataset contains a series of annotated recordings from a tri-axial accelerometer worn in a subject's pants pocket. Data was recorded while subjects travelled through the interior of a building.

Fig. 2 compares salient indexes (includes only salient segments) to full indexes (includes both salient and non-salient segments). Fig. 2 A shows the the number of true results returned by the salient index over the number of true results returned by the full index. An increase in buffer improves the results for all three datasets. However, the increase in buffer also increases the number of pruned results as shown in Fig. 2 B. The ECG and GAIT datasets are optimal with a buffer size of 40 (for a salient segment at t_i, index all segments from t_{i-20} to t_{i+20}). For a buffer of 40, the ECG and GAIT datasets retrieve 92% and 89% respectively of the full results and prunes only 15% and 8% respectively than that of the full LSH index. Results for the WALK dataset are best with a buffer of 10-20 with 70%-85% of the full results set with 18%-37% of the number of pruning operations.

The percentage of pruning results increases much faster for the WALK dataset than that of the ECG and GAIT datasets since the segment size is much smaller for WALK. For instance, one cycle (or step) in the WALK dataset consists of approximately 50-60 time points. Therefore, a buffer of size 40 results in an index that is very close in size to the full index.

The overall performance of the WALK dataset is not as good as the ECG and GAIT datasets due to two reasons. First, accelerometer data is far more diverse than the GAIT and ECG datasets. This means that the measured difference (Euclidean distance) of the accelerometer values between two similar steps is larger on average between two similar heartbeats in ECG or two similar steps in GAIT. Second,

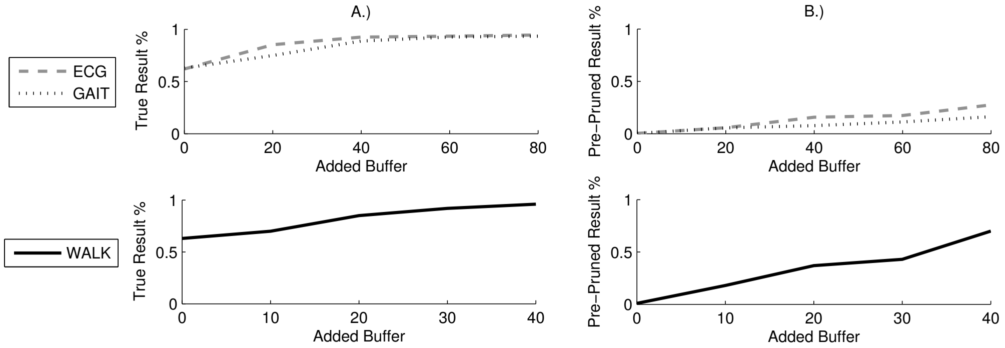

Fig. 2. A.) Displays the number of true results returned by the salient index over the number of true results returned by the full index with an increasing amount of buffer. B.) Displays the number of LSH results (pre-pruned) of the salient index over the number of LSH results for the full index with an increasing amount of buffer.

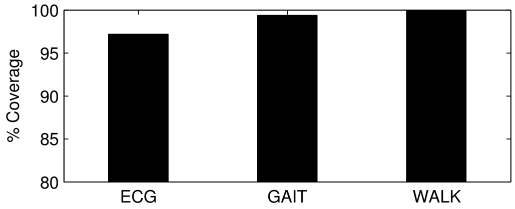

Fig. 3. Displays the percentage of signal covered by salient segmentation for the ECG, GAIT, and WALK datasets.

the WALK dataset is much smaller than the ECG and GAIT dataset. The full WALK dataset consists of about .4M indexed segments versus the ECG and GAIT datasets with 61M and 11.4M indexed segments respectively. Given a larger diversity with a smaller number of potential matches, the experiments must be run with a relatively larger R for WALK (note that all datasets use the same R, but WALK has a much smaller segment size). A smaller R will result in extremely small result sets for both the salient and full indexes. This small R will therefore yields an artificially high performance for the salient index. For example, result sets may be of size 1 where the results include only the search segment.

For the previous experiment, the search segment was randomly selected from the group of salient segments. In order for this experiment to be valid, salient segments should cover a large amount (or all) of their respective time series signal. Fig. 3 displays the percentage of signal covered by salient segmentation for the ECG, GAIT, and WALK datasets. All three datasets have coverage of at least 97% and therefore, their respective salient indexes consist of a large majority of the original time series signals.

V. Conclusion

This paper presented a highly efficient technique for indexing medical time series signals. Time series signals were indexed using Locality Sensitive Hashing (LSH). LSH has a provable sub-linear computational search complexity. LSH's search complexity is heavily dominated by pruning, and therefore, can be improved by removing redundant segments. Redundancy is reduced by employing Salient Segmentation. Salient Segmentation is the process of extracting unlikely (or interesting) segments from a time series signal. The index is populated with only salient segments while all other non-salient segments are ignored.

The proposed technique was tested on three publicly available physiological datasets. The amount of pruning was reduced by up to 95% while producing near identical search results to a complete index. In addition, Salient Segmentation was shown to produce segments with high coverage. The indexed segments covered more than 97% of the original time series for all three datasets.

References

[1] M. Wood and K. Ellenbogen, "Cardiac pacemakers from the patients perspective," *Circulation*, vol. 105, no. 18, pp. 2136–2138, 2002.

[2] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the International Conference on Very Large Data Bases*, 1999, pp. 518–529.

[3] R. Weber, H. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proceedings of the International Conference on Very Large Data Bases*. IEEE, 1998, pp. 194–205.

[4] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[5] J. Woodbridge, M. Lan, M. Sarrafzadeh, and A. Bui, "Salient segmentation of medical time series signals," in *Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on*. IEEE, 2011, pp. 1–8.

[6] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *ACM SIGMOD international conference on Management of data*, vol. 23, no. 2. ACM, 1994, pp. 419–429.

[7] Y. Cai and R. Ng, "Indexing spatio-temporal trajectories with chebyshev polynomials," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 599–610.

[8] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[9] J. Hausdorff, A. Lertratanakul, M. Cudkowicz, A. Peterson, D. Kaliton, and A. Goldberger, "Dynamic markers of altered gait rhythm in amyotrophic lateral sclerosis," *Journal of applied physiology*, vol. 88, no. 6, pp. 2045–2053, 2000.