QRS MORPHOLOGY CLASSIFICATION: QUANTITATIVE EVALUATION OF DIFFERENT STRATEGIES

S. H. Rappaport, L. Gillick, G. B. Moody, and R. G. Mark

Biomedical Engineering Center
Massachusetts Institute of Technology
Cambridge, Mass, U.S.A.

A number of QRS Morphology Analysis and Clustering algorithms were quantitatively evaluated through the use of an annotated ECG database. Algorithm parameters of data representation, beat similarity measurement, and normalization procedure were addressed. With the establishment of a systematic algorithm comparison system, algorithms were rank-ordered in terms of clustering performance. The rankings indicated the marked superiority of certain normalization procedures. Performance contributions of other algorithm parameters were less dramatic.

Introduction

QRS Morphology Analysis and Clustering (QRSMAC) is a critical process in automated cardiac arrhythmia detectors. It separates QRS waveforms into classes based on their degree of morphologic similarity. Many different approaches to the QRSMAC task have been described in the literature over the past two decades. They have traditionally been divided into the two broad classes of template matching [1-6] and feature extraction/clustering [7-14]. "Template matching" has usually referred to systems in which QRS data is represented by time-serial samples. QRS similarity comparisons are performed by means of either a cross correlation [2,4] or "area differences" computation. In "feature extraction" systems, the QRS data is represented by a set of either heuristic descriptors such as QRS amplitude, area, offset, width, etc., or formal features such as coefficients of orthonormal vector sets. The features are considered to represent each QRS as a point in N-dimensional space, where N is the number of features measured. Similarity between beats is then related to the distance separating points in N-space.

In the past, it has been very difficult to assess the relative merits of the many different QRSMAC algorithms which have been proposed. The lack of a generally acceptable annotated database has made it impossible to compare evaluations done by different groups. Additionally, there has been no generally accepted experimental methodology by which to make such comparisons. The present study made use of the MIT/BIH annotated ECG database[15] to systematically compare a number of different QRSMAC strategies. The problem, we soon discovered, was much more complex than we had initially imagined, even when restricted primarily to real-time algorithms.

Figure 1 diagrams a generalized QRSMAC system. The metric element determines the morphologic similarity between incoming QRS complexes and previously established QRS clusters. Its function is regulated by the cluster control element, which also controls the assignment of incoming QRS complexes to existing or new clusters based upon morphologic similarity.

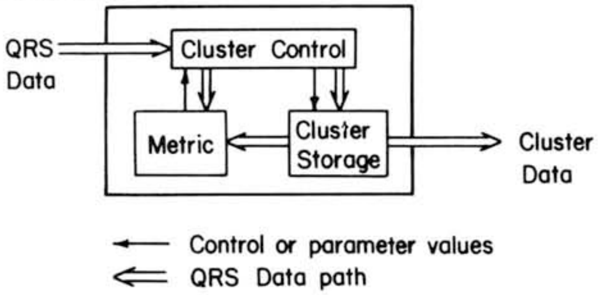


Figure 1: The Generalized QRSMAC Process

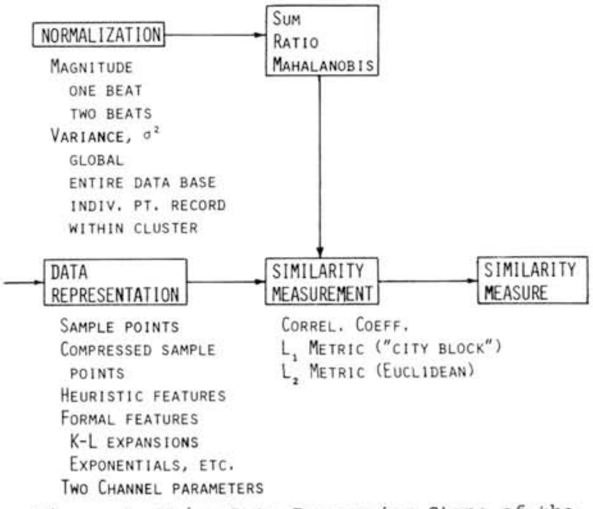


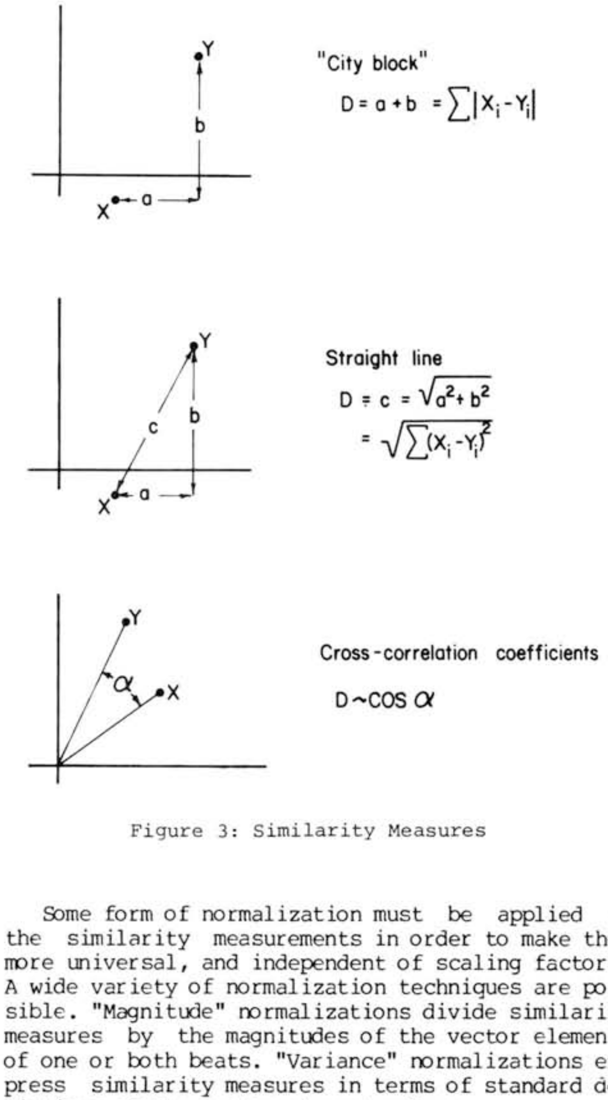
Figure 2: Major Data Processing Steps of the Metric Element

The heart of the QRSMAC lies in the metric element. This is a step for which many implementation possibilities exist. The major data processing steps within the metric element are shown in figure 2. In the "data representation" step, individual QRS's are represented as a series of numbers which are stored in the elements of a "data vector". Typ-

0276-6574/83/0000/0033$01.00 © 1982 IEEE

ically, elements will contain either QRS time sample values or else measured QRS features. Time sample vectors may contain all samples of a QRS or a specifically chosen subset of QRS data. Feature vectors may contain many types of either heuristic or formal features. Note that a feature vector may contain only a chosen few of the total possible features in a set.

Once the data vectors are defined, beat similarity is measured (see fig 3). This may involve calculating the correlation coefficient between two data vectors (the N-dimensional cosine between the vectors). Alternatively, data vectors may be considered to point to locations in N-space. Beat similarity may then be expressed as either the "straight line" (L2) or "city block" (L1) distances between N-space locations.



Y
b
X
a
"City block"
$D = a + b = \sum |X_i - Y_i|$
Y
c
b
X
a
Straight line
$D = c = \sqrt{a^2 + b^2}$
$= \sqrt{\sum (X_i - Y_i)^2}$
Y
α
X
Cross-correlation coefficients
$D \sim \cos \alpha$

Figure 3: Similarity Measures

Some form of normalization must be applied to the similarity measurements in order to make them more universal, and independent of scaling factors. A wide variety of normalization techniques are possible. "Magnitude" normalizations divide similarity measures by the magnitudes of the vector elements of one or both beats. "Variance" normalizations express similarity measures in terms of standard deviations of the vector element values.

The normalized similarity measures are used to assign each beat to a morphologic cluster (see fig 4). The candidate beat is assigned to the nearest cluster unless the corresponding similarity measure is greater than a preset threshold. In that case a new cluster is formed. The threshold setting will therefore determine the number of clusters formed by a particular algorithm.

The number of possible QRSMAC algorithms is enormous. Any data representation method can be combined with any similarity measure normalized by any one of many schemes. The traditional separation of QRSMAC algorithms into either feature extraction/clustering or template matching approaches is clearly an oversimplification.

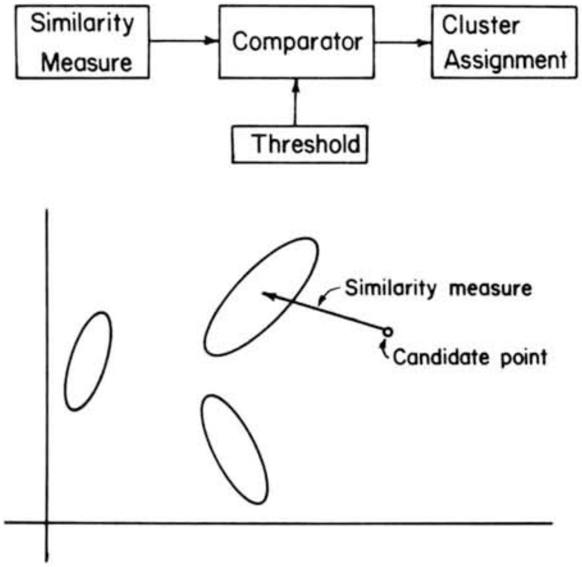


Figure 4: Cluster Assignments

Methods

Algorithms

Of the hundreds of possible QRSMAC algorithms, 32 were chosen to be implemented and compared. Each algorithm combined a specific data representation, measurement strategy, and normalization procedure. The completed algorithms were selected to simulate many of the procedures commonly used in arrhythmia analysis systems today. Several algorithms were also synthesized to allow for the evaluation of certain major algorithm variations, such as comparing an L1 to an L2 similarity measure, or comparing time samples to heuristic features. The 32 algorithms constructed are indicated in the following table.

Table 1: QRSMAC Algorithms

- Heuristic features
  - 8 single channel features
    - L2 similarity measure
      - No normalization (2,3.a)
      - Variance normalization (4.or,5.odr,6.cr,7.car,8.cdr)
      - Magnitude normalization (30.s2,31.as2)
    - L1 similarity measure
      - Magnitude normalization (9.s2,10.s1,11.r2,12.r1)
  - 16 feature, 8 from each of two channels
    - L2 similarity measure
      - No normalization (15)
      - Variance normalization (16.or,17.wr)
- Time samples
  - 45 single channel samples
    - L1 similarity measure
      - Magnitude normalization (18.s2,19.as2,20.ws2,21.was2,24.as1)
    - L2 similarity measure
      - Magnitude normalization (32.s2)
    - Correlation similarity measure (25,26.a)
  - 46 samples, 23 from each of two channels
    - L1 similarity measure
      - Magnitude normalization (29.s2)
  - 12 time samples (every 5th original sample)
    - L1 similarity measure
      - Magnitude normalization (22.s2)
    - Correlation similarity measure (27)
  - 5 time samples (every 9th original sample)
    - L1 similarity measure
      - Magnitude normalization (23.s2)
    - Correlation similarity measure (28)

Minor codes

a - averaging used in cluster representation
w - wiggling used in sample extraction
2 - data from both vectors being compared used in normalization
1 - data from only one of the vectors being compared used in normalization
d - variances modified by discriminant analysis derived results
o - overall type variances
c - averaged within-cluster type variances
r - ratio type calculations
s - sum type calculations

The table describes the algorithms organized by data representation, similarity measure, and normalization. Numbers in parentheses display algorithm identifying numbers and also (following the decimal points) additional codes describing minor algorithm variations. A few algorithms were left out of the table for simplicity.

Algorithm variations were generally constrained to the metric stage. The principal variations substituted QRS time samples for heuristic features in data representation, used either L1, L2, or correlation calculations in similarity measurement, and used either variance or magnitude type normalizations (when normalizations were being applied).

The 8 heuristic features consisted of [16]

Peak to trough QRS height
Center of mass offset from the baseline
QRS width derived from height and area
ST segment slope
Normalized time interval between peaks
Signed area
Unsigned area
QRS width as determined by locating QRS beginning and end points

QRS time samples were extracted beginning 55 msec prior to the largest QRS peak (either positive or negative). Samples were extracted for 125 msec following this start point using a sampling frequency of 360 Hz, 11 bit resolution. In the two-channel, time sample experiment the sampling frequency was 180 Hz. When "wiggling" was used, five data vectors were extracted for each beat, the vectors differing from one another by the extraction start locations. Variations of -2, -1, 0, +1, and +2 sample points from the standard start point were used to create the five vectors. Each of the five vectors was matched to clusters during similarity measurements, and the best was chosen.

Database

The MIT/BIH annotated ECG database was used as the ECG test source in the experiments. Thirty of the 48 records of the database were selected for use in the study. 18 records were rejected because the incidence of abnormal beats was too low (< 1%). Each record contained one-half hour of two-channel ECG data from one patient. Annotations indicated the physiologic (though not necessarily morphologic) identity of all database beats. All data processed by the QRSMAC routines consisted of actual QRS complexes as determined by the annotation codes. No artifacts or otherwise false QRS's were evaluated. Beats processed by the algorithms were selected to be of seven types: normal, PVC, left and right bundle branch block, ventricular escape, aberrently conducted SVPB, and paced beats.

Performance measurement and algorithm comparison

Algorithms were tested on each of the thirty records of the database. Algorithm output consisted of a set of QRS clusters, each cluster being detailed as to the number of beats of each physiologic type it contained. In general, algorithm performance reflected an algorithm's ability to create clusters containing beats primarily of only one physiologic type.

Performance measurement involved four tasks. First, clusters were named to indicate the physiologic beat type the cluster most likely represented (figure 5). Second, the clusters produced by each algorithm were combined to generate a classification error matrix. The matrix indicated for each beat type the percentage of beats that had been assigned to all possible cluster types (figure 5). Third, from each algorithm's error matrix, sensitivity and positive predictive accuracies were determined for each of the seven possible beat types. Fourth, the 14 error measures so derived for each algorithm were normalized and then averaged together to produce a single final performance indicator (figure 6). Due to the normalization, the

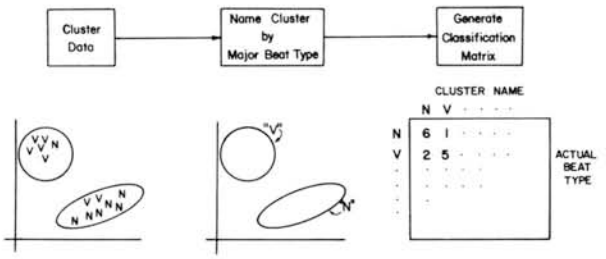


Figure 5: Cluster Naming and Error Classification

final performance measure is relevant only in the confines of this study. Although it allows us to assess the particular strengths and weaknesses of the algorithms under investigation, it has no use in comparing results to those of other investigators.

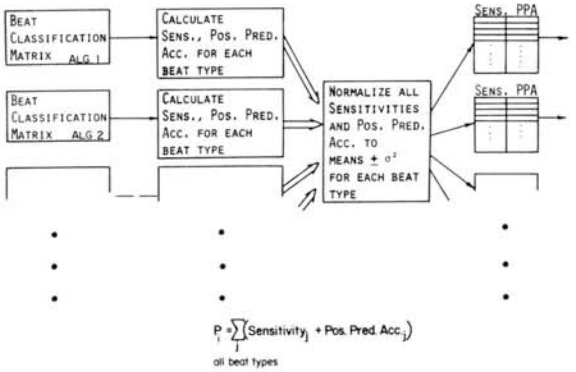


Figure 6: Performance Measure Determination

The performance measure is a function of cluster count (the number of clusters created by an algorithm in processing the database). If an algorithm creates a large number of clusters it has less chance to make classification errors as when it uses only a few clusters. Its performance will therefore increase with cluster count. An algorithm's actual merit is related to its ability to perform well while using only a minimal number of clusters. To compare this aspect in different algorithms, one is therefore required to compare algorithm performance vs. cluster count relationships (performance curves). Several algorithms were examined over a range of cluster counts to create typical performance curves (figure 7). Algorithms were judged superior when their curves lay relatively closer to the upper left-hand corner of such plots.

Performance curve determinations were a very time consuming process. The determination of six points on one curve required at least two days of continuous computing time. Many algorithms were therefore compared by a simplified process in which only one point was found for an algorithm. This point was plotted against the background of existing performance curves, and algorithms were compared based on how such individual points fell in relation to the existing curves.

Results

The following major observations were made in analyzing the data.

1. Performance curves for a variety of different algorithms were well-behaved, monotonically increasing functions of cluster count, which seemed to begin to reach plateaus at cluster counts of 650 (totaled over 30 records, about 20 clusters per record). Therefore, it seems reasonable to use performance curves as an experimental technique by which to compare algorithms -- in fact it is required.

2. The best algorithm of all those tested incorporated:

QRS sample data points, L1 similarity measure,
2 beat magnitude sum normalization with
averaging.

The worst performance was observed with:

Heuristic features, L2 similarity measure,
variance ratio normalization, within-cluster
variance calculations.

The performance curves for these two algorithms are shown as lines 19 and 6, respectively, in figure 8. Any or all of the algorithm parameters could have been responsible for the performance disparity of algorithms 6 and 19. Several possibilities are examined below.

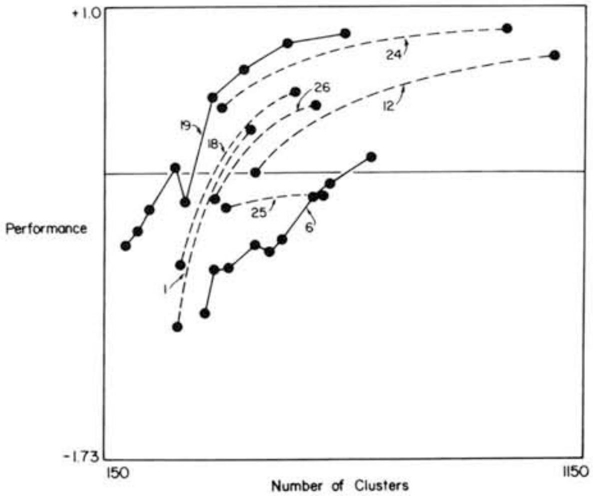


Figure 7: Actual Performance Curves

3. L1 similarity measures were found to perform slightly better than or as well as L2 measures. Algorithms in the pairs <30 and 9>, and <32 and 19> are identical to one another except that 30 and 32 use an L2 approach whereas 9 and 19 use an L1. In figure 8, it can be observed that both algorithms in a pair displayed similar performance to one another. Both measures can therefore produce equally good results. L1 is, however, simpler to compute.

4. Time samples were found to perform slightly better than or as well as heuristic features. In pairs <9 and 18> and <30 and 32>, 9 and 30 use features whereas 18 and 32 use samples. A modest performance gain when using time samples is apparent. The time sample computations, however, used 45 sample points whereas feature calculations used only 8 features. Time sample approaches were therefore much more computationally expensive and memory intensive.

5. Only minor performance losses were observed when using a reduced number of time samples in L1 type similarity calculations. Algorithms 18, 22, and 23 are identical except that 45, 12, and 5 samples, respectively, were used by the algorithms in calculations. Relatively insignificant performance differences among the algorithms are apparent in figure 8. Thus, the use of a large number of sample points in calculations is unnecessary. Results 4 and 5 combined show that equal numbers of time samples and features can be expected to perform equally well at equal cost.

6. Correlation coefficient measures were found to perform more poorly than L1 or L2 measures. In figure 6, pairs <18 and 25> and <19 and 26> (18 and 19 using L1, 25 and 26 using correlation) demonstrate a marked performance reduction due to the correlation approach.

7. Cluster representation based on the average of all beats contained in a cluster, as opposed to representation based on only the first beat entered into the cluster, provided time-sample experiments with significant performance increases while only minimally affecting, if not harming, feature based strategies. Algorithms 18 and 19 are identical time-sample routines except that 19 uses averaging. In figure 7, performance is seen to be noticeably increased with averaging. Algorithm 3 is an averaging version of algorithm 2, both evaluating feature data. In figure 8, a minor drop in performance with averaging can be noted.

8. Magnitude type normalization was found to be significantly better than variance normalization. No normalization was found to be intermediate. Figure 9 graphs all algorithms based on their normalization procedure and shows the dramatic superiority of the magnitude approach. Three algorithms differing only by their normalization, 2, 6, and 30 (unnormalized, variance and magnitude normalized, respectively), in figure 8 also demonstrate the same relationship.

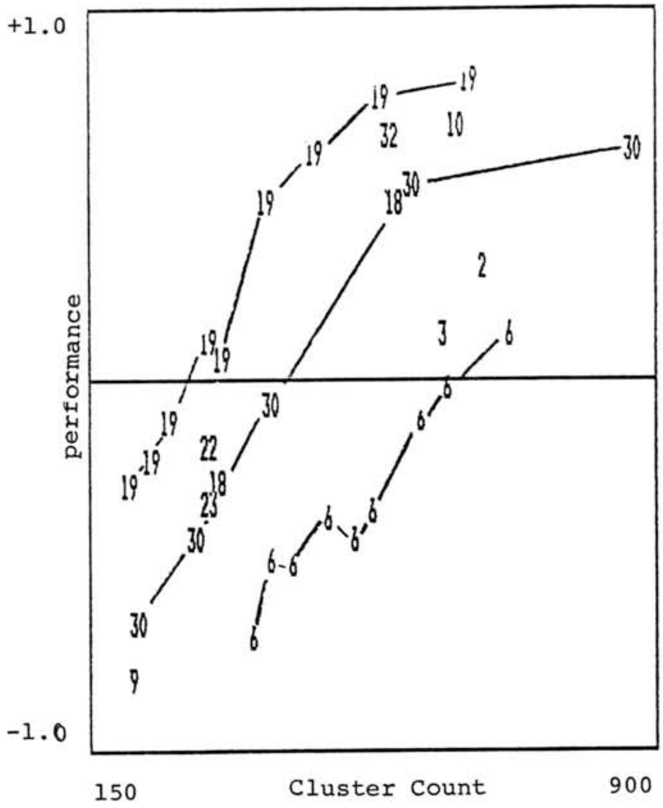


Figure 8: Performance Curves with Additional Experimental Results

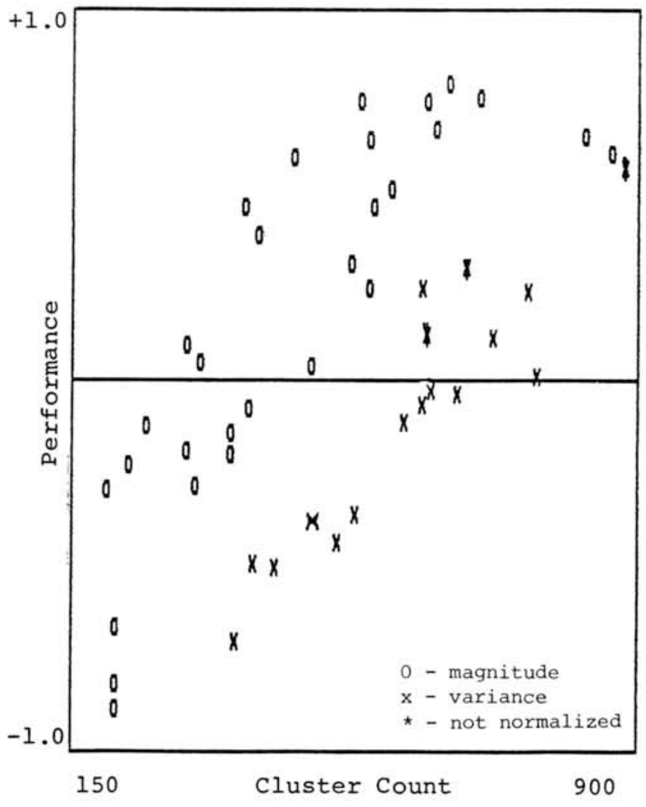


Figure 9: Algorithms by Normalization

Disscussion

Three conclusions are evident from the results. First, it is possible to devise a workable methodology for performing QRSMAC algorithm comparisons. Second, algorithm performance appears to be most affected by the normalization strategy in use. Variations in data representation or similarity measure appear to have less impact on algorithm functioning. In particular, "magnitude" type normalization was seen to be much more effective in the QRSMAC process as compared to "variance" methods. The use of template averaging in time sample algorithms was also associated with marked increases in strategy performance. Third, performance of the best QRSMAC algorithm tested could not be significantly improved through algorithm embellishments such as wiggling or adding a second ECG channel to the analysis. Finally, although the study was very illuminating, we recognize that it is in no sense an exhaustive analysis of the QRSMAC process. Many areas of algorithm variation are yet to be explored.

References

1. Hansmann, D.R., Sheppard, J.J., Yeshaya, A. Evaluation of the Dyna-Gram Holter ECG analysis system. Computers in Cardiology, pp 171-81. Long Beach, California: IEEE Computer Society, 1976.

2. Feldman, C.L., Amezeen, P.G., Klein, M.D., et. al. Computer detection of ectopic beats. Comput. Biomed Res. 3:666-74. 1971.

3. Balm, G.J. Crosscorrelation techniques applied to the electrocardiogram interpretation problem. IEEE Transactions on Biomedical Engineering BME-14:258-62. 1967.

4. Dillman, R., Judell, N., Kuo, S. Replacement of Aztec by correlation for more accurate VPB detection. Computers in Cardiology, pp. 29-32. Long Beach, California: IEEE Computer Society. 1978.

5. Otterstrom, L. Automated monotoring of cardiac arrhythmias using template matching. In Proceedings of the IC Nordic Meeting on Medical and Biological Engineering. pp. 61-1, 66-2, June 1977.

6. Arnold, J.M., Shah, P.M., Clark, W.B. Artifact rejection in a computer system for the monitoring of arrhythmias. Computers in Cardiology, pp. 163. Long Beach, California: IEEE Computer Society. 1975.

7. Nolle, F.M. Argus, a climical computer system for monitoring electrocardiographic rhythms. D.Sc. dissertation, Washington University, St. Louis, 1972.

8. Mead, C.N., Moore, S.M., Spenner, B.F., Hitchens, R.E., Clark, K.W., Thomas, L.J. Detection of multiform PVCs using a combination of time-domain and frequency domain information. Computers in Cardiology, pp. 343-6. Long Beach, California: IEEE Computer Society. 1978.

9. Watanabe, S. Karhunen-Loeve expansion and factor analysis: Theoretical remarks and applications. Transactions of the 4th Prague Conference on Information Theory, pp. 635-60, 1965.

10. Mark, R.R., Moody, G.B., Olson, W.H., Peterson, S.K., Schluter, P.S., Walters, J.B., Jr. Real-time ambulatory arrhythmia analysis with a microcomputer. Computers in Cardiology. Long Beach, California: IEEE Computer Society. 1979.

11. Lovelace, D.E., Knoebel, S.B., Zipes, D.P. Recognition of ventricular extrasystoles in sedentary versus ambulatory populations. Computers in Cardiology, pp. 9-11. Long Beach, Ca: IEEE Computer Society. 1976.

12. Kneobel, S.B., Lovelace, D.E., Rasmussen, S., et. al. Computer diagnosis of supraventricular arrhythmias: a new esophageal approach. Circulation 60: 977-87. 1979.

13. Ritter, J.A., Thomas, L.J., Ripley, K.L. ARGUS /RT: a microputer system for clinical arrhythmia monitoring. Computers in Cardiology, pp. 79-84. Long Beach, California: IEEE Computer Society. 1977.

14. Sanders, W., Alderman, E., Tedklenberg, P., Harrison, D.C. The Stanford computer-based arrhythmia monitoring system. Computers in Cardiology, pp. 199. Long Beach, California: IEEE Computer Society. 1974.

15. Mark, R.G., Schluter, P.S., Moody, G.B., Devlin, P., Chernof, D. An annotated ECG database for evaluating arrhythmia detectors. In Frontiers of Engineering in Health Care: Proceedings of the 4th Annual Conference of the IEEE Engineering in Medicine and Biology Society, September 1982.

16. Moody, G.B., Mark, R.G. Development and evaluation of a two-lead ECG analysis program. Computers in Cardiology, pp. 39-44. Long Beach, California: IEEE Computer Society. 1983. This Proceedings.

17. Rappaport, S.H. Electrocardiogram morphology clustering analysis: an evaluation of various strategies. M.I.T. EECS Masters Thesis September 1982.