IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 2, 2019, accepted November 5, 2019, date of publication November 8, 2019, date of current version November 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952399

Advanced Real-Time Dynamic Programming in the Polygonal Approximation of ECG Signals for a Lightweight Embedded Device

SEUNGMIN LEE[1], YOOSOO JEONG[2], JUNHO KWAK[2], DAEJIN PARK[2], AND KIL HOUM PARK[2]

[1]Advanced Dental Device Development Institute, Kyungpook National University, Daegu 41566, South Korea
[2]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding authors: Daejin Park (boltanut@knu.ac.kr) and Kil Houm Park (khpark@ee.knu.ac.kr)

This work was supported in part by the BK21 Plus Project funded by the Ministry of Education, South Korea, under Grant 21A20131600011, in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1A6A3A01011035, and in part by the Ministry of Science and ICT under Grant NRF-2019R1A2C2005099.

ABSTRACT Arrhythmia is less frequent than a normal heartbeat in an electrocardiogram signal, and the analysis of an electrocardiogram measurement can require more than 24 hours. Therefore, the efficient storage and transmission of electrocardiogram signals have been studied, and their importance has increased recently due to the miniaturization and weight reduction of measurement equipment. The polygonal approximation method based on dynamic programming can effectively achieve signal compression and fiducial point detection by expressing signals with a small number of vertices. However, the execution time and memory area rapidly increase depending on the length of the signal and number of vertices, which are not suitable for lightweight and miniaturized equipment. In this paper, we propose a method that can be applied in embedded environments by optimizing the processing time and memory usage of dynamic programming applied to the polygonal approximation of an ECG signal. The proposed method is divided into three steps to optimize the processing time and memory usage of dynamic programming. The first optimization step is based on the characteristics of electrocardiogram signals in the polygonal approximation. Second, the size of a data bit is used as the threshold for the time difference of each vertex. Finally, a type conversion and memory optimization are applied, which allow real-time processing in embedded environments. After analyzing the performance of the proposed algorithm for a signal length L and number of vertices N, the execution time is reduced from $O(L^2N)$ to $O(L)$, and the memory usage is reduced from $O(L^2N)$ to $O(LN)$. In addition, the proposed method preserve a performance of fiducial point detection. In a QT-DB experiment provided by Physionet, achieving values of -4.01 ± 7.99 ms and -5.46 ± 8.03 ms.

INDEX TERMS Dynamic programming, electrocardiogram, embedded system, fiducial point, optimization, polygonal approximation, signal compression.

ABBREVIATION

The following abbreviations are used in this manuscript:

ECG	electrocardiogram
PA	polygonal approximation
DP	dynamic programming

I. INTRODUCTION

With the development of life science and technology, the percentage of deaths from heart disease is gradually increasing as society ages due to the increased average life expectancy. Research on electrocardiogram (ECG) signals is actively being carried out for the early diagnosis of heart disease, especially since the development of hardware has resulted in the miniaturization and weight reduction of equipment [1] and the size of the related market is gradually increasing. As a result, various studies have been being conducted on real-time analysis of ECG signals in embedded environments [2].

ECG signals are electronically converted from the depolarization and repolarization of the atria and ventricle [3], and P-wave, QRS complex and T-wave waveforms are periodically repeated [4], [5]. Since arrhythmia causes changes in the

The associate editor coordinating the review of this manuscript and approving it for publication was Donghyun Kim.

shape of the waveform, it is possible to detect arrhythmia by analyzing the waveform's characteristic values [6]. The onset, peak, and offset of the P-wave, QRS complex, and T-wave waveforms are called the fiducial points of the waveform. Because these points are used to acquire the feature values of the waveform, the detection of a correct fiducial point is the basis for ECG signal analysis [7]–[10].

Fig. 1 shows the fiducial points and feature values of each waveform.

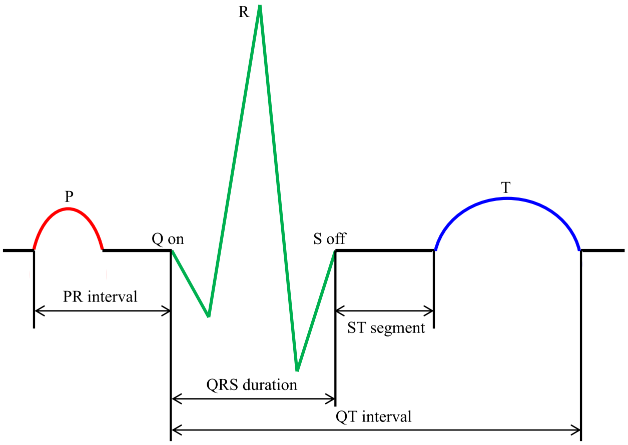


FIGURE 1. **The fiducial points and features of an ECG signal.**

Arrhythmia is rare and appears to be nonperiodic in ECG signals. It takes a long time to obtain enough arrhythmia data for an accurate analysis of a heart disease, sometimes more than 24 hours. ECG signals are sampled using high frequencies above 100 Hz, so a vast amount of data is recorded in a short time. Therefore, signal compression techniques are required to effectively store and transmit the data. However, conventional signal compression techniques, such as the Fourier transform, Walsh transform [11], wavelet transform [12], [13] and Karhunen-Loeve transform [14], result in loss in during the data compression process. In particular, signal distortion causes the nondetection or false detection of the fiducial point [15].

Polygonal approximation (PA)-based fiducial point detection [16] has been proposed as a method to express an ECG signal as a small number of vertices to determine the fiducial points. The advantage of this approach is that ambiguous fiducial points can be represented as vertex points using features that are boundaries between the baseline area, with small amplitude changes, and the waveform area, with large amplitude changes. Fig. 2 illustrates the PA in which ambiguous areas are simplified by vertices.

As shown in Fig. 2(b), the number of candidates of the fiducial point is decreased, and the features of the vertices are highlighted.

The PA not only enables an effective signal compression but also emphasizes the feature value with the fiducial point included as the vertex; it makes it easier to detect the fiducial point and has the advantage of not requiring a signal restoration. Fig. 3 is a result of PA in the ECG signal.

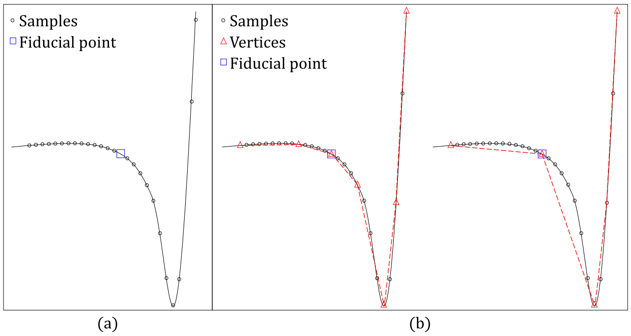


FIGURE 2. **Illustration of the PA: (a) the existing method and (b) the PA.**

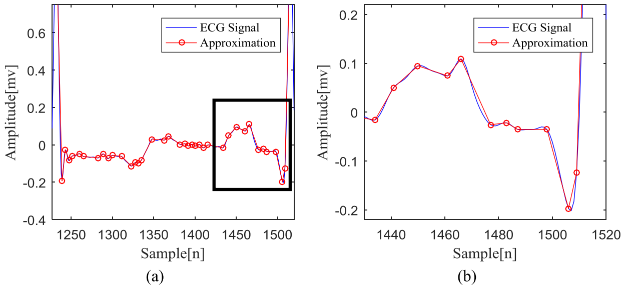


FIGURE 3. **Result of the PA for an ECG signal: (a) result in the R-R interval and (b) zoom-in of the black box region in (a).**

Approximately 300 samples of the signal are compressed by the PA into approximately 30 vertices. The onset, peak, and offset of the waveform are well represented as vertices, and the small approximation error maintains the shape information of the signal well.

However, the PA has difficulty in real-time processing under low-power and low-capacity constraints, such as in an embedded environment, because the optimization technique is based on the dynamic programming (DP) method [17], which requires more memory area and a longer execution time with an increase in the signal length and number of vertices.

In addition, the method used to record the time information of each vertex is inefficient. The number of approximated vertices is small, but the time differences between the vertices are irregular. Therefore, additional memory is required to store the vertex time information. In particular, ECG signals require a long measurement time, which significantly increases the number of bits allocated to the time information, resulting in a lower compression ratio.

To solve this problem, in this paper, we propose an improved PA method that enables real-time processing in an embedded environment by optimizing the DP based on the characteristics of ECG signals.

Fig. 4 compares the complexity problem of conventional DP and the improved result of the proposed DP method in this paper. The time and space complexity of the algorithm are greatly reduced.

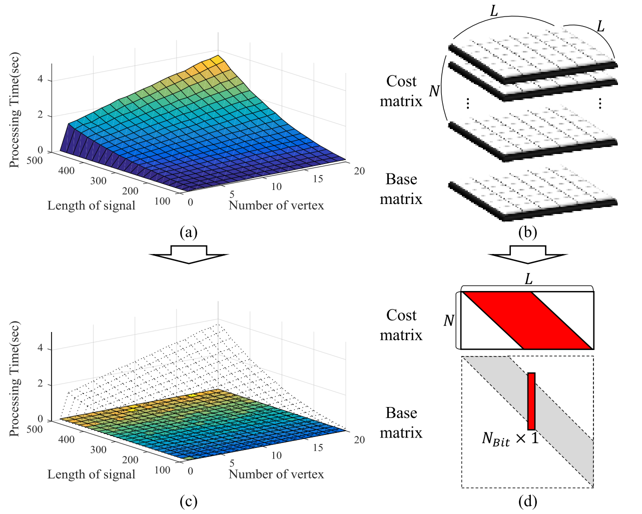


FIGURE 4. The problem of the conventional DP method and the improved performance of the proposed DP method: (a) the execution time of conventional DP ($O(L^2N)$), (b) memory usage of conventional DP ($O(L^2N)$), (c) execution time of the proposed DP method ($O(L)$), and (d) memory usage of the proposed DP method ($O(LN)$).

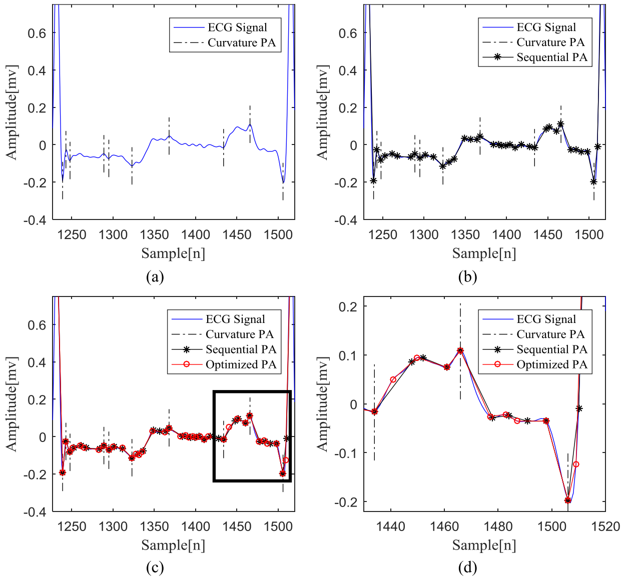


FIGURE 5. The results of the PA according to the algorithm flow for the (a) curvature-based PA, (b) sequential PA, (c) optimization, and (d) zoom-in of the black box region in (c).

The proposed method consists of three stages. First, the computation and memory usage are optimized based on the characteristics of ECG signals and a bottom-up operation. Then, the time information of a vertex is stored as the time difference between the vertices with a given time threshold. In this step, the time-difference threshold improves the performance time by reducing the DP computation. Additionally, the threshold minimizes the memory usage through the type conversion of data and a memory optimization during the calculation.

The composition of this paper is as follows. First, the existing DP method and the problem are briefly described in Section II, and each step in the DP optimization is described in Section III. In Section IV, the performance of the optimized DP method is verified through experiments in an embedded environment. Finally, we conclude the paper in Section V.

II. REVIEW OF THE EXISTING METHODS

A. CONVENTIONAL POLYGONAL APPROXIMATION

The algorithm flow of the PA for an input ECG signal is summarized as follows, and Fig. 5 shows the results of each step.

1) The R-R section of the input signal is separated.
2) After calculating the curvature of the separated R-R section, the curvature-based PA [18] is applied to select the initial vertices. However, many of the fiducial points, specifically the onsets and offsets, are not represented by vertices due to their similar features with the samples around them and their low curvature.
3) The sequential PA [19] is applied to the interval between each initial vertex to select additional vertices. The fiducial points are well expressed through the additional vertices, but the large error is a problem.
4) DP is applied to the additional vertices to optimize their positions.
5) Steps 2–4 are repeated to proceed with the PA for the entire input signal.

B. DYNAMIC PROGRAMMING

In the PA, DP optimizes the location information of the vertices selected in the sequential PA. This not only minimizes the error between the approximated signal and the input signal but also helps to represent the fiducial point as a vertex, which is the boundary point separating the baseline region from the waveform region.

DP is a global optimization technique in which the optimal path between two points is optimized based on the optimal principle of Bellman as the global optimal path between any two points on the global optimal path. The top-down recursive approach simplifies and optimizes the problem, especially by using memoization to remember the computational results, which eliminates redundant operations to enable a high-speed global optimization. In this case, the size of the cost matrix and path matrix required for memoization is $O(L^2N)$ when L is the length of the input signal and N is the number of vertices.

Algorithm 1 shows the existing DP method using the recursive method.

Fig. 6 represents the cost matrix for DP. $C_k(i,j)$ represents the cost optimization result for k vertices in the partial signal for the samples from i to j, and C_0 is the base matrix, which represents the approximation error if there is no vertex inside the partial signal and a practical error operation is performed.

For a signal with length L, the optimization of the partial signal from i to j, including the k vertices, is recursively

Algorithm 1 Conventional DP

```
Goal: Calculate $C_N(1, L)$
$S$ : input signal
$L$ : length of the signal
$N$ : number of vertices
$C_k$ : cost matrix of size $L \times L$,   $k = 1, \cdots, N$
$C_0$ : base matrix of size $L \times L$
$R$ : range of $v_k$

% By using the recursive function $DP$, $C_N(1, L)$ is calculated
$C_N(1, L) = DP(1, L, N)$

Function: $C_k(i, j) = DP(i, j, N)$
if $N$ is 0 then
    if $C_0(i, j)$ is $\infty$ then
        Calculate the linear approximation error between $i$ to $j$ and save as $C_0(i, j)$
    else
        Return $C_0(i, j)$ % Memoization
else
    if $C_0(i, j)$ is $\infty$ then
        $R = [1, \cdots, L]$
        $C_k(i, j) = \min_{v_k \in R}\{DP(i, v_k, N-1) + DP(v_k, j, 0)\}$
    else
        Return $C_k(i, j)$ % Memoization
```

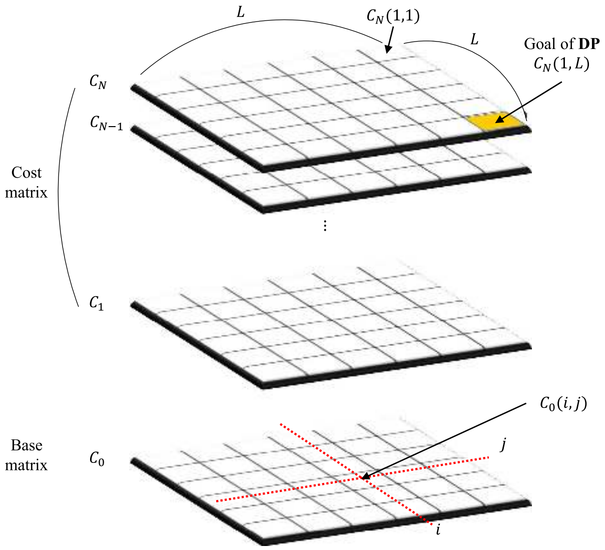


FIGURE 6. The composition of the cost matrix for DP.

computed as (1).

$$C_k(i, j) = \min_{v_k \in [1, \cdots, L]} (C_{k-1}(i, v_k) + C_0(v_k, j)), \tag{1}$$

where v_k denotes the position of the k^{th} vertex. $C_k(i, j)$ divides the partial signal including $k-1$ vertices and the partial signal that does not include vertices based on v_k and stores the result in $C_k(i, j)$ when the sum of the partial signal is at its minimum. At this time, the path v_k is stored in the path matrix $P_k(i, j)$, as shown in (2).

$$P_k(i, j) = \underset{v_k \in [1, \cdots, L]}{arg\,min} \ (C_{k-1}(i, v_k) + C_0(v_k, j)) \tag{2}$$

Therefore, the DP method using memoization has a disadvantage in memory usage is needed to record the cost and path matrices. Fig. 7 shows the execution time of the conventional DP method according to the signal length and number of vertices.

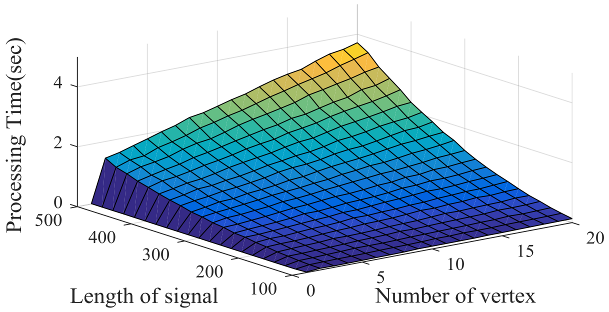


FIGURE 7. The execution time of conventional DP.

As shown in Fig. 7, it is difficult to apply real-time processing with conventional DP in embedded environments due to the rapid increase in the execution time with the signal length and number of vertices.

III. OPTIMIZATION OF DYNAMIC PROGRAMMING FOR AN ECG SIGNAL

In this paper, the proposed improvement for the real-time application of DP in embedded environments has three stages. First, the calculation and memory usage are optimized based on the characteristics of ECG signals and the PA, and then a bottom-up method is performed to improve the processing time and memory usage instead of a top-down method. The time difference between vertices is used to effectively record a vertex's time information, which further improves the processing time by reducing the computational range of the cost matrix according to the threshold N_{Bit} for the time difference. Here, N_{Bit} is a number determined by the number of allocated bits. For example, when the number of allocated bits is 3, N_{Bit} is $8 = 2^3$. Finally, a conversion of the data type and an adaptive determination of the weight value are proposed to optimize the memory usage. Then, by modifying the bottom-up method's computational sequence, the memory optimization of the base matrix is performed to minimize the memory usage.

A. CHARACTERISTICS OF THE POLYGONAL APPROXIMATION OF AN ECG SIGNAL

As shown in Fig. 7, DP increases the amount of memory and operations required according to the signal length and number of vertices. However, the characteristics of the PA for ECG signals can substantially reduce the memory required for the operation.

The approximation error in ECG signal does not change when the signal is inverted, and the cost matrix can be expressed as (3):

$$C_k(i,j) = C_k(j,i) \tag{3}$$

Thus, the cost matrix is a symmetric matrix, and we denote that the cost matrix has a *symmetry* characteristic. The area in which the operation cost is actually required in the cost matrix becomes the area of the upper triangular matrix, as shown in Fig. 8.

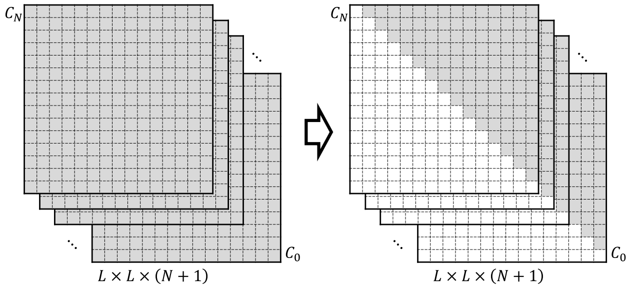


FIGURE 8. Decrease in the memory usage in the cost and base matrices according to the *symmetry* characteristic.

In addition, the *symmetry* is enhanced by the characteristic in the PA applied to the ECG signal. The ECG signal is inputted over time, and time information at each vertex of the PA is monotone increasing; we denote that the vertices of the PA have a *monotone* characteristic. The monotone increasing of time information of each vertex can be expressed as (4):

$$x_1 = x_{v_0} < x_{v_1} < \cdots < x_{v_N} < x_{v_{N+1}} = x_L \tag{4}$$

where x_i and x_{v_k} denote the time information of the i^{th} sample and k^{th} vertex, respectively, and N denotes the number of vertices.

Therefore, the range of v_k in (1) is computed only between i and v_{k+1}, and (1) is modified as shown in (5):

$$C_k(i,j) = \min_{i<v_k<v_{k+1}} (C_{k-1}(i,v_k) + C_0(v_k,j)) \tag{5}$$

In addition, for the ECG signal, DP is applied to the signal inside the initial vertices and always starts and ends at the first and last samples of the input signal, which are the initial vertices. Therefore, only the first row of each layer of the cost matrix, except the base matrix, is used for the computation. That is, the computational component used for $C_k(i,j)$ in the cost matrix is always $i = 1$, and the existing cost matrix $C_k(i,j)$ is expressed as $C(k,j)$, as shown in (6).

$$C(k,j) = \min_{1<v_k<j} (C(k-1,v_k) + C_0(v_k,j)) \tag{6}$$

The modified cost matrix can be represented as shown in Fig. 9. Accordingly, the improved algorithm can be expressed as Algorithm 2. Since the path matrix can be expressed in the same form, we can confirm that the spatial complexity improves from $O(L^2N)$ to $O(L^2)$.

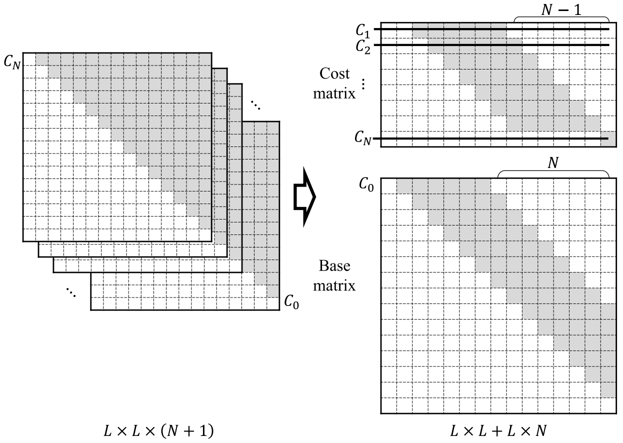


FIGURE 9. Decrease in the memory usage in the cost and base matrices according to the *monotone* characteristic.

Algorithm 2 Advanced DP by Using the Characteristics of an ECG Signal

```
Goal: Calculate $C(N, L)$
$S$ : input signal
$L$ : length of the signal
$N$ : number of vertices
$C$ : cost matrix of size $N \times L$
$C_0$ : base matrix of size $L \times L$
$R$ : range of $v_k$

% Initialize the base matrix
foreach $i$ from 1 to $L$ do
    foreach $j$ from $i+1$ to $L$ do
        Calculate the linear approximation error
        between $i$ to $j$ and save as $C_0(i,j)$

% Calculate the first row of cost matrix
foreach $j$ from 3 to $L-N+1$ do
    $R = [2, \cdots, j-1]$
    $C(1,j) = \min_{v_k \in R}\{C_0(1,v_k) + C_0(v_k,j)\}$

% Calculate the second to $N-1^{th}$ row of cost matrix
foreach $d$ from 2 to $N-1$ do
    foreach $j$ from $2+d$ to $L-N+d$ do
        $R = [d+1, \cdots, j-1]$
        $C(d,j) = \min_{v_k \in R}\{C(d-1,v_k) + C_0(v_k,j)\}$

% Calculate $C(N, L)$
$R = [N+1, \cdots, L-1]$
$C(N,L) = \min_{v_k \in R}\{C(N-1,v_k) + C_0(v_k,L)\}$
```

B. TIME CONSTRAINT BETWEEN VERTICES

The PA transmits the information at the vertices, which also has the effect of signal compression. The existing ECG signals are periodically sampled signals, but the vertices of the approximated signals are chosen nonperiodically, which results in the need to store additional time information. However, ECG signals generally involve a long-term measurement, resulting in a larger number of bits being needed to express the time information, which undermines the signal compression performance. Especially in embedded environments, these problems need to be solved because an improved signal compression performance is necessary to minimize the power consumption.

For this purpose, we suggest storing the time information of the current vertex as the time difference from the previous vertex. The current vertex's time information can be computed by accumulating the time difference information based on the initial vertex's time information. In general, the time difference between vertices is approximately 30 samples (based on 250 Hz signals), so allocations of 5 bits in size will result in a sufficient time information expression. However, when using the simplified form of an ECG signal, the number of vertices decreases, resulting in a sharp increase in the time difference between vertices. To avoid such an exceptional case, we add a time constraint to the sequential PA and DP steps.

First, during the PA step, the additional vertex selection stage using the sequential PA is improved. The sequential PA is a technique that is used to add vertices, with previous samples added as vertices if the approximation error is larger than a threshold. At this time, when adding a vertex, a condition is added such that the time difference between the fixed point and the vertex does not exceed the threshold value N_{Bit} corresponding to the given number of bits. Additionally, the DP method must be modified to maintain the time difference threshold in the optimization process. By modifying the search interval of the k^{th} vertex in (5) as (7), the optimization value is calculated only when the interval between the k^{th} vertex and the $k+1^{th}$ vertex does not exceed N_{Bit}:

$$C_k(i,j) = \min_{Thr_L < v_k < v_{k+1}} (C_{k-1}(i, v_k) + C_0(v_k, j)), \qquad (7)$$

where $Thr_L = max(i, v_{k+1} - N_{Bit})$.

Since the time difference between the two vertices is limited by N_{Bit}, the computation for the base matrix (C_0) is reduced, as shown in Fig. 10.

C. MEMORY OPTIMIZATION

1) TYPE CONVERSION

The result of the approximation error calculation usually includes a decimal point, so a data type such as a float or a double is used. We propose to reduce the memory to less than half by converting a 32-bit or 64-bit data structure to a 16-bit unsigned int type. However, when a simple type cast is applied, decimal deviations of less than 0 are recognized as a difference of 0 or 1 in the quantization process, so a normal error calculation cannot be applied. To solve this problem, it is necessary to consider appropriate weights. Small weights may still cause a large distortion of the matrix due to the small deviation, and large weights may cause the matrix to exceed the limit of the 16-bit unsigned int type–65,535, in this case. Each component of the cost matrix represents an approximation error, but the maximum component cannot be obtained until the cost matrix is completed. In this study, we suggest a scaling weight ω based on the maximum error limit of the signal. The maximum error limit is determined as shown in Fig. 11.

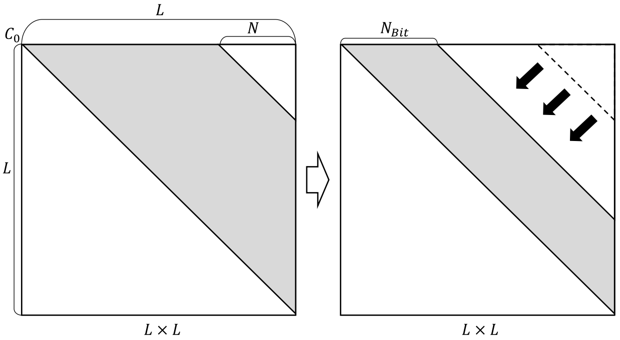


FIGURE 10. Decrease in the memory usage in the base matrix according to the threshold of time difference N_{Bit}.

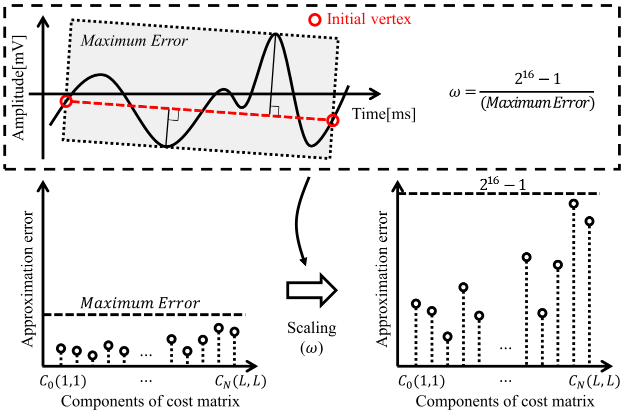


FIGURE 11. Calculation of the maximum error limit and determination of the scaling weight ω.

The two points with the maximum distance difference from the straight line connecting the two ends of the signal are obtained in the + and - directions. Then, the rectangles passing through both the end points of the signal and the two points with the maximum distance difference are obtained. The area of the obtained rectangle becomes the upper limit of the approximation error. The area of the rectangle is used as the maximum error to determine the scaling weight ω.

By appropriately determining the weights, it is possible to greatly reduce the memory usage while maintaining the deviation between the data in the type conversion.

2) MEMORY OPTIMIZATION

The existing bottom-up operation computes the base matrix and then computes the cost matrix row-by-row. Since each row of the cost matrix uses the previous row of the cost matrix

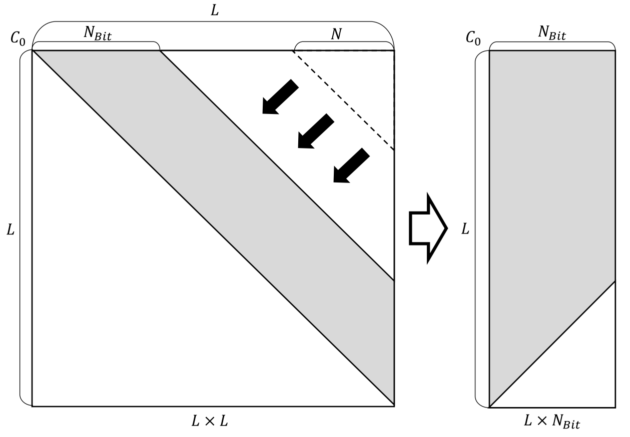


FIGURE 12. Minimizing the base matrix in a row-wise bottom-up operation.

in the computation process, it can be reduced to a memory space of two rows in size, and the base matrix can compress the column length, as shown in Fig. 12

However, the base matrix, which accounts for most of the memory, will not be significantly improved, because the memory usage reduction rate is not large. To improve the effective memory usage, we propose to reorganize the order of the bottom-up operations. To obtain the components of the cost matrix $C(k, j)$, the $k - 1^{th}$ row of the cost matrix and the j^{th} column of the base matrix are required. In other words, the j^{th} column of the base matrix is only used when calculating the j^{th} column of the cost matrix. Therefore, when the cost matrix is computed in units of columns instead of in units of rows, the base matrix can be represented by a column with a size of $L \times 1$ instead of a matrix with a size of $L \times L$.

Algorithm 3 shows a DP scheme in which the memory is optimized by applying an operation in units of columns. Additionally, cost matrix can be minimized to $N_{Bit} \times 1$ in size, as shown in Fig. 13.

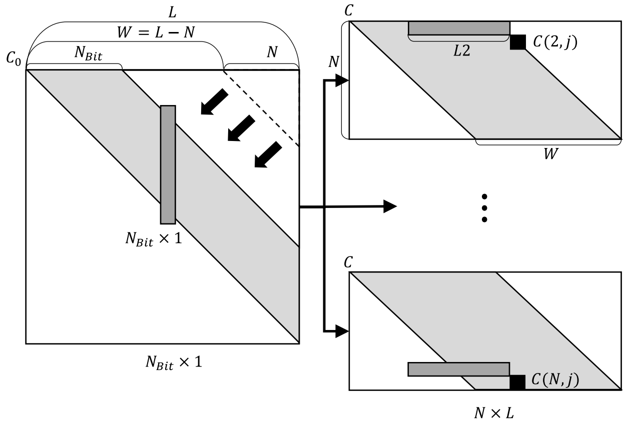


FIGURE 13. Column-wise bottom-up operation and minimized base matrix.

As shown in Fig. 12, additional memory can be reduced by compressing the upper triangular matrix component of the cost matrix into a trapezoid shape. Therefore, the sizes of the cost matrix and base matrix are reduced from $L \times L \times N$ and $L \times L$ to $N \times W$ and $N_{Bit} \times 1$, respectively, through the proposed memory improvement of DP.

Algorithm 3 Advanced DP by Optimizing the Memory Usage

```
Goal: Calculate $C(N, L)$
$S$ : input signal
$L$ : length of the signal
$N$ : number of vertices
$C$ : cost matrix of size $N \times L$
$C_0$ : base matrix of size $N_{Bit} \times 1$ column vector
$C_T$ : temporary row vector used in the first row of the cost matrix
$R$ : range of $v_k$
% Calculate the $C_T$
foreach $j$ from 3 to $1 + \min(N_{Bit}, L - N - 1)$ do
    Calculate the linear approximation error between 1 to $j$ and save as $C_T(j - 1)$
% Calculate the cost matrix until $L - 1^{th}$ column
foreach $j$ from 3 to $L - 1$ do
    % Update the base matrix according to $j^{th}$ column of cost matrix
    foreach $i$ from $\max(1, j - N_{Bit})$ to $j - 1$ do
        Calculate the linear approximation error between $i$ to $j$ and save as $C_0(i - (j - N_{Bit}) + 1, 1)$
    % Calculate each row of $j^{th}$ column of cost matrix
    foreach $d$ from $\max(1, j - W - 1)$ to $\min(N - 1, j - 2)$ do
        if $d$ is 1 then
            $R = [\max(2, j - N_{Bit}), \cdots, j - 1]$
            $C(1, j) = \min_{v_k \in R}\{C_T(v_k - 1) + C_0(v_k - j + N_{Bit} + 1)\}$
        else
            $R = [\max(d + 1, j - N_{Bit}), \cdots, j - 1]$
            $C(d, j) = \min_{v_k \in R}\{C(d - 1, v_k) + C_0(v_k - j + N_{Bit} + 1)\}$
% Update the base matrix according to $L^{th}$ column of cost matrix
foreach $i$ from $\max(1, L - N_{Bit})$ to $L - 1$ do
    Calculate the linear approximation error between $i$ to $L$ and save as $C_0(i - (L - N_{Bit}) + 1)$
% Calculate $C(N, L)$
$R = [\max(d + 1, L - N_{Bit}), \cdots, L - 1]$
$C(N, L) = \min_{v_k \in R}\{C(N - 1, v_k) + C_0(v_k - (L - N_{Bit}) + 1)\}$
```

IV. EXPERIMENT AND ANALYSIS OF THE RESULTS

A. EMBEDDED SYSTEM

For the experiment with the proposed algorithm, a Raspberry Pi 3 Model B is used for wireless transmission with a

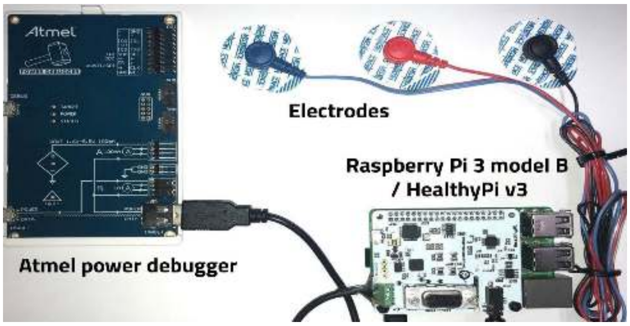


FIGURE 14. Raspberry Pi circuit with Healthy Pi and a power debugger.

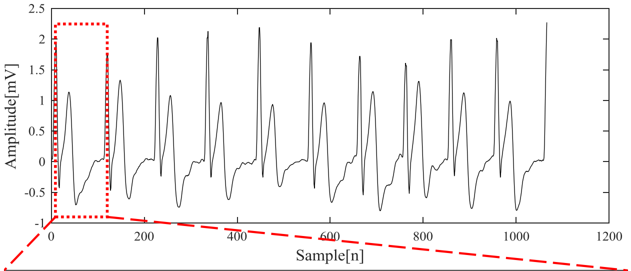
microcontroller unit (MCU), and Healthy Pi v3 is used as the sensor, configured as shown in Fig. 14.

Healthy Pi v3 is well designed to measure ECG signals using 3-lead electrodes, and the Raspberry Pi is also compatible with Healthy Pi. A power debugger is used to measure the change in the power consumption of the proposed algorithm. The data used in the experiments are roughly divided into three types. First, an experiment is carried out with a signal lasting 10 seconds acquired at a sampling frequency of 125 Hz in the embedded system. To emphasize the excellence of the proposed algorithm, the same experiment is also performed on a MIT-BIH ADB record [20]. The MIT-BIH ADB is recorded for approximately 30 minutes at a high frequency of 360 Hz. Finally, an experiment is conducted on the QT-DB of Physionet [21] in the same manner, and the results are compared to verify that the fiducial point detection performance of the conventional PA is preserved during the improvement process. Each signal used in the experiment is preprocessed by applying a 1-25 Hz Butterworth bandpass filter to suppress the baseline wander (0.15 up to 0.3 Hz) and power line interference (30 Hz or 60 Hz). The polygonal approximation is performed after detecting the R-peak in the filtered signal [7].

B. TIME COMPARISON

This section provides experimental results on data obtained directly at a 125 Hz sampling frequency and on MIT-BIH ADB record obtained at a 360 Hz sampling frequency to confirm the improved performance in an embedded system. Fig. 15 shows the results of a detailed recording of the variations in the execution time according to the improved DP for each of the initial vertex intervals within one R-R interval in the signal with a 125 Hz sampling frequency in the embedded system.

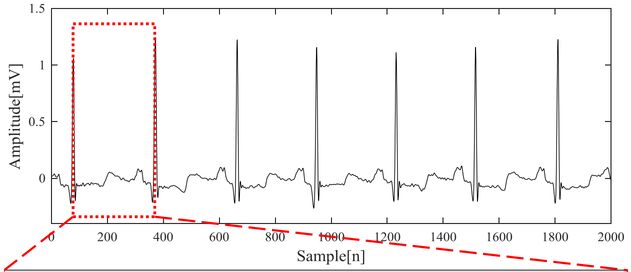
In the case of interval 8, the length of the interval is 10 and the number of vertices is 0, but in the experiment of Section III-C, the execution time increases because the vertex is added by the time difference threshold of $N_{Bit} = 8$. However, because there are more intervals where the execution time is reduced and the reduction ratio is also large, the overall execution time is greatly reduced.



Interval	1	2	3	4	5	6	7	8	9	10
Interval length	7	2	8	21	16	13	31	10	10	9
# of vertices	4	0	5	5	8	2	4	0	1	6
Conv. DP	0.112	0.001	0.174	2.080	1.177	0.490	5.904	0.002	0.148	0.224
DP in Sec. III-A	0.053	0.001	0.071	0.940	0.463	0.246	2.832	0.002	0.124	0.096
DP in Sec. III-C	0.054	0.001	0.073	0.359	0.251	0.170	0.573	0.109	0.108	0.094

*(N_{Bit} : 8, Unit : ms)

FIGURE 15. Processing time for the data measured at 125 Hz in an embedded system.



Interval	1	2	3	4	5	6	7	8	9	10
Interval length	7	5	24	7	177	16	14	8	11	23
# of vertices	4	2	6	0	20	2	1	2	1	6
Conv. DP	0.120	0.047	3.096	0.002	1156.8	0.858	0.273	0.134	0.173	2.729
DP in Sec. III-A	0.060	0.024	1.416	0.001	504.5	0.454	0.312	0.073	0.163	1.231
DP in Sec. III-C	0.060	0.024	1.402	0.002	48.19	0.443	0.305	0.070	0.159	1.226

*(N_{Bit} : 32, Unit : ms)

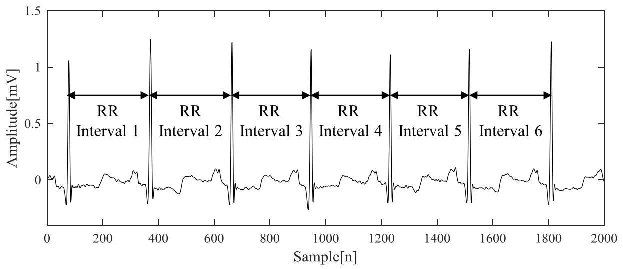
FIGURE 16. Processing time for the MIT-BIH ADB record at 360 Hz.

The experimental results for the MIT-BIH ADB record with a sampling frequency of 360 Hz are shown in Fig. 16.

In this way, the execution time is improved through the improvement of the DP. The longer the input signal and the larger the number of vertices, the greater the improvement in the execution time is, as in the case of interval 7 of Fig. 15 and interval 5 of Fig. 16.

C. MEMORY COMPARISON

Regarding the memory usage, the initial space complexity is $O(L^2N)$ but is improved to $O(L^2)$ and finally to $O(NL)$, as shown in Sections III-A and III-C.2.



RR Interval	1	2	3	4	5	6
Maximum interval	177	186	81	163	67	171
# of maximum vertex	20	17	9	16	9	17
Conv. DP	5,141	4,866	513	3,530	351	4,113
DP in Sec. III-A	150	160	31	124	22	137
DP in Sec. III-C	14	13	3	11	3	12

*(N_{Bit} : 32, Unit : KB)

FIGURE 17. Memory usage of the MIT-BIH ADB record for each RR interval.

Fig. 17 shows the memory usage in applying the improved DP method in Fig. 16 when the transform is added and after the final optimization.

The existing DP method increases the memory usage as the interval length increases and cannot be applied in a low-capacity embedded environment due to the memory overflow. However, as a result of the proposed improvement to the DP method, the memory usage of the first interval, which has the largest memory usage, is greatly reduced from 5,141 KB to 14 KB. Thus, the memory usage of DP is greatly improved, and the memory usage can be maintained in a stable manner since the memory usage for a long interval is reduced more, as shown in Fig. 17.

D. POWER CONSUMPTION

As shown in the previous experiment, the proposed DP improvement can greatly reduce the execution time and memory usage. This can be expected to reduce the power consumption, and we conduct an experiment as shown in Fig. 18 for the same data used in Fig. 15.

As shown in Fig. 18, the improved DP reduces the total power consumption due to the reduction in the execution time. Thus, we confirm that the proposed DP method can be effectively applied even in a low-power embedded environment.

E. SUMMARY OF THE EXPERIMENTS

Fig. 19 summarizes the processing time and memory usage performance according to the DP improvement stages.

Regarding the change in the memory usage, the spatial complexities of the cost matrix and base matrix are $O(L^2N)$ and $O(L^2)$, respectively, but the proposed algorithm significantly reduces the complexities to $O(NL)$ and $O(N_{Bit})$, respectively. In the case of the execution time, the time complexity of conventional DP with a top-down operation is $O(L^2N)$, which is similar to the spatial complexity of the cost matrix. In Sections III-A and III-B, however, the time complexities of $O(L^2)$ and $O(N_{Bit}L)$ corresponding to the memory usage of the base matrix are shown. Since N_{Bit} is given as a constant, the execution time is not influenced by the number of vertices, as shown in Fig. 19. In Section III-C, the type conversion and the operation order are changed to optimize the memory usage. Since the amount of operation does not change, the execution time result is the same. Regarding the power consumption, a large amount of power is required, because the execution time and memory usage are large. However, the power consumption is reduced due to the reductions in the execution time and memory usage.

Fig. 20 shows a graph of the execution time and memory usage changes for each RR interval of the data used in Fig. 17.

For interval 1, the execution time and memory usage decrease by 95.44% and 97.08%, respectively. Comparing the results of Sections III-A based on the characteristics of the ECG signal, the execution time and memory usage decrease by 89.54% and 90.67%, respectively.

Fig. 21 shows the experimental results comparing the measured memory usage and processing time for each RR interval in a 30-minute MIT-BIH ADB record consisting of 2271 RR intervals.

As shown in Fig. 20 and Fig. 21, the proposed method dramatically improves the DP in terms of the execution time and memory area.

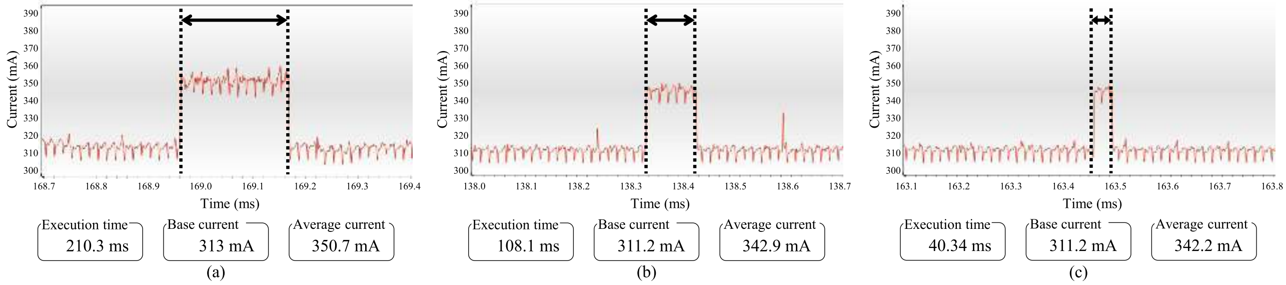


FIGURE 18. Power consumption of Fig. 15: (a) conventional DP, (b) the DP method in Section III-A, and (c) the DP method in Section III-B.

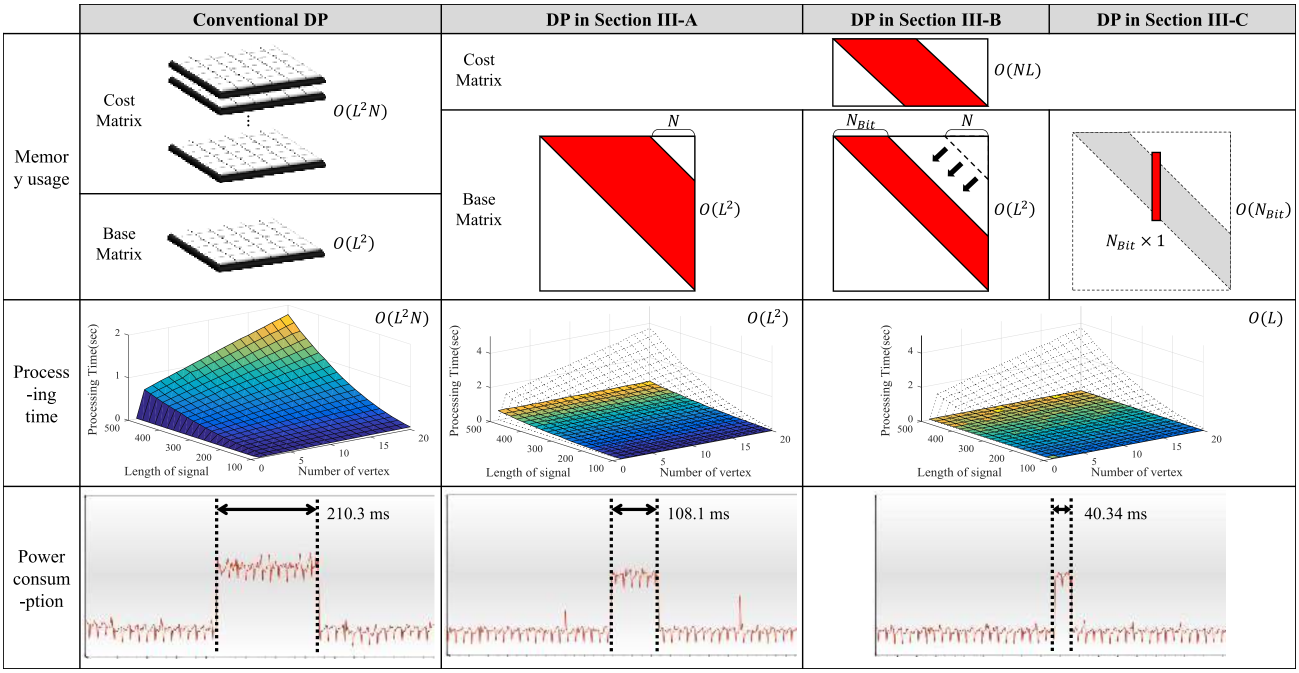


FIGURE 19. Performance comparison with the improved DP method.

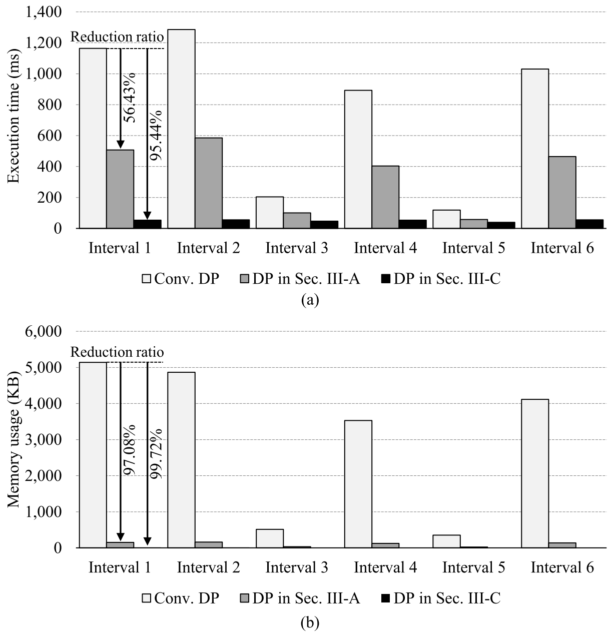


FIGURE 20. Comparison of the changes in the execution time and memory usage of DP in Fig. 17: (a) execution time and (b) memory usage.

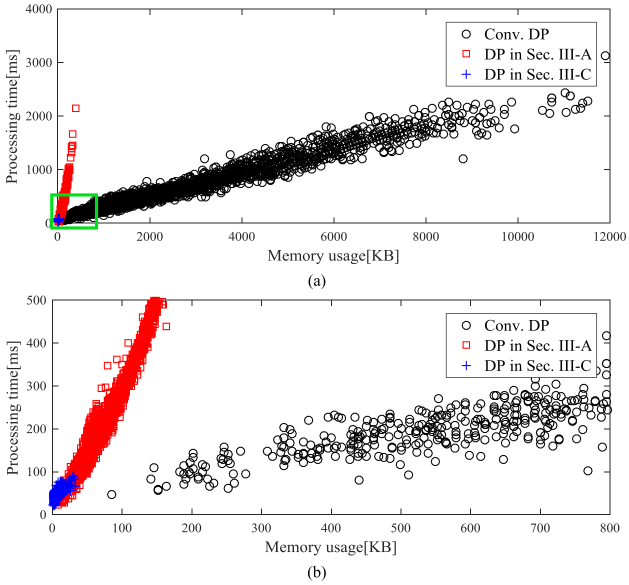


FIGURE 21. Comparison of the changes in the execution time and memory usage of DP in 2271 heartbeats of the MIT-BIH ADB record: (a) total results and (b) zoom-in of the green box region in (a).

F. FIDUCIAL POINT DETECTION

Finally, to verify the effects on the detection of the fiducial points, we apply the improved PA and compare the detection results when applying the same fiducial point-detection method. A fiducial point is detected by analyzing three feature values: the amplitude difference (A), time difference (T), and angles with neighboring vertices (θ), as shown in Fig. 22.

The experiment is conducted in the same way as with the existing PA for QT-DB provided by Physionet, and the results are shown in Table 1.

The experimental results show no meaningful error in the fiducial point detection, and the conventional DP method can be sufficiently replaced with the improved DP method.

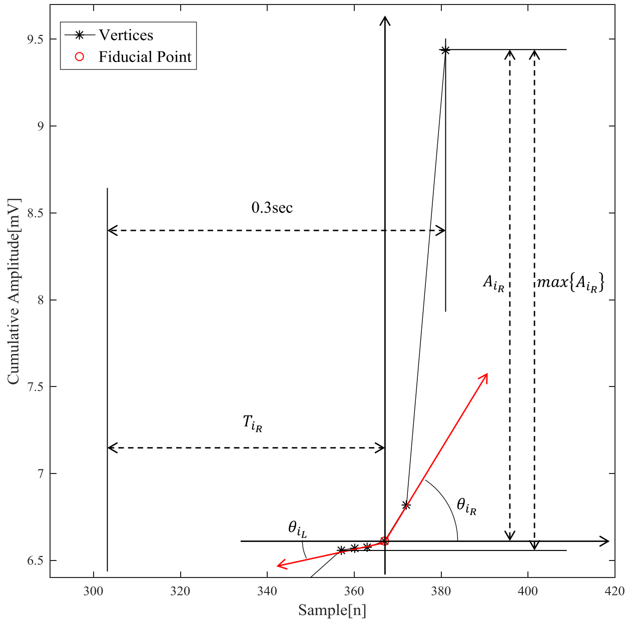


FIGURE 22. Three features of each vertex.

TABLE 1. QRS segmentation performance comparison in the QT-DB.

Method	Ref.	QRS onset (ms)	QRS offset (ms)
This work	-	−4.01 ± 7.99	−5.46 ± 8.03
Lee et al.	[16]	−4.02 ± 7.99	−5.45 ± 8.04
Yazdani and Vesin	[22]	6.16 ± 8.3	1.5 ± 4.2
Martinez et al.	[23]	−0.2 ± 7.2	2.5 ± 8.9
Ghaffari et al.	[24]	−0.6 ± 8.0	0.3 ± 8.8
Manriquez and Zhang	[25]	−2.6 ± 7.1	0.7 ± 8.0
Manriquez and Zhang	[26]	0.58 ± 7.18	−0.95 ± 8.25
Dumont et al.	[27]	0.3 ± 6.6	−1.9 ± 8.3
Martinez et al.	[9]	4.6 ± 7.7	0.8 ± 8.7
Jane et al.	[28]	−7.82 ± 10.86	−3.64 ± 10.74
Laguna et al.	[29]	−3.6 ± 8.6	−1.1 ± 8.3
Tolerance	[30]	6.5	11.6

V. CONCLUSION

The PA requires a long execution time and a large memory usage during the application of DP. It is inefficient to apply in embedded environments. Accordingly, in this paper, DP is improved in three steps to enable real-time application in embedded environments. In the first step, the characteristics of the PA are effectively analyzed in ECG signals to ensure that the DP performance is maintained while the execution time and memory usage are improved. In the second step, the time information at the vertex is expressed using the time differences with the previous vertex, and time difference thresholds are applied together. The second step significantly improves the execution time while maintaining most of the fiducial point detection performance, enabling a real-time application in embedded environments. In the third step, by determining the adaptive thresholds based on the maximum error when calculating the optimization errors, the error is effectively scaled to maintain the error deviation in the type conversion, enabling a stable memory improvement. In addition, in the bottom-up operation, the size of the base matrix is reduced significantly after replacing the row-by-row operation of the cost matrix with a column-by-column operation, which can operate stably even in an embedded environment with a low memory capacity.

The proposed method can improve the performance by effectively analyzing the characteristics of ECG signals and the DP method for a one-dimensional signal. This result is expected to be applicable not only to ECG signals but also to similar one-dimensional signals. In particular, it is expected that this approach will be useful for signal compression and transmission in an application to signals such as photoplethysmography (PPG) and electroencephalography (EEG) signals.

REFERENCES

[1] G. S. P. Gomes and L. H. C. Ferreira, "A lightweight embedded solution for ECG filtering and QRS complexes localization using wavelets," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2018, pp. 1–6.

[2] B. Mishra, N. Arora, and Y. Vora, "A wearable device for real-time ECG monitoring and cardiovascular arrhythmia detection for resource constrained regions," in *Proc. 8th Int. Symp. Embedded Comput. Syst. Des.*, Cochin, India, Dec. 2018, pp. 48–52, doi: 10.1109/ISED.2018.8704072.

[3] R. J. Huszar, *Basic Dysrhythmias: Interpretation & Management*. Maryland Heights, MO, USA: Mosby, 2007, pp. 1–15.

[4] H. L. Chan, W. S. Chou, S. W. Chen, S. C. Fang, C. S. Liou, and Y. S. Hwang, "Continuous and online analysis of heart rate variability," *J. Med. Eng. Technol.*, vol. 29, no. 5, pp. 227–234, 2005.

[5] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods And Tools for ECG Data Analysis*. Norwood, MA, USA: Artech House, 2006.

[6] B. M. Oussama, B. M. Saadi, and H. S. Zine-Eddine, "Extracting features from ECG and respiratory signals for automatic supervised classification of heartbeat using neural networks," *Asian J. Inf. Technol.*, vol. 14, no. 2, pp. 53–59, 2015. [Online]. Available: https://medwelljournals.com/abstract/?doi=ajit.2015.53.59

[7] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[8] M.-E. Nygårds and L. Sörnmo, "Delineation of the QRS complex using the envelope of the e.c.g.," *Med. Biological Eng. Comput.*, vol. 21, no. 5, pp. 538–547, Sep. 1983, doi: 10.1007/BF02442378.

[9] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, "A wavelet-based ECG delineator: Evaluation on standard databases," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 570–581, Apr. 2004.

[10] J. P. V. Madeiro, P. C. Cortez, J. A. L. Marques, C. R. V. Seisdedos, and C. R. M. R. Sobrinho, "An innovative approach of QRS segmentation based on first-derivative, Hilbert and Wavelet Transforms," *Med. Eng. Phys.*, vol. 34, pp. 1236–1246, Nov. 2012.

[11] M. Elsayed, M. Mahmuddin, A. Badawy, T. Elfouly, A. Mohamed, and K. Abualsaud, "Walsh transform with moving average filtering for data compression in wireless sensor networks," in *Proc. IEEE 13th Int. Colloq. Signal Process. Appl. (CSPA)*, Batu Ferringhi, Malaysia, Mar. 2017, pp. 270–274, doi: 10.1109/CSPA.2017.8064964.

[12] A. S. Abdulbaqi, S. A.-D. M. N. Saif, F. M. M. Falath, and N. A. I. Nawar, "A proposed technique based on wavelet transform for electrocardiogram signal compression," in *Proc. 1st Annu. Int. Conf. Inf. Sci.*, Fallujah, Iraq, Nov. 2018, pp. 229–234, doi: 10.1109/AiCIS.2018.00049.

[13] M. L. Hilton, "Wavelet and wavelet packet compression of electrocardiograms," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 5, pp. 394–402, May 1997, doi: 10.1109/10.568915.

[14] S. Olmos, M. Millan, J. Garcia, and P. Laguna, "ECG data compression with the Karhunen-Loeve transform," in *Proc. Comput. Cardiol.*, Indianapolis, IN, USA, Sep. 1996, pp. 253–256.

[15] T.-H. Kim, S.-Y. Kim, J.-H. Kim, B.-J. Yun, and K.-H. Park, "Curvature based ECG signal compression for effective communication on WPAN," *J. Commun. Netw.*, vol. 14, no. 1, pp. 21–26, Feb. 2012, doi: 10.1109/JCN.2012.6184547.

[16] S. Lee, Y. Jeong, D. Park, B.-J. Yun, and K. H. Park, "Efficient fiducial point detection of ECG QRS complex based on polygonal approximation," *Sensors*, vol. 18, no. 12, p. 4502, Dec. 2018, doi: 10.3390/s1812450.

[17] R. Bellman, and S. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 2015.

[18] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1376–1381, Dec. 1998.

[19] K. J. O'Connell, "Object-adaptive vertex-based shape coding method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 251–255, Feb. 1997, doi: 10.1109/76.554440.

[20] G. B. Moody and R. G. Mark, "The MIT-BIH arrhythmia database on CD-ROM and software for use with it," in *Proc. Comput. Cardiol.*, Chicago, IL, USA, Sep. 1990, pp. 185–188.

[21] P. Laguna, R. G. Mark, A. Goldberg, and G. B. Moody, "A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG," in *Proc. Comput. Cardiol.*, Lund, Sweden, Sep. 1997, pp. 673–676.

[22] S. Yazdani and J.-M. Vesin, "Extraction of QRS fiducial points from the ECG using adaptive mathematical morphology," *Digit. Signal Process.*, vol. 56, pp. 100–109, Sep. 2016, doi: 10.1016/j.dsp.2016.06.010.

[23] A. Martínez, R. Alcaraz, and J. J. Rieta, "Application of the phasor transform for automatic delineation of single-lead ECG fiducial points," *Physiol. Meas.*, vol. 31, no. 11, pp. 1467–1485, Sep. 2010, doi: 10.1088/0967-3334/31/11/005.

[24] A. Ghaffari, M. R. Homaeinezhad, M. Akraminia, M. Atarod, and M. Daevaeiha, "A robust wavelet-based multi-lead electrocardiogram delineation algorithm," *Med. Eng. Phys.*, vol. 31, no. 10, pp. 1219–1227, Dec. 2009.

[25] A. Illanes-Manriquez and Q. Zhang, "An algorithm for robust detection of QRS onset and offset in ECG signals," in *Proc. Comput. Cardiol.*, Bologna, Italy, Sep. 2008, pp. 857–860.

[26] A. Illanes-Manriquez and Q. Zhang, "An algorithm for QRS onset and offset detection in single lead electrocardiogram records," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 541–544.

[27] J. Dumont, A. I. Hernandez, and G. Carrault, "Parameter optimization of awavelet-based electrocardiogram delineator with an evolutionary algorithm," in *Proc. Comput. Cardiol.*, Lyon, France, Sep. 2005, pp. 707–710.

[28] R. Jane, A. Blasi, J. Garcia, and P. Laguna, "Evaluation of an automatic threshold based detector of waveform limits in Holter ECG with the QT database," in *Proc. Comput. Cardiol.*, Lund, Sweden, Sep. 1997, pp. 295–298.

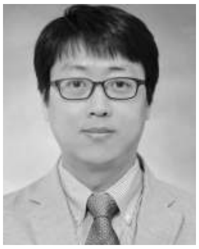
[29] P. Laguna, R. Jané, and P. Caminal, "Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database," *Comput. Biomed. Res.*, vol. 27, no. 1, pp. 45–60, Feb. 1994, doi: 10.1006/cbmr.1994.1006.

[30] "Recommendations for measurement standards in quantitative electrocardiography," *Eur. Heart J.*, vol. 6, no. 10, pp. 815–825, Oct. 1985. [Online]. Available: 10.1093/oxfordjournals.eurheartj.a061766

SEUNGMIN LEE received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2010, 2012, and 2018, respectively. He expanded his research topics to bioinspired signal processing algorithms and electronics systems. He holds a postdoctoral position at KNU. His research interests include signal processing, image processing, bioinspired signal processing, and compact system implementation.

YOOSOO JEONG received the B.S. degree in mathematics and the M.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree in electronics engineering with KNU. His research interests include image signal processing and pattern recognition.

JUNHO KWAK received the B.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2019, where he is currently pursuing the integrated M.S and Ph.D. degrees in electronics engineering. His research interests include biosignal processing and biosignal processor chip implementation.



DAEJIN PARK received the B.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2003 and 2014, respectively. From 2003 to 2014, he was a Research Engineer with SK Hynix Semiconductor and Samsung Electronics, designing high-performance signal processors. He is currently a full-time Professor with the School of Electronics Engineering, KNU. He has explored the accelerated signal processor architecture and low-power chip implementation with a custom-designed software algorithm optimization.

KIL HOUM PARK received the B.S. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1984 and 1990, respectively. He is currently a full-time Professor with Kyungpook National University. He has widely published studies in the areas of computer vision, image processing, electrocardiogram signal processing, and signal compression.