의공학회지 : 제17권, 제1호, 1996
J. of KOSOMBE : Vol. 17, No. 1

연 구 논 문
96 - 17 - 1 - 15

적응 프랙탈 보간을 이용한 심전도 데이타 압축

전 영 일*·윤 영 로**

=Abstract=

ECG Data Compression Using Adaptive Fractal Interpolation

Young Il Jun*, Young Ro Yoon**

This paper presents the ECG data compression method referred the adaptive fractal interpolation algorithm. In the previous piecewise fractal interpolation(PFI) algorithm, the size of range is fixed. So, the reconstruction error of the PFI algorithm is nonuniformly distributed in the part of the original ECG signal. In order to improve this problem, the adaptive fractal interpolation(AFI) algorithm uses the variable ranges. If the predetermined tolerance was not satisfied, the range would be subdivided into two equal size blocks. Large ranges are used for encoding the smooth waveform to yield high compression efficiency, and the smaller ranges are used for encoding rapidly varying parts of the signal to preserve the signal quality.

The suggested algorithm was evaluated using MIT/BIH arrhythmia database. The AFI algorithm was found to yield a relatively low reconstruction error for a given compression ratio than the PFI algorithm. In applications where a PRD of about 7.13% was acceptable, the AFI algorithm yielded compression ratio as high as 10.51, without any entropy coding of the parameters of the fractal code.

Key words : ECG data compression, Piecewise fractal interpolation, Adaptive fractal interpolation

서 론

심전도 신호 분석과 데이타 처리 알고리듬에 관한 연구가 활발히 진행됨에 따라 원격지로의 데이타 전송과 휴대용 Holter 등의 효율적인 데이타 저장 등을 위해 심전도 데이타 처리의 한 분야로서 심전도 데이타 압축에 관한 많은 연구가 이루어져 왔다[1].

최근 발표된 분할 프랙탈 보간(piecewise fractal interpolation:PFI)을 이용한 심전도 데이타 압축 알고리듬[2]은 임의의 신호가 자기 자신의 유사한 부분들로 표현될 수 있다는 프랙탈 모델(fractal model) 이론에 근거하여 일차원 신호를 데이타 자신의 부분들에 대한 수축적 유사 변환(contractive affine transformation)으로 구성하고 그 변환식의 계수만을 저장하는 압축 방법으로, 기존의 직접 데이타 압축 방법보다 임상적으로 중요한 부분에 대한 재생오차를 줄이고 원래신호의 미세한 파형들까지 제대로 복원됨을 보였다. 하지만 이 방법은 전체 데이타를 같은 크기의 구간들로 나누어 변환하기 때문에 정해진 구간 크기에 따라 압축률이 일정하여 다소 효율이 떨어지는 단점이 있다. 이를 개선하기 위해 본 논문에서는 변환할 구간의 크기가 신호에 따라 가변되는 적응 프랙탈 보간(adaptive fractal interpolation:AFI) 압축 알고리듬을 제안한다. 기존의 PFI 알고리듬에서 가장 유사한 부분을 찾아내기 위한 분할 구간의 크기가 알고리듬을 적용할 때 고정되어지는 것에 반해, AFI 알고리듬은 부호화 후의 재생오차를 부호화 과정에서 계산하여 신호의 복잡도가

〈속보논문〉

* 삼성종합기술원 의료기기연구팀
** 연세대학교 보건과학대학 의용전자공학과
통신저자 : 전영일, 윤영로, (222-749) 강원도 원주시 홍업면 매지리 234번지, Tel. (0371)760-2440, Fax. (0371)760-2197
본연구는 1994년 연세대학교 연구비로 이루어졌음

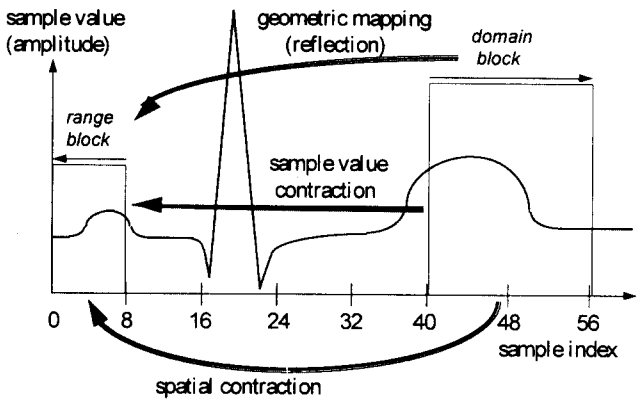

그림 1. 일차원 신호에서의 부분 수축 변환
Fig. 1. Piecewise contractive mapping of 1-D signal

낮은 부분에서는 분할 구간을 크게 하고 신호의 복잡도가 높은 부분에서는 분할 구간의 크기를 작게 하는 방법으로 재생오차를 줄이게 된다.

제안된 AFI 알고리듬을 MIT/BIH 데이터를 이용하여 최근 발표된 PFI 알고리듬[2]과 비교하였다.

프랙탈 이론

70년대 후반, Mandelbrot에 의해 프랙탈 개념이 도입된 이후, Barnsley[3]와 Jacquin[4] 등에 의해 반복 함수계 (iterated function system : IFS) 이론과 이를 발전시킨 분할된 반복 함수계 (partitioned iterated function system : PIFS) 등의 프랙탈 이론이 정립되어 영상 데이타 압축 등의 신호처리 분야에 적용되어 왔다[3-6]. Barnsley에 의해 제안된 IFS 이론은 수축 변환 (contractive mapping)의 성질을 갖는 유사 변환 (affine transformation)들의 모임으로 표현된다.

$$w_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \qquad (1)$$

유사 변환식은 식 (1)과 같이 나타내며, 두 데이타 간의 자기 유사성을 표현해 주는 좌표 변환식이다. 여기서 변환 매개변수인 a_i, b_i, c_i, d_i는 원래의 데이타를 확대, 축소, 회전시키는 역할을 하고, e_i, f_i는 좌표축 방향으로 이동시키는 역할을 하게 된다. 이러한 변환식이 주어진 데이타에 대해 수축적일 때 이를 수축 변환이라 하며, 이 변환식을 임의의 데이타에 반복적으로 적용하게 되면 간단한 알고리듬으로 매우 복잡한 프랙탈 형상을 만들어 내는 것이 가능하게 된다. 역으로 복잡한 프랙탈 형상에서 그 변환식을 찾아내는 것이 프랙탈 부호화 과정이다.

일차원 신호에서의 프랙탈 부호화 방법은 주어진 신호를 여러개의 부분으로 나누고 각 부분에 대해 그 보다 큰 또 다른 부분들과의 유사성을 계산하여 가장 오차가 작은 부분들의 변환식 계수를 구해내는 방법이다. 그림 1은 심전도 신호를 예로들어 임의의 부분들로 나누어진 일차원 신호의 프랙탈 부호화를 위한 부분 수축 변환을 나타낸다. 실제 심전도 신호에 대한 프랙탈 부호화는 그림 1에 나타낸 것보다 훨씬 더 많은 수의 블록들로 나뉘어 진다.

일반적으로, 프랙탈 부호화에서 주어진 신호는 부호화 과정을 거쳐 레인지(range)라 부르는 정의역 블럭과 부호화 과정에서 도메인(domain) 이라 불리는 치역 블럭들로 나누어진다. 예를 들어, 그림 1에서, 주어진 신호를 8개의 샘플을 갖는 부분들과 이 보다 큰 16개의 샘플을 갖는 부분들로 나눈다면, 각각은 정의역 블록과 치역 블럭에 해당된다. 전체 신호는 여러 개의 중첩되지 않는 치역 블럭들로 나뉘고, 각 정의역 블럭은 임의의 영역으로부터 중첩되어 선택되어질 수 있는 치역 블럭들 중에서 자기와 가장 유사한 치역 블럭과 정합되어진다. 즉 치역 블럭을 정의역 블럭 크기 만큼씩 중첩시켜 설정한다면 그림에서 첫번째 정의역 블럭은 6번째 치역 블럭을 x축과 y축 방향으로 축소시키고 x축에 대해 반사시킨 것과 가장 유사하게 된다. 이때 x축과 y축 방향의 수축도와 반사시킨 모양임을 나타내는 부호만 저장하면 이들 변환식의 계수로부터 원래의 정의역 블럭 신호를 재현할 수 있게된다.

프랙탈 보간

Barnsley는 임의 신호의 구간에 대한 함수의 보간을 위해 IFS의 사용을 제안하였다[3]. 제안된 프랙탈 보간 함수는 일차원 신호에 대한 부호화 방법을 제공한다.

일차원 신호에서 n 번째 데이타의 x 좌표와 y 좌표 값이 각각 u_n과 v_n인 데이타 집합을 $\{(u_n, v_n) : n=0, 1, N;\ u_n<u_n+1\}$ 이라 하고, 주어진 데이타 집합에 대한 보간 점들(interpolation points)의 집합을 $\{(x_i, y_i) : i=0, 1, \cdots, M;\ M \le N\}$ 이라 하면, 공간 X는 $x \times y$ 평면이고, 보간 함수(interpolation function)는 다음과 같은 형태의 M개의 유사 변환들로 구성된다.

$$w_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \quad i = 1, 2, \cdots\cdots, M \qquad (2)$$

이 식에서 d_i는 변환 i에 대한 수축도 라고 하는데 이것은 실수이고 구간 $(-1, 1)$ 사이의 값이다. 이 값은 보간 점들에 무관하고, 보간 점들 사이의 보간 함수 모양을 조정하는 역할을 한다. 일반적으로 자기-유사 프랙탈 모델은 식 (1)의 유사 변환에서 b_i에 0을 대입하면 식 (2)의 계수 행렬과 같이되어 선형 프랙탈 보간 함수가 단일 값을 갖도록 한다.

각 유사 변환은 보간 점들 집합의 두 끝 점을 두 연속한 보간 점 쌍으로 정합시킨다. 즉,

$$w_i \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} \text{ and } w_i \begin{pmatrix} x_M \\ y_M \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

$$i = 1, 2, \cdots\cdots, M \qquad (3)$$

따라서 각 변환에 대한 수축도 d_i가 선택되면, 나머지 매개변수들은 수식 (3)을 사용하여 다음과 같이 주어진다. 수축도를 구하는 방법은 PFI 알고리듬[2]과 마찬가지로 Mazel과 Hayes에 의해 제안된 방법[7]을 사용하였다.

$$a_i = \frac{x_i - x_{i-1}}{x_M - x_0} \qquad (4)$$

$$l_i = \frac{x_M x_{i-1} - x_0 x_i}{x_M - x_0} \qquad (5)$$

$$c_i = \frac{y_i - y_{i-1}}{x_M - x_0} - d_i \frac{y_M - y_0}{x_M - x_0} \qquad (6)$$

$$f_i = \frac{x_M y_{i-1} - x_0 y_i}{x_M - x_0} - d_i \frac{x_M y_0 - x_0 y_M}{x_M - x_0} \qquad (7)$$

변환 매개변수들이 결정된 후에, 선형 프랙탈 보간 함수는 다음에 설명하는 알고리듬에 의해 구현될 수 있다.

A_0를 점, 선분, 사각형 등과 같은 $x \times y$ 평면내의 콤팩트 집합이라 하자. 그런 다음 새로운 집합 A_1을 만들기 위해 A_0에 변환을 적용하면

$$A_1 = \bigcup_{i=1}^{M} w_i(A_0) \qquad (8)$$

가 된다. 이제 A_1을 임시로 저장하고 A_0를 버린 다음 A_2를 만들기 위해 A_1에 변환을 적용하면

$$A_2 = \bigcup_{i=1}^{M} w_i(A_1) \qquad (9)$$

가 되고 A_1은 버린다. 이런 식으로 만들어진 집합 $\{A_0, A_1, A_2, \cdots\}$의 시리즈는 선형 프랙탈 보간 함수에 수렴하게 된다. 여기서 A_0는 심전도 등의 일차원 신호에서 분할된 각 치역 블럭을 뜻하고, A_0에 변환을 적용하여 얻어진 최종 결과가 일차원 신호의 임의의 정의역 블럭에 수렴하면 이 때의 변환에 사용된 매개변수가 각 정의역 블럭의 부호화 결과가 되는 것이다.

프랙탈 보간 함수에서 저장되어야 할 매개변수는 수축도 d_i와 변환하고자 하는 구간의 왼쪽 끝점의 함수값, 그리고 상응하는 치역 블럭의 위치와 그 양쪽 끝 점의 함수값 등이다. 그 중에서 수축도 값은 변환의 수축 여부를 결정짓는 매개변수로 반드시 1보다 작은 값을 택해야 한다.

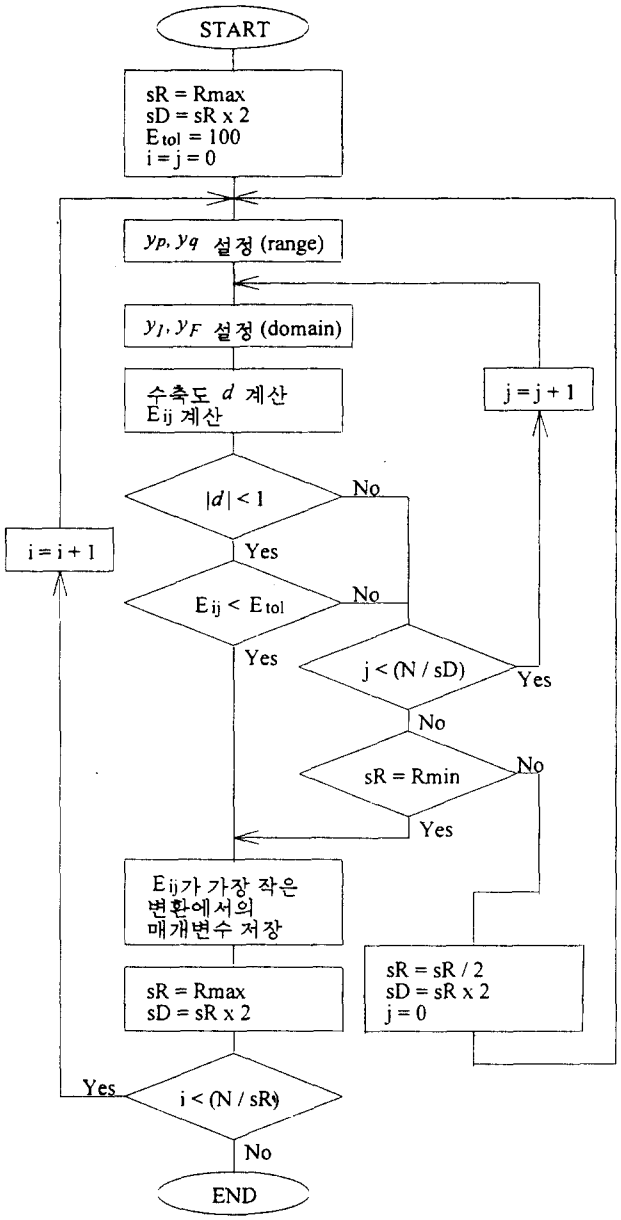

그림 2. AFI 알고리듬의 부호화 과정
Fig. 2. Encoding of AFI algorithm

적응 프랙탈 보간 알고리듬

기존의 PFI 알고리듬은 압축을 수행하기 전에 이미 정의역 블럭 크기가 정해진다. 즉, 압축된 신호를 복원할 때 오차가 커지는 부분에 대해서도 같은 크기의 정의역 블럭들로 나누어 부호화하기 때문에 재생 오차가 특정 부분에서 크게 나타나게 된다. 이러한 단점을 개선하기 위해, 주어진 신호의 복잡도에 의한 자기 유사성의 정도에 따라 변환하고자 하는 정의역 블럭의 크기를 가변하는 적

응 프랙탈 보간 방법을 구현하였다. 자기 유사성의 정도는 위에서 설명한 수축도 d_i 값에 의해 비교되어질 수 있다. 그림 2는 적응 프랙탈 알고리듬의 부호화 과정을 나타낸 것으로, 부호화 과정에서 재생후의 오차를 계산하고 그 값이 미리 정한 조건을 만족하지 못하면 분할되는 정의역 블럭 크기를 줄여 조건이 만족될 때까지 수축도와 재생오차를 다시 계산한다.

전체 샘플 수가 N 개인 주어진 입력 신호에 대해, AFI 알고리듬은 정의역 블럭의 크기 sR을 최대 정의역 블럭 크기 R_{max}로 설정하고 부호화하기 시작한다. 주어진 크기의 i번째 정의역 블럭에 대해 가능한 모든 j번째 치역 블럭 들의 프랙탈 보간 함수를 각각 계산하고, 수축도 d가 1보다 작은 치역 블럭들의 프랙탈 보간 함수와 정의역 블럭 내의 원래 함수를 비교하여 그 오차 E_{ij}중에서 가장 작은 값이 미리 정한 허용오차 E_{tol}보다 작으면, 그 변환의 매개 변수들을 저장하고 i를 증가시켜 다음 정의역 블럭들에 대해 반복한다. 만약 미리 정한 허용오차를 만족하지 못할 경우, 정의역 블럭는 더 작은 두 개의 정의역 블럭로 나뉘고 각 정의역 블럭에 대해 허용오차 조건이 만족되거나 정의역 블럭의 크기가 미리 정한 최소 정의역 블럭 크기 R_{min}에 이를 때까지 위의 과정을 반복한다.

각 변환식에서 저장되어야 하는 매개변수는 정의역 블럭의 크기와 왼쪽 끝점의 함수값, 관련된 치역 블럭의 위치와 양쪽 끝점의 함수값, 수축도 등이다.

AFI 알고리듬에서 정의역 블럭의 크기를 분할하는 기준이 되는 허용오차 E_{tol}에 따라 압축률이 달라진다. 허용오차가 크면 주어진 신호는 대부분 크기가 큰 정의역 블럭들로 나뉘어 압축 효율은 높아지지만 재생오차가 커지고, 허용오차가 작으면 주로 크기가 작은 정의역 블럭들로 나뉘게 되어 압축 효율은 떨어지지만 재생오차가 작아지게 된다.

심전도 신호에서 복잡도가 높은 부분은 주로 QRS complex이고 복잡도가 낮은 부분은 P 파나 T 파등이다. 따라서 AFI 알고리듬을 적용한 결과 QRS complex 부분은 여러 개의 작은 정의역 블럭들로 부호화 되고 그 밖의 부분들은 더 큰 정의역 블럭들로 부호화 되게 된다.

실험결과 및 고찰

프랙탈 보간 함수를 이용하여 일차원 데이타를 부호화하는 데는 매개변수의 선택에 따라 많은 구현 방법이 가능하므로 데이타 압축에 응용하기 위해서는 매개변수의 선택에 제한을 가하고 각 매개변수를 양자화 할 필요가 있다.

PFI 알고리듬에서는, 치역 블럭의 크기를 정의역 블럭 크기의 두 배로 설정하고 치역 블럭의 시작점을 각 정의역 블럭의 시작점과 일치시킴에 의해 관련된 치역 블럭의 양 끝점의 함수값은 저장하지 않아도 된다. 또한 각 정의역 블럭의 시작점을 이전 정의역 블럭의 끝점과 중첩시킴에 의해 관련된 치역 블럭의 양 끝점의 함수값은 저장하지 않아도 된다. 또한 각 정의역 블럭의 시작점을 이전 정의역 블럭의 끝점과 중첩시킴에 의해 정의역 블럭의 오른쪽 끝점의 함수값도 저장할 필요가 없게 된다. 하지만 AFI 알고리듬에서는 정의역 블럭의 크기가 가변되기 때문에 정의역 블럭의 크기와 관련된 치역 블럭의 양쪽 끝점의 함수값을 추가로 저장해야 한다. 때문에 PFI 알고리듬보다 각 변환에서 저장해야 할 매개변수의 수는 많아지지만, 주어진 신호에 대해 보다 큰 정의역 블럭들을 적용하기 때문에 전체적으로 변환의 수가 적어 실제 저장되는 데이타 양은 훨씬 작아진다.

본 논문에서는 최대 정의역 블럭 크기 Rmax와 최소 정의역 블럭 크기 Rmin을 각각 64와 8로 설정하고 각 변환에서의 정의역 블럭의 크기를 8, 16, 32, 64의 네 종류로 제한하여 저장시 2bits 만을 할당하도록 하였다. 즉, PFI 알고리듬은 프로그램이 시작되기 전에 정해진 하나의 정의역 블럭 크기를 사용하지만, AFI 알고리듬에서는 프로그램이 시작된 후에 신호에 따라 위의 네 가지 정의역 블럭 중 가장 큰 크기의 정의역 블럭를 사용해서 미리 정한 허용오차를 만족하지 못하면 반으로 나눈 크기의 정

표 1. PFI에서 하나의 변환에 대한 매개변수들
Table 1. Map parameters of PFI algorithm

PFI의 변환 매개변수들	저장되는 bits 수
정의역 블럭의 왼쪽 끝점의 함수값	12
해당 치역 블럭의 위치	8
수축도	6
저장되는 전체 bits 수	26

표 2. AFI에서 하나의 변환에 대한 매개변수들
Table 2. Map parameters of AFI algorithm

AFI의 변환 매개변수들	저장되는 bits 수
정의역 블럭의 왼쪽 끝점의 함수값	12
치역 블럭의 왼쪽 끝점의 함수값	12
치역 블럭의 오른쪽 끝점의 함수값	12
해당 치역 블럭의 위치	8
수축도	6
정의역 블럭의 크기	2
저장되는 전체 bits 수	52

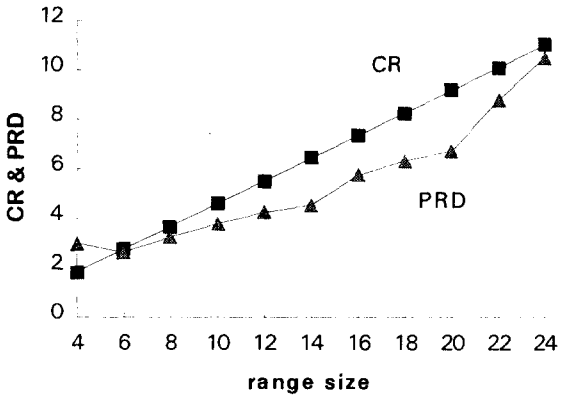

그림 3. AFI의 허용오차 E_{tol} 에 따른 압축률과 실효치 차이
Fig. 3. CR and PRD with tolerance of AFI

의역 블럭를 사용하여 부호화 한다.

표 1과 2는 두 알고리듬에서 하나의 변환에 대해 저장되어야 할 매개변수들의 bit 수를 나타낸다. 각 알고리듬에서 정의역 블럭과 치역 블럭의 양쪽 끝점의 함수값들은 원래 데이타의 한 샘플이 12 bits이므로 같은 bit수를 할당하였다. 해당 치역 블럭의 위치를 나타내는 데는 가능한 치역 블럭의 수를 256 개로 제한하여 8 bits를 할당하였다. 매개변수 중 수축도는 Mazel과 Hayes에 의해 발표된 논문[7]의 실험 결과에 따라 6 bits의 정수로 양자화하여 저장하였다.

두 알고리듬의 압축 성능을 비교하기 위해서, 압축 정도는 압축된 신호와 원래 신호의 비로 그 정도를 나타내는 압축률(compression ratio : CR)을 사용하였고, 재생 오차는 다음 식으로 주어지는 실효치 차이(percent rms difference)를 사용하였다.

$$PRD = \sqrt{\frac{\sum_{i=1}^{n}[x_{org}(i)-x_{rec}(i)]^2}{\sum_{i=1}^{n}x^2_{org}(i)}} \times 100 \qquad (10)$$

여기서 x_{org} 와 x_{rec} 는 원래데이타와 복원된 데이타를 나타내고 n은 오차를 구하고자 하는 구간내의 데이타 수를 나타낸다.

실험에서 데이타는 샘플링 주파수를 400 Hz로 하고, 12 bits의 해상도를 갖는 MIT/BIH 데이타베이스를 사용하였다.

그림 3은 AFI 알고리듬에서 정의역 블럭 분할을 위한 허용오차 E_{tol} 에 따른 압축률과 재생 오차를 나타낸다. 허용오차를 크게 할 수록 원래 신호의 보다 큰 정의역 블럭들로 표현되기 때문에 더 높은 압축률을 얻을 수 있으며, 반면에 재생 오차는 증가하게 된다. 허용오차를 작게 설정하면 이를 만족시키기 위해 주어진 신호의 대부분이 보다 작은 정의역 블럭 들로 나뉘어 변환되기 때문에 압축률은 낮아지고 재생 오차는 작아지게 된다. 만약 정의역 블럭 분할을 위한 허용오차를 0으로 정하면, AFI 알고리듬은 주어진 신호를 모두 미리 정한 가장 작은 정의역 블럭 들로 변환하게 되고 따라서 구간의 크기가 고정되어 있는 PFI 알고리듬과 같아지게 된다.

표 3. MIT/BIH 데이터에 대한 AFI 알고리듬의 결과
Table 3. Results of AFI about MIT/BIH data

MIT/BIH Database	PFI 알고리듬		AFI 알고리듬	
	CR	PRD	CR	PRD
105	7.38	5.68	10.24	6.80
106	7.38	18.66	11.82	8.53
111	7.38	8.24	9.60	8.08
117	7.38	17.34	12.05	8.13
122	7.38	7.24	9.75	6.83
208	7.38	6.11	9.60	4.41
평균	7.38	10.54	10.51	7.13

표 3은 정의역 블럭 분할을 위한 허용오차를 32으로 선택한 AFI 알고리듬과 정의역 블럭 크기를 16으로 선택한 PFI 알고리듬으로 MIT/BIH 데이터베이스 105번, 106번, 111번, 117번, 122번, 208번 데이터에 대한 압축률과 재생 오차를 나타낸다. MIT/BIH 데이터베이스 105, 106, 117번은 대부분 정상적인 파형을 나타내는 데이터로 이들에 대한 압축률은 그림 4의 MIT/BIH 데이타베이스 208번 데이터의 경우보다 높은 값을 나타내고 있다. 데이터 208번의 경우는 압축률이 낮은 대신 재생 오차가 작게 나타나고 있어 전체적인 성능은 비슷한 것으로 나타났다. 또한 폭이 좁은 QRS 파형에서 오차가 커지는 것을 볼 수 있고, 대부분의 오차 신호는 원래 신호에 포함되어 있는 잡음 성분에 기인하고 있는 것을 볼 수 있다. 데이터 208번을 포함한 6개 데이터에서 PFI 알고리듬의 평균 압축률은 7.38 이고 평균 PRD는 10.54 였으며, AFI 알고리듬의 평균 압축률은 10.51 이고 평균 PRD는 7.13 이였다.

그림 4는 MIT/BIH 데이타베이스 208번 데이타에 대해 각각 비슷한 압축률을 갖는 PFI와 AFI 알고리듬을 사용하여 압축 재생한 결과이다. PFI 알고리듬에서 압축률과 재생오차는 정의역 블럭 크기에 비례하게 되는데, 이 때 압축률을 좌우하는 정의역 블럭 크기가 정수이어야 하는 제한 때문에 AFI 알고리듬과 똑같은 압축률을 만들 수는 없다. 비슷한 압축률에서 PFI 알고리듬으로 재생된 파형은 미세한 신호들도 매우 작은 오차로 재생되는 것을 볼 수 있으나 신호의 복잡도가 큰 QRS 파형 부분에서 재생 오차가 커지는 것을 볼 수 있다. AFI 알고리듬의 경우 미세한 신호들이 완전하게 재생되지는 않지만 재생

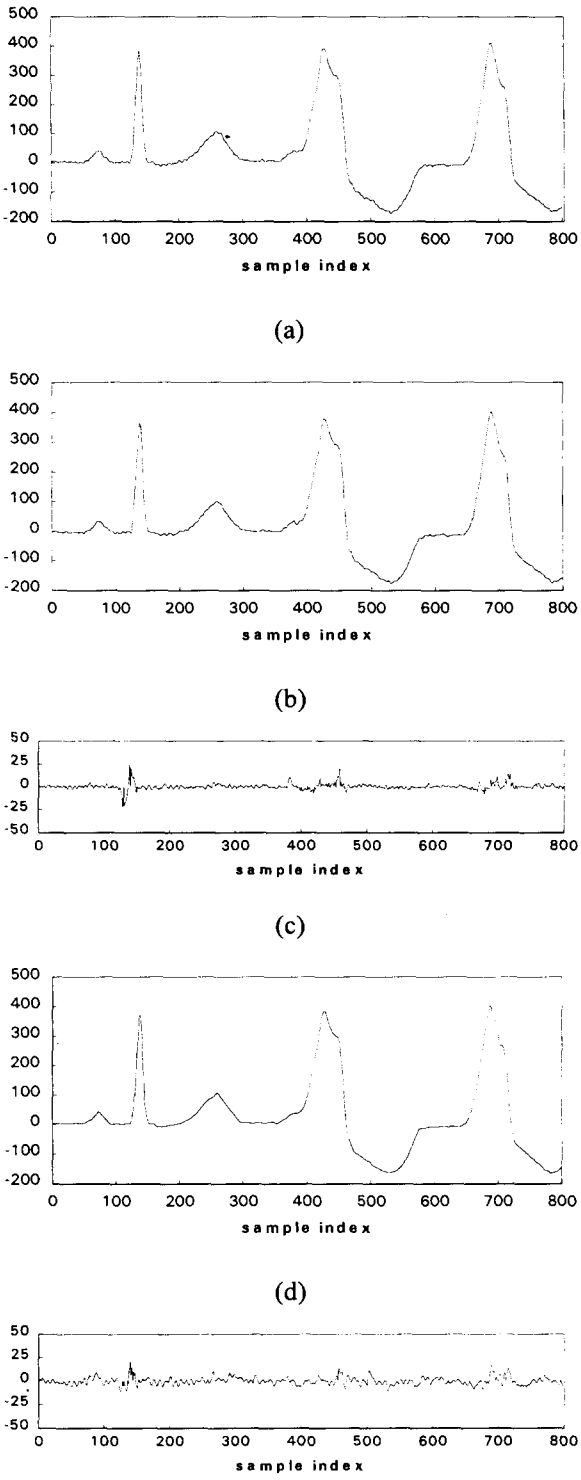

그림 4. PFI와 AFI 알고리듬의 비교
(a)심전도 파형
(b)PFI로 재생된 파형
(c)PFI의 재생 오차
(d)AFI로 재생된 파형
(e)AFI의 재생 오차

Fig. 4. Comparison of PFI and AFI algorithms
(a)Original ECG
(b)Reconstructed signal with PFI
(c)Reconstructed error of PFI
(d)Reconstructed signal with AFI
(e)Reconstructed error of AFI

표 4. PFI, AFI알고리듬의 압축률과 재생 오차
Table 4. CR and PRD of PFI and AFI algorithms

PFI 알고리듬			AFI 알고리듬		
Range size	CR	PRD	허용오차	CR	PRD
8	3.69	3.28	8	5.30	3.67
16	7.38	6.11	16	7.88	4.09
24	11.08	9.97	100	11.59	5.62

오차가 고르게 분포하고 같은 압축률에서 전체적으로 훨씬 낮은 재생 오차를 나타낸다.

표 4는 압축률을 좌우하는 파라미터에 따라 PFI와 AFI 알고리듬의 압축률과 재생 오차를 비교하기 위해 MIT/BIH 데이타 208번을 사용하여 실험한 결과이다. 비교를 위한 두 알고리듬의 압축률이 서로 일치하지 못하는 이유는 PFI 알고리듬의 압축률을 좌우하는 파라미터인 정의역 블럭 크기가 정수라야 하는 제한 때문이다. PFI 알고리듬과 AFI 알고리듬으로 데이타를 압축하는데 걸리는 시간은 400Hz 심전도 데이타 한 개 채널 2048 샘플에 대해 486 PC에서 약 10초 정도이다. 압축하는데 걸리는 시간은 대부분 자기 유사성을 찾기 위해 변환하고자 하는 정의역 블럭과 그 대상이 되는 치역 블럭 사이의 오차를 계산하는 부분에서 소요되는데 이는 허용오차에 비례하게 된다. 아직은 계산시간을 개선시켜야 하는 문제점이 남아있어 실시간 심전도 데이타 압축에는 부적합하고, 이미 받아놓은 데이타를 백업하거나 전송하기 위한 압축 알고리듬으로 사용되어질 수 있다.

결 론

본 논문에서는 반복 수축 변환 이론에 근거한 적응 프랙탈 보간 알고리듬을 제안하고 이를 심전도 데이타 압축에 적용하였다.

기존의 분할 프랙탈 보간 알고리듬은 프랙탈 변환을 구하고자 하는 데이타 구간의 크기가 고정된 방법으로, 압축률을 높일 경우 신호의 복잡도가 커지는 곳에서 재생 오차가 심해지는 현상이 나타난다. 이를 보완하기 위해, 신호의 복잡도에 따라 변환하고자 하는 데이타 구간의 크기가 가변 되는 적응 프랙탈 보간 알고리듬은 신호가 서서히 변화하는 부분에서 압축률을 높이기 위해 폭이 넓은 정의역 블럭를 사용하고, 신호가 급격히 변화하는 부분에서는 재생 오차를 줄이기 위해 폭이 좁은 정의역 블럭를

사용하는 방법으로, 분할 프랙탈 보간 알고리듬의 재생 오차가 특정 부분에 집중되는 것에 반해 적응 프랙탈 보간 알고리듬의 재생 오차는 전 구간에 걸쳐 균일하게 분포하였다.

비슷한 압축률에서 적응 프랙탈 보간 알고리듬은 분할 프랙탈 보간 알고리듬보다 훨씬 낮은 재생 오차를 나타냈다. 이 결과는 엔트로피 코딩을 사용하지 않은 상태에서 비교한 것으로, 저장된 매개변수에 대해 허프만 코딩이나 산술 코딩 등의 처리를 더 거친다면 보다 높은 압축률을 얻을 수 있다. 또한 유사성을 비교하기 전에 DPCM (differencial pulse code modulation) 등의 전처리 과정을 더 거친다면 보다 높은 압축률을 기대할 수 있다.

참 고 문 헌

1. S. Jalaleddine, C. Hutchens, R. Strattan, and W. Coberly, *"ECG data compression techniques-A unified approach," IEEE Trans. Biomed. Eng.*, vol. BME-37, pp. 329-343, Apr. 1990.
2. 전영일,정현민,윤영로,윤형로, *"Holter Data* 압축 알고리즘에 관한 연구," 의공학회지, 제16권 제1호, pp. 17-24, 1995
3. M. F. Barnsley, Fractals Everywhere. New York: Academic, 1988.
4. A. E. Jacquin, *"Image coding based on a fractal theory of iterated contractive image transformations,"* IEEE Trans. Image Processing, vol. 1(1), pp. 18-30, Jan. 1992.
5. E. W. Jacobs, R. D. Boss, Y. Fisher, *"Fractal-Based Image Compression, Ⅱ,"* NOSC TR-1362, Naval Ocean Systems Center, San Diego, CA., June 1990.
6. E. W. Jacobs, Y. Fisher and R. D. Boss, *"Image Compression: A study of the iterated transform method,"* Signal Processing, vol. 29, no. 13, Dec. 1992.
7. D. S. Mazel, M. H. Hayes, *"Using iterated function systems to model discrete sequences,"* IEEE Trans. Signal Processing, vol. 40. no. 7, pp. 1724-1734, July 1990.

=국문초록=

본 논문은 심전도 데이터 압축을 위해 적응 프랙탈 보간 (AFI) 알고리듬 방법을 제안한다. 기존의 분할 프랙탈 보간 (PFI) 알고리듬은 고정된 크기의 정의역 블럭을 사용한다. 따라서 그 재생오차가 원래의 심전도 신호의 특정 부분에 집중된다. 이 문제를 해결하기 위해 적응 프랙탈 보간 알고리듬에서는 가변 정의역 블럭을 사용한다. 만약 미리 정한 허용오차가 만족되지 않으면 정의역 블럭은 두 개의 더 작은 정의역 블럭들로 나뉘어지게 된다. 큰 정의역 블럭들은 높은 압축 효율을 얻기 위해 굴곡이 적은 파형을 부호화 하는데 사용되고, 더 작은 정의역 블럭들은 신호의 질을 유지하기 위해 급격히 변화하는 파형을 부호화 하는데 사용된다.

제안된 알고리즘은 MIT/BIH 데이터베이스를 사용하여 평가되었다. AFI알고리듬은 주어진 압축률에서 기존의 PFI알고리듬보다 상대적으로 적은 재생 오차를 나타냈다. 약 7.13% 정도의 실효치 차이가 허용되는 응용에서, AFI알고리듬은 매개변수들에 대한 엔트로피 코딩 없이 10.51 이상의 압축률을 얻을 수 있었다.