

# Zero-Order 압축 방식에서 ECG 데이터 압축률 예측과 적용에 관한 연구

안형민, 김영길

아주대학교 공과대학 전자공학과, uP&DSP 실험실

(Tel) 0331-219-2378 / (Fax) 0331-212-9531

## A Study on Prediction and Application of ECG Data Compression Rate at Zero-Order Compression

H. M. Ahn, Y. K. Kim

uP&DSP Lab, Dept. of Electronics Engineering, Ajou University

e-mail : goodmin@unitel.co.kr

### Abstract

There are many kinds of method to compress data. To very simple methods from very complex methods, a kind is various. In this study, the simplest form of the Tolerance-Comparison method, zero-order method is used. Using this method, despite using low speed CPUs, it is possible to compress real time data. So this method is suitable for ECG holter system.

In this study, to complement zero-order method, it is needed to develop prediction technique and to research ways to apply the technique.

### 서론

ECG 압축 알고리즘은 ECG 홀터 시스템의 등장으로 인해 그 필요성이 증가되었다.

24시간 동안 ECG 데이터를 저장하기 위해 서는 상당히 큰 저장 용량을 갖는 장치가 있어야 한다. 요즘은 메모리 기술의 발전으로 ECG 데이터를 압축하지 않고도 충분히 24시간의 ECG 데이터를 저장할 수 있다. 그러나 ECG 데이터를 압축하여 저장하였을 때 본래 ECG 신호와 별 차이를 나타내지 않고 데이터 양을 줄일 수 있다 먼 더 긴 시간, 더 많은 채널을 저장할 수 있기 때문에

압축 알고리즘의 효율성을 무시 할 수 없을 것이다.

많은 ECG 압축 알고리즘들 중에서 소위 Tolerance-Comparison 방식으로 분류되는 압축 알고리즘이 있는데, 이 압축 알고리즘의 압축률을 결정하는 가장 중요한 요소는 Tolerance Value( $\epsilon$ )이고, 이 값의 크기에 따라 압축률과 신호 왜곡의 크기가 변하게 된다.

이 압축 알고리즘의 단점은 어떠한 신호에 대해 가장 적절한  $\epsilon$  값을 정할 기준이 없다는 것이다. 단지 이 알고리즘을 사용하는 사람이 실험 결과에 의해 자신이 만족하는 결과를 보여주는  $\epsilon$  값을 선택하여 그 값을 압축에 이용하는 방식이다.

그러나 ECG 신호는 각 사람마다 ECG 신호의 크기가 다를 뿐 아니라 한 사람의 각 채널마다 나오는 ECG 신호의 크기 또한 다르기 때문에 ECG 신호를 압축하기 위해 T-C 방법을 사용할 때 고정된  $\epsilon$  값을 사용하는 것은 매우 안 좋은 결과를 가져올 수 있다.

본 연구에서는 T-C 방법을 사용하여 ECG 신호를 압축할 때, 사용자가 정한  $\epsilon$  값에 의해 얻어질 압축률의 대략적인 크기를 미리 알아내는 방법과 이러한 방식을 토대로 다양하게 변하는 ECG 신호의 크기에 상관없이 압축률을 일정하게 유지할 수 있는 방법을 연구하였다.

### Tolerance-Comparison 압축 알고리즘

T-C 알고리즘은 크게 2가지 종류로 나뉜다. 첫 번째는 Predictor 방법, 두 번째는 Interpolator 방법이 있다. 두 방법의 가장 큰 차이는 현재 시점을 기준으로 어느 시점의 데이터를 처리할 것인가이다.

먼저 Predictor 방식은 그 뜻과 같이 현재까지 들어온 데이터를 가지고 다음에 들어올 데이터를 예측한다. 이 예측된 값과 실제 들어온 데이터의 값의 차를 구해 그 차이가 미리 정한 Tolerance Value( $\epsilon$ )보다 작으면 데이터를 저장하지 않는다. 그러나 그 차이가  $\epsilon$ 보다 크다면 데이터는 저장된다.

그 다음 Interpolator 방식은 과거의 데이터와 현재 데이터 사이에 있는 데이터들의 저장 여부를 결정하는 것이다. 소위 보간법이라고 불리는 방법을 이용하여 과거 데이터와 현재 데이터 사이에 있는 데이터들을 추정한다. 그리고 각 위치의 실제 데이터와 차를 구해 그 차이가  $\epsilon$ 보다 작으면 데이터를 저장하지 않고  $\epsilon$ 보다 크다면 데이터는 저장된다.

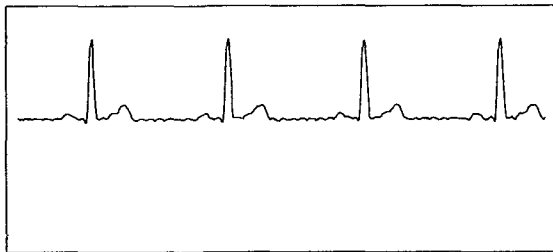


그림 1. 128Hz로 샘플링하고 전처리과정을 거친 ECG 신호

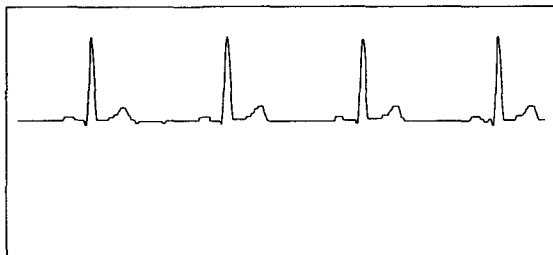


그림 2(a).  $\epsilon$  값을 30으로 하고 압축/복원한 ECG 신호

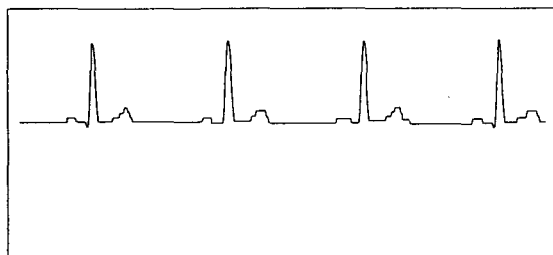


그림 2(b).  $\epsilon$  값을 50으로하고 압축/복원한 ECG 신호

이와 같이 T-C 알고리즘은 손실 압축 알고리즘이고 이 알고리즘에 가장 큰 영향을 주는 요소는  $\epsilon$  값이다.  $\epsilon$  값이 커질 수록 압축률은 높아지겠지만 복원된 신호는 본래 신호와의 오차율도 높아질 것이다. 그러나  $\epsilon$  값이 작아 질 수록 압축률은 떨어지겠지만 본래 신호와의 오차율은 작아질 것이다.

### Zero-Order Predictor & Interpolator

T-C 알고리즘의 Predictor와 Interpolator 방식은 크게 zero-order(ZOP, ZOI)와 first-order(FOP, FOI) 방식이 많이 사용되나 본 연구에서는 zero-order 방식만을 사용하였다.

기본적으로 Predictor 방식은 예측식이 필요하다. ZOP는 예측식이 영차 함수임을 의미한다. 다시 말해 ZOP는 예측한 점이 이전에 기준이 된 데이터 값이 되는 것이다.

ZOI는 현재 점과 이전에 기준이 된 데이터 값을 비교한다. 결국 ZOI와 ZOP는 동일한 알고리즘이다. 단지 비교할 데이터의 시점을 어디에 두고 있는지 만 다를 뿐이다.

그림1은 전처리 과정을 거친 ECG 신호이다. 여기서 전처리과정을 거친 신호란 전력선 잡음제거와 이동평균을 거친 신호를 말한다.

그림2는  $\epsilon$  값을 변화시키며 ZOP로 원 신호를 압축하고 복원한 ECG 신호 파형이다. 그림2를 보면 알 수 있듯이  $\epsilon$  값이 작은 경우의 복원된 파형이 원 신호와 더 비슷함을 알 수 있다.

### 압축률 추정 알고리즘

ZOP나 ZOI는 시점이 다른 데이터와 데이터의 차를 계산하고 그 값이  $\pm \epsilon$  안에 있는지를 보고 데이터의 저장 여부를 결정하게 되는 사실을 알고 있다.

그리고 간단한 프로그래밍 작업으로 어떤 신호의 데이터와 데이터 사이의 차이 값에 대한 분포도 얻을 수 있다.

그림3은 그림1에 보여준 ECG 원 신호에서 바로 옆 데이터들끼리의 차이 값에 대한 분포도이다. 그렇다면 그림2에 복원된 신호에서 바로 옆 데이터들끼리의 차이 값에 대한 분포도는 어떻게 될까?

그림4를 보면 알 수 있듯이 ECG 원 신호에서 Tolerance-Value( $\epsilon$ ) 이하의 차이 값들은 모두 0이 되었다는 것을 알 수 있다. 이 사실은 분포도에 나온 데이터 개수를 비교해 보면 된다.

이 말의 의미는 데이터간의 차이 값이  $\epsilon$  보다 작다는

것은 압축이 될 수 있다는 뜻이고 차이 값이  $\epsilon$ 보다 크다는 것은 zero-order 알고리즘을 통해 전혀 압축이 될 수 없다는 뜻이다.

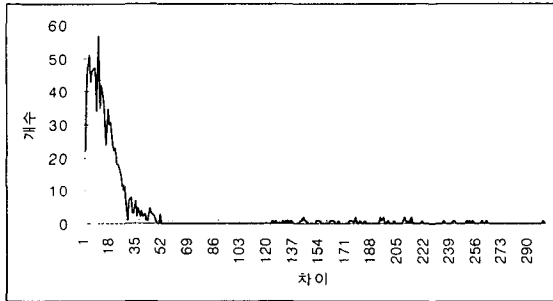


그림 3. ECG 원신호의 분포도

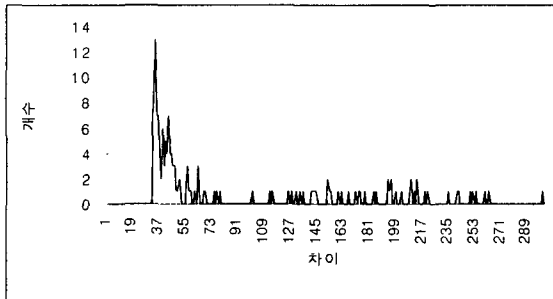


그림 4(a).  $\epsilon=30$ 인 경우 복원된 신호의 분포도(차이 0에서 개수는 818개)

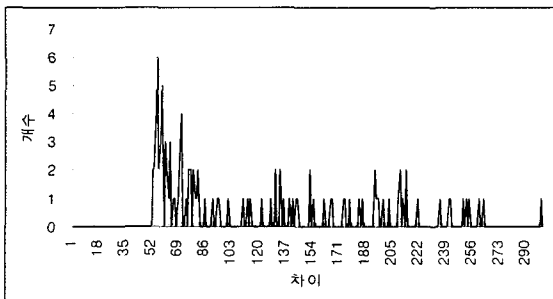


그림 4(b).  $\epsilon=50$ 인 경우 복원된 신호의 분포도(차이 0에서 개수는 871개)

만약에 데이터를 저장하기 위한 코드로 Run-Length 코드를 사용했다면, 압축이 전혀 안되는 한 개 데이터를 저장하기 위해서는 데이터 크기와 연속된 개수를 표현하기 위해 2개의 코드가 필요하게 된다. 결국 이웃하는 데이터들끼리의 차이 값이  $\epsilon$ 값보다 큰 데이터들은 모두 2개의 코드로 구성되어 있을 것이다.

그러므로  $\epsilon$  값과 어느 구간에서의 ECG 신호의 이웃하는 데이터들끼리의 차이 값에 대한 분포를 안다면 그 구간에서의 대략적인 압축률을 압축 해보지 않고 예측할

수 있다는 뜻이다.

표1은 실제 zero-order 압축 후 생성된 데이터 개수와 예측한 압축량을 비교해 놓았다.

(표 안의 숫자는 데이터 개수)

	ECG 신호	zero-order 압축	압축량 예측
$\epsilon = 30$	1000	366	366
$\epsilon = 50$	1000	262	260

표 1. ECG 신호 데이터, 압축 후 데이터, 압축량 예측 데이터 각각의 개수

표를 보면 실제 압축과 예측한 압축률에 차이가 있음을 알 수 있다. 사실 차이가 있어야 정상이다. 왜냐하면 그림4(a),(b) 분포도에서 압축량을 예측할 때 차이가 0인 곳의 데이터 개수가 단지 2개의 코드로 저장 됐다고 가정했기 때문이다. 그럼에도 불구하고 매우 비슷한 결과가 나올 수 있는 것은 ECG 신호 파형의 특징 때문이다.

ECG 신호는 T파 끝부터 P파 시작점까지 수평선의 형태를 갖는다. 또한 그림3의 분포도에서 이웃하는 데이터들 사이의 차이는 0에서 35사이에 대부분 존재한다. 이 범위에 있는 대부분의 데이터는 ECG 신호의 T파와 P파 사이에 있고 그러한 부분은 zero-order 압축했을 때 몇 개의 코드만으로 압축 될 수 있다. 그러므로 실제 압축량과 예측된 압축량의 크기는 비슷할 것이다.

### 압축률 추정 알고리즘 적용

압축률 추정 알고리즘을 적용할 수 있는 시스템은 ECG 데이터를 저장하는 홀터 시스템이고 그러한 시스템 중에서 AZTEC 압축 알고리즘을 이용하는 시스템이 있는데, AZTEC 알고리즘은 zero-order를 기반으로 하는 압축 알고리즘이다.

서론에서도 언급했듯이 ECG 신호는 각 사람마다, 각 채널마다 크기와 형태가 다르기 때문에 고정된  $\epsilon$ 을 사용하는 zero-order 기반의 압축 알고리즘을 적용했을 때 어떤 사람은 적당히 압축이 됐으나 그렇지 않은 사람도 있을 것이다. 또한 각 ECG 채널에 대해서도 동일한 현상이 일어날 수 있다.

또한 압축시작부터 끝까지 ECG 신호 변화에 관계없이 일정한 압축률을 유지하고 싶다면 고정된  $\epsilon$ 을 사용하는 안될 것이다.

이러한 한계를 극복하기 위해 압축률 추정 알고리즘을 적용하면  $\epsilon$  값만으로 일정 구간 압축률을 추정할 수 있

고 ECG는 비슷한 패턴이 주기적으로 발생하므로 추정된 압축률은 더 긴 구간에 대해서도 신뢰성을 확보할 수 있다.

그러므로 초기에 설정한 ECG 홀터 시스템의 목표에 벗어나지 않는지 확인하기 위해 주기적으로 압축률 추정 알고리즘을 적용하여  $\epsilon$  값을 조정해 주면 고정된  $\epsilon$  값을 사용하는 압축 알고리즘의 한계를 극복할 수 있다.

## 참고문헌

- [1] 송근국, 이경중, 윤영로, 윤형로, "플래쉬 메모리 카드를 이용한 홀터 심전계의 설계", 대한의용생체공학회, pp251-259, 19권, 3호, 1998년
- [2] Sateh M. S. Jalaleddine, Robert D. Strattan, "ECG Data Compression Techniques - A Unified Approach", pp329-343, IEEE Transaction on Biomedical, Vol37, No4, April. 1990
- [3] Willis J. Tompkins, "Biomedical Digital Signal Processing", pp193-2-13, Prentice-Hall, 1993