

무선센서네트워크에서 센서 데이터 손실과 오류 감소를 위한 데이터 압축 방법

신동현*, 김창화**

Data Compression Method for Reducing Sensor Data Loss and Error in Wireless Sensor Networks

DongHyun Shin^{*}, Changhwa Kim^{**}

ABSTRACT

Since WSNs (Wireless Sensor Networks) applied to their application areas such as smart home, smart factory, environment monitoring, etc., depend on sensor data, the sensor data is the most important among WSN components. The resources of each node consisting of WSN are extremely limited in energy, hardware and so on. Due to these limitation, communication failure probabilities become much higher and the communication failure causes data loss to occur. For this reason, this paper proposes 2MC (Maximum/Minimum Compression) that is a method to compress sensor data by selecting circular queue-based maximum/minimum sensor data values. Our proposed method reduces sensor data losses and value errors when they are recovered. Experimental results of 2MC method show the maximum/minimum 35% reduction efficiency in average sensor data accumulation error rate after the 3 times compression, comparing with CQP (Circular Queue Compression based on Period) after the compressed data recovering.

Key words: CQP (Circular Queue based on Period), 2MC (Maximum/Minimum Compression), IoT (Internet of Things), WSN (Wireless Sensor Network) Failures, Communication Error, Sensor Data Compression

1. 서 론

무선센서네트워크는 Fig. 1과 같이 센서, 센서 노드, 게이트웨이 노드, 싱크 노드 등으로 구성된다. 각 노드의 센서로부터 센싱(Sensing)된 센서 데이터를 상위 노드로 릴레이 하면서 모니터링 센터로 데이터를 전송하고, 모니터링 센터에서는 수집된 센서 데이

터를 이용하여 유의미한 정보를 찾는다. 특히 무선센서네트워크는 IoT (Internet of Things)의 기반이 되는 기술로써, 현재 스마트 홈, 스마트 자동차, 화재 경보 시스템, 해양 환경 등 다양한 분야에 활용되고 있다[1-5].

하지만 무선센서네트워크는 상용 전원의 사용이 어려운 특징을 가지고 있다. 이로 인해 에너지 효율

* Corresponding Author : Changhwa Kim, Address: (26403) 150, Namwon-ro, Heungeop-myeon, Wonju-si, Gangwon-do, Korea. TEL : +82-33-760-8663, FAX : +82-33-760-8718, E-mail : kch@gwnu.ac.kr

Receipt date : Jan. 11, 2016, Approval date : Jan. 28, 2016

[†] Department of Computer Science & Engineering, Gangneung-Wonju National University (E-mail : dhshin@cs.gwnu.ac.kr)

^{**} Department of Computer Science & Engineering, Gangneung-Wonju National University

* This research was a part of the project titled "Development of the wide-band underwater mobile communication systems" funded by the Ministry of Oceans and Fisheries, Korea and this is also a little bit partly supported by "Full-time Graduate Student Support Work" from Graduate School of Gangneung-Wonju National University.

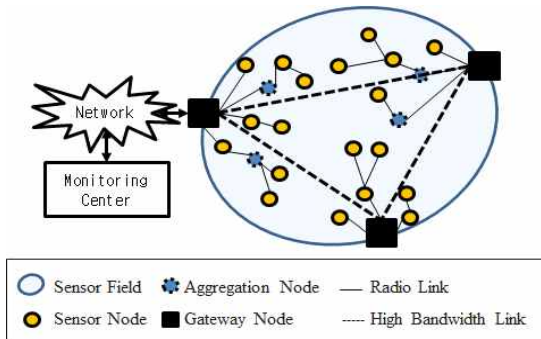


Fig. 1. Components of Wireless Sensor Network.

이 중요하여 저 전력 장치에서도 장기간 작동해야 한다[3,4,6,7]. 궁극적으로 노드는 초소형으로 제작되어야 하기 때문에 하드웨어 자원 또한 매우 제약적이다. 따라서 무선센서네트워크 기술을 사용하기 위해서는 에너지 및 하드웨어 등의 극도로 제한적인 자원을 고려해야 한다.

무선센서네트워크는 기본적으로 센서 데이터를 이용한 정보 수집을 기본으로 하여 활용되고 있다. 하지만 무선통신의 경우 유선통신에 비해 노드의 무선 전원 고갈, 하드웨어 부품과 안테나, 하우징 등이 안개, 바람, 눈, 비 등의 영향을 더 많이 받게 되는데, 이 경우 노드에서 일시적인 혹은 중·장기적인 통신장애가 발생할 수 있다[8].

통신장애가 발생하게 되는 경우 센서 데이터를 릴레이 하는 과정에서 센서 데이터가 손실될 수 있으며, 통신장애 기간이 길어질수록 센서 데이터의 손실로 더 큰 손해가 발생하게 된다. 특히 해양환경에서의 수중통신은 느린 전송속도와 제한된 대역폭을 가지며, 해양의 극한 환경 등으로 인해 전파를 사용하는 통신보다 더 많은 통신장애가 발생할 가능성이 크다[9,10,11]. 통신장애 환경으로 인해 발생한 센서 데이터 손실 문제를 해결하기 위한 여러 방법들이 제안되어왔지만 기존의 센서 데이터 압축 혹은 복구 방법에서는 센서 데이터의 특징을 고려하지 않아 데이터 손실이 많을수록 오차율이 매우 커지게 되는 문제점이 발생했다.

본 논문에서는 이와 같은 문제를 해결하기 위해 무선센서네트워크 통신장애 환경에서 센서 데이터의 손실을 감소시키고, 센서 데이터의 특징을 이용해 센서 데이터를 선택하는 2MC (Maximum/Minimum Compression)방법을 제안한다. 이 방법은 통신장애

가 발생하면 센서 데이터를 원형 큐 자료구조 기반에 저장하고, 메모리에 가득 차게 되면 압축을 진행한다. 센서 데이터 압축 시 센서 주기에 따른 압축 구간 내에서 최대/최소값 중심의 센서 데이터 중심으로 압축한다. 이후 통신이 복구되면 압축된 센서 데이터를 통해 실제 데이터를 복구한다. 센서 데이터는 MCU (Micro Controller Unit)의 플래시 메모리에 저장하여 전원이 끊겨도 저장된 데이터가 보존되며, 최근에는 512KB 이상의 플래시 메모리를 가진 MCU가 출시됨으로써 기존에 비해 원활하게 센서 데이터를 저장할 수 있다. 저장 가능한 양은 센서 데이터와 센싱 주기에 따라 달라진다.

2MC 방법에 대한 실험결과, 압축 횟수를 최대 3회까지 증가시켰을 때 CQP (Circular Queue Compression based on Period)방법과 비교한 평균 누적 오차율은 3회 압축에서 최대 35%감소하였으며, 실험에 사용된 해수온도, 상대습도, 해수 pH, 대기온도 모두 2MC 방법이 CQP 방법보다 평균 누적 오차율이 더 적었다. 4회 압축 시 2MC 방법과 CQP 방법 모두 평균 누적 오차율이 큰 오차를 보였지만, 2MC 방법은 CQP 방법보다 Linear 보간법과 Spline 보간법에 대해 평균 누적 오차율이 약 30% 감소하였다.

본 논문은 2장에서 관련연구, 3장에서 압축을 위한 센서 데이터 선택 방법인 최대/최소값 중심의 2MC (Maximum/Minimum Compression) 알고리즘, 4장에서 2MC 방법의 성능평가, 5장에서 결론으로 마무리를 짓는다.

2. 관련연구

멀티미디어는 정보를 전달하는 매개체인 문자(Text), 이미지(Image), 음성(Sound), 동영상(Video) 등을 두 가지 이상 결합하여 정보를 표현하는 것을 말한다[12]. 멀티미디어의 품질이 높아질수록 높은 저장 용량이 요구되는데, 이때 멀티미디어 압축 기법을 사용하여 데이터의 저장 용량을 줄일 수 있다. 음성 압축의 경우 대표적으로 PCM (Pulse Code Modulation)방식을 사용하여 아날로그 신호를 디지털 신호로 변환하는데, 압축을 위해 PCM 방법에 기반을 둔 예측 기법인 DPCM (Differential Pulse Code Modulation)과 ADPCM (Adaptive Differential Pulse Code Modulation)압축 방법을 사용한다.

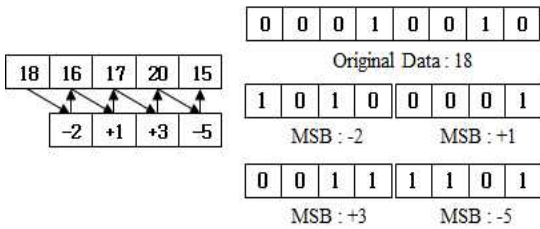


Fig. 2 A Method for DPCM Compression.

DPCM 압축 방법은 처음의 원본 데이터에서 차이 값만 저장하여 압축하는 방법이다[13]. Fig. 2와 같이 첫 데이터 값이 18이고 16, 17, 20, 15의 데이터가 있을 때, DPCM 방법은 첫 데이터 값 18은 PCM 방식에 기반을 두어 저장하고, 그 이후의 값은 차이 값만 저장한다. 차이 값의 경우 가장 첫 번째 비트는 부호 비트로 0인 경우 증가, 1인 경우 감소를 나타낸다. 원본 데이터가 1바이트 단위로 저장되어 있을 때, DPCM 방법을 사용하여 4비트씩 차이 값만 저장하면, 절반 정도의 압축률을 가질 수 있다.

하지만 기울기 과부하(Slope Overload)와 중간 데이터가 손실되면 데이터 복구 시 기울기 과부하가 발생한 이후의 데이터는 모두 오차가 발생한다[12].

ADPCM 방법은 DPCM의 기울기 과부하 문제를 해결하기 위한 방법이다. 기본적인 압축 방법은 DPCM과 동일하며, 기울기 과부하 문제를 해결하기 위해 ADPCM 방법에서는 데이터의 차이 값이 클 때는 차이 값을 저장하는 비트의 수를 증가시키고 그렇지 않은 경우는 감소시켜 적응적(Adaptive)으로 사용할 수 있는 방법이다[14].

손실된 데이터의 복구 방법 중 하나로 센서 데이터가 손실되었을 때 PCA (Principle Component Analysis) 기법을 이용하여 센서 데이터를 복구하는 방법이 연구되었다[15]. PCA는 서로 연관성을 갖는 데이터를 하나의 그룹으로 그룹화 시키면서 데이터를 분석하는 통계학적 모델링 기법이다. 이 기법은 손실된 데이터를 복구하거나 오류를 검출하는데 많이 사용되지만, 기존의 데이터를 분석하여 손실된 데이터를 복구하기 때문에 데이터의 수가 충분하지 않다면 데이터 복구 시 오차율이 커진다.

CQP (Circular Queue Compression based on Period) 방법은 센싱 주기를 기반으로 센서 데이터를 압축하고, 보간법을 이용해 센서 데이터를 복구하는 방법이다[16]. 이 방법은 센서 데이터의 압축 횟수

c 에 따른 센서 데이터의 센싱 주기 2^c 에 해당하는 홀수 번째의 센서 데이터만 선택하여 압축하는 방법으로 압축 방법은 굉장히 간단하지만 이 역시 센서 데이터의 특징을 고려하지 않고 단순히 센서 데이터의 저장 순서에 따라 압축하기 때문에 데이터 복구 시 많은 오차가 발생하게 된다. 이 방법은 본 논문에서 제안하는 방법의 기반이 된다.

3. 2MC (Maximum/Minimum Compression) 알고리즘

3.1 센서 데이터 선택을 위한 기본 알고리즘

센서 데이터 선택을 위한 기본 알고리즘은 센서 데이터의 저장, 압축, 복구를 위한 전체적인 흐름을 다루며, C 프로그램의 main(); 함수와 같은 역할을 한다. 이 알고리즘은 IEEE802.11 DCF 통신프로토콜과 인터럽트 기반으로 작성되었다.

송신 노드에서 수신 노드로 RTS 전송을 시작으로 통신이 시작되고, ACK를 수신하기 전까지의 일련의 조건에 의해 통신장애 판단, 복구 판단, 센서 데이터 복구 등의 절차를 수행하게 된다. 센서 데이터의 저장과 압축은 Save_Comp_Sensor_Data(); 함수에서 수행되며, 압축된 센서 데이터의 복구는 게이트웨이 노드와 모니터링 센터 사이의 서버에서 수행된다. 압축 데이터의 복구 시 센서 데이터 전송은 에너지 효율을 위해 통신장애가 발생한 노드 그룹의 자식 노드에서 통합(Aggregation) 후 부모 노드로 전송한다. 통신 링크가 끊어진 상위 노드는 통신장애가 발생한 노드 그룹의 부모 노드, 통신 링크가 끊어진 하위 노드는 통신장애가 발생한 노드 그룹의 자식 노드로 정의한다.

이 알고리즘에 사용되는 함수와 기호에 대한 설명은 Table 1, 알고리즘은 Fig. 3과 같다. 이 알고리즘은 인터럽트 기반의 프로그램으로 작성되어있기 때문에 Sleep Mode에서 메시지, 데이터 등의 수신 이벤트가 발생하면 Wake-Up 상태로 전환된다. 알고리즘에 대한 자세한 절차는 다음과 같다.

Step 1. 센싱 주기 설정 후 Sleep Mode로 진입한다.

Step 2. Step 3. ~ Step 6.을 계속 반복한다.

Step 3. 센서 데이터를 수신하면 Save_Comp_Sensor_Data(); 함수를 수행한다. 이 함수는 센서 데이터의 저장 및 압축을 담당하며, 압축 방법에 따른

Table 1 Function and Symbol Using Sensor Data Selection Algorithm

Function and Symbol	Explanation
Save_Comp_Sensor_Data(<i>S</i>);	Storing and Compression Function for Sensor Data <i>S</i>
T_Out();	Check CTS or ACK arrival after send RTS or DATA, respectively, during a given time limit and if the time limit is expired, then it returns 'true', otherwise it returns 'false'
Send(<i>S</i> , <i>Src</i> , <i>Dst</i>);	Function for Sensor Data Sending Source to Destination
Send(RTS)	Function for Sending
Recv(CTS)	Function for Receiving
<i>Rcv</i>	Variable for Receiving
Delete	Deleting Sensor Data

알고리즘은 3.2절에서 자세히 다룬다.

Step 4. 송신 노드는 수신 노드에 RTS를 전송하고, Time Out 시간 측정을 시작한다.

Step 5. 송신 노드는 수신 노드로부터 Time Out

동안 CTS를 수신했다면, 센서 데이터를 도착노드로 전송하고, ACK를 기다린다. 만약 CTS를 수신하지 못했다면, Sleep Mode로 진입한다.

Step 6. 송신 노드는 수신 노드로부터 ACK를 수신하였다면, 큐에 저장했던 센서 데이터를 삭제 후 주기를 기본 값으로 설정하고 Sleep Mode로 진입한다.

```

Algorithm Sensor Data Manager(){
Step 1. Set of Global Variables for Period;
Step 2. While(1){
    Rcv_Data ← False;

Step 3. IF Rcv_Data ← Recv(S) THEN
    Save_Comp_Sensor_Data(S);
    IF Queue == Empty THEN
        Sleep Mode;

Step 4. ELSE
        Send (RTS) and Start Time Out;
        Rcv_CTS ← False;
        Rcv_T_Out ← False;

Step 5. While (!(Rcv_CTS←Recv(CTS)) && !
        (Rcv_T_Out←T_Out()));
        IF Rcv_CTS ← Recv(CTS) THEN
        Send (S, Src, Dst);
        While (!(Rcv_ACK←Recv(ACK))&&!
        (Rcv_T_Out←T_Out()));

Step 6. IF Rcv_ACK ← Recv(ACK) THEN
        Delete Data in Queue;
        P ← default;
        ELSE
        Sleep Mode;
        ELSE
        Sleep Mode;
        ELSE
        Sleep Mode;
    }
}
    
```

Fig. 3 Basic Algorithm for Sensor Data Selection.

3.2 센서 데이터 선택 규칙

센서 데이터를 그래프로 표현하면 센서가 설치된 환경, 센서의 종류, 주위 환경, 예기치 않은 상황의 발생 등으로 각 센서 데이터는 다른 형태의 패턴으로 나타날 수 있다. 이러한 센서 데이터의 패턴은 빅 데이터 분석을 통해 앞으로 일어날 상황을 예측할 수 있을 뿐만 아니라 상황이 발생한 원인을 분석하는데 사용될 수 있다.

일반적인 자연 환경에서 온도 센서 데이터 값의 변화는 Fig. 4와 같이 시간에 일정한 증·감을 보이고,

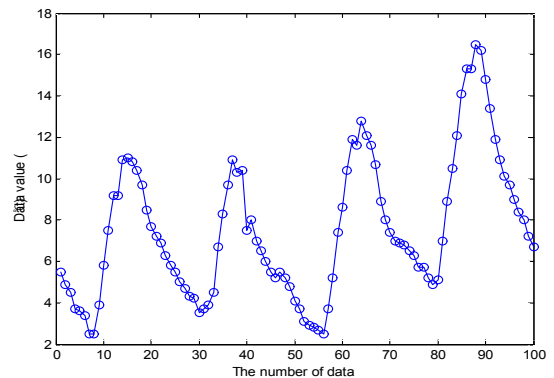


Fig. 4 Temperature Sensor Data.

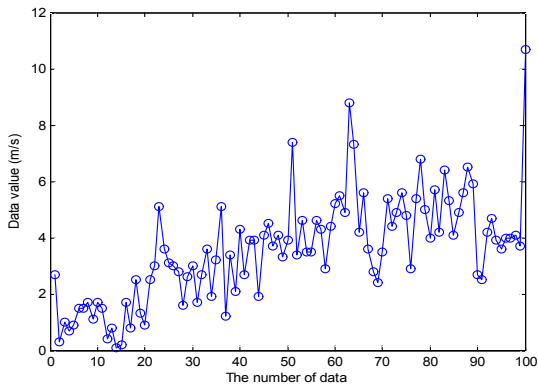


Fig. 5. Wind Speed Sensor Data.

풍속의 센서 데이터 값은 Fig. 5와 같이 온도의 센서 데이터 값에 비해 증·감이 변화하는 주기가 훨씬 짧다.

센서 데이터의 패턴은 각 센서마다 모두 다르지만 센서 데이터의 값이 변화하는 구간을 가지고 있다. 센서 데이터를 선택할 때 이러한 부분을 고려하지 않고 저장한다면, 압축으로 인해 손실된 센서 데이터 복구 시 센서 데이터의 오차가 발생하게 된다.

본 절에서는 센서 데이터 값이 변화하는 부분을 중심으로 최대/최소값의 센서 데이터를 선택, 압축하여 센서 데이터 복구 시 평균 누적 오차율을 감소시키기 위한 센서 데이터의 선택 방법을 제시한다.

센서 데이터 압축 시 저장을 위한 센서 데이터를 선택하기 위해서는 이전 구간에서 $2i-1$ 번째 센서 데이터가 선택되었는지를 먼저 고려한다. 만약 $2i-1$ 번째 센서 데이터가 선택되었다면 Fig. 6과 같이 크게 9가지 상황에 따라 센서 데이터를 선택한다. 각 상황 별 센서 데이터는 $i \geq 1$ 일 때, 각 압축구간별 $2i-1, 2i, 2i+1$ 을 비교한다(단, i 는 데이터의 순서를 나타낸다). 이 때, 첫 번째 데이터는 반드시 선택한다.

Case 1. IF $2i-1 > 2i > 2i+1$ THEN $2i+1$ 번째 데이터 선택

Case 2. IF $2i-1 < 2i < 2i+1$ THEN $2i+1$ 번째 데이터 선택

Case 3. IF $2i-1 = 2i = 2i+1$ THEN $2i+1$ 번째 데이터 선택

Case 4. IF $2i-1 > 2i < 2i+1$ THEN $2i$ 번째 데이터 선택

Case 5. IF $2i-1 < 2i > 2i+1$ THEN $2i$ 번째 데이

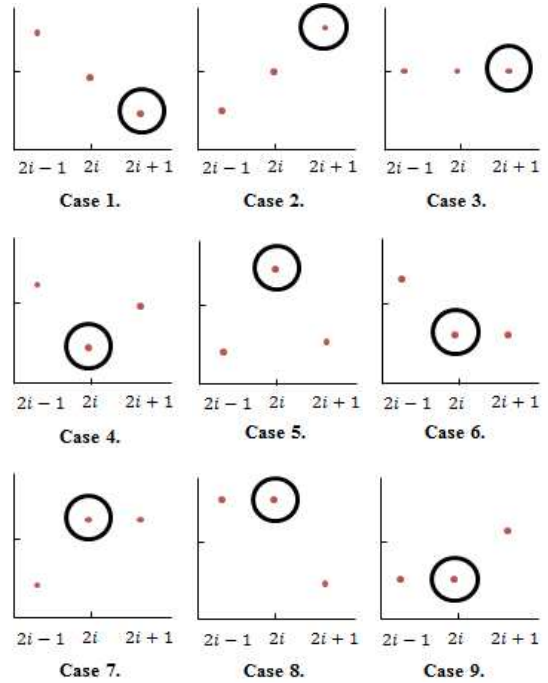


Fig. 6. Patterns of Sensor Data.

터 선택

Case 6. IF $2i-1 > 2i = 2i+1$ THEN $2i$ 번째 데이터 선택

Case 7. IF $2i-1 < 2i = 2i+1$ THEN $2i$ 번째 데이터 선택

Case 8. IF $2i-1 = 2i > 2i+1$ THEN $2i$ 번째 데이터 선택

Case 9. IF $2i-1 = 2i < 2i+1$ THEN $2i$ 번째 데이터 선택

만약, Fig. 7과 같이 압축구간 내에서 $2i-1$ 번째 센

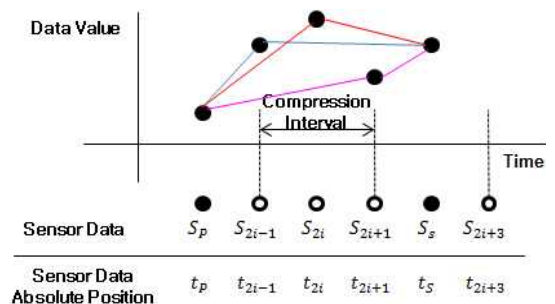


Fig. 7. Example for Sensor Data Selection when Not Selection ($2i-1$)th Data.

서 데이터가 선택되지 않았다면 가장 먼저, 다음 압축 구간에서 $2i-1$ 번째 센서 데이터가 선택되었을 경우의 센서 데이터 선택 규칙에 따라 센서 데이터를 선택한다. 다음 압축 구간의 센서 데이터 선택 후 이전 압축 구간에서 선택된 센서 데이터와 $2i-1, 2i, 2i+1$ 번째의 센서 데이터, 그리고 다음 압축 구간에서 선택된 센서 데이터와의 직선의 방정식을 구한 후 오차의 합이 가장 작은 센서 데이터를 선택한다.

$2i-1$ 번째의 센서 데이터가 선택되지 않은 경우 현재 압축구간에서 압축을 위한 센서 데이터를 선택하는 절차는 다음과 같다. 단, i 번째 센서 데이터는 S_i , 이전 압축구간에서 선택된 센서 데이터를 S_p , 다음 압축구간에서 선택된 센서 데이터를 S_s , 이전 압축구간에서 선택된 센서 데이터의 절대 위치를 t_p , 다음 압축구간에서 선택된 센서 데이터의 절대 위치를 t_s 라 한다.

Step 1. 다음 압축 구간에 해당하는 센서 데이터를 $2i-1$ 번째 센서 데이터가 선택된 경우에 기반을 두어 S_s 를 선택한다.

Step 2. $2i-1, 2i, 2i+1$ 번째 센서 데이터에 대해 각각 S_p 와 S_s 사이의 직선의 방정식을 구한다.

$$y_{2i-1} = \frac{S_s - S_{2i-1}}{t_s - t_{2i-1}}(t - t_{2i-1}) + S_{2i-1} \quad (1)$$

$$y_{2i(1)} = \frac{S_{2i} - S_p}{t_{2i} - t_p}(t_{2i-1} - t_p) + S_p \quad (2)$$

$$y_{2i(2)} = \frac{S_s - S_{2i}}{t_s - t_{2i}}(t_{2i+1} - t_{2i}) + S_{2i} \quad (3)$$

$$y_{2i+1} = \frac{S_{2i+1} - S_p}{t_{2i+1} - t_p}(t - t_p) + S_p \quad (4)$$

Step 3. 각 직선의 방정식 식(1)~(4)를 통해 실제 데이터와의 오차 E 를 계산한다.

$$E_{2i-1} = \sum_{k=2i}^{2i+1} \left| \frac{S_s - S_{2i-1}}{t_s - t_{2i-1}}(t_k - t_{2i-1}) + S_{2i-1} - S_k \right| \quad (5)$$

$$E(1)_{2i} = \left| \frac{S_{2i} - S_p}{t_{2i} - t_p}(t_{2i-1} - t_p) + S_p - S_{2i-1} \right| \quad (6)$$

$$E(2)_{2i} = \left| \frac{S_s - S_{2i}}{t_s - t_{2i}}(t_{2i+1} - t_{2i}) + S_{2i} - S_{2i+1} \right| \quad (7)$$

$$E_{2i} = E(1)_{2i} + E(2)_{2i} \quad (8)$$

$$E_{2i+1} = \sum_{k=2i-1}^{2i} \left| \frac{S_{2i+1} - S_p}{t_{2i+1} - t_p}(t_k - t_p) + S_p - S_k \right| \quad (9)$$

Step 4. 오차 E 를 나타내는 식(5)~(9) 중 합이 가장 작은 센서 데이터를 선택한다.

$$k = \underset{t \in \{2i-1, 2i, 2i+1\}}{\operatorname{argmin}} (E_t) \quad (10)$$

센서 데이터를 복구 할 때, 센서 데이터의 변화하는 주기의 수 보다 저장한 센서 데이터가 적으면 큰 오차가 발생한다. 이 경우 센서 데이터의 샘플링 주기를 높이면 해결할 수 있지만, 통신장애 발생 시 메모리 부족현상이 더욱 가속화될 수 있는 단점이 있다. 따라서 적절한 샘플링 주기를 설정해야 한다.

3.3 센서 데이터 선택정보 관리

2MC 압축 방법에서는 메모리가 가득 차게 되면 메모리 압축을 위해 저장되어 있는 데이터의 순서일 때 $2i-1, 2i, 2i+1$ 번째의 데이터 중 3.2절의 센서 데이터 선택 규칙에 따라 센서 데이터를 선택한 후 원형 큐에 재배치한다. 만약, 압축 이후 각 데이터에 대해 압축하기 이전의 원본 데이터의 정확한 위치를 알 수 없으면 센서 데이터를 복구하기 위해 보간법을 적용할 수 없다. 그러므로 압축 후 각 센서 데이터에 대해 원본 데이터에서의 정확한 위치를 구하는 것이 필요하다.

원본 데이터의 정확한 위치를 구하기 위해 2MC 알고리즘에서는 압축 횟수 c 에 따라 $c+1$ 개의 비트를 이용하여 센서 데이터 저장순서에 대한 압축 구간별 기준 센서 데이터(압축 구간의 첫 번째 센서 데이터)로부터 상대 위치 정보를 관리할 필요가 있다.

Fig. 8은 압축구간에 대해 데이터 중 선택된 데이터에 대한 위치 비트열을 만드는 방법의 예를 소개한다. Fig. 8에서 ●과 ○는 데이터를 나타내고, ●는 압축 진행 시 선택된 데이터, ○는 선택되지 않은 데이터를 나타낸다. 구간에 대한 선택 비트는 각 압축구간 내에서 선택된 데이터의 위치를 나타내는 정보를 나타내고, 전체 데이터 비트열은 원형 큐에 저장된 센서 데이터들이 각 압축구간 내에서 몇 번째 위치에 존재하는지에 대한 정보를 나타낸다.

압축이 진행 될 때마다 압축구간(Compression Interval) 내에서 압축을 위한 센서 데이터를 하나 선택하고 선택된 데이터의 번호에 대한 비트를 따로

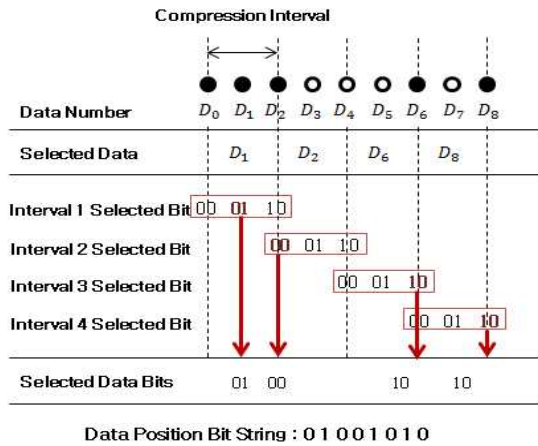


Fig. 8. Example of Made Data Position Bit String.

저장한다. 예를 들어, Fig. 8에서는 압축을 위해 첫 번째 구간에서 D_1 , 두 번째 구간에서 D_2 , 세 번째 구간에서 D_6 , 네 번째 구간에서 D_8 을 선택한다면 전체 데이터에 대한 비트열을 01001010으로 저장할 수 있다. 이때 첫 구간에서의 첫 번째 데이터는 반드시 선택되기 때문에 첫 번째 압축 구간에서는 첫 번째 데이터를 선택하더라도 구간 내에 다른 데이터를 선택하여 저장하고, 첫 번째 데이터에 대한 비트는 따로 관리하지 않는다.

센서 데이터 저장순서에 대한 정보를 관리하는 절차는 다음과 같다.

Step 1. 데이터를 선택하기 위한 압축구간 (Compression Interval)을 구한다.

Step 2. 첫 번째 압축구간에서부터 센서 데이터 선택 기준에 따라 센서 데이터를 하나씩 선택한다. 이때, 센서 데이터 선택에 따라 위치에 따른 비트를 저장한다. 이때, 압축 횟수에 따른 비트를 새롭게 계산한다.

3.4 2MC (Maximum/Minimum Compression) 알고리즘

센서 데이터의 경우 데이터 값에 대해 그래프로 표현하면 데이터의 특성에 따라 다양한 형태로 나타낼 수 있다. 하지만 공통적으로 센서 데이터 값은 증가→감소, 감소→증가, 증가→증가, 감소→감소로 변화하는 부분이 존재한다. 센서 데이터 압축 시 이러한 데이터를 고려하여 저장한다면 통신이 복구된

후 센서 데이터를 복구할 때 오차를 줄일 수 있다.

이러한 아이디어에 기반을 두어 2MC 방법에서는 압축 구간 내에서의 최대/최소값 중심으로 데이터를 저장하여 센서 데이터 복구 시 오차를 줄일 수 있도록 한다. 또한 CQP 방법과 마찬가지로 센서 데이터 압축 시 첫 번째 센서 데이터는 반드시 남겨두어야 하고, 센서 데이터 압축 횟수 c 에 따라 센싱 주기는 2^c 로 증가한다.

2MC 방법은 기본적으로 CQP 방법을 기반의 압축 방법이지만 몇 가지 다른 점이 존재한다. CQP 방법은 압축 횟수에 따라 홀수 번째의 센서 데이터를 선택하여 압축하지만, 2MC 방법은 압축 구간 내에서 최대/최소값 중심으로 압축할 센서 데이터를 선택하여 압축한다. 센서 데이터는 $2i-1, 2i, 2i+1$ (단, i 는 압축 구간 내의 센서 데이터의 저장 위치, $i \geq 1$)를 비교하여 선택한다. CQP와 2MC 방법에 대한 비교는 Table 2와 같다.

2MC 방법을 수행하기 위한 알고리즘을 설명하기 전에 알고리즘에 사용되는 기호를 Table 3에서 설명한다.

2MC 알고리즘 Fig. 9의 절차는 다음과 같다.

Step 1. Save_Comp_Sensor_Data(S); 로부터 넘겨받은 데이터 S 를 $rear$ 가 가리키는 위치에 저장한다.

Table 2 Comparisons of CQP and 2MC

	CQP	2MC
Compression Period	2^c (c is number of compression)	
Selecting Data	Odd's Sensor Data	cf. 3.2

Table 3 Symbol Using 2MC Algorithm

Symbol	의 미
S_i	Received Sensor Data
$front$	Position for Deleting Sensor Data
$rear$	Position for Inserting Sensor Data
s_{rear}	Space for Storing Sensor Data
s_i	In Circular Queue for Storing i th Space
$Capacity$	Circular Queue Size
i	Position of Sensor Data after c th compression
t_i	Absolute Position of i th Sensor Data

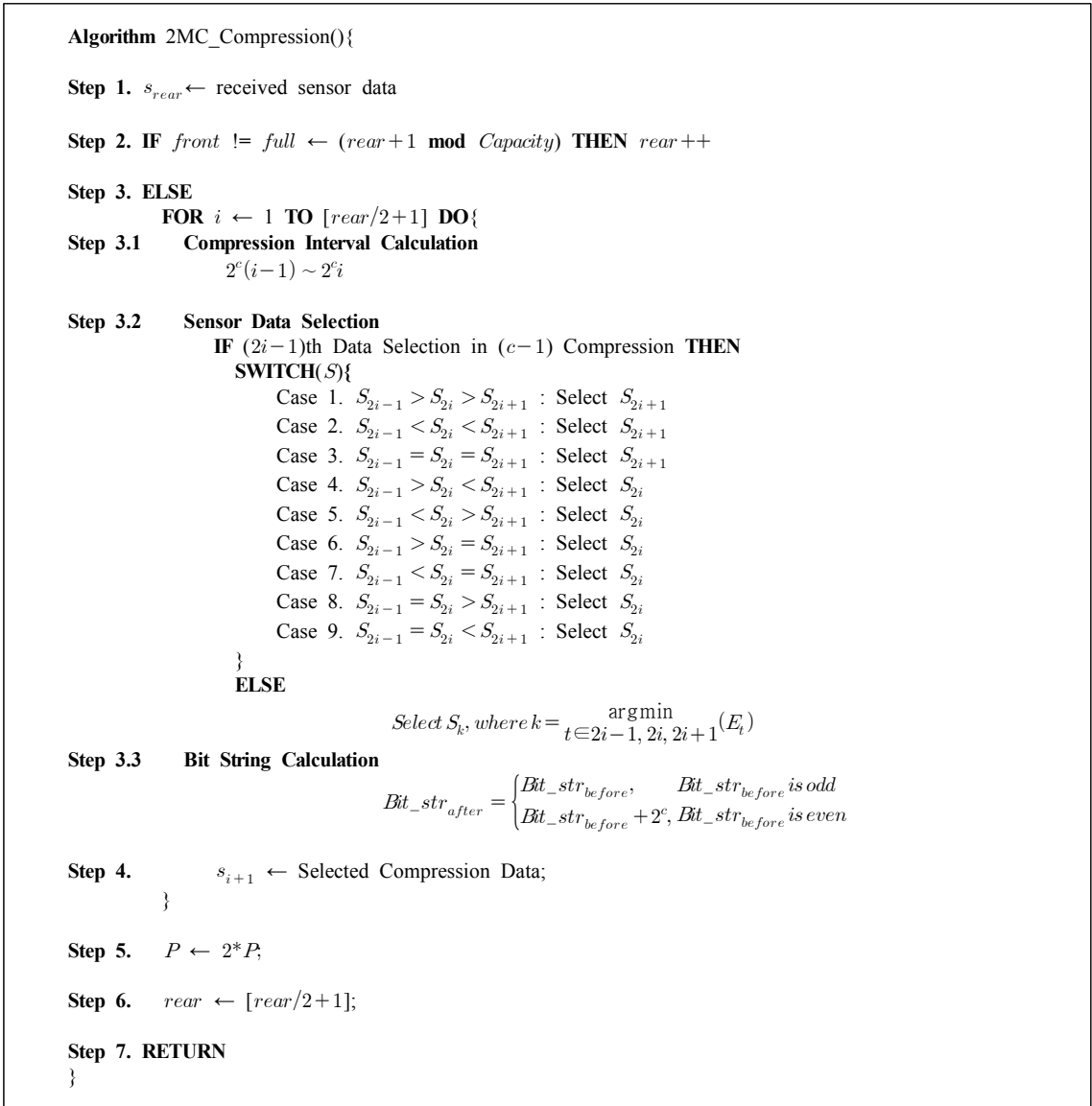


Fig. 9. 2MC Algorithm.

Step 2. 원형 큐가 가득 차지 않은 경우 $rear$ 를 증가시킨 후 **Step 6.**을 수행한다.

Step 3. 압축 구간 내에 존재하는 센서 데이터 중 센서 데이터 선택 규칙에 따라 압축을 위한 센서 데이터를 선택한 후 비트열을 저장한다(단, c 는 압축 횟수, i 는 압축 횟수에 따른 센서 데이터의 위치).

Step 4. 선택된 센서 데이터를 원형 큐의 두 번째 공간부터 차례대로 재배치한다.

Step 5. 현재 센싱 주기 P 를 2배로 증가시킨다.

Step 6. $rear$ 의 위치를 새롭게 지정한다.

Step 7. RETURN 한다.

3.5 통신장애 상황 전파 및 센서 데이터 복구 절차

통신장애 상황이 복구되면 압축된 센서 데이터를 복구해야 한다. 통신장애가 복구된 것은 통신장애가 발생한 링크의 부모 노드와 자식 노드간의 IEEE802.11 DCF이 정상적으로 수행되면 통신이 복구된 것으로 판단한다. 이때 링크가 끊어진 후손 노드들은 통

신장애가 복구된 것을 알 수 없기 때문에 메시지를 통해 상황을 알릴 필요가 있다. 마찬가지로 통신장애가 발생한 경우에도 메시지 전송을 통해 상황을 알려야 한다.

예를 들어, 통신장애가 발생하면 ‘Communication Failure Message’ 전송을 통해 이 메시지를 수신한 노드는 계속해서 단말노드까지 메시지를 전송하고 각 노드에서 센서 데이터의 압축 및 저장을 진행할 수 있으며, 통신장애가 발생한 그룹의 부모 노드와 자식 노드 사이에 통신이 복구되면 통신장애 그룹의 자식 노드는 후손 노드에게 통신장애 전과 방법과 마찬가지로 ‘Recovery Message’ 전송을 통해 통신장애 복구 상황을 전파할 수 있다. 이때, 각 노드에서 저장 및 압축하고 있던 센서 데이터를 최종적으로 모니터링 센터로 전송하게 되는데 센서 데이터의 복구는 모니터링 센터로 전송되기 직전의 서버에서 이루어지게 하여, 노드의 에너지 소모를 줄일 수 있도록 한다.

압축으로 인해 손실되었던 센서 데이터의 복구 절차는 다음과 같이 수행할 수 있다. 첫째, 서버는 각 노드로부터 수신한 압축 데이터의 압축 횟수를 식별한다. 압축 횟수 식별을 통해 위치 비트열의 정보를 파악할 수 있다. 둘째, 센서 데이터의 절대 위치를 계산한다. 압축 횟수 식별을 통해 수집한 비트의 수를 통해 실제 센서 데이터의 절대 위치를 계산한다. 셋째, 보간법을 이용해 센서 데이터를 복구한다. 절대 위치 계산 후 보간법을 사용하여 센서 압축으로 인해 손실된 센서 데이터를 복구한다.

4. 실험결과 및 분석

본 장에서는 2MC 알고리즘을 사용하여 센서 데이터 압축 후 복구하였을 때, 복구된 센서 데이터의 평균 누적 오차율에 대한 성능을 평가한다. 4.1절에서는 2MC 알고리즘의 성능을 평가하기 위한 환경, 4.2절에서는 성능평가 결과에 대해 서술한다.

4.1 평가환경

2MC 알고리즘을 사용하여 센서 데이터를 압축하여 복구하였을 때 원본 데이터와의 오차에 대한 성능을 평가하기 위해 가장 먼저 기존에 연구되었던 CQP 알고리즘과 2MC 알고리즘을 이용하여 센서 데이터를 압축한다. 센서 데이터의 압축 횟수는 4회까지 증가시키고, 각 압축 횟수에 따라 압축된 센서 데이터를 이용하여 압축으로 인해 손실된 센서 데이터를 복구한다. 복구는 보간법을 이용하여 진행되며, 사용된 보간법의 종류는 Linear와 Spline이다.

성능을 평가하기 위해 사용된 데이터 종류는 Table 4와 같이 기상청에서 제공받은 2014년 4월 서울특별시 온도 데이터와 상대습도 데이터, 실제 연근해 수산양식을 위한 센서 네트워크를 설치하여 수집한 2015년 1월 경상남도 통영시 육지도의 해수의 온도와 pH 데이터를 사용하였다. 센서 데이터 수집을 위해 설치한 센서 네트워크의 구성은 Table 5와 같이 게이트웨이 노드 1개, 릴레이 노드 2개, 복합센서 2개를 사용하였다. 게이트웨이 노드와 통합모니터링 센터는 CDMA (Code Division Multiple Access) 통신을 사용하고, 센서 노드간 및 센서 노드와 게이

Table 4 Environment of Performance Evaluation

Type of Data	04. 2014 Temperature, Relative Humidity 01. 2015 Underwater Temperature, Underwater pH
Number of Data	400 (100, respectively)
Using Tool	MATLAB R2014a
Using Interpolation	Linear, Spline

Table 5 Consists of Sensor Networks

G/W MCU	Cortex-M4 1MB Flash Memory
S/N MCU	Cortex-M3 512KB Flash Memory
Kinds of Sensor	Underwater pH, Underwater Temp., Underwater Turbidity
Communication Modules	CDMA, Zigbee, RS-485

트웨이 노드는 Zigbee 모듈, 센서 노드와 센서는 RS-485 모듈을 사용하였다. 이 성능 평가를 위한 실험은 MATLAB R2014a에서 수행하였다.

성능을 평가하기 위한 절차는 첫째, 압축의 횟수를 1회부터 4회까지 증가시키면서 2MC와 CQP알고리즘에 따라 압축할 센서 데이터를 선택한다. 둘째, 압축 횟수에 따라 선택된 센서 데이터들을 이용하여 센서 데이터의 절대 위치를 구한다. 셋째, Linear와 Spline보간법을 이용하여 압축으로 인해 손실된 센서 데이터를 복구한다. 넷째, 원본 데이터와 보간법을 사용하여 복구된 센서 데이터의 평균 누적 오차율을 비교한다. 이 절차는 해수온도, 상대습도, 해수 pH, 온도에 대해 각각 수행한다.

4.2 평가결과

본 절에서는 4.1절의 성능 평가 수행 절차에 따라

얻은 결과를 분석한다. 성능 평가 결과를 분석하기 위해 CQP와 2MC 압축 방법을 사용하여 진행 된 압축으로 인해 손실된 센서 데이터를 복구한 센서 데이터와 원본 데이터를 비교한다. 이때, 그래프를 통해 복구된 센서 데이터가 원본 데이터와 일치하는지 확인한다. 복구된 센서 데이터와 원본 데이터는 1회 압축부터 4회 압축까지 비교하여 평균 누적 오차율을 계산하고, 각각에 대해 평균 누적 오차율을 4회 압축한 후 센서 데이터를 복구한 결과를 Fig. 10부터 Fig. 17까지 나타내었다. 각 그림의 센서 데이터는 순서대로 해수온도, 상대습도, 해수 pH, 대기온도 이다.

압축 횟수가 2회까지 증가한 경우 원본 데이터와 복구된 센서 데이터는 CQP방법과 2MC방법 모두 거의 일치하는 것을 볼 수 있다. 하지만 압축 횟수가 3회로 증가하는 경우 CQP 방법으로 압축한 경우 실제 데이터와 큰 오차가 발생하였으며, 2MC 방법으

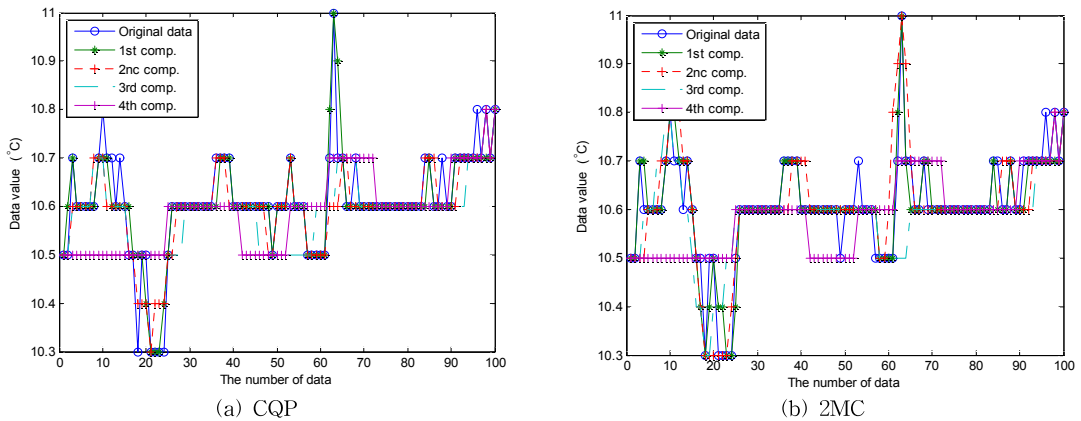


Fig. 10. Accuracy after Linear Interpolation in Case of Underwater Temp. (a) CQP, (b) 2MC Method.

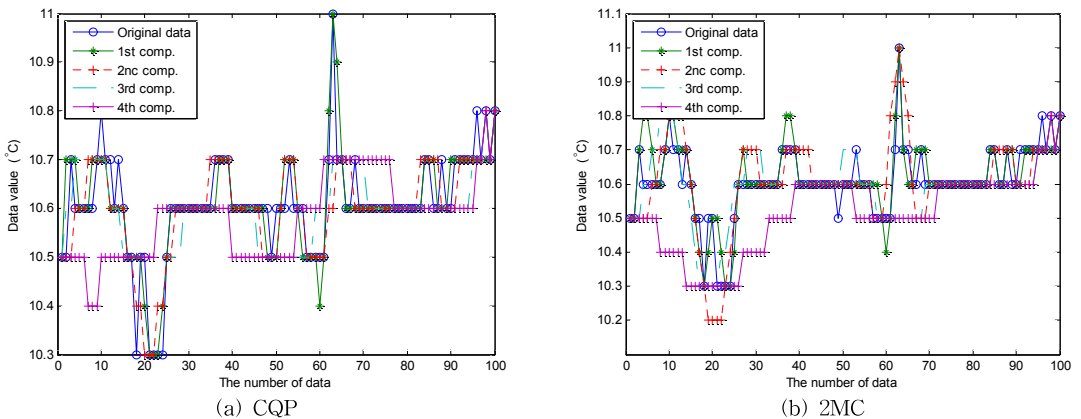
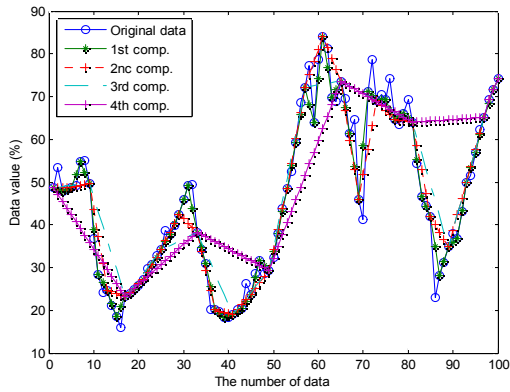
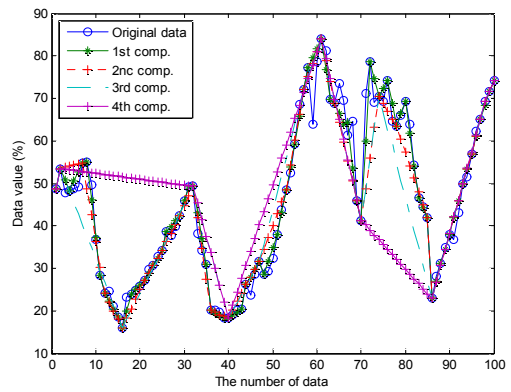


Fig. 11. Accuracy after Spline Interpolation in Case of Underwater Temp. (a) CQP, (b) 2MC Method.

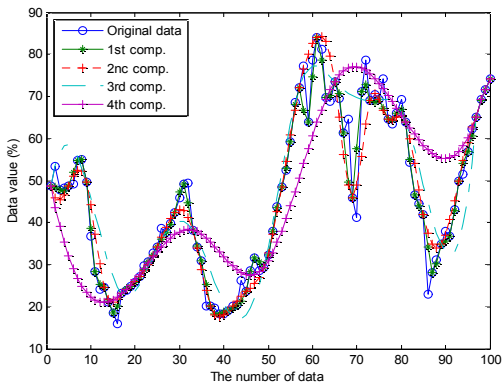


(a) CQP

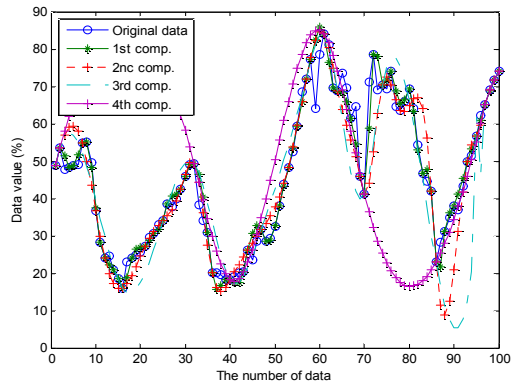


(b) 2MC

Fig. 12 Accuracy after Linear Interpolation in Case of Relative Humidity. (a) CQP, (b) 2MC Method.

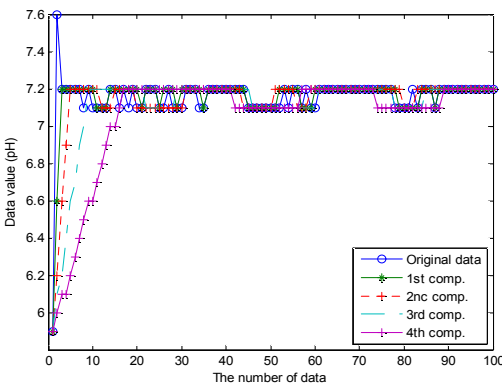


(a) CQP

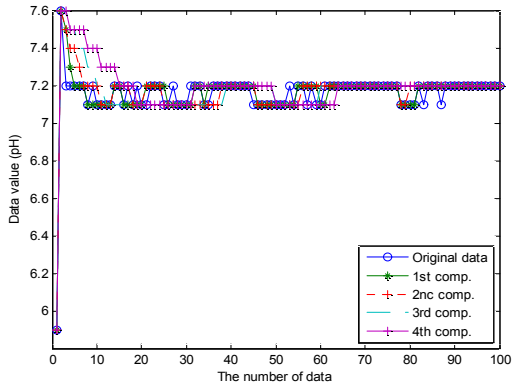


(b) 2MC

Fig. 13 Accuracy after Spline Interpolation in Case of Relative Humidity. (a) CQP, (b) 2MC Method.

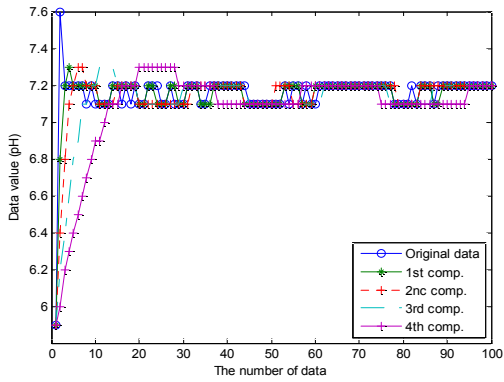


(a) CQP

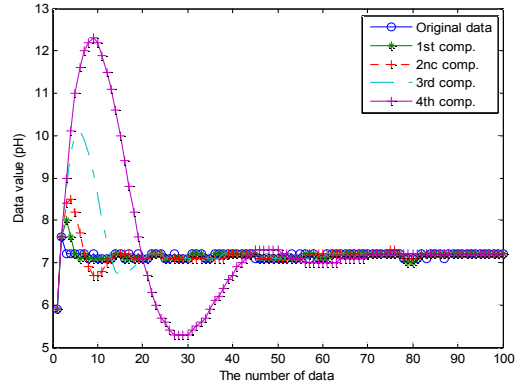


(b) 2MC

Fig. 14 Accuracy after Linear Interpolation in Case of Underwater pH. (a) CQP, (b) 2MC Method.

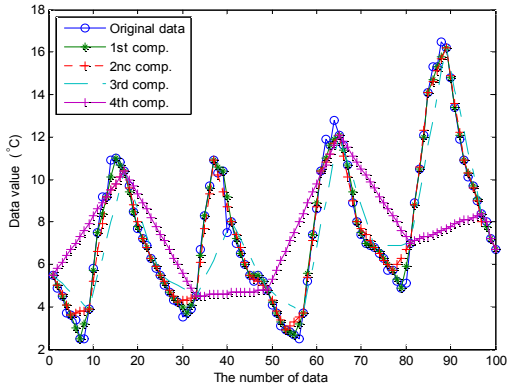


(a) CQP

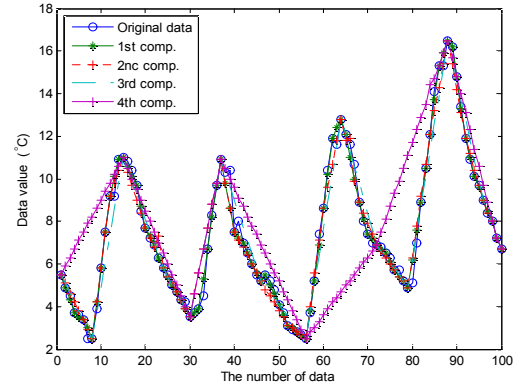


(b) 2MC

Fig. 15 Accuracy after Spline Interpolation in Case of Underwater pH. (a) CQP, (b) 2MC Method.

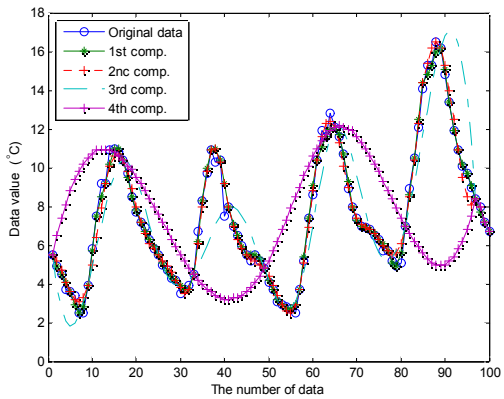


(a) CQP

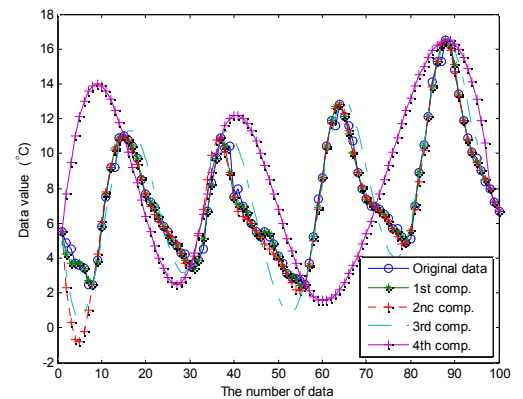


(b) 2MC

Fig. 16 Accuracy after Linear Interpolation in Case of Temp. (a) CQP, (b) 2MC Method.



(a) CQP



(b) 2MC

Fig. 17 Accuracy after Spline Interpolation in Case of Temp. (a) CQP, (b) 2MC Method.

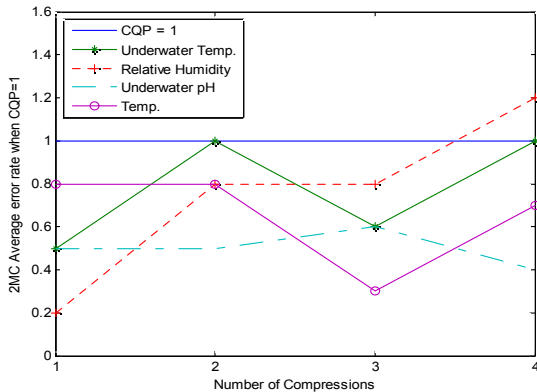


Fig. 18. 2MC Average Error Rate when CQP=1.

로 압축한 경우 Spline보간법은 오차가 크게 발생하였지만 Linear 보간법은 실제 데이터와 거의 일치했다. 실제 오차의 경우 원본 데이터와 복구된 센서 데이터 사이에 발생하는 면적만큼 오차가 발생한다. 대체로 Linear와 Spline 보간법 중 Linear 보간법으로 센서 데이터를 복구하였을 때 성능이 더 좋았다.

CQP 방법 대비 2MC 방법의 압축 횟수에 따른 평균 누적 오차율은 Fig. 18과 같다. Fig. 18은 CQP 방법의 평균 누적 오차율을 1로 보았을 때의 2MC 방법의 평균 누적 오차율을 나타낸다. 보간법은 대체로 Linear 보간법 사용 시 성능이 더 좋기 때문에 Linear 보간법을 사용한 경우만 표기하였으며, 각 센서 데이터에 대한 CQP와 2MC 방법의 평균 누적 오차율은 Table 6, Table 7과 같다.

압축 횟수 1회에서 각 센서 데이터의 평균 누적 오차율은 0.1-2.9%로 실제 데이터와 거의 차이가 없었으며, 압축 횟수가 2회로 증가해도 2MC 방법으로 압축한 경우 Spline 보간법을 제외한 나머지 평균 누적 오차율은 0.3-6.3%로 실제 데이터와 큰 차이가 없었다. 하지만 압축 횟수가 3회로 증가하면 CQP 알고리즘을 사용한 압축 방법은 Linear, Spline방법 모

Table 6. CQP: Linear Interpolation Result(Average Error Rate)

Data Set \ No.	1(2x)	2(4x)	3(8x)	4(16x)
Underwater Temp.	0.2	0.3	0.6	0.8
Relative Humidity	3.6	7.6	13.9	27.1
Underwater pH	0.4	0.7	1.1	1.9
Temp.	2.6	6.1	17.2	42.7

Table 7. 2MC: Linear Interpolation Result(Average Error Rate)

Data Set \ No.	1(2x)	2(4x)	3(8x)	4(16x)
Underwater Temp.	0.1	0.3	0.4	0.8
Relative Humidity	2.9	6.3	11.8	34.9
Underwater pH	0.2	0.4	0.7	0.9
Temp.	2.1	5.2	6.7	31.8

두 평균 누적 오차율이 최대 32%로 크게 증가하였다. 하지만 2MC 방법으로 압축한 경우 Linear 보간법으로 센서 데이터 복구 시 평균 누적 오차율이 0.4-11.8%로 3회 압축(16개의 센서 데이터 중 하나를 선택하여 압축)까지 유효성을 보였다.

상대습도의 경우 압축구간 내에서 센서 데이터를 선택하는 양에 비해 데이터 값이 변화하는 주기가 더 많기 때문에 다른 센서 데이터에 비해 오차가 조금 더 컸다.

CQP 알고리즘을 이용한 압축 방법은 압축 횟수가 증가할수록 실제 분석할 수 있는 데이터가 줄어들어 구간 별 최대/최소값을 찾지 못하거나, 압축구간 내에서 데이터 크기가 크게 변하는 경우 오차가 굉장히 크게 발생하는 요인이 된다. 하지만 2MC 방법의 경우 압축구간 내에서 최대/최소값 중심으로 데이터를 복구하기 때문에 8개의 센서 데이터 중 하나를 선택하는 3회 압축까지 압축이 유효하다.

4회 압축의 경우 CQP 방법과 2MC 방법 모두 센서 데이터 값의 차이가 작은 경우를 제외하고 약 30% 이상의 큰 평균 누적 오차율을 보였지만, 2MC방법의 경우 CQP 방법에 비해 평균 누적 오차율이 Linear 보간법과 Spline 보간법에 대해 약 30% 이상이 각각 감소하였다.

5. 결 론

본 논문에서는 압축구간 내에서 센서 데이터의 최대/최소값 중심의 압축방법인 2MC 압축 방법을 제안하고, 그 성능을 평가했다. 센서 데이터는 특징에 따라 그래프로 표현하면 여러 패턴으로 나타나며 센서 데이터를 선택하는데 있어 오차가 크게 나타나는 구간인 최대/최소값 중심으로 압축하는 2MC 압축 방법이 CQP 압축 방법보다 3회 압축에서 성능이 최대 35% 더 좋았다.

실제 MATLAB을 통한 실험을 통해 센서 데이터의 압축 후 센서 데이터를 복구하여 원본 데이터와 비교한 결과 압축 횟수가 3회까지 증가하여도 2MC 압축 방법은 평균 누적 오차율이 3회 압축까지 유효했다. 상대습도의 경우 3회 압축에서 약 11%의 평균 누적 오차율을 보였지만, 다른 센서 데이터에 비해 약간 높은 정도이며 실제 사용할 수 있을 정도로 유효함을 보였다. CQP 압축 방법은 압축 횟수가 1~2회인 경우에만 원본 데이터와 유사하게 나타났다. 4회 압축의 경우 CQP와 2MC 방법 모두 30%이상의 평균 누적 오차율의 큰 오차가 발생하였지만 2MC 방법의 경우 CQP에 비해 Linear 보간법과 Spline 보간법으로 센서 데이터를 복구한 경우 각각 약 30% 이상의 평균 누적 오차율이 더 적게 나타났다.

CQP 방법은 압축 횟수가 증가하게 되면 저장된 센서 데이터의 수는 줄어들기 때문에 센서 데이터 패턴의 극대/극소 값을 찾지 못하여 평균 누적 오차율이 커지지만, 2MC 방법은 압축구간 내에서 최대/최소값을 선택하여 오차 값이 커지는 극대/극소 값을 선택하여 평균 누적 오차율이 훨씬 작아진다. 실험결과, 8개 중 하나의 센서 데이터를 선택하여 데이터를 복구 하더라도 실제에 가까운 값을 얻을 수 있었다.

멀티미디어 압축기법의 DPCM 압축 방법을 적용할 경우 압축 횟수에 따라 차이 값을 저장하는 비트의 수를 절반으로 사용한다면 압축 횟수의 증가에 따라 압축률이 4배씩 증가하게 된다. 실제 최근 출시되는 MCU의 플래시 메모리의 크기가 512KB이고, 단순히 데이터를 저장할 수 있는 크기가 300KB, 센서 데이터의 크기가 1Byte일 때, 약 30만개 이상의 센서 데이터가 저장 가능하며, 10초마다 저장 시 34.7일 저장이 가능한 용량이다. 여기에 DPCM 압축 방법을 적용한다면 약 69.4일간 센서 데이터를 저장할 수 있다.

향후 연구로 압축 횟수가 많아져도 오차율을 줄이고, 2MC 방법이 유효하기 위한 데이터의 수를 찾는 방법에 대한 연구를 진행할 필요가 있다. 이 외에도 센서 데이터의 각 패턴에 따른 오차 허용 범위를 적용한 센서 데이터 압축 기법에 관한 연구를 진행할 필요가 있다.

REFERENCE

- [1] H.W. Nam, S.S. Shin, C.H. Kim, S.H. Park, "Remote Monitoring System Based on Ocean Sensor Networks for Offshore Aquaculture," *Oceans-St. John's*, pp. 14-19, 2014.
- [2] E. Borgia, "The Internet of Things Vision: Key Features, Applications and Open Issues," *Computer Communications*, Vol. 54, pp. 1-31, 2014.
- [3] A. Bonastre, J.V. Capella, R. Ors, "In-Line Monitoring of Chemical-Analysis Processes Using Wireless Sensor Networks," *Trends in Analytical Chemistry*, Vol. 34, pp. 111-125, 2012.
- [4] Andrei Maciuca, Mircea Strutu, Dan Popescu, Grigore Stamatescu, "Cell-based Sensor Network for Complex Monitoring at Home of Patients with Chronic Diseases," *Electrical and Electronics Engineering*, pp. 11-13, 2013.
- [5] D.K. Lee and D.J. Choi, "Implementation of Zigbee-based Publish/Subscribe System for M2M/IoT Services.", *Journal of Korea Multimedia Society*, Vol. 17, No. 12, pp.1461-1462, 2014.
- [6] J.G. Lee and Y.J. Song, *Ubiquitous Sensor Network*, Hansan, Seoul, 2009.
- [7] Huseyin Ugur Yildiz, Kemal Bicakci, Bulent Tavli, Hakan Gultekin, Davut Incebacak, "Maximizing Wireless Sensor Network Lifetime by Communication/Computation Energy Optimization of Non-Repudiation Security Service: Node Level Versus Network Level Strategies," *Ad Hoc Networks*, Vol. 37, pp. 301-323, 2016.
- [8] C.H. Kim, *Technology Development for USN-based Energy Management, Marine Sensors, Sensor Nodes and Middleware for Efficiencies and Enhancement of Marine Industry*, Ministry of Science, ICT and Future Planning, Se-jong, 2015.
- [9] S.J. Park, S.H. Park, S.K. Kim, C.H. Kim, "Underwater Communications and Underwater

Sensor Network Technology,” *Communications of the Korean Institute of Information Scientists and Engineers*, Vol. 28, No. 7, pp. 79-88, 2010.

[10] J.E. Kim, N.Y. Yun, Y.P. Kim, S.Y. Shin, S.H. Park, J.H. Jeon, et al., “Design and Performance Evaluation of Hierarchical Protocol for Underwater Acoustic Sensor Networks,” *The Korea Society for Simulation*, Vol. 20, No. 4, pp. 157-166, 2011.

[11] Sodolfo W.L. Coutinho, Azzedine Boukerche, Luiz F.M. Vieira, Antonio A.F.Loureiro, “A Novel Void Node Recovery Paradigm for Long-term Underwater Sensor Networks,” *Ad Hoc Networks*, Vol. 34, pp. 144-156, 2015.

[12] M.H. Kim, Y.H. Lee, Y.D. Jeon, *Multimedia System*, Hongreung Science, Seoul, 2006.

[13] K.S. Choi, “Bit Plane Modification for Improving MSE-near Optimal DPCM-based Block Truncation Coding,” *Digital Signal Processing*, Vol. 23, Issue 4, pp. 1171-1180, 2013.

[14] Yun Zhao, Xiaoming Li, Lingxu An, Jian Sun, “Research on Encoding/Decoding Method of Electric Physical Information Based on LMS-ADPCM Algorithm,” *Advanced Power System Automation and Protection*, pp. 795-800, 2011.

[16] U.S. Uk and S.H. Kim, “Data Reconstruction Scheme Using PCA in Sensor Network Environment,” *Institute of Control, Robotics and Systems Conference*, pp. 20-24, 2007.

[16] D.H. Shin and C.H. Kim, “A Method for Storing and Recovering Sensing Data Using Queue in Wireless Sensor Network Communication Failures,” *The 2014 Fall Conference of the KIPS*, pp. 207-210, 2014.

[17] B.A. Forouzan, *Data Communications and Networking*, Fifth Edition, Mc Graw Hill, Singapore, 2010.



신 동 현

2014년 강릉원주대학교 컴퓨터공학 공학사
 2014년~현재 강릉원주대학교 컴퓨터공학 석사과정
 관심분야: IoT (Internet of Things), WSN (Wireless Sensor Network), Data Mining



김 창 화

1985년 고려대학교 수학교육과 이학사
 1987년 고려대학교 전산학전공 이학석사
 1990년 고려대학교 전산학전공 이학박사

1994년~1995년 University of Toronto, Enterprise Integration Lab. Post-Doc. & Visiting Professor
 2002년~2004년 미국 Texas A&M대학 Visiting Scholar
 1989년~현재 강릉원주대학교 컴퓨터공학과 교수
 2005년~현재 강릉원주대학교 해양센서네트워크시스템 기술연구센터 센터장
 관심분야: Internet of Things, Distributed System, Intelligent System, Underwater Communication and Sensor Network