

지리정보 데이터 압축률 향상을 위한 Run-Length/Byte-Packing 압축 알고리즘 설계 및 구현

윤석환¹ · 양승수² · 박석천^{3*}

Design and Implementation of Run-Length/Byte-Packing Compression Algorithm to Improve Compressibility of Geographic Information Data

Seok-Hwan Yoon¹ · Seung-Su Yang² · Seok-Cheon Park^{3*}

¹Department of Computer Sciences, Semyung University, Chungbuk 27136, Korea

²Department of IT Convergence Engineering, Gachon University, Gyeonggi 13120, Korea

^{3*}Department of Computer Engineering, Gachon University, Gyeonggi 13120, Korea

요 약

최근 압축 알고리즘이 지리정보 데이터를 압축하기 위한 방법으로 가장 많이 사용되고 있다. 그러나 이와 같은 압축 알고리즘은 지리정보 데이터 압축에 실제 적용하기에는 지도 데이터의 연속성이 미흡하고, 단일 데이터로 압축할 수 없기 때문에 압축률이 저하된다는 문제점이 있다. 따라서 본 논문에서는 이러한 문제점을 개선하기 위해 압축 알고리즘들의 장점을 취합해 지리정보 데이터 압축을 가능하게 하고, 압축 및 복원 속도를 향상시킨 Run-Length/Byte-Packing 압축 알고리즘을 설계 및 구현하였다. 구현한 알고리즘을 평가한 결과 기존 압축 알고리즘에 비해 제안 알고리즘이 평균 약 5% 향상된 것을 확인하였으며, 압축률과 복원 속도가 향상되었다는 것을 확인하였다.

ABSTRACT

Recently, a compression algorithm is widely used as a method for compressing geographic information data. However, such a compression algorithm has a problem that the continuity of the map data is insufficient to actually apply it to the geographic information data compression, and the compression rate is lowered because it can not be compressed into a single data. Therefore, in this paper, we have designed and implemented a Run-Length / Byte-Packing compression algorithm that enables compression of geographic information by combining the advantages of compression algorithms and improves compression and restoration speed. As a result of evaluating the implemented algorithm, it was confirmed that the proposed algorithm improved about 5% on average compared with the existing compression algorithm, and it was confirmed that the compression rate and the restoration speed were improved.

키워드 : 지리정보, 압축 알고리즘, Run-Length, Byte-Packing, 융합 알고리즘

Key word : Geographic Information, Compression Algorithm, Run-Length, Byte-Packing, Fusion Algorithm

Received 30 August 2017, Revised 08 September 2017, Accepted 19 September 2017

* **Corresponding Author** Seok-Cheon Park(E-mail: scpark@gachon.ac.kr, Tel:+82-31-750-5747),
Department of Computer Engineering, Gachon University, Gyeonggi 13120, Korea

Open Access <https://doi.org/10.6109/jkiice.2017.21.10.1935>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

최근 스마트폰을 비롯한 스마트 디바이스들의 발전과 더불어 지리정보를 이용한 시스템이 일상생활에도 도입되어 다양하게 사용되고 있다[1,2].

지리정보 시스템이란 기존 아날로그 방식의 지도 데이터를 컴퓨터로 수집·분석·가공하여 지형과 관련된 모든 분야에서 사용하기 위해 만들어진 종합정보 시스템이다. 실제로 지리정보 시스템은 교통정보, 도면 자동화, 도시정보 등 다양한 곳에서 활용되고 있으며 다양한 지형정보를 데이터베이스화하여 통합 및 관리하고 있다[3,4]. 이러한 지리정보 시스템은 지리정보 데이터를 기반으로 운영되고 있으며 요구하는 데이터의 정밀도가 점차 증가하기 때문에 지리정보 데이터의 용량이 증가하는 문제점이 있다. 이를 해결하기 위해 지리정보 데이터를 압축시키는 압축 알고리즘에 대한 연구가 활발히 이루어지고 있으며, 주로 Run-Length 알고리즘을 통해 데이터 압축을 실행하고 있다[5].

그러나 Run-Length 알고리즘은 실제 적용하기에는 지도 데이터의 연속성이 미흡하다는 문제점이 있다. 이외에도 Byte-Packing 등과 같은 압축 알고리즘이 있으나, 지리정보 데이터를 단일 데이터로 압축할 수 없기 때문에 압축률이 저하된다는 문제가 있다.

따라서 본 논문에서는 지리정보 데이터를 압축하고 복원하는 속도를 개선시킨 Run-Length/Byte-Packing 융합 알고리즘을 설계하고 평가하였다. 기존 압축 알고리즘의 문제점을 개선하기 위해 제안된 알고리즘은 압축 알고리즘들의 장점을 취합하여 지리정보 데이터 압축이 가능하도록 하였다.

본 논문의 구성은 1장 서론에 이어 2장 관련연구로 압축 알고리즘에 대해 분석하였다. 3장에서는 분석한 기술을 기반으로 Run-Length/Byte-Packing 융합 알고리즘을 설계하고, 4장에서는 설계한 알고리즘을 구현 및 테스트하여 성능을 평가하였다. 마지막으로 5장에서는 결론을 기술하였다.

II. 관련연구

2.1. Run-Length 압축 알고리즘

Run-Length 압축 알고리즘은 정보원 데이터 내에 계

속하여 반복되는 문자열에 대한 데이터 크기를 물리적으로 축소할 수 있는 방법을 의미한다. 계속 반복되는 데이터에 대한 반복횟수를 측정하여 반복되는 데이터와 해당 데이터의 반복 횟수를 표기하는 방법으로 데이터 용량을 압축한다. Run-Length 압축 예시는 그림 1과 같다[6,7].

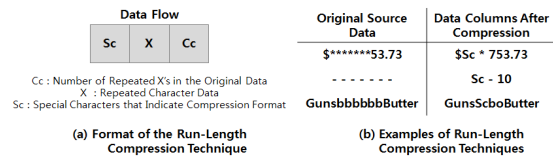


Fig. 1 Example of Run-Length Compression

2.2. Byte-Packing 압축 알고리즘

Byte-Packing 압축 알고리즘은 최소 단위인 bit 단위로 공간 데이터를 저장하여 압축된 데이터를 원본으로 완벽하게 복원하는 무손실 압축 방법을 의미한다[8].

Byte-Packing 압축 알고리즘은 공간 데이터의 저장 공간 효율성을 높이기 위해 실제 좌표 정보를 저장하지 않고, 공간 객체의 시작점을 기준으로 디퍼런셜 벡터 값을 구해 상대 좌표로 변환 및 저장한다. Byte-Packing을 이용한 정보 저장 방법은 그림 2와 같다[9].

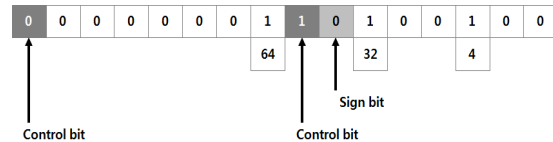


Fig. 2 Saving Information Using Byte-Packing

III. Run-Length/Byte-Packing 융합 압축 알고리즘 설계

3.1. 융합 압축 알고리즘 요구사항

본 논문에서 설계한 Run-Length/Byte-Packing 융합 압축 알고리즘을 설계하기 위해선 기본적으로 기존 연구에 비해 압축률 및 복원시간이 개선되어야 한다. 알고리즘을 설계하기 위한 요구사항은 다음과 같다.

- 원본 좌표 및 압축 좌표를 이진수로 변환하고, bit 압축을 적용을 통해 연속되는 데이터 표현이 가능하도록

록 함

- 128까지 중복 값이 표현되도록 하며 2개 이상의 좌표 뿐만 아니라 단일 또는 다수의 데이터 압축이 모두 가능하게 함
- 좌표 데이터를 표현하기 위해 무손실 압축으로 정수 및 실수 표현이 가능하도록 함

3.2. 융합 압축 알고리즘 블록다이어그램

본 논문에서는 Run-Length 압축 알고리즘과 Byte-Packing 압축 알고리즘을 융합하여 성능을 향상시킨 압축 알고리즘을 설계하였다. 설계한 알고리즘은 DB에서 데이터를 서버에 전송하면 Byte-Packing 과정을 통해 압축을 진행하며, Byte-Packing 알고리즘은 압축정보의 중복저장이 많기 때문에 Byte-Packing 알고리즘으로 데이터가 압축 되지 않을 경우 Run-Length 알고리즘을 통해 재압축을 실행한다.

이와 같이 압축과정을 모두 거치면 단말기에 데이터를 전송한다. 설계한 융합 압축 알고리즘의 블록다이어그램은 그림 3과 같다.

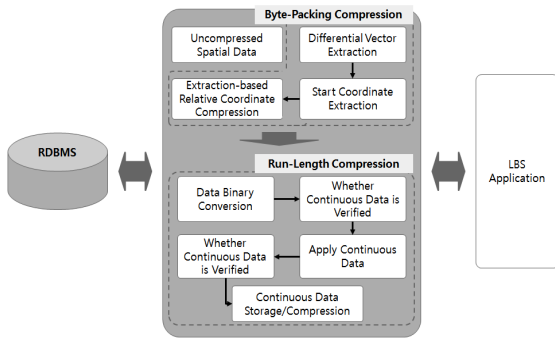


Fig. 3 Fusion Compression Algorithm Block Diagram

3.3. Run-Length 압축 알고리즘 설계

본 논문에서 설계한 Run-Length/Byte-Packing 융합 압축 알고리즘은 Run-Length 압축 알고리즘과 Byte-Packing 알고리즘으로 나누어진다.

Run-Length 압축 알고리즘은 데이터가 입력되면 입력된 데이터가 연속되는 값인지 아닌지를 확인한다. 데이터가 연속되면 최대값이 될 때까지 연속 길이 값이 증가하고, 최대값이 되기 전에 데이터가 끝나면 그 시점까지의 연속 길이 값을 데이터로 저장하게 된다. 설

계한 Run-Length 압축 알고리즘은 그림 4와 같다.

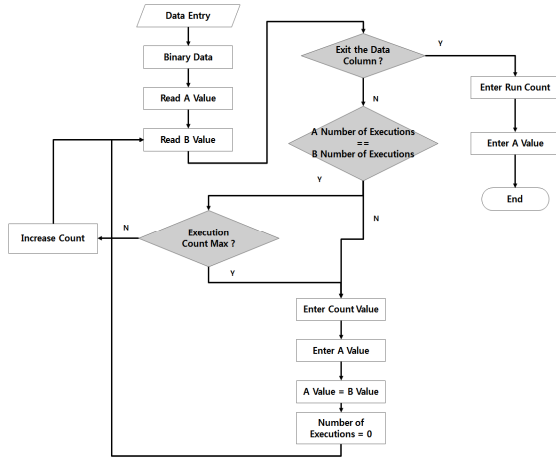


Fig. 4 Designed Run-Length Algorithm

3.4. Byte-Packing 압축 알고리즘 설계

Byte-Packing 압축 알고리즘은 입력되는 데이터를 통해 기준점인지 아닌지를 판단하고, 기준점일 경우 해당되는 기준점의 좌표를 추출한다. 좌표를 추출하면 디퍼런셜 벡터 값을 추출하기 위해 해당되는 기준점과 현재 좌표를 연산한다. 또한 디퍼런셜 벡터 값을 추출하기 위해 추출한 기준 좌표 값과 현재 좌표 값 연산을 수행한다.

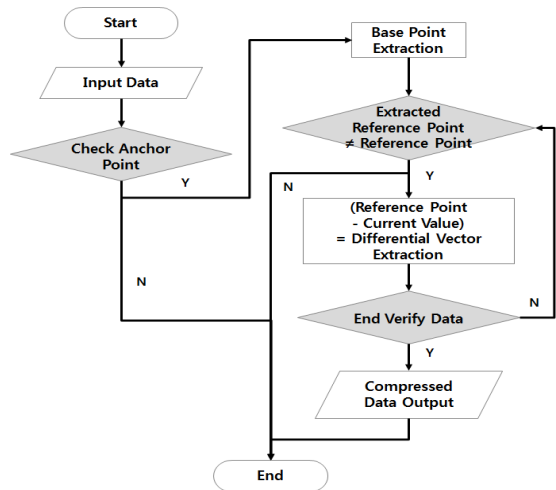


Fig. 5 Designed Byte-Packing Algorithm

만약 기준점이 없을 경우 데이터의 개수를 확인 후 데이터가 한 개이면 종료한다. 설계한 Byte-Packing 압축 알고리즘은 그림 5와 같다.

3.5. 융합 압축 알고리즘 설계

본 논문에서는 지도 데이터를 효율적으로 압축하기 위해 Run-Length 압축 알고리즘과 Byte-Packing 압축 알고리즘을 설계하였고, 두 가지 압축 알고리즘을 융합하여 새로운 알고리즘을 설계하였다. 설계한 융합 압축 알고리즘은 그림 6과 같다.

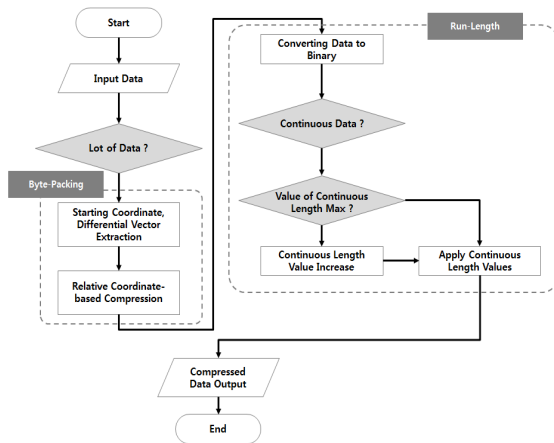


Fig. 6 Designed Fusion Compression Algorithm

설계한 알고리즘은 Run-Length 압축 알고리즘으로 연속되는 데이터를 찾아 압축한다. 이에 계산량이 비교적 간단한 Run-Length 압축 알고리즘을 이용함으로써 압축 및 복원시간을 향상시키고, 압축률을 증대시켰다.

Byte-Packing 압축 알고리즘은 시작점 좌표를 추출하는 과정과 디퍼런셜 벡터 값을 추출하는 과정으로 구분된다. 시작점 좌표를 추출하는 경우에는 절대 좌표를 가지고 있는 객체의 시작점과 디퍼런셜 벡터 값을 통해 Byte-Packing을 재구성한다.

3.5.1. 지리정보 데이터 확인 방법

데이터를 압축하기 전에 Byte-Packing을 통해 데이터가 압축할 수 있는지를 확인하고, 시작되는 기준점으로 디퍼런셜 벡터 값을 구한다. 이러한 과정을 통해 구한 디퍼런셜 벡터 값에 Byte-Packing 압축 알고리즘을 적용하고, 수집한 정보를 Run-Length에 적용하여 재압

축을 수행한다.

그러나 Byte-Packing의 경우 단일 데이터를 압축할 수 없기 때문에 Run-Length 압축 알고리즘을 통해 단일 데이터 및 시작점 정보를 압축한다.

3.5.2. 연속 데이터 확인 방법

Run-Length 압축 알고리즘은 연속 데이터가 없으면 압축을 하지 못하기 때문에 사전에 연속 데이터 여부를 확인한다. 연속되는 문자 값과 연속 횟수를 기준으로 Run-Length 압축 및 재구축을 진행하며, 연속된 데이터가 아닐 경우 Run-Length 적용 대상에 포함되지 않는다. 연속 데이터의 Max 값인 127을 기준으로 연속 데이터 값의 초과 여부를 확인하고, 이러한 체크과정을 통해 연속 데이터는 임시 데이터에 카운터 및 연속 길이 값을 증가시킬 수 있게 된다.

IV. Run-Length/Byte-Packing 융합 압축 알고리즘 테스트 및 평가

본 논문에서는 설계한 융합 압축 알고리즘의 성능 평가를 위해 원본데이터, Run-Length, Byte-Packing 압축 알고리즘과 비교 평가를 수행하였다.

4.1. Run-Length 압축 알고리즘 테스트

Run-Length 압축 알고리즘은 이진수로 변환한 좌표를 알고리즘에 적용하며, Binary 값은 0과 1로 구성된다. 실험 데이터는 표 1과 같다.

Table. 1 Experimental Data

Point	Absolute Coordinate	Binary
1	283327, 533543	1000101101101101101, 10000010110011010101
2	283250, 533416	1000101101100100000, 10000010110001010110
3	283173, 533399	1000101101011010011, 10000010101111010111
4	283196, 533162	1000101101010000110, 10000010101101011000

이러한 객체를 Run-Length 압축 알고리즘에 적용한 값은 표 2와 같다.

Table. 2 Result of Run-Length

Apply Run-Length	
105101101101101101	10710110011010101
105101101100105	1071011031010110
105101101011010011	107101014010111
1051011010104110	1071010110101103

Run-Length 압축 알고리즘 테스트는 우선적으로 이진수로 변환한 다음 Run-Length 압축 알고리즘을 적용하며, 모든 데이터에서 극대화한 연속성을 갖게 되어 연속 길이 값을 통해 압축을 진행한다.

4.2. Byte-Packing 압축 알고리즘 테스트

Byte-Packing 압축 알고리즘은 절대 위치를 가진 공간 객체의 시작점을 기준으로 나머지 점의 디퍼런셜 벡터 추출 및 비트 단위 저장을 수행한다.

테스트 데이터는 4개의 포인트를 기준으로 각 포인트를 상대좌표로 변환 및 Byte-Packing 압축 알고리즘에 따라 Hexa Code로 저장한다. 테스트 데이터는 표 3과 같다.

Table. 3 Test Data

Point	Absolute Coordinate
1	283327, 533543
2	283250, 533416
3	283173, 533399
4	283196, 533162

Byte-Packing 압축 알고리즘은 저장할 수 있는 데이터의 크기가 Max 63이 단점이지만, 8191 이내의 수를 2 바이트에 저장할 수 있는 장점이 있으며 Hexa Code 해석을 통해 원본 데이터로 완벽히 복원이 가능하다. Byte-Packing 압축 알고리즘을 적용한 결과는 표 4와 같다.

Table. 4 Result of Byte-Packing Application

Point	Absolute Coordinate	Byte-Packing
1	283327, 533543	2883327, 533543
2	283250, 533416	95, 145
3	283173, 533399	172, 72
4	283196, 533162	59, 209

4.3. Run-Length/Byte-Packing 압축 알고리즘 테스트

본 논문에서는 Run-Length/Byte-Packing 융합 압축 알고리즘을 제안하였다. Run-Length 압축 알고리즘 적용을 통해 시작점 외의 데이터들을 압축하고, Byte-Packing 압축 알고리즘을 통해 시작점 외의 나머지 데이터들을 압축한다.

Run-Length 압축 알고리즘은 연속 데이터가 적으므로 압축률도 낮지만, 이진화를 통해 데이터의 연속성 및 Run-Length 압축률을 증가한다.

Byte-Packing 압축 알고리즘은 이진화로 변환하며, 문자 값 1과 반복 개수 7을 표현하여 압축이 된다. 제안한 압축 알고리즘 적용 객체는 표 5와 같다.

Table. 5 Design Algorithm Applied Objects

Point	Absolute Coordinate	Run-Length	Byte-Packing
1	283327, 533543	105101101101101101, 10710110011010101	283327, 533543
2	283250, 533416	1001101, 15	59, 109
3	283173, 533399	10011010, 150	136, 236
4	283196, 533162	110011, 101301	219, 369

4.4. Run-Length/Byte-Packing 융합 압축 알고리즘 평가 및 결과

4.4.1. 압축률 평가

본 논문에서 설계한 Run-Length/Byte-Packing 융합 압축 알고리즘을 통해 수도권 지역의 테스트 데이터를 압축하였으며, 결과는 표 6과 같다.

Table. 6 Result of Test Data Compression

Algorithm	Compression Size (MB)	Compressibility (%)
Run-Length	100,314	53
Byte-Packing	103,802	55
Supposed Algorithm	111,166	59

저장 방법은 파일 포맷과 DB 방식을 비교하였으며, 기존 Run-Length 압축 알고리즘과 Byte-Packing 압축 알고리즘을 설계한 융합 알고리즘과 비교 평가하였으

며, 평가 결과 압축률이 향상되었음을 확인하였다.

4.4.2. 복원시간 평가

설계한 융합 압축 알고리즘의 복원시간을 평가하기 위해 수도권 지역의 공간 데이터를 복원하여 원본 데이터로 회복되기까지의 시간을 포인트에 따라 분석하였다. 분석 결과는 그림 7과 같다.

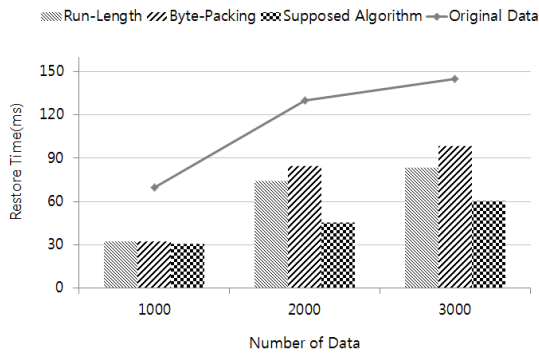


Fig. 7 Results According to Points

그림 7은 기준 데이터로 복원시간을 평가한 결과이며, 기존 압축 알고리즘보다 설계한 융합 압축 알고리즘의 복원시간이 향상된 것을 확인하였다.

다음으로 기준 데이터에 따라 복원시간을 평가하였으며, 설계한 융합 압축 알고리즘이 기존 알고리즘보다 복원시간이 향상된 것을 확인하였다. 기준 데이터에 따른 복원시간 평가 결과는 그림 8과 같다.

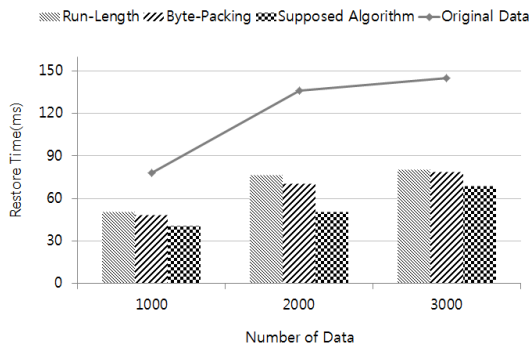


Fig. 8 Evaluation Results based on Reference Data

마지막으로 폴리곤 데이터에 따라 복원시간을 평가

하였으며, 설계한 융합 압축 알고리즘이 기존 알고리즘에 비해 폴리곤 복원시간이 향상된 것을 확인하였다. 폴리곤에 따른 복원시간 평가 결과는 그림 9와 같다.

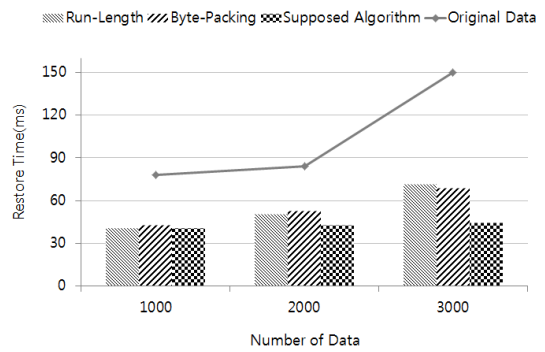


Fig. 9 Evaluation Results According to Polygon

4.4.3. 평가 결과

지리정보 데이터는 서로 다른 데이터 값으로 구성되며 본 논문에서 설계한 Run-Length/Byte-Packing 알고리즘과 기존 알고리즘들은 라인 데이터가 먼저 검색된다. 이에 라인 데이터 압축이 먼저 진행되기 때문에 포인트와 폴리곤 데이터의 중복 데이터 발생이 적으며 데이터 수가 1000개 일 때 포인트와 폴리곤 데이터의 복원시간이 비슷한 시간으로 측정되었다. 이러한 포인트 및 폴리곤 데이터는 지리정보 데이터가 누적될수록 많은 중복 데이터를 발생하기 때문에 다수의 데이터 압축이 발생하여 복원시간이 점차 상승하게 된다.

테스트 결과, 본 논문에서 설계한 Run-Length/Byte-Packing 융합 압축 알고리즘은 기존 알고리즘에 비해 압축률은 약 5% 증가하였으며, 기존 알고리즘에 비해 평균 복원시간도 향상된 것을 확인하였다. 평가 결과는 표 7과 같다.

Table. 7 Result of Evaluation

Algorithm	Compressibility (%)	Average Restore Time		
		Point (ms)	Standard (ms)	Polygon (ms)
Run-Length	53	63	68.6	53.6
Byte-Packing	55	71.3	65.3	54
Supposed Algorithm	59	45	52.6	42

V. 결 론

최근 스마트 디바이스의 보급률 증가와 함께 지리 정보를 이용한 시스템이 실생활에 접목되어 다양한 분야에 적용되어 사용되고 있다. 지리정보 시스템은 컴퓨터로 지도 데이터를 수집·분석·가공하여 지형을 사용하는 모든 분야에서 활용하기 위한 종합정보 시스템으로 교통정보, 도시정보 등 다양한 곳에서 활용되고 있다.

그러나 이러한 지리정보 시스템을 운영하기 위해서는 지리정보 데이터의 용량이 막대하다는 문제점과 지도 데이터의 정밀도는 점차 증가한다는 문제점이 있다. 이를 해결하기 위해 지리정보를 압축시킬 수 있는 압축 알고리즘에 대한 연구가 활발히 진행되고 있다.

그러나 Run-Length 알고리즘은 지도 데이터의 연속성이 미흡하기 때문에 실제 적용하기가 어렵고, Byte-Packing 알고리즘의 경우 단일 데이터로 지리정보 데이터를 압축할 수 없다는 문제점이 있다.

따라서 본 논문에서는 이러한 문제점을 개선하기 위해 압축 알고리즘들의 장점을 취합해 지리정보 데이터 압축이 가능하도록 하였다. 또한 지리정보 데이터에 대한 압축 및 복원시간을 개선시킨 Run-Length/Byte-Packing 융합 압축 알고리즘을 설계하여 이를 평가하였다.

평가 결과 기존 압축 알고리즘 대비 압축률은 약 5% 향상되었으며, 평균 복원시간 또한 기존 알고리즘에 비해 향상된 것을 확인하였다.

REFERENCE

[1] M. S. Gong, B. Kwon, J. W. Kim, S. H. Lee, "A Novel Error Detection Algorithm Based on the Structural Pattern of

LZ78-Compression Data," *Journal of the Korea Institute of Communication Sciences*, vol. 41, no. 11, pp. 1356-1363, Nov. 2016.

[2] Z. O. T. Ouibrahim, H. Benali, S. Medini, A. Belmouhoub, "Developing a geographic information system (GIS) for mapping and analyzing the polymetallic deposits of M'Sirda volcanic province, Northwest Algeria," *Arabian Journal of Geosciences*, vol. 7, no. 6, pp. 2107-2117, Jun. 2014.

[3] S. Moore, D. Haviland, W. Moore, M. Tran, "Preparing Teachers to Use GIS: The Impact of a Hybrid Professional Development Program on Teachers' Use of GIS," *Journal of Science Education and Technology*, vol. 25, no. 6, pp. 930-946, Dec. 2016.

[4] N. K. Koo, K. W. Lee, "Prediction of Wildfire Spread and Propagation Algorithm for Disaster Area," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no. 8, pp. 1581-1586, Aug. 2016.

[5] M. Guo, Y. Huang, Z. Xie, "A balanced decomposition approach to real-time visualization of large vector maps in CyberGIS," *Frontiers of Computer Science*, vol. 9, no. 3, pp. 442-455, Jun. 2015.

[6] H. Yuan, K. Guo, X. Sun, Z. Ju, "A Power Efficient Test Data Compression Method for SoC using Alternating Statistical Run-Length Coding," *Journal of Electronic Testing*, vol. 32, no. 1, pp. 59-68, Feb. 2016.

[7] D. H. Shin, C. H. Kim, "Data Compression Method for Reducing Sensor Data Loss and Error in Wireless Sensor Networks," *Journal of Korea Multimedia Society*, vol. 19, no. 2, pp. 360-374, Feb. 2016.

[8] J. H. Park, S. H. Lee, E. J. Lee, K. R. Kwon, "Efficiency Comparison of GIS Vector Map Data Compression by Vector Map Components," *Proceeding of Institute of Electronics Engineers of Korea*, pp. 153-154, Jun. 2014.

[9] G. S. Moon, Y. Kim, "Edge Image Compression Algorithm based on Characteristics of Edge Image," *Proceeding of Korea Information Science Society*, pp. 962-964, Dec. 2016.



윤석환(Seok-Hwan Yoon)

1996년 8월: 아주대학교 공학박사(정보 시스템)
 1986년 1월 ~ 1997년 12월: 한국전자통신연구원(ETRI) 책임연구원
 1998년 1월 ~ 2005년 2월: 정보통신진흥원 책임연구원
 1998년 1월 ~ 2011년 12월: 한국정보처리학회 부회장 및 편집위원장
 2013년 3월 ~ 현재: 세명대학교 컴퓨터학부 교수
 ※관심분야: 클라우드 컴퓨팅, 개발체계, 공개 소프트웨어 상용화



양승수(Seung-Su Yang)

2012년 2월: 평생교육진흥원 컴퓨터공학 공학사
2015년 2월: 가천대학교 전자계산학과 공학석사
2015년 3월 ~ 현재: 가천대학교 IT융합공학과 박사과정
※관심분야: 인터넷 통신, 의료정보 서비스, 컴퓨터 네트워크



박석천(Seok-Chon Park)

1982년 2월: 고려대학교 컴퓨터공학 공학석사
1989년 2월: 고려대학교 컴퓨터공학 공학박사
1989년 2월 ~ 현재: 가천대학교 컴퓨터공학과 정교수
※관심분야: ICMB, 멀티미디어 통신, 압축 알고리즘