

논문 2012-49CI-3-3

빈 분류기반 공간에너지집중기법을 이용한 GIS 벡터맵 압축 (GIS Vector Map Compression using Spatial Energy Compaction based on Bin Classification)

장 봉 주*, 이 석 환**, 권 기 룡***

(Bong-Joo Jang, Suk-Hwan Lee, and Ki-Ryong Kwon)

요 약

최근 공간정보표현을 위해 벡터데이터 기반 디지털 맵의 활용도의 증가와 측량기술의 발전에 기인하여 높은 수준의 해상도를 갖는 고용량의 GIS(geographic information service) 서비스가 활발히 이루어지고 있다. 본 논문은 방대한 범위에서 높은 해상도를 갖는 1cm(센티미터) 단위의 정밀 벡터맵의 효율적인 압축을 위해 공간 영역 상에서의 에너지 집중 기법(spatial energy compaction, SEC)을 이용한 벡터맵 압축 기법을 제안한다. 제안 기법은 벡터맵에서 공간정보를 표현하기 위해 주로 사용되는 폴리곤(polygon) 및 폴리라인(polyline) 오브젝트에 대해 압축을 수행한다. 제안 기법에서는 전체 벡터맵을 국부영역으로 나눈 후, 각 국부영역 내의 오브젝트 인접성 및 방향성을 이용하여 세 가지 종류의 빈으로 분류하여 에너지 집중을 수행하며, 미리 정의된 가변길이부호화를 이용해 부호화하여 압축한다. 실험결과 동일한 벡터맵에 대해 1m 정밀도의 벡터간소화 기법이 약 13%의 압축율을 갖는 데 반해, 제안 기법은 1cm 단위의 정밀도에서 원본 데이터의 80%이상 압축이 수행됨을 확인하였다. 또한 이것은 기존에 제안한 SEC 기반 압축 기법보다 높은 압축율을 가지면서 낮은 계산 복잡도를 유지하며, 기존의 벡터 근사화 기법보다 높은 정밀도에서 높은 압축률로 압축할 수 있음을 실험을 통해 확인하였다.

Abstract

Recently, due to applicability increase of vector data based digital map for geographic information and evolution of geographic measurement techniques, large volumed GIS(geographic information service) services having high resolution and large volumed data are flowing actively. This paper proposed an efficient vector map compression technique using the SEC(spatial energy compaction) based on classified bins for the vector map having 1cm detail and high range. We encoded polygon and polyline that are the main objects to express geographic information in the vector map. First, we classified 3 types of bins and allocated the number of bits for each bin using adjacencies among the objects. and then about each classified bin, energy compaction and or pre-defined VLC(variable length coding) were performed according to characteristics of classified bins. Finally, for same target map, while a vector simplification algorithm had about 13%, compression ratio in 1m resolution we confirmed our method having more than 80% encoding efficiencies about original vector map in the 1cm resolution. Also it has not only higher compression ratio but also faster computing speed than present SEC based compression algorithm through experimental results. Moreover, our algorithm presented much more high performances about accuracy and computing power than vector approximation algorithm on same data volume sizes.

Keywords : 벡터맵 압축(Vector map compression), 공간영역압축(spatial domain encoding), 빈 분류(Bin classification) 공간에너지집중(Spatial energy compaction)

* 학생회원, *** 정회원-교신저자, 부경대학교 정보보호협동과정
(Interdisciplinary Program of Information Security, Pukyong National University)

** 정회원, 동명대학교 정보보호학과
(Department of Information Security, TongMyong University)

※ 이 논문은 2011년도 교육과학기술부의 재원으로 한국연구재단의 일반연구자지원사업(2011-0010902) 및 한국산업 기술진흥원의 지역혁신인력양성사업(1345-136-198)의 지원을 받아 수행된 것임.

접수일자: 2012년2월10일, 수정완료일: 2012년5월4일

I. 서 론

현재 컴퓨터 및 네트워크의 발전에 기인하여 높은 해상도의 영상 및 2D/3D 그래픽의 전송 및 처리가 손쉽게 이루어지고 있다. 이러한 높은 해상도를 갖는 고용량의 데이터를 손쉽게 전송 및 처리할 수 있는 배경에는 각각의 콘텐츠에 적합한 압축과정이 필연적으로 포함되어 있음을 간과할 수 없다. 네트워크와 저장매체의 발전에도 불구하고, 더 높은 해상도와 더 현실감 있는 수준의 콘텐츠에 대한 대중의 욕구에 비례하여 높은 수준의 압축 기술의 발전을 가속화시키고 있다. 그로인해 현재까지 일반적인 파일 데이터를 비롯하여 음악, 정지 영상, 동영상 및 그래픽 정보를 담고 있는 콘텐츠를 표현하는 데이터의 압축을 위해 무수히 많은 종류의 압축 기술들이 연구되어 왔다. 대표적인 범용 데이터 압축 기술로는 Lempel-Ziv(LZ77, LZ78) 알고리즘을 이용한 zlib^[1] 압축기술이 있으며, 이를 개선하여 LZMA(Lempel-Ziv-Marcov Algorithm)을 이용한 더 높은 압축효율을 갖는 7-zip^[2]이 현재 널리 사용되고 있다. 이것들은 무손실 압축을 수행하는 알고리즘들로서 인코딩 및 디코딩과정을 거친 데이터가 완벽히 일치하므로 손실을 허용하지 않는 데이터의 압축에 적합하다. 반면 손실압축은 원본 데이터와 인코딩 및 디코딩을 거친 데이터가 완벽히 일치하지 않아도 되는 데이터의 압축효율을 극대화하기 위해 주로 사용된다. 멀티미디어 콘텐츠를 위한 손실압축에서는 인간시각시스템(human visibility system) 및 인간청각시스템(human audio system) 특성을 이용하여 인간의 감각에 예민하지 않은 성분을 양자화를 통해 제거함으로써 압축효율을 높이는 것이 가능하다. 대표적인 콘텐츠 압축 기술들에 대해 살펴보면, 정지영상에 대한 압축 기술로 DCT(discrete cosine transform) 및 DWT(discrete wavelet transform)과 같은 주파수변환영역에서의 중복 에너지 집중, 양자화 및 엔트로피(entropy) 코딩을 각각 수행하여 압축하는 JPEG 및 JPEG2000과 같은 기술이 널리 사용된다^[3-6]. 특히 음성, 음악 등의 소리정보와 영상이 혼재되어 동영상에 대한 압축 기법들을 하나의 표준으로 규합하여 정의한 MPEG-1, MPEG-2, 및 MPEG-4 등이 널리 사용되어 왔다. 이 중 동영상 압축을 위한 MPEG 시리즈의 Part-2 및 H.263, MPEG4-AVC/H.264, SVC 및 MVC등이 정지영상압축에서 나아가 시간적 중복성을 제거하여 압축효율을 높이는 기술들로서

개발되어왔다^[7-14]. 아울러 현재 이들보다 월등히 우수한 성능의 동영상압축 기술인 HEVC (high efficiency video coding)에 대한 연구가 활발히 진행 중이다^[15]. 한편 벡터좌표계를 갖는 그래픽데이터를 위한 압축 기법으로 벡터데이터 특성 상 앞서 언급한 일반적인 무손실 데이터 압축을 주로 사용해왔으며, 손실압축을 위해, 다양한 벡터 근사화 기법을 통해 필요한 최소한의 꼭짓점만을 남겨둔 채, 콘텐츠의 형상을 유지시키는 동시에 불필요한 절점을 제거하여 데이터양을 감소시키는 방법을 주로 사용해왔다. 대표적인 벡터 근사화 기법으로 Douglas-Peucker 알고리즘^[16]이 있으며, 이들 알고리즘은 컴퓨터 비전, 영상처리 등의 분야에서도 널리 사용된다.

한편, 오늘날의 GIS 서비스는 산업, 편의, 안전 및 행정 등에 해당하는 우리 사회 전반에 걸쳐 일상에 깊숙이 침투해있다. 다른 멀티미디어 콘텐츠와 마찬가지로, 보다 높은 정밀도와 현실 세계와 매우 흡사한 지도데이터에 대한 요구와 그에 따른 오브젝트의 추가 및 정밀도 향상으로 인해 점차 데이터양이 급격히 증가하고 있다. 실제 내비게이션 및 스마트폰, PC 등을 이용하는 GIS 서비스 사용자 입장에서, 필요한 GIS 데이터를 단 시간에 인터넷, 모바일 단말, 개인 PMP 등으로 전송 및 저장을 수행하고자 할 경우에 큰 데이터 용량으로 인해 많은 시간을 투자해야 하며 때때로 저장매체의 제약받기도 한다. 또한, 군사, 정부, 토목과 관련된 GIS 서비스 공급자의 경우, 높은 수준의 정밀도와 현실세계의 재현을 구현한 방대한 지리정보 데이터의 용량 문제는 GIS의 발전과 아울러 꾸준히 야기되고 있다. 더욱이 향후 보다 진보한 3차원 및 실사 형태의 GIS 서비스가 출현한다면 GIS 데이터의 용량성에 관한 문제는 여러 S/W 및 H/W적인 제약을 초래할 것이다. 앞서 언급한^[1-2]와 같은 일반적인 데이터 압축들을 이용할 경우, 특정 지역에 대한 벡터데이터의 임의접근(random access)을 위해 모든 압축된 데이터를 복원해야 하며, 부동소수점을 갖는 복잡한 정밀 벡터맵 데이터들의 상관성을 고려하지 않으므로 높은 압축효율을 기대하기 어렵다. 따라서 이러한 문제를 예측하고 보완하기 위해 벡터맵 데이터 압축에 대한 연구 역시 앞서 언급한 벡터 근사화 기법을 비롯한 다양한 기법들로서 제안되어왔다. Kolesnikov^[17]은 현대적 네트워크 환경에서 벡터맵 전송효율을 높이기 위해 폴리곤으로 구성된 벡터맵에 대해 벡터 근사화 기법을 이용하여 공간적 다해상도 계층

으로 분해하여, 저화질 데이터와 양자화 오차에 대해 산술 엔트로피 코딩을 수행하여 저해상도 우선순위로 점진적 전송을 가능케 했다. 하지만 높은 압축률과 빠른 수행속도에도 불구하고, 이 기법은 복잡하고 높은 정밀도를 갖는 벡터맵의 경우 연산속도와 압축률이 현저히 저하되는 문제점을 가지고 있다. Shekhar 등^[18]은 사전 검색 알고리즘을 이용하는 손실 압축 기법을 제안하였으나, 역시 정밀 지도 데이터에 대해서는 한계가 있으며, 방대한 지역을 표현하는 벡터맵 압축을 위해 생성되는 사전 데이터의 용량 역시 크게 증가하는 문제점을 가지고 있다. 이처럼 높은 정밀도를 유지하면서 벡터맵을 효과적으로 압축하기 위해서 압축율과 정밀도의 Trade-off라는 문제점과 직면함을 알 수 있다. 본 논문의 이전 연구로 제안하였던 [19]의 기법은 벡터맵 압축 기법은 공간영역 상에서 벡터 분포특성을 이용한 SEC 기법을 사용하여 정밀도를 가변하는 방법으로 고정밀 벡터맵에서 높은 수준의 압축율을 나타내는 기법을 제안하였다. 하지만 이 알고리즘은, 최종 압축 비트열을 만들기 위해 7-Zip 엔트로피 코딩기법을 사용하므로 이 때, 높은 계산복잡도를 가지게 되며 이로 인해 연산속도가 저하되는 문제점이 있었다.

본 논문은 앞서 언급한 벡터맵 데이터 압축 기법들이 정밀도가 높은 벡터맵에 적용될 경우 발생할 수 있는 문제점들을 보완하는 동시에, 더 우수한 정밀도와 압축율 및 연산속도를 갖는 벡터맵 압축 알고리즘을 제안하고자 한다. 또한 이 기법은 다양한 크기와 정밀도, 폴리곤 및 폴리라인에 구분없이 모두 적용할 수 있는 장점이 있다. 본 논문의 구성은 다음과 같다. II장에서는 기존의 GIS 보안 기술 및 관련 연구에 대하여 살펴보고, III장에서는 제안한 선택적 암호화 기법에 대하여 자세히 살펴본다. 그리고 IV장에서는 제안 알고리즘을 구현하여 실험 및 고찰한 결과를 분석하며, 마지막 V장에서는 본 논문의 결론을 맺는다.

II. 관련 연구

본 논문과 관련하여, GIS 벡터맵 압축을 위한 여러 알고리즘들을 앞서 서론에서 살펴보았다. 벡터맵 압축을 위해 이러한 기법들은 크게, 일반적인 파일 데이터 압축 기술과 벡터 근사화 기법 및 에너지 집중 등의 벡터 변환을 이용한 기법들로 분류됨을 알 수 있다. 일반적인 상용 데이터파일 압축 기법을 제외하고, 벡터 근

사화 기법의 대표적인 알고리즘인 Douglas-Peucker 기법과 SEC 기반의 정밀압축 알고리즘에 대해 아래에 간략하게 나타내었다.

1. Douglas-Peucker 벡터 근사화 알고리즘

대표적인 벡터 근사화 알고리즘으로써 [16]의 Douglas-Peucker 알고리즘은 복잡한 형태의 선 구조를 단순화 시키는 선형 단순화 기법이다. 이 알고리즘은 여러 꼭짓점으로 구성된 폴리곤, 폴리라인 등의 벡터 데이터에 대해 비 중요 꼭짓점을 제거함으로써 근사화를 수행한다. 이 기법은 다른 선형 단순화 기법들에 비해 정점 제거 및 형상 보존에서 우수하며, 구현 및 알고리즘이 단순한 장점으로 인해 벡터 그래픽스, 벡터 데이터 압축, 디지털 지도 생성 등의 기술에서 현재까지 널리 사용되고 있는 알고리즘이다. 아래 그림 1과 식 (1)로 Douglas -Peucker 알고리즘의 기본 원리를 나타내었다.

$$v_n \begin{cases} \text{Preservation, if } d_{\max} \leq \epsilon \\ \text{Elimination, if } d_{\max} < \epsilon \end{cases} \quad (1)$$

이 기법은 우선, 해당 벡터 오브젝트의 시작점과 끝점을 연결한 선분과 각 꼭짓점들 간의 수선의 길이를 각각 계산하여, 최대 수선의 길이 d_{\max} 를 찾는다. 여기서 식 (1)과 같이 최대 수선의 길이 d_{\max} 와 임계치 ϵ 을 비교하여 최대수선을 기준으로 다시 두 선분을 분할하여 각각의 최대수선을 찾는다. 더 이상 분할되지 않을 때까지 이 과정을 재귀적으로 수행하여 최종적으로 시작점, 끝점, ϵ 에 따라 선택된 분할 점들만으로 구성된 단순화 된 벡터 오브젝트를 생성하는 것으로 벡터 근사화가 수행된다.

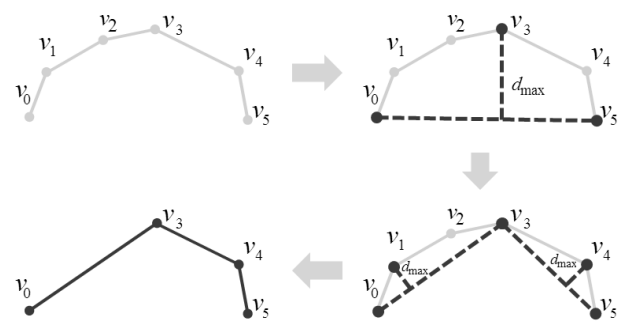


그림 1. Douglas-Peucker 알고리즘 수행 과정의 예
Fig. 1. Example of processing of Douglas-Peucker Algorithm.

2. SEC 기반 벡터맵 압축

방대한 지리영역과 높은 정밀도를 갖는 GIS 벡터 데이터의 압축을 위하여, [19]의 기존 연구에서는 그림 2에서와 같이 정수부에 대해서 SEC를 수행하며, 소수부에 대해 정밀도를 가변함으로써 벡터맵 데이터를 압축한다.

이 기법은 TM좌표계를 사용하는 정밀 벡터맵 내의 각각의 폴리라인 및 폴리곤을 최소 압축 단위로서 MCA(minimum Coding Attribute)로 정의한 후, 각 MCA 내 각 꼭짓점 좌표(x,y)들에 대해 64bit 부동소수점 값에 대한 정수부와 소수부를 분리하여 정렬한다. 벡터 공간영역에서 좌표를 표현하기 위해 중요한 성분인 정수부에 대해서는 레이어 경계영역 및 MCA 경계영역을 이용하여 2단계 에너지 집중(2-step spatial energy compaction, 2S-SEC)을 수행하며 상대적으로 정밀도에 크게 영향을 주지 않는 소수부에 대해서 사용자 정의(user-defined) 정밀도를 적용한다. 그 후, 처리된 정수부와 소수부를 각각의 MCA의 압축 비트열(payload)로서 재배열하여 엔트로피 부호화를 수행한다. 이 압축 기법은 정수부의 SEC를 위해 최적의 압축효율을 위하여 벡터맵의 폴리곤 및 폴리라인이 가지는 방향

성의 유사성을 이용한다. 각 MCA의 정수부에 대해, 꼭짓점들의 평균 좌표값을 DC로 간주하며, 이 때 각 꼭짓점과 DC가 이루는 벡터의 부호 및 크기 성분들을 분류하여 각 꼭짓점들의 유사도를 높인다. 이 과정을 수행함으로써 일반 데이터 압축기법보다 에너지 집중의 효과가 우수함이 [19]에서 실험을 통해 증명되었다.

III. 제안한 벡터맵 압축 기법

[19]에서 제안한 기법은 MCA 단위의 SEC를 수행하여 각 꼭짓점들이 이루는 벡터의 방향성과 크기의 유사성을 압축함으로써 높은 압축율을 나타내었다. 하지만, 이 논문은 벡터맵을 구성하는 오브젝트 단위의 국부 특성만을 이용하는 한계를 가지고 있다. 제안 기법은 방대한 벡터맵을 큰 지역 단위로 나눈 다음, 그 지역에 대해 각각 CB(category bin), DB(direction bin) 및 AB(accuracy bin)에 해당하는 세 종류의 빈(bin)을 이용하여 전역 특성 및 국부특성을 계층적으로 압축함으로써 [19]의 한계를 극복하는 알고리즘을 제안한다. 본 논문에서 CB는 하나의 국부지역을 여러 개의 영역으로 나누는 단위로 정의되었으며, DB는 하나의 CB에 대해 여러 개의 더 작은 영역으로 나누는 단위로 정의하였다, 또한 AB는 하나의 DB 내에 포함된 각 꼭짓점 좌표값들이 갖는 정밀도 값을 표현하는 단위로 정의하였다. 압축은 이들 각각 빈에 대해 독립적으로 수행되며, 따라서 제안 기법은 그림 3과 같이 크게 벡터맵 압축을 위한 전처리와, CB, DB 및 AB에 대해 각각의 빈 특성을 이용해 압축을 수행하는 단계들로 구성된다.

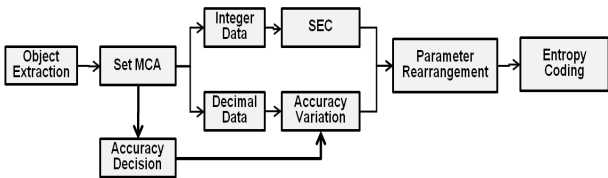
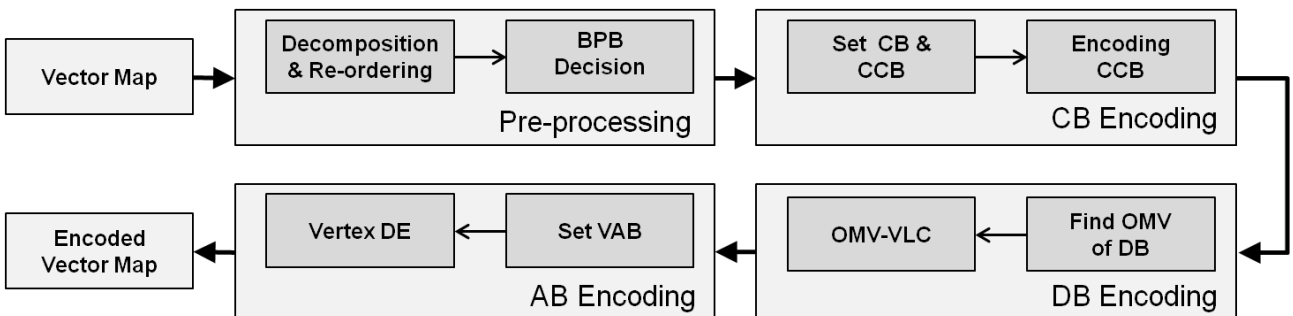


그림 2. [19]에서 제안한 SEC 기반 벡터맵 압축 알고리즘
Fig. 2. SEC based vector map compression algorithm proposed by [19].



*BPB: Bits Per Bin *CB: Category Bin *DB: Direction Bin *AB: Accuracy Bin *CCB: Connectivity of Category Bin
*OMV: Object Moving Vector *VLC: Variable Length Coding *VAB: Vertices in Accuracy Bin *DE: Differential Encoding

그림 3. 제안한 벡터맵 압축기법의 블록다이어그램
Fig. 3. Block diagram of proposed vector map compression algorithm.

1. Pre-Processing

제안 알고리즘은 벡터맵 DB의 빠른 임의접근 (random access)을 용이하게 하기 위해 벡터맵을 여러 국부지역으로 나눈 후 압축을 수행한다. 우선 압축을 위한 전처리 과정 중 전체 벡터맵으로부터 N_A 개의 국부영역들 $A = \{a_0, a_1, a_2, \dots, a_{N_A}\}$ 와 각 국부영역에 포함된 N_L 개의 레이어들 $a_i = \{l_1, l_2, \dots, l_{N_L}\}$ 로 분해하는 것을 그림 4에 나타내었다. 그림 4에서 보듯이 전체 벡터맵에 대해 동등한 범위 ($R_{A,x}, R_{A,y}$)로 나누어지는 각 국부영역 a_i 가 분리된다. 또한 각 국부영역 a_i 대해 오브젝트 속성에 따라 폴리곤, 폴리라인 및 포인트 중 한 성분의 특징을 갖는 N_P 개의 오브젝트들로 구성된 각 레이어 $l_j = \{p_0, p_1, p_2, \dots, p_{N_L}\}$ 를 분리해내는 분해 과정이 이루어진다.

이 때, 분해된 각 레이어 l_j 내에서 N_V 개의 좌표를 갖는 각 오브젝트 $p_k = \{v_0, v_1, v_2, \dots, v_{N_V}\}$ 의 시작점 v_0 를 이용하여 l_j 내의 모든 오브젝트들을 ‘좌->우’(수평방향), ‘하->상’(수직방향)의 순서로 재배열하는 과정을 함께 수행한다. 이것은 벡터맵을 제작하는 단계에서 생성되는 오브젝트들의 순서가 오브젝트의 위치와 상관없이 결정되는 것에 기인한다. 그림 5에서 오브젝트 재배열 과정을 예를 통해 나타내었다. 실제 지리학적으로 주거지역, 상업지역 등과 같은 오브젝트의 밀집도가

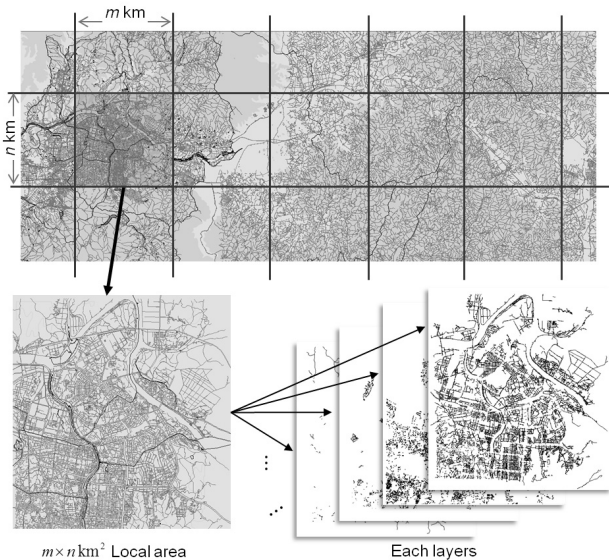


그림 4. 전체 벡터맵으로부터 국부 지역 분해과정
Fig. 4. Decomposition of local area from whole vector map.

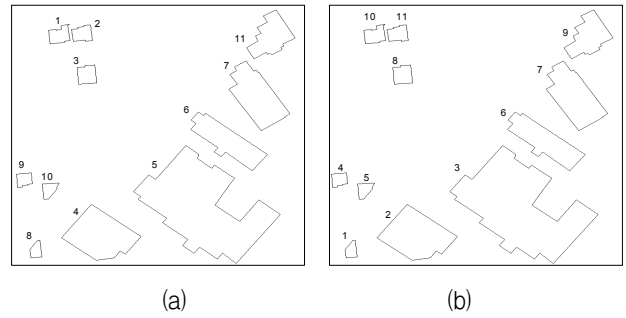


그림 5. 레이어 내의 오브젝트 재배열 과정의 예
(a) 원본, (b)재배열 후의 오브젝트 순서
Fig. 5. Example of re-arrangement of objects in a layer, (a) original and (b) object order after re-arrangement.

높은 영역의 오브젝트 인접성에 따라 그림 5와 같이 오브젝트를 재배열함으로써 압축효율을 높일 수 있다.

제안 기법에서는 지리학적으로 인접한 오브젝트를 재배치하여 각각 CB, DB, AB로서 귀속시키고, 이들 각 bin의 특성을 이용하여 각 bin에 따른 계층적 압축을 수행함으로써 압축효율의 증대를 이루었다. 따라서 각 bin들이 갖는 범위를 결정하는 것이 압축 효율에 중요한 영향을 미치므로, 벡터맵의 정밀도 c 에 따라 각 bin에 할당되는 비트 수(bits per bin, BPB)를 결정하는 것이 중요하다. 먼저, 최종 분해된 지역의 각 레이어 내에서 꼭짓점좌표가 갖는 각각의 비트수를 식 (1)로서 결정한다.

$$nBit_p = \lceil \log_2(R_{A, \max} \cdot c) \rceil ,$$

$$\text{where } R_{A, \max} = \begin{cases} R_{A, x} & \text{if } R_{A, x} \geq R_{A, y} \\ R_{A, y} & \text{if } R_{A, x} < R_{A, y} \end{cases} \quad (1)$$

이 때, TM 좌표계를 기준으로 정밀도 단위 $c = 1$ 이면 해당 벡터맵은 1m 정밀도로 압축됨을 의미한다. 또한, 여기서 계산된 $nBit_p$ 로서 CB, DB 및 AB에 할당되는 각각의 비트 수 $nBit_{CB}$, $nBit_{DB}$ 및 $nBit_{VB}$ 를 결정할 수 있다. 각 bin의 비트 수 $nBit_{CB}$, $nBit_{DB}$ 및 $nBit_{VB}$ 는 전체 비트수와 오브젝트들 간의 밀집도를 바탕으로 결정된다. CB는 벡터맵 오브젝트들 간의 연결성을 탐색하여 압축되며, 이때 연결성 탐색을 위한 CB의 비트 수를 식 (2)로서 할당하였다.

$$nBit_{CB} = \lceil \log_2(dist_{CB}) \rceil , \quad (2)$$

$dist_{CB}$, $dist_{DB}$ 및 $dist_{AB}$ 는 각각의 bin이 갖는 범위로서 실제 TM 좌표계상의 거리를 나타낸다. 한편, DB는 각 오브젝트를 표현하는 좌표값들 사이의 변위를 압축하

기 위해 사용된다. DB에 할당된 비트수 $nBit_{DB}$ 는 CB의 범위 내에서 정밀좌표값 표현을 위한 $dist_{AB}$ 를 제외하므로 식(3)과 같이 결정되어진다.

$$nBit_{DB} = \lceil \log_2(dist_{DB}) \rceil, \quad (3)$$

$$\text{where } dist_{DB} = \frac{dist_{CB}}{dist_{AB}}$$

AB는 벡터맵의 정밀도를 위해 할당되는 비트들로서 정의되며, 각 좌표들 간의 차등부호화 압축에 사용된다. AB에 할당된 비트수는 식(4)으로서 결정된다.

$$nBit_{AB} = \lceil \log_2(dist_{AB}) \rceil \quad (4)$$

벡터맵의 정밀도에 따라 $dist_{CB}$, $dist_{DB}$ 및 $dist_{AB}$ 를 결정함으로써 최적의 압축효율이 얻어지며, 본 논문에서는 실험적으로 통계화한 값을 사용한다. 따라서 식 (1)로부터 결정된 $nBit_p$ 과 각 빈의 BPB로서 $nBit_{CB}$, $nBit_{DB}$ 및 $nBit_{VB}$ 을 이용하여, 각각의 벡터맵 오브젝트는 CB, DB, 및 AB의 각 빈을 통과하면서 압축이 수행된다.

2. Category Bin Encoding

카테고리 빈 압축은 CB들에 대해 각 CB의 수직 및 수평 방향 연결성을 이용함으로써 수행된다. 이 단계에서는 CB내에 포함된 모든 오브젝트 좌표들에 대해 하나의 대푯값과 명세 좌표값들로 치환함으로써 압축이 가능하다. 우선, 분해된 레이어 l_j 내 N_p 개의 오브젝트들 p_k 의 시작점 v_0 를 하나 이상 포함하는 CB를 모두 탐색한다. 제안 방법에서의 탐색 방법을 표 1로서 나타내었다. 탐색은 현재 CB를 기준으로 ‘down’, ‘Right’, ‘Up’, ‘Left’ 방향의 순서로 이루어지며, 현재 CB가 종속된 방향은 표 1의 ‘CheckFromWhere()’ 루틴을 통해 탐색에서 제외한다.

표 1의 알고리즘을 이용하여 CB의 연결성을 탐색한 예를 그림 6에 나타내었다. 그림 6 (a)는 폴리라인으로 구성된 정밀벡터맵의 도로 레이어를 나타낸 것이며, (b), (c), (d), 및 (e)는 각각 $dist_{CB}$ 에 대해 320m, 160m, 80m 및 40m 범위를 갖는 CB에 대해 연결성을 탐색한 결과이다.

그림 6에서 알 수 있듯이, $nBit_{CB}$ 에 따라 CB의 연결성이 결정된다. $dist_{CB}$ 가 작을수록 식 (2)에 의해 CB에 포함된 오브젝트가 갖는 좌표값들의 비트수는 줄

표 1. CB 연결성 탐색을 위한 Pseudo Code
Table 1. Pseudo Code for searching CB Connectivity.

```

Function SerchConnectivity()
{
  From = CheckFromWhere()

  IF (CB != Isolated){
    IF(From != DOWN)
      IF(isConnetDown){
        SetDonwFlag()
        MoveToDown()
        SerchConnectivity()
      }
    IF(From != RIGHT)
      IF(isConnetRight){
        SetLeftFlag()
        MoveToRight()
        SerchConnectivity()
      }
    IF(From != UP)
      IF(isConnetUp){
        SetUpFlag()
        MoveToUp()
        SerchConnectivity()
      }
    IF(From != LEFT)
      IF(isConnetLeft){
        SetLeftFlag()
        MoveToLeft()
        SerchConnectivity()
      }
  }
}

```

어드는 반면, CB의 연결성이 저하되며 하나의 CB 내 포함되는 오브젝트 수가 줄어드는 결과를 초래한다. 이것은 이후 설명될 CB 압축을 저하에 기인하므로, 벡터맵 특성에 따라 $dist_{CB}$ 로서 CB의 크기를 선택하는 것이 중요하다. 한편, 표 1과 그림 6으로부터 각 탐색된 CB의 특성에 따라 표 2와 같이 분류하였다.

여기서 오브젝트가 속하지 않은 CB를 CB_{null} 로 정의하고, 압축이 필요 없는 국소영역으로 간주한다. CB에 포함된 오브젝트가 하나 이상일 경우, 카테고리 빈 압축단위로서 $CB_{isolated}$, CB_{head} , CB_{follow} 등을 정의하여 압축을 수행한다. 먼저, 연결성이 없는 독립된 CB의 경우 $CB_{isolated}$ 로 정의하며, 이 때, $CB_{isolated}$ 에 포함된 오브젝트의 수 N_{P,CB_i} 와 각 오브젝트 내 좌표값의 수 N_V 를 정의하면, $CB_{isolated}$ 에서 압축되는 총 비트수 $\epsilon_{isolated}$ 는 각 $CB_{isolated}$ 의 대푯값 $2 \cdot nBit_{CB}$ 과 나머지 각 좌표값들에 필요한 비트수에 의해 아래 식 (5)로서 추정할 수 있다.

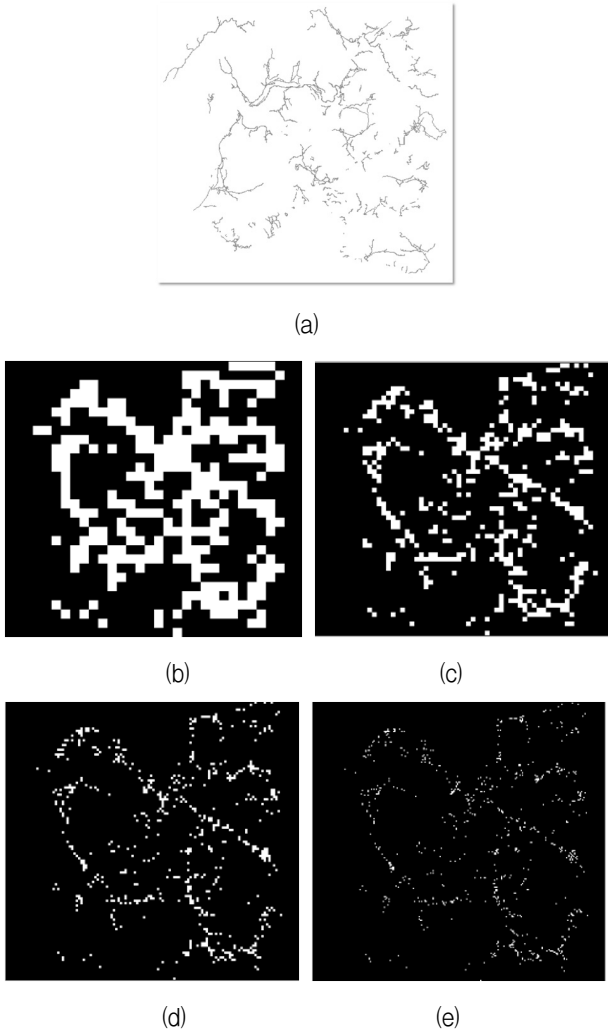


그림 6. $nBit_{CB}$ 따른 카테고리 bin의 연결성 탐색 예, (a)대상 폴리라인 레이어, (b) $dist_{CB} = 320m$ 인 경우, (c) $dist_{CB} = 160m$ 인 경우, (d) $dist_{CB} = 80m$ 인 경우, 및 (e) $dist_{CB} = 40m$ 인 경우

Fig. 6. Exampel of CB connectivity searching by $nBit_{CB}$, (a)Target polyline layer, (b) in case of $dist_{CB} = 320m$, (c) in case of $dist_{CB} = 160m$, (d) in case of $dist_{CB} = 80m$, and (e)in case of $dist_{CB} = 40m$.

$$\epsilon_{isolated} = 2 \cdot \left\{ nBit_P \cdot \sum_{a=0}^{N_{P,CB}-1} N_{V,a} - \left((nBit_P - nBit_{CB}) + nBit_{CB} \cdot \sum_{a=0}^{N_{P,CB}-1} N_{V,a} \right) \right\} \quad (5)$$

한편, 연결성을 갖는 CB들에 대해서 각각 CB_{head} , CB_{follow} 를 정의하였다. CB_{head} 는 연결성을 갖는 최초

표 2. 연결성 탐색 후 CB의 분류

Table 2. CB Classification after CB connectivity searching.

CB 종류	오브젝트 수	종속성	압축 비트 수
CB_{null}	0	No	0
$CB_{isolated}$	≥ 1	No	$\epsilon_{isolated}$
CB_{head}	≥ 1	No	ϵ_{head}
CB_{follow}	≥ 1	Yes	ϵ_{follow}

의 CB를 의미하며, 이때 압축 가능한 비트 수 ϵ_{head} 는 $CB_{isolated}$ 와 동일한 $2 \cdot nBit_{CB}$ 크기의 대푯값을 정의하므로 식 (6)에서처럼 $\epsilon_{isolated}$ 에 더하여 ‘Down’, ‘Right’, ‘Up’ 및 ‘Left’의 순서로 연결성 정보 3 비트가 추가되어 추정된다.

$$\epsilon_{head} = \epsilon_{isolated} + 3 \quad (6)$$

반면 CB_{follow} 는 CB_{head} 에 대해 종속성을 갖는 모든 CB를 정의하며, 이 CB_{follow} 들은 표 1의 알고리즘에 따라 대푯값으로서 연결성 정보만을 가진다. 여기서 각 CB의 연결성을 나타내는 정보는 현재 CB의 종속된 방향을 제외하고 ‘down’, ‘Right’, ‘Up’, ‘Left’ 순서로 3비트를 할당된다. 따라서 CB_{follow} 내에 포함된 N_{P,CB_f} 개의 오브젝트들의 총 압축 비트 수는 식 (7)과 같이 추정된다.

$$\epsilon_{follow} = 2 \cdot nBit_P \cdot \sum_{a=0}^{N_{P,CB}-1} N_{V,a} - (3 + 2 \cdot nBit_{CB} \cdot \sum_{a=0}^{N_{P,CB}-1} N_{V,a}) \quad (7)$$

그림 6의 예에서와 일반적으로 CB의 연결성을 고려하기 위해서 각 $CB_{isolated}$ 및 CB_{head} 가 갖는 대푯값이 $2 \cdot nBit_{CB}$ 이 CB_{follow} 의 연결성 정보 3비트보다 큰 값을 가진다. 따라서 최종적으로 CB의 연결성 정보를 이용한 레이어 전체의 압축 효율을 표 2로부터 분류된 각 CB의 수 $N_{CB_{isolated}}$, $N_{CB_{head}}$, 및 $N_{CB_{follow}}$ 에 따라 식 (8)과 같이 얻을 수 있다.

$$\epsilon_{CB} = N_{CB_{isolated}} \cdot \epsilon_{isolated} + N_{CB_{head}} \cdot \epsilon_{head} + N_{CB_{follow}} \cdot \epsilon_{follow} \quad (8)$$

식 (5) 와 식 (7)로부터, $nBit_{CB}$ 를 이용하여 CB들의 연결성을 높임으로서 $N_{CB_{isolated}}$, $N_{CB_{head}}$ 의 값을 줄이고, $N_{CB_{follow}}$ 를 증가시키는 방법으로 CB의 압축율을 향상시킬 수 있다. 반면, CB의 연결성 향상은 $nBit_{CB}$ 에 대해 많은 수의 비트를 할당하므로 CB 내의 오브젝트 좌표값들이 갖는 비트수를 상대적으로 증가시키는 경우가 발생한다. 따라서 CB의 연결성과 할당된 비트수 사이의 Trade-Off관계를 고려하여 $dist_{CB}$ 를 결정해야 한다.

3. Direction Bin Encoding

각 CB의 연결성 탐색과 압축이 완료되면 각 CB가 포함하는 각각의 오브젝트에 대해 오브젝트가 지나는 DB의 변위를 추정함으로써 DB 압축을 수행한다. 각 CB는 식 (3)에 의해 $N_{DB} \times N_{DB}$ 개의 $nBit_{DB}$ 비트를 갖는 DB들로 나누어지며, 이 때, CB에 포함된 임의의 오브젝트 p_k 는 여러 DB에 걸쳐 존재하게 된다. 그림 7 로써 4×4 개의 DB를 갖는 CB내에서 여러 DB에 걸친 폴리라인 오브젝트의 예를 나타내었다.

그림 7에서 보듯이, 하나의 오브젝트는 여러 개의 DB에 걸쳐 존재하며, 오브젝트에 의해 선택된 DB들은 각각 하나 이상의 좌표값들을 포함하고 있다. DB에 대

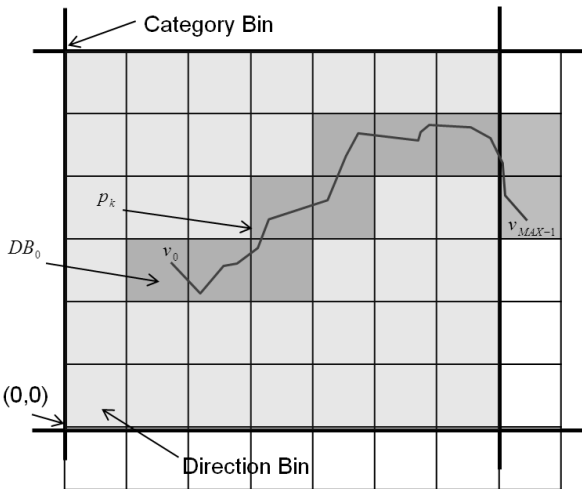


그림 7. 4×4 개의 DB를 갖는 CB 내에서 오브젝트 단위의 압축을 위해 선택된 DB들
Fig. 7. Selected DBs in a CB having 4×4 numbers of DB for vector object compression.

한 압축은 하나의 오브젝트에 대해 연속적인 DB에 할당된 비트수 $nBit_{DB}$ 를 각 DB의 변위로 표현함으로써 수행될 수 있다. 그림 7의 예에서, v_0 가 포함된 $DB_0 = (1,2)$ 를 기준으로 DB의 진행 방향과 크기를 표현하면 식 (9)를 이용하여 표 3으로서 얻어진다.

$$DDB_q = DB_q - DB_{q-1}, \quad 0 < q \leq |v_{MAX-1} \in DB_q| \quad (9)$$

표 3으로 알 수 있듯이, DB의 진행방향은 식 (9)에 의해 DB들 간의 차등부호화의 결과로서 표현되며, 그림 7의 예에서 오브젝트 일부가 CB를 벗어나더라도 모든 CB가 동일한 크기의 DB를 가지므로 연산의 결과에 영향을 미치지 않는다.

표 3에 나타난 결과로 미루어 DB의 진행방향으로서 압축을 수행하는 것은 수치지도 상에서 연결된 두 좌표 사이의 실제 거리가 한정적임에 기인한다. 그림 6-(a)를 이용하여 $dist_{DB} = 10m$ 로 정의하였을 때 모든 DB들의 진행방향과 크기에 대한 빈도 수를 그림 8에 나타내었다.

그림 8로부터 $dist_{DB} = 10m$ 로 결정했을 때, 일반 도로를 나타내는 그림 6-(a)에 대해 전체 DB에 대한

표 3. 그림 7의 예를 이용한 DB의 변위 검색 결과
Table 3. Searching results of DB differentials using Fig. 7.

order	0	1	2	3	4	5	6	7	8	9
DB	1,2	2,2	3,2	3,3	4,3	4,4	5,4	6,4	7,4	7,3
DDB	1,2	1,0	1,0	0,1	1,0	0,1	1,0	1,0	1,0	0,-1

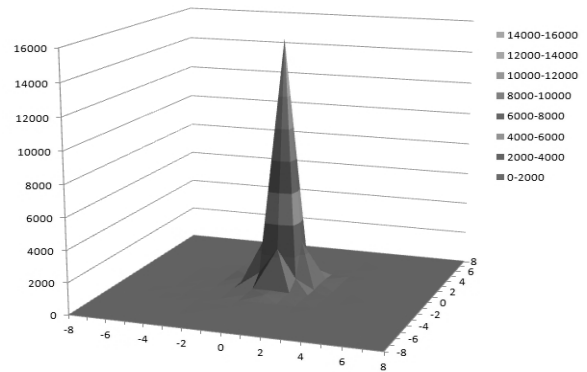


그림 8. 그림 6-(a)에 대해 $dist_{DB} = 10m$ 일 때의 모든 DB에 대한 진행방향과 크기에 대한 빈도 수
Fig. 8. Frequency of all DBs' directions and magnitudes when $dist_{DB} = 10m$ in Fig. 6-(a).

변위 $|DDB| \leq 1$ 인 경우가 전체의 96.2% 이상을 차지하며, $|DDB| \leq 2$ 인 경우는 전체의 99.5 % 이상임을 확인할 수 있다. 이로써 하나의 오브젝트에 대해 큰 비트수로 할당된 DB라 하더라도, DB들 간의 인접성을 이용하여 그 변위들만 표현함으로써 적은 수의 비트로 압축이 가능함을 알 수 있다. 또한 각 DB로부터 계산된 변위 DDB에 대한 통계는 비디오 압축에서 사용되는 MPEG2-Part2^[7]의 움직임벡터(MV, motion vector)와 유사한 결과를 가짐을 알 수 있다. 따라서 제안 기법에서는 DB의 최종적인 압축을 위해, 각각의 DDB에 대해 []에서 사용되는 MV를 위한 VLC(variable length coding) 테이블을 이용하여 VLC를 수행한다. 제안한 압축 기법에서는 DB 압축과정을 통해 가장 큰 압축효율을 얻을 수 있다.

3. Accuracy Bin Encoding

CB로부터 오브젝트들의 인접성을 이용하여 1차적인 압축을 수행하며, DB로부터 각 오브젝트의 좌표점들의 대략적인 변위를 추정함으로써 2차적 압축을 수행할 수 있다. 벡터맵 압축을 위해서는 CB 및 DB로부터 압축된 데이터 이외에 $dist_{AB}$ 에 해당하는 각 좌표점이 갖는 세부적인 위치값들에 대한 고려가 필요하다. 제안한 AB 압축 기법에서는 식 (4)로 표현되는 세부 좌표값에 대한 비트 수 $nBit_{AB}$ 로써 식 (1)에서 정의한 정밀도의 벡터맵을 저장할 수 있는 방법을 사용한다. 3.3절로부터 하나의 DB에 포함된 좌표값들의 정밀 데이터에 대해 계층적 부호화를 수행함으로써 압축된 데이터 내에서의 정밀도 가변 및 손실/무손실 압축을 수행할 수 있다. DB 내 좌표값

들은 최대 $nBit_{AB}$ 개의 AB를 가지며 이들 AB들은 그림 9와 같이 각 좌표의 x, y 값에 대해 비트 슬라이싱(bit slicing)된 값으로서 저장된다.

제안 기법에서, AB에 할당된 비트값들은 정밀한 국소지역을 명세하기 위해 사용되므로 실질적인 에너지 집중을 이용한 압축은 이루어지지 않는다. 그 대신, 벡터맵의 사용 목적 및 연산속도와 압축을 향상을 위해 그림 9에서 나타내었듯이 각AB들을 계층적으로 전송 또는 저장함으로써, 벡터맵의 손실압축에 사용할 수 있도록 설계하였다.

IV. 실험 결과

제안한 bin 분류기반 벡터맵 압축 기법에 대한 성능 평가를 위한 실험을 수행하였다. 실험을 위해 그림 10과 같이, $10km \times 10km$ 의 국부 영역을 갖는 벡터맵 레이어를 생성하였으며, 이 중 폴리곤으로 구성된 ‘건물’ 레이어와 폴리라인으로 구성된 ‘일반도로’ 레이어에 대해 각각의 실험을 수행하였다. 또한 TM 좌표계에 의해 표현되는 실측치 1cm 이내의 오차를 갖도록 식 (1)의 $c = 100$ 으로 설정하여 정밀 벡터맵 압축에 대해 결과를 도출하였다. 또한 $10km \times 10km$ 영역에서 1cm 정밀도를 갖는 벡터맵 데이터의 최적의 압축효율을 위해, 실험적으로 $dist_{DB} = 10m$ 및 $dist_{AB} = 0.64m$ 로 결정하였다.

제안 기법의 객관적 비교를 위해 일반적인 데이터 압축기법으로서 7-zip 알고리즘[2], 대표적인 벡터 간소화 알고리즘으로서 Douglas-Peucker 알고리즘[16] 및 기존의 공간 에너지집중 기반에서의 압축 알고리즘[19] 등과 압축 결과를 비교하였다. 표 4에 각각의 알고리즘에 대한 압축 결과를 나타내었다.

표 4의 결과로부터 벡터맵의 특성을 고려하지 않은 7-Zip과 비교하였을 때, 각 좌표점들의 오차범위로 0.01m 미만을 갖는 벡터맵 bin 분류 기반의 제안 기법이 월등히 높은 압축효율을 가짐을 확인하였다. 한편, 벡터 간소화 기법을 사용하는 Douglas-Peucker 알고리즘과 제안 알고리즘은 압축원리가 상이하므로 RMSE(root mean square error) 등의 방법을 통한 직접적인 비교는 어렵다. 또한 제안 기법과의 비교를 위해 Douglas-Peucker 알고리즘의 정밀도 범위 $\epsilon = 0.01m$ 로 설정하게 되면, 각 좌표점들의 간소화 조건이 대부분 만족하지 않으므로 실제 간소화를 통한 압축은 거의 이루어지

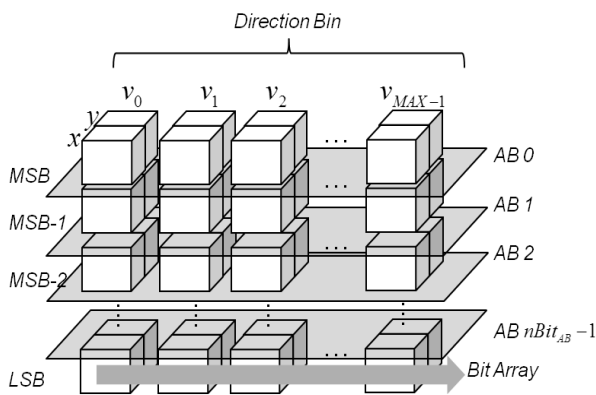


그림 9. 오브젝트가 걸친 DB에 포함된 상세 좌표값들에 대한 AB 구조.

Fig. 9. AB structure about detailed coordinates in DB having components of an object.



그림 10. 실험에 사용한 10km×10km 벡터맵 레이어, (a) 폴리곤으로 구성된 ‘건물’ 레이어 및 (b) 폴리라인으로 구성된 ‘일반도로’ 레이어.

Fig. 10. 10km×10km vector map layers used in experiments, (a) ‘Building’ layer composed of polygons, (b) ‘Main Street’ layer composed of polylines.

표 4. 제안 알고리즘의 압축 결과 비교

Table 4. Comparison results of proposed compression algorithm with others.

Layer		Original	Proposed method (error < 0.01m)	7-zip	Duglas-Peucker ($\epsilon = 1m$)	present SEC (error < 0.01m)
10-(a)	Byte	1,127,552	210,442	747,315	986,736	482,119
	rate	100%	18.66%	66.28%	87.51%	42.76%
10-(b)	Byte	1,448,400	272,721	863,444	798,816	580,315
	ratio	100%	18.83%	59.61%	55.15%	40.07%

지 않는다. 따라서 그림 10의 벡터맵 레이어에서 시각적으로 용인할 수 있을 것으로 간주되는 적정 임계치 $\epsilon = 1m$ 로 설정하여 압축을 수행하였다. 표 4의 실험 결과는 이때의 압축 결과를 나타낸 것이다. 그림 11은 벡터맵을 확대하였을 때, 표 4의 압축 효율에 대해 제안 알고리즘과 Douglas-Peucker 알고리즘의 시각적 오차를 비교한 것이다.

표 4와 그림 11로부터 제안 기법이 Douglas-Peucker 알고리즘보다 월등히 적은 시각적 오차를 가짐에도 불구하고 보다 높은 압축효율을 나타냄을 확인할 수 있다. 더욱이, 표 4에서 Douglas-Peucker 알고리즘의 특성상 연속한 좌표점들이 급격히 꺾히는 오브젝트들의 집

합인 ‘건물’ 레이어에서 압축효율의 저하가 발생한다. 반면, 제안 기법은 Douglas-Peucker 알고리즘에서와 같은 압축효율의 저하는 발생하지 않음을 확인하였다. 공간영역에서의 에너지 집중(SEC)을 이용한 기존방법과의 비교에서 역시 제안 알고리즘이 보다 우수한 압축효율을 가짐을 알 수 있었다. 또한 에너지 집중 이후 실질적 데이터 압축을 위한 엔트로피 코딩으로써 zlib 또는 7-zip 알고리즘을 이용하는 기존 기법과는 달리, 제안 기법은 압축 과정의 일부로서 정의된 단순한 허프만 테이블을 참조하므로 계산 복잡도에서 큰 이점을 얻을 수 있다.

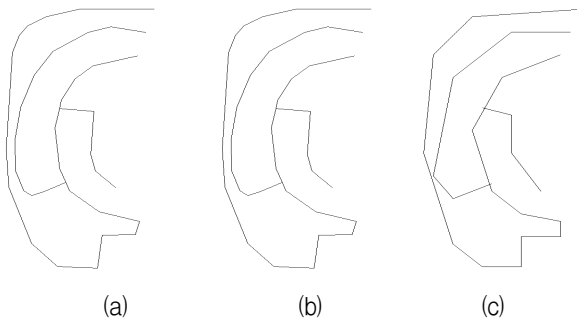


그림 11. 벡터맵 확대 시, 정밀도 0.01m를 갖는 제안 알고리즘과 $\epsilon = 1m$ 를 갖는 Douglas-Peucker 알고리즘의 시각적 오차 비교, (a)원본 벡터맵, (b) 제안 알고리즘 (c) Douglas-Peucker 알고리즘

Fig. 11. After enlarging vector map, comparison of visible artifacts of proposed algorithm with 0.01m accuracy and Douglas-Peucker algorithm with $\epsilon = 1m$, (a) uncompressed vector map, (b) compressed map by proposed algorithm and (c) Douglas-Peucker algorithm.

V. 결 론

본 논문에서는 높은 정밀도를 갖는 벡터 맵 데이터의 압축을 위해, 국부 벡터맵 영역에 대해 CB, DB 및 AB 등 세 개의 빈 영역으로 세분화하여 각각의 빈의 특징에 대해 독자적인 압축을 수행함으로써 압축효율을 최대화하는 기법을 제안하였다. 제안 기법을 사용하여 0.01m 정밀도를 갖는 벡터맵에 대해 압축을 수행한 결과 원본 벡터맵의 80%이상의 데이터가 압축됨을 실험을 통해 확인하였다. 또한, 제안 기법이 다른 여러 압축 기법들과 비교하여 최대 4배 이상의 압축효율을 얻음으로서 제안 기법의 우수성을 증명하였다.

제안 알고리즘을 사용함으로써 높은 정밀도를 요구하는 벡터맵을 효율적으로 실시간 압축을 수행할 수 있으므로, 관공서 군사용, 네비게이션 E등의 다양한 분야에서 활용할 수 있을 것으로 기대한다. 또한 2차원 벡터데이터를 갖는 다양한 CAD 시스템으로의 확장 또한 가능할 것으로 기대한다.

참 고 문 헌

[1] <http://zlib.net/>, January, 2012.
 [2] <http://7-zip.org/>, January, 2012.
 [3] W.B. Pennebaker and J.L. Mitchell, "JPEG: Still Image Data Compression Standard," *Van*

Nostrand Reinhold, 1993.

- [4] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 Still Image Compression Standard," *IEEE Signal Processing Magazine*, Vol. 18, Issue. 5, pp. 36-58, Sept. 2001.
- [5] M. Boliek, C. Christopoulos, and E. Majani(editors), "JPEG2000 Part I Final Draft International Standard," *ISO/IEC JTC1/SC29 WG1 N1855*, Aug. 2000.
- [6] 장원우, 조성대, 강봉순, "JPEG 및 JPEG2000을 이용한 영상 압축과 색역 변화의 관계," *대한전자공학회논문지*, 제 46권, 제 4호, pp. 1-8, 2009년 7월
- [7] "Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2), 1994.
- [8] "Video Coding for Low Bit Rate Communication," ITU-T, ITU-T Recommendation H.263 version 1, 1995.
- [9] "Coding of audio-visual objects - Part 2: Visual," in *ISO/IEC 14496-2 (MPEG-4 Visual Version 1)*, Apr. 1999.
- [10] 김우식, 박인수, 박수현, 이병욱, 박래홍, "시험 비트열을 이용한 MPEG-4 텍스처 디코딩의 정적 Conformance Test," *대한전자공학회논문지*, 제 38권, 제 1호, pp.69-78, 2001년 1월.
- [11] "Advanced video coding for generic audiovisual services," ITU-T Rec. & ISO/IEC 14496-10 AVC, 2005.
- [12] 최용일, 전병우, "MPEG4-SVC의 SNR 스케일러블 부호화 기술 동향," *대한전자공학회논문지*, 제 32권, 제 9호, 100-110쪽, 2005년 9월.
- [13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Extension of the H.264/MPEG-4 AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, to appear, p. Draft, Summer 2007.
- [14] M. Droese and C. Clemens, "Results of CE1-D on multiview video coding," *ISO/IEC JTC1/SC29/WG11 Doc. m13247*, Montreux, Switzerland, 2006.
- [15] "Vision and requirements for High performance video coding(HVC) Codec," *ISO/IEC JCT1/SC29/WG11, MPEG Doc. N10175*, Pusan, KR, Oct. 2010.
- [16] D. H. Douglas and T. K. Peucker. "Algorithms for the reduction of the number of points required to represent a line or its caricature," *The Canadian Cartographer*, Vol.10, Issue. 2, pp.112-122, 1973.

[17] Alexander Kolesnikov, "Vector Maps Compression for Progressive Transmission," *Proc. of the 2nd IEEE International Conference on Digital Information Management (ICDIM'07)*, pp. 81-86. Lyon, France, Oct. 2007.

[18] S. Shekhar, Y. Huang, and J. Djughash, "Dictionary Design Algorithm for Vector Map Compression," *Proceedings of the Data Compression Conference IEEE Computer Society*, 2002.

[19] 장봉주, 문광석, 이석환, 권기룡, "GIS 디지털 맵의 안전한 전송 및 저장을 위한 효율적인 압축 기법," 한국멀티미디어학회논문지, 제 14권, 제 2호, pp.210-218, 2011년 2월.

저 자 소 개



장 봉 주(학생회원)
 2002년 부산외국어대학교
 전자공학과 학사 졸업.
 2004년 부산외국어대학교 전자
 컴퓨터공학과 석사 졸업.
 2007년~현재 부경대학교 정보보
 호협동과정박사과정수료.
 <주관심분야 : 영상압축, 멀티미디어 정보보호>



이 석 환(정회원)
 1999년 경북대학교 전자공학과
 학사 졸업.
 2001년 경북대학교 전자공학과
 석사 졸업.
 2004년 경북대학교 전자공학과
 박사 졸업.
 2005년~현재 동명대학교 정보보호학과 조교수
 <주관심분야 : 워터마킹, DRM, 영상신호처리>



권 기 룡(정회원)-교신저자
 1986년 경북대학교 전자공학과
 학사 졸업.
 1990년 경북대학교 전자공학과
 석사 졸업.
 1994년 경북대학교 전자공학과
 박사 졸업.

1996년~2006년 부산외국어대학교 디지털정보
 공학부 부교수
 2006년~현재 부경대학교 IT융합응용공학과 교수
 <주관심분야 : 멀티미디어 정보보호, 영상처리,
 멀티미디어통신 및 신호처리>