K평균 군집화를 이용한 벡터 데이터 압축 방법

Vector Data Compression Method using K-means Clustering

이동헌*, 전우제**, 박수홍***
Dong-Heon Lee, Woo-Je Chun, Soo-Hong Park

요 약 최근 이동전화, PDA, 텔레매틱스 단말기 등과 같은 모바일 기기의 사용이 늘어나고 있다. 모바일 기기들에서 지원하는 서비스 중 큰 부분을 차지하는 것으로는 위치추적, 경로 탐색 등의 서비스가 있다. 이러한 서비스를 제공하기 위하여 모바일 환경에서의 공간데이터에 대한 사용이 증가하고 있다. 하지만 모바일 기기의 저장 공간이 늘어났음에도 불구하고 여전히 공간데이터에 대한 요구를 수용하기에는 한계가 따른다. 따라서 본 연구에서는 모바일 환경에서 사용되는 공간 데이터에 대한 손실 압축 기법을 제시하고, 실험을 통한 압축률, 데이터 손실 정도를 분석하고자 하였다. 이렇게 제시된 공간 데이터 압축 기법을 실제 데이터에 적용하여 실험해 보고 동일 데이터에 대하여 선행 연구에서 제시한 방법을 적용한 결과와 비교·분석을 통하여 제시된 압축 방법이 높은 위치 정확도를 요구하는 데이터에 적용 하였을 때 더 나은 성능을 보이는 것을 제시할 수 있었다.

Abstract Nowadays, using the mobile communication devices, such as a mobile phone, PDA, telematics device, and so forth, are increasing. The large parts of the services with these mobile devices are the position tracking and the route planning. For offering these services, it is increasing the use of the spatial data on the mobile environment. Although the storage of mobile device expands more than before, it still lacks the necessary storage on the spatial data. In this paper, lossy compression technique on the spatial data is suggested, and then it is analyzed the compression ratio and the amount of loss data by the test. Suggested compression technique on the spatial data at this paper is applied to the real-data, and others methods, suggested at the previous studies, is applied to same data. According as the results from both are compared and analyzed, compression technique suggested at this study shows better performance when the compression result is demanded the high position accuracy.

주요어 : 벡터 데이터, 압축, K평균 군집화
KeyWords : Vector data, Compression, K-means Clustering

1. 서 론

최근 이동전화, PDA, 텔레매틱스 단말기 등 모바일 기기의 사용이 빠른 속도로 증가하고 있다. 이들 모바일 기기에서는 경로탐색, 지도 서비스 등을 위해서 공간데이터의 사용이 필수적으로 요구된다. 하지만 모바일 기기는 여전히 데스크톱 환경에 비하여 제한적인 연산 수행능력과 저장 공간의 한계성을 갖는다. 이러한 이유로 모바일 환경에서는 래스터 데이터에 비하여 상대적으로 적은 저장 공간을 차지하는 벡터 데이터 조차도 여전히 큰 부담이 된다. 모바일 환경에서 사용되는 대표적인 벡터 데이터에는 실제로 거리 측정 또는 경로탐색을 위한 네트워크 데이터와 배경으로 사용되는 지도 데이터가 있다. 맵 데이터는 경우에

* 인하대학교 지리정보공학과 석사 dheon.lee@samsung.com
** 인하대학교 지리정보공학과 석사 woojechun@hotmail.com
*** 인하대학교 지리정보공학과 조교수 shpark@inha.ac.kr

는 약간의 위치 오차가 포함되더라도 그 오차정도가 눈으로 구분할 수 없을 정도의 수준이라면 받아들여질 수 있다[1]. 실제로 모바일 환경에서의 맵 데이터는 지형지물의 강조를 위하여 단순화 등의 방법이 사용된다. 사전기반[2]의 벡터 데이터 압축 기법에 대한 선행 연구가 진행 되었지만 이들은 적용하기에는 공간 데이터의 위치 오차 수준이 너무 커지게 되거나, 오차를 줄인다면 사전이 너무 커지는 단점이 있어 실제 적용이 어려운 문제점을 가지고 있다. 본 연구의 목적은 맵 데이터에 대하여 K평균 군집화 기법을 이용한 손실 압축방법을 설계하고 그 효율성을 제시 하는 것이다. 세부 목적으로는 사전기반의 접근방법을 이용하고, 군집화 기법인 K평균 군집화를 이용하여 사전을 제작한다. 세 번째는 전체 데이터에 대한 압축률 보다는 실제 사용 가능하도록 데이터의 손실률을 최소화 하는 방향으로 압축한다. 마지막으로 설계된 압축 알고리듬과 선행 연구와의 비교 실험을 통하여 본 연구의 타당성과 적용가능성에 대하여 평가한다.

2. 선행연구

현재까지 데이터 압축 분야에서 벡터 데이터에 대한 압축 기법은 래스터 데이터에 비하여 연구가 다양하지 못하였다. 축척 변환을 위하여 데이터의 구조를 변환하는 과정에서 적용 가능한 제거, 통합, 단순화, 치환 등의 방법[3]은 전체 데이터의 크기를 줄이는데 사용 가능하다. 또 벡터 데이터에 대한 일반화 과정, 예를 들면 선에 대한 일반화 방법인 Douglas-Peucker[4] 방법 역시 압축 방법으로 이용 가능하다. 대표적인 사전기반의 벡터 데이터 압축에 관한 연구로는 2000년에 발표된 "Design Algorithms for Vector Map Compression"[5] 가 있다. 이 연구에서는 벡터 데이터를 각각 상대적인 위치를 나타내는 디퍼런셜 벡터로 나눈 후, Freeman coding[6][7]을 기반으로 하는 FHM(Fibonacci, Huffman, and Markov) 방법을 이용하여 미리 제작된 사전에 근사화 하는 방법을 통하여 데이터 량을 줄이는 방법이 제시 되었다. 이 연구에서는 미리 제작된 사전을 사용하는 방식으로 사전을 제작하는 과정을 생략할 수 있지만 사전이 데이터의 특성을 반영할 수 없으므로 복원 시에 많은 위치 오차를 갖는다. 또 다른 연구로는 데이터 특성을 찾아낼 수 있는 데이터 마이닝 기법 중 K평균 군집화 기법을 적용하여 사전을 제작하고, 이를 통하여 벡터 데이터를 압축하는 방법에 대하여 연구한 "Vector Map Compression: A Clustering Approach"[1] 가 있다. 이 연구의 결과는 FHM방법을 이용하는 것 보다 더 좋은 성능을 보이지만 실제 데이터에 적용 했을 때 여전히 만족할만한 위치 정확도를 얻을 수 없었다.

3. 적용 기술

3.1 사전기반 압축

사전기반 압축방법은 주어진 데이터를 대표할 수 있는 값을 추출하여 사전을 제작하고 이들을 가리키는 포인터 집합을 이용하여 중복되는 값을 제거함으로써 전체 데이터 량을 줄이는 방법이다. 이는 여러 가지 방법으로 공간 데이터에 적용할 수 있으며, 가장 간단한 예로 아래와 같은 활용이 가능하다. OGC에서 발표한 공간 데이터베이스의 표준인 Simple Features Specification For SQL1.1[8]에 따라 임의의 두 기하 객체를 저장하면 POLYGON((1 1, 1 2, 2 2, 2 1, 1 1)), POLYGON((1 2, 2 2, 2 3, 1 3, 1 2))으로 표현할 수 있다. 여기에서 중복되는 점을 제거하여 POINT(1 1), POINT(2 1), POINT(1 2), POINT(2 2), POINT(1 3), POINT(2 3)와 같이 6개의 좌표를 이용하여 사전을 구성할 수 있다. 이처럼 폴리곤에서는 실제 좌표를 저장하지 않고 사전의 각 좌표를 가리키는 포인터만을 가지고 객체를 저장하는 방식이 사전기반 압축방식이다. 연구에서는 중복되는 모든 점을 저장하지 않고 유사한 좌표의 점을 하나의 대푯값으로 근사화하는 과정을 통하여 데이터의 크기를 줄인다. 좌표를 근사화 하였을 때 위치오차가 최소가 되도록 사전을 제작하는 것을 목표로 한다.

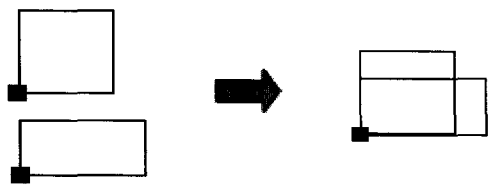
3.2 K평균 군집화

K평균 군집화 기법은 데이터 마이닝 분야에서 일련의 지식을 추출하는 방법의 한가지로, 미리 기준을 정하지 않고 유사도가 높은 개체들을 하나로 묶어주는 방법이다[9][10]. 본 연구에서는 여러 가지 군집화 기법 중 K평균 군집화 기법을 사용하였다. 이 방법은 군집 중심의 개수를 입력하여 각 데이터를 대표할 수 있는 K개의 군집으로 나누는 방법이다. 알고리듬이 비

교적 간단하여 공간데이터와 같은 대용량 데이터에 적용이 쉽고, 초기 값 K 이외에 다른 사전정보를 필요로 하지 않는 것이 장점이다[11]. 하지만 적절한 K의 크기를 정하는 것이 어렵다는 단점이 있다. 연구에서는 K의 크기를 변화시키면서 적절한 값을 찾고자 하였다. 물론 K 값을 크게 한다면 위치정확도는 향상이 된다. 하지만 공간 데이터의 위치 정확도를 높이고자 K 값을 한없이 늘려 준다면 압축률이 떨어지는 것을 피할 수 없다. 이렇게 K 값이 커짐에 따라 압축률이 저하되는 것을 막기 위하여 연구에서는 개별적인 2차원 좌표를 군집화에 적용하지 않고, 길이와 각도의 두 인자로 쪼개어서 군집화 기법을 적용하였다.

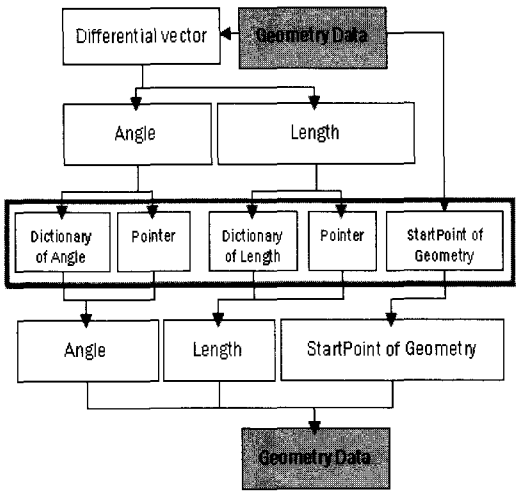
4. 압축 알고리듬

벡터 데이터를 압축하는 데에 K평균 군집화 기법을 적용하기 위해서는 기하 객체를 여러 개의 디퍼런셜 벡터로 나누는 과정이 필요하다. 디퍼런셜 벡터는 현재 점의 좌표를 표현하기 위하여 이전 점의 좌표 혹은 해당 객체가 시작하는 점 좌표와의 차이를 이용하여 상대적인 위치를 표현하는 벡터이다[1]. 연구에서는 해당 좌표와 이전 좌표의 차이를 이용하여 디퍼런셜 벡터를 추출 하였다. 도시계획이 체계적으로 이루어진 도심지에서는 디퍼런셜 벡터가 도로 중심선과 유사한 각도를 이루며 반복해서 나타나는 경향을 보이고 있었다. 이런 디퍼런셜 벡터의 각도를 따로 추출하여 군집화를 수행한다면 보다 작은 K 값을 이용하더라도 좋은 결과를 얻을 수 있을 것이다. 또한 연구에서 공간 객체의 시작점을 나타내는 첫 번째 점의 좌표는 군집화를 적용하는 데에 포함시키지 않도록 압축되지 않은 상태로 따로 저장하게 된다. 객체의 시작점은 이전 점이 존재하지 않으므로 디퍼런셜 벡터를 만들 수 없고, 원점과의 차이를 이용하여 만들어진 벡터를 시작점들만을 따로 군집화 기법에 적용시키더라도 <그림 1>과 같이 인접 객체와 합쳐지는 문제가 발생하기 때문이다.



<그림 1> 인접 시작점이 같은 군집이 된 경우

다음은 이렇게 추출된 디퍼런셜 벡터를 길이와 각도로 분리하여 각각 K평균 군집화를 적용하는 과정이다. 디퍼런셜 벡터를 2차원 공간상에서 K1개의 군집 중심을 갖는 군집화를 적용하면 사전이 가질 수 있는 경우의 수는 K1개가 된다. 하지만 두 개의 인자로 분리하여 각각 길이 K2개, 각도 K3개의 사전 두 개를 제작한다면, 전체 사전이 가질 수 있는 디퍼런셜 벡터의 수는 K2 x K3 개가 된다. 이렇게 하나의 공간 데이터를 두 요소로 나눈다면 적은 크기를 갖도록 사전을 제작하더라도 좌표를 근사화하는 단계에서 위치 오차를 줄일 수 있게 된다. 이런 과정을 거쳐 객체의 시작점, 두 개의 사전, 두 개의 사전을 가리키는 포인터의 총 다섯 부분으로 최종 압축된 결과를 얻을 수 있다. 위 과정의 역 과정을 통하여 위치오차가 포함된 데이터를 복원할 수 있다. <그림 2>는 일련의 압축과 복원 과정을 표현한 그림이다.



<그림 2> 압축/복원 알고리듬

5. 실험 및 분석

5.1 실험 데이터

실험 데이터는 서울시 강서구, 양천구의 1:1,000 수치지도에서 면으로 표현되는 지형, 지물 중 건물을 나타내는 레이어를 추출하여 제작하였다<그림 3>. 이렇게 추출된 데이터는 72016개의 폴리곤 형태를 가지며, 개별적인 위치를 나타내는 점의 수는 모두 566607개 이다.

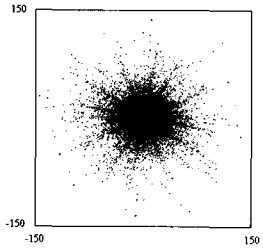
<그림 3> 실험 데이터

5.2 실험 내용

실험 데이터를 절대적인 위치를 표현하는 각 객체의 시작점 좌표와 군집화 기법을 적용하기 위한 디퍼런셜 벡터로 분리 하였다. 연구에서 사용한 디퍼런셜 벡터는 현재 점 좌표에서 이전 점 좌표와의 차이를 이용하여 추출 하였다. 그리고 각 객체의 시작점은 실제 좌표를 이용하였다. <그림 4>는 각 객체의 시작점 좌표를 분리하여 나타낸 그림이다.

<그림 4> 시작점 집합

다음 과정은 이렇게 분리된 시작점을 제외한 나머지 점 좌표에 대한 디퍼런셜 벡터를 계산하는 과정이다. 계산된 디퍼런셜 벡터의 수는 전체 객체가 가지는 점의 수에서 객체의 시작점의 개수, 즉 객체의 수만큼을 뺀 것과 같다. <그림 5>는 시작점을 제외한 디퍼런셜 벡터를 확대한 그림이다.



<그림 5> 디퍼런셜 벡터 집합

이렇게 얻어진 결과를 길이와 각도로 분리한 후 각각 K평균 군집화 기법을 적용하여 두 개의 사전과 두 개의 포인터를 얻는 과정을 통하여 압축이 완료된다. 각각의 디퍼런셜 벡터에 대하여 기존 연구의 방법을 이용하였을 경우와 연구에서 제시하는 방법을 적용 하였을 때 얻을 수 있는 표현 가능한 디퍼런셜 벡터 수와 사전이 차지하는 저장 공간의 크기를 비교하였다.

<표 1> 연구에서 제시하는 방법을 이용한 사전 제작

길이 사전 엔트리 수	각도 사전 엔트리 수	표현 가능한 디퍼런셜 벡터 수	전체 사전 크기 (byte)	포인터 크기(bits)
16	16	256	256	4 × 2
32	32	1024	512	5 × 2
64	64	4096	1024	6 × 2
128	128	16384	2048	7 × 2
256	256	65536	4096	8 × 2
512	512	262144	8192	9 × 2
1024	1024	1048576	16384	10 × 2

<표 2> 기존 연구에서 제시하는 방법을 이용한 사전 제작

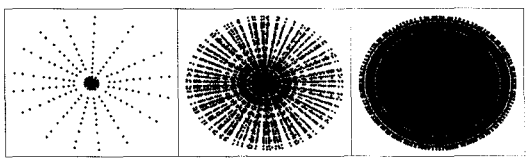
사전 엔트리 수	표현 가능한 디퍼런셜 벡터	전체 사전 크기(byte)	포인터 크기(bits)
256	256	4096	8
512	512	8192	9
1024	1024	16384	10
2048	2048	32768	11
4096	4096	65536	12
8192	8192	131072	13
16384	16384	262144	14
32768	32768	524288	15
65536	65536	1048576	16

기존 연구에서 제시하는 방법을 이용하여 사전을 제작할 경우 사전이 차지하는 공간이 두 배가 될 때 표현 가능한 디퍼런셜 벡터의 수 역시 두 배가 된다. 하지만 본 연구에서 제시하는 방법을 적용하여 사전을 제작할 경우 사전이 차지하는 공간이 두 배가 될 때 표현 가능한 디퍼런셜 벡터의 크기는 4배가 된다. 또한 동일한 수의 사전 엔트리를 갖도록 할 경우 연구에서 제시하는 방법을 적용하였을 때 사전이 차지하는 저장 공간이 더 적게 나타나는 것을 확인할 수 있다. 표 1과 표2에서 마지막 부분의 포인터 크기는 제작되는 사전의

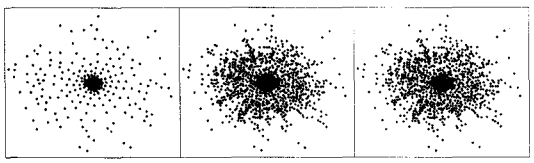
엔트리 수와 연관이 있다. 따라서 사전의 엔트리 수를 모두 포함 할 수 있는 최소의 포인터가 필요하게 된다. 예를 들면 256개의 엔트리를 모두 가리킬 수 있는 포인터의 최소 크기는 8bits 이다. 연구에서는 포인터의 크기를 결정하고 그 포인터가 가리킬 수 있는 최대의 수만큼을 엔트리로 갖는 사전을 제작하여 실험 하였다. 연구에서 제안하는 방법의 경우에는 각각 길이와 각도에 대하여 두 개의 포인터가 필요하게 되므로 크기가 두 배가 된다. 이렇게 크기를 줄여 사전을 제작할 때 위치정확도의 차이가 미미하다면 기존의 방법보다 향상된 압축 방법이라 할 수 있다.

5.3 실험 결과

실험에서 K평균 군집화 과정은 통계 패키지인 SPSS v10 을 이용하여 수행 되었다. 연구에서 제안하는 방법과 기존 연구의 방법을 비교하기 위하여 동일한 데이터에 대하여 거리와 각도 모두 16, 32, 64, 128, 256, 512, 1024 개로 늘려가면서 군집화 수행해 보고 2차원상의 디퍼런셜 벡터를 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536개의 K 값으로 갖는 군집화를 수행 하였다. 이렇게 해서 생성된 거리와 각도의 사전으로 조합 가능한 경우의 수를 표현해 보면 아래의 그림과 같다<그림 6><그림7>.



<그림 6> 제안된 방법을 조합 가능한 벡터 사전



<그림 7> 기존 연구 방법을 적용하여 제작한 벡터 사전

<그림 6>에서 좌로부터 각도, 길이의 사전 수가 각각 16, 64, 256개로 가질 수 있는 모든 조합의 수는 256, 4096, 65536개가 된다. 이것은 그림 5에서 나타나는 디퍼런셜 벡터가 그림 6의 사전에 가장 가까운 값으로 근사화 될 때 사전이 클수록 적은 위치 오차를 포함할 수 있는 것이다. <그림 7>은 선행연구 방법을 실험 데이터에 적용 하였을 때 가질 수 있는 사전의 분포이다. 마찬가지로 좌로부터 256, 4096, 65536개의 엔트리 수를 갖도록 사전을 제작한 결과이다. 두 방법에 대하여 위와 같이 전체 디퍼런셜 벡터를 대표하는 사전을 제작하고 원 데이터를 사전의 엔트리로 근사화 하는 과정을 통하여 데이터를 압축해 보고 결과를 비교하였다. 결과비교를 위하여 각 디퍼런셜 벡터를 사전에 근사화 하는 방법으로 데이터를 압축하고 이를 이용하여 압축된 데이터의 크기와 압축률을 계산하였다. 그리고 원 데이터로의 재구성을 통하여 복원된 데이터를 압축되기 전 원 데이터와의 위치 정확도 손실 정도를 계산해 보았다. 실험은 두 방법에 대하여 각각 9번과 7번을 수행하였는데 기존 연구에 대하여 k값을 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536로 증가시키면서 실험 하였고, 연구에서 제안하는 방법을 길이와 각도에 대하여 k값을 16, 32, 64, 128, 256, 512, 1024로 증가시키면서 실험 하였다.

<표 3> 연구에서 제안된 방법 적용 결과

엔트리수		사전 크기 (byte)	포인터 크기 (bits)	압축된 크기 (byte)	RMSE	압축률 (%)
16	16	128 x 2	3380600	1575087	2.676862	82.63
32	32	256 x 2	4225750	1680986.75	1.493162	81.46
64	64	512 x 2	5070900	1787142.5	0.955886	80.29
128	128	1024 x 2	5916050	1893810.25	0.464277	79.11
256	256	2048 x 2	6761200	2001502	0.220999	77.92
512	512	4096 x 2	7606350	2111241.75	0.104137	76.71
1024	1024	8192 x 2	8451500	2225077.5	0.052501	75.46

<표 4> 기존 연구 방법 적용 결과

엔트리수	사전 크기 (byte)	포인터 크기 (bits)	압축된 크기 (byte)	RMSE	압축률 (%)
256	4096	3380600	1578927	2.255	82.58
512	8192	3803175	1635844.875	1.712879	81.96
1024	16384	4225750	1696858.75	1.263176	81.28
2048	32768	4648325	1766064.625	0.898792	80.52
4096	65536	5070900	1851654.5	0.62569	79.58
8192	131072	5493475	1970012.375	0.432236	78.27
16384	262144	5916050	2153906.25	0.299722	76.24
32768	524288	6338625	2468872	0.208773	72.77
65536	1048576	6761200	3045982	0.146055	66.4

동일한 데이터에 대하여 두 가지 다른 손실 압축 방법을 적용하여 압축 성능을 분석하고자 할 때, 가장 중요한 변수로는 전체 데이터가 얼마나 작아졌는지에 대한 판단 기준인 압축률과 데이터의 손실 정도를 나타내는 데이터 손실률이 있다. 연구에서는 공간 데이터에 대한 손실 압축 기법을 제시하였기 때문에 데이터의 손실률은 위치 정확도의 손실로서 측정이 가능하다. 위의 표에서 압축된 데이터의 크기는 폴리곤의 시작점을 저장하는 부분, 디퍼런셜 벡터의 거리와 각도를 대표하는 두 개의 사전, 사전을 가리키는 두 개의 포인터의 다섯 부분을 합한 것이다. 기존 연구와의 데이터 크기 비교를 위하여 압축률을 사용하였고, 위치정확도를 비교하기 위하여 RMSE를 사용하였다. 압축률은 압축하기 전 원본 데이터의 크기와 압축이 완료된 데이터의 크기에 대한 비율로 계산 되었고, 위치정확도를 비교하는 척도로는 RMSE를 사용하였다. 이 두 계산에 사용된 수식은 아래와 같다.

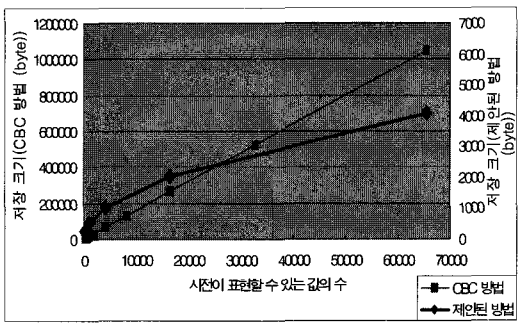
$$\text{압축률}(\%) = \frac{\text{원본 데이터 크기} - \text{압축 데이터 크기}}{\text{원본 데이터 크기}} \times 100$$

$$SE = \sqrt{\frac{e_1^2 + e_2^2 + e_3^2 + \cdots + e_n^2}{n-1}}$$

또한 포인터의 크기는 <표 1>과 <표 2>에서 개별 포인터의 크기가 정해지면 이것과 디퍼런셜 벡터의 수의 곱 형태로 계산이 가능하다. 개별적인 디퍼런셜 벡터마다 포인터가 필요하게 되고, 전체 디퍼런셜 벡터의 수는 전체 포인트 수에서 각 객체의 시작점을 제외한 수와 같게 된다. 따라서 다음 식과 같이 계산이 가능하다.

포인터크기=
(전체포인트수-전체폴리곤수)× $\lceil \log_2(\text{사전엔트리수}) \rceil$

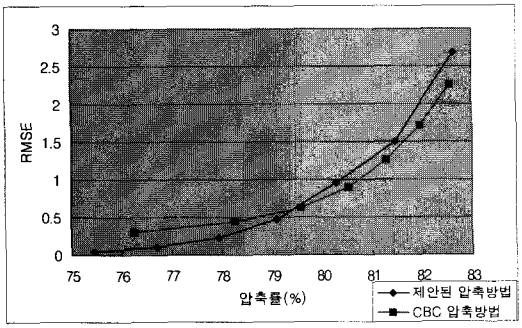
두 가지 방법에 대하여 실험한 결과 사전이 표현할 수 있는 디퍼런셜 벡터의 수 측면에서 비교를 해 보았다. 실제로 디퍼런셜 벡터를 표현할 수 있는 경우의 수를 동일하게 만들어줄 경우 기존 압축 방법이 RMSE로 표현되는 위치 정확도의 손실이 적은 것으로 나타났다. 표 3과 표 4에서 사전이 가질 수 있는 경우의 수를 65536개로 하였을 때 기존 압축 방법은 사전의 엔트리 수는 모든 경우를 엔트리로 작성해야 하므로 65536개가 필요하다. 하지만 제안된 압축 방법에서는 두 개의 사전 엔트리 수의 곱만큼을 수용할 수 있다. 따라서 256개의 길이 사전과 256개의 각도 사전을 이용하여 65536개의 경우를 표현할 수 있다. 그러므로 사전이 차지하는 저장 공간의 크기는 기존 방법에서 1048576바이트가 필요하던 것이, 제안된 방법에서는 4096바이트만으로도 저장이 가능하다. 결과를 살펴본다면 기존 압축 방법에서의 사전의 크기는 사전이 가질 수 있는 수가 두 배로 늘어나면 저장 공간 역시 두 배로 늘어나게 된다. 하지만 제안된 방법에서는 사전이 가질 수 있는 경우의 수가 4배가 될 때 사전이 차지하는 크기는 2배가 된다.



<그림 8> 사전의 표현 범위와 저장 공간 크기 관계

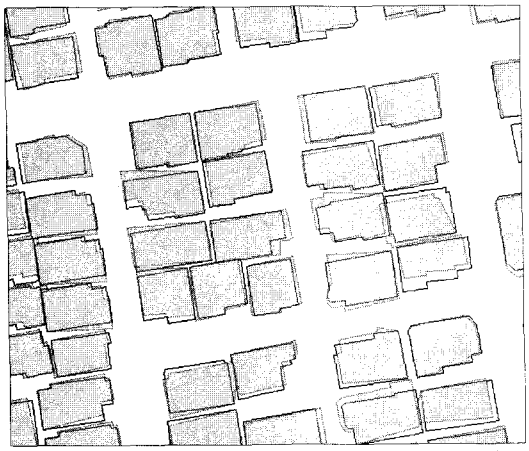
<그림 8>은 사전이 가질 수 있는 범위가 커질 때, 사전이 차지하는 저장 공간의 크기를 나타낸 그래프이다. 그래프에서와 같이 사전의 표현 범위가 커짐에 따라 두 방법을 따르는 사전이 차지하는 저장 공간의 차이가 점점 커진다. 사전의 엔트리 수가 표현할 수 있는 경우의 수가 동일할 경우 기존 방법의 사전이 좀 더 정확한 위치 정확도를 보여주고 있다. 하지만 위치 정확도를 높이기 위하여 제안된 방법보다 기존 방법을 이용하기에는 위치 정확도의 향상이 너무 미미한 수준이고, 반사적으로 손해 보아야 하는 압축률의 차이가 너무 크다. 실험 데이터에 두 방법을 적용한 경우 사전의 크기를 각각 65536과 256,256으로 하였을 때, 약 0.15정도의 위치정확도 향상을 위하여 압축률을 75.46%에서 28.86% 로 줄여야 한다. 이러한 좁은 지역에 대해서는 동일한 경우의 수를 갖는 기존 방법을 적용하기보다는 제안된 방법을 적용하여 사전의 크기를 한 단계 크게 만드는 편이 압축률과 위치 정확도 양쪽에서 향상된 효과를 볼 수 있다.

<그림 9>는 두 가지 방법을 건물 데이터에 적용한 경우의 압축률과 위치 정확도의 관계이다.

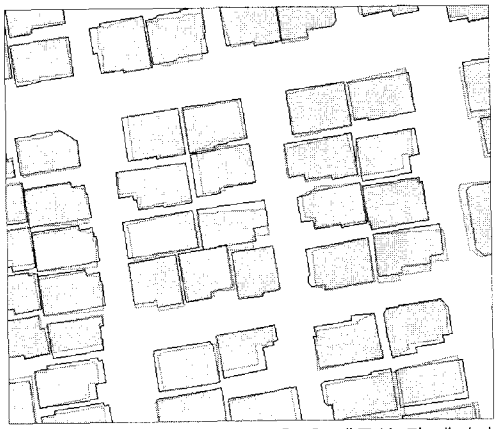


<그림 9> 압축률과 위치 정확도 관계

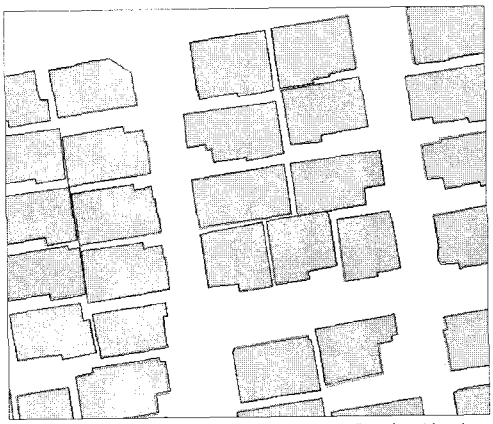
두 데이터의 경우에서 사전의 크기를 작게 하여 압축률을 높였을 경우는 재구성시 위치 정확도가 제안된 방법에 비하여 기존 방법의 성능이 높게 나타났다. 하지만 위치 정확도를 높여 사전의 크기를 증가시킬 수록 위치 정확도의 차이는 비슷한 수준을 유지하게 되고, 압축률은 크게 향상된 결과를 확인할 수 있다. 그림 10과 그림 11은 사전이 가질 수 있는 경우의 수를 갖게 하여 두 방법을 데이터에 적용한 결과이다. 두 경우에서 RMSE는 각각 1.493162, 1.263176 로 기존 방법이 다소 좋은 위치 정확도를 보이고, 압축률은 81.46%, 81.28%로 제안된 방법이 좋은 압축률을 보였다. 그림상에서는 큰 차이를 느끼기 힘들지만 재구성시의 위치 정확도에서 기존 방법이 좋은 성능을 보였다.



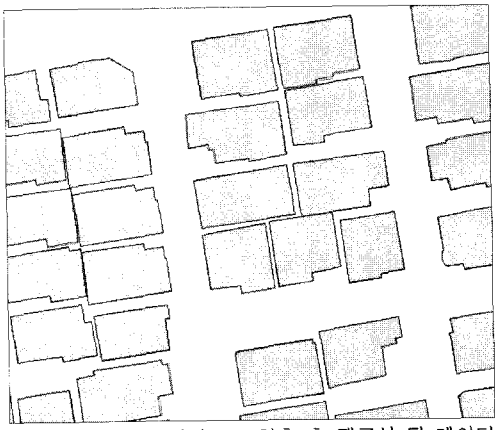
<그림 10> 제안된 방법으로 압축 후 재구성 된 데이터 (사전 크기 : 32 x 32)



<그림 11> 기존 방법으로 압축 후 재구성 된 데이터 (사전 크기 : 1024)



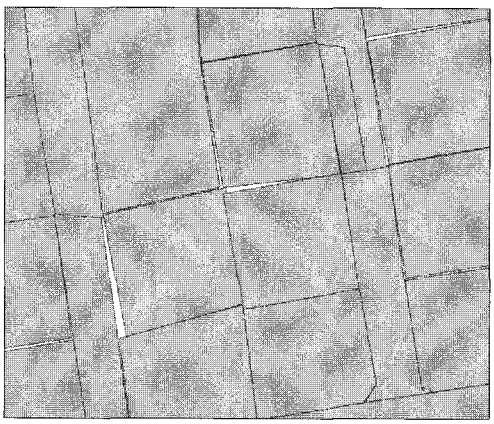
<그림 12> 제안된 방법으로 압축 후 재구성 된 데이터 (사전 크기 : 256 x 256)



<그림 13> 기존 방법으로 압축 후 재구성 된 데이터 (사전 크기 : 65536)

사전의 크기를 늘려서 두 방법에 적용하여 <그림 12>와 <그림 13>의 결과를 얻었다. 위의 두 결과는 사전의 엔트리가 구성할 수 있는 값의 범위를 65536으로 유지하고 실험을 하였다. 이 경우 역시 기존 압축 방법이 재구성시 위치 정확도가 제시된 방법에 비하여 수치적으로는 좋은 것을 확인 할 수 있다. 하지만 육안으로는 거의 유사한 정도의 위치 정확도를 보인다. 이러한 경우에 압축률 면에서 기존 방법은 66.4%를 제안된 방법은 77.92%를 보이고 있다. 제안된 방법은 기존 방법과 비교하여 구별하기 힘들 정도의 위치 정확도를 가지고, 압축률에서 10% 정도의 우위를 보이는 것으로 나타난다.

실험 데이터의 경우에는 인접 폴리곤과 점을 공유하고 있지 않는 데이터이다. 제안된 방법에서는 객체의 시작점만을 절대 좌표로 저장하며 나머지 점의 좌표는 시작점으로부터의 상대적인 위치를 표현한다. 따라서 실험 데이터와는 달리 인접 폴리곤과 점을 공유하는 필지와 같은 데이터의 경우 연구에서 제안하는 손실 압축 방법을 적용한다면 인접한 폴리곤과 공유하는 점의 좌표가 두 폴리곤 사이에 다른 점으로 저장된다. 이러한 현상은 압축률을 낮추어 사전을 아무리 크게 제작하더라도 <그림 14>와 같이 필연적으로 발생하게 된다.



<그림 14> 인접 폴리곤과 점을 공유하는 경우 발생하는 왜곡

따라서 손실 압축을 적용하면서 이러한 문제를 완벽히 해결하기 위해서는 압축 기법 이외에 위상 정보를 복원할 수 있는 연구가 필요하다. 연구에서는 디스플레이용 데이터 압축 기법을 제시 하고자 하였다. 이러한 목적을 위해서는 압축률 저하시키더라도 육안으로 왜곡정도를 느끼기 힘든 정도라면 가능할 것이다. 실험 데이터에 두 압축 방법을 적용하여 가능성을 판단해 보았다. 표 3과 표 4에서와 같이 사전의 엔트리 수를 256×256과 65536으로 늘려주었을 각각 위치정확도 수준이 0.220999와 0.146055로 나타났다. 이러한 경우 <그림 12> <그림 13>에서 보는 것처럼 육안으로는 이격 정도를 인식하기 힘든 정도의 결과를 낳았다.

6. 결 론

본 연구에서는 모바일 환경에 적절한 벡터 데이터의 손실 압축 기법을 설계하고 결과와 선행 연구를 비교 분석 하여 위치 정확도와 압축 효율성을 검증 하였다. 연구에서는 사전 기반의 접근방식을 사용하였고, 사전을 제작하는 방법으로는 데이터에서 의미 있는 지식을 추출하는 데이터 마이닝 기법 중 K평균 군집화 기법을 이용하였다. 연구의 궁극적인 목적은 의미 있는 지식을 추출하는 것이 아닌 전체 데이터를 가장 잘 대표할 수 있는 엔트리를 추출하는 것이다. 따라서 공간 데이터의 일반적인 특성을 이용하여 디퍼런셜 벡터들을 길이와 각도로 나누어 K평균 군집화를 수행 하는 방법을 적용하였다. 기존에 제시된 CBC(Clustering-Based Compression) 방법과 비교 실험을 한 결과, 높은 위치 정확도가 요구되는 즉 왜곡의 정도가 육안으로 쉽게 파악되는 대상에 대해서는 CBC 방법보다 더 향상된 압축률을 보이는 것으로 나타났다. 하지만 위치 정확도가 덜 요구되는 대상에 대해서는 그다지 좋은 성능을 보이지 못하는 문제점이 나타났다. 또한 상대적인 좌표를 나타내는 디퍼런셜 벡터를 이용하여 군집화를 하기 때문에 인접 객체와 공유하는 동일한 하나의 점을 두 개로 나뉘는 경우가 발생하였다. 이는 곧 위상 정보의 변형이 가해지므로 위상 정보의 손실이 발생하게 되는 것이다. 이러한 문제점에도 불구하고 기존에 제시된 CBC 방법에 비하여 좋은 압축 성능을 보이고 위치 정확도를 높이는데 더욱 유연하게 작동이 가능하다. 또한 위치 정확도를 계속 높이면서 실험한 결과 기존의 방법을 적용하였을 때 보다 데이터의 증가 속도가 낮은 것을 확인하였다. 따라서 제시된 방법을 적용한다면 위치 정확도가 요구되는 공간 데이터의 압축에 효율적으로 사용 할 수 있을 것이다. 제시된 방법을 적용 하는 데에 걸리는 압축 시간과 복원 시간에 대해서는 차후 연구과제로 남겨두었다.

참고문헌

[1] Shashi Shekhar, Yan Huang, Judy Djugash, Changqing Zhou, "Vector Map Compression: A Clustering Approach", Proceedings of the tenth ACM international symposium on Advances in geographic information systems, 2002, pp. 74-80.

[2] Clarke, Keith C., Analytical and computer Cartography, Practice-Hall, 1990.

[3] Weibel, R. "An adaptive methodology for automated relief generalization ," Proceedings, AUTOCARTO 8, Eighth International Symposium on Computer-Assisted Cartography, Baltimore, MD, 1981, pp. 42-49.

[4] Douglas, D. H., Peucker, T. K., "Algorithms for the reduction of the number of points requred to represent a digitized line or its caricature", Canadian Cartographer, vol. 10, 1973, pp. 110-122.

[5] D. Salomon. Data Compression: the Complete Reference. Springer-Verlag, 2nd edition, 2000.

[6] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IRE Trans. Electronic Computers, Vol. EC-10, 1961, pp. 260-268.

[7] C. C. Lu and J. G. Dunham. "Hightly Efficient Coding Schemes for Contour Lines Based on Chain Code Representations",. IEEE Transactions on Communications, 39(10), 1991, pp. 1511-1514.

[8] David Beddoe, Paul Cotton, Robert Uleman, Sandra Johnson, Dr. John R., Herring, "OpenGIS Simple Features Specification for SQL Revision 1.1", OpenGIS Consortium., 1999.

[9] Macqueen, J. "Some methods for classification and analysis of multivariate observations.", In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281-297.

[10] Jiawei Jan, Micheline Kanber, Datamining comcepts and Techniques, Morgan Kaufmann, 2000.

[11] Anil K. Jain, M. Narasimha Murty, Patricia J Flynn, "Data Clustering: A Review", ACM Computing Surveys, Volume 31, Issue 3, 1999, pp. 264-323.

이동헌

2003년 인하대학교 지리정보공학과 졸업(공학사)

2005년 인하대학교 대학원 지리정보공학과 졸업(공학석사)

2005년 ~ 현재 삼성SDS

관심분야 : Spatial Databases, Telematics, Vector Data Compression, Computational Geometry

전우제

2002년 인하대학교 지리정보공학과 (공학사)

2005년 인하대학교 지리정보공학과 (공학석사)

2005년~현재 삼성SDS

관심분야 : Spatial Databases, Telematics, Vector Data Compression, Computational Geometry

박수홍

1989년 서울대학교 지리학과 졸업 (학사)

1991년 서울대학교 대학원 지리학과 졸업(석사)

1996년 Univ. of South Carolina 졸업(박사)

1996년~1997년 Indiana University, Research Associate

1998년~2000년 서울시정개발연구원 지리정보연구센터 연구위원

2000년~현재 인하대학교 공과대학 지리정보공학과 (조교수)

관심분야 : Spatial Databases, Telematics, Vector Data Compression