

지리정보 표현 및 압축 방법에 대한 비교

이효종*, 우승용*

*전북대학교 전자정보공학부

e-mail: hlee@chonbuk.ac.kr, psywoo@hanmail.net

Comparison of Compression Methods for Geological Information

Hyo-Jong Lee*, Seung-Yong Woo*

*Division of Electronics & Information Engineering, Chonbuk University

요 약

지리 정보는 여러 분야에서 활용되고 그 데이터양도 방대하여 효율적으로 저장하여 관리해야 할 필요가 있다. 본 논문은 균등분할 방식과 비균등분할 방식에 의한 두 가지 지리정보 표현 및 압축방법을 수행하는 연산회수와 자료의 효율성 등을 중심으로 비교하였다. 두 가지 방법 모두 효율적인 활용이 가능하며 상황에 따라 기본 방법에서 수정을 가하여 사용할 수도 있다.

1. 서론

국토를 개발하고 관리하는 데에는 많은 정보들이 필요하다. 이에 많은 국가들은 오래전부터 지리 정보 시스템(GIS : Geographic Information System)을 구축하여 활용하고 있으며, 또한 효율적인 지리정보 획득을 위해 여러 가지 기술들을 개발하고 있다. 특히 최근에는 날씨에 영향을 받지 않고 3차원 정보를 쉽게 획득할 수 있는 LiDAR(Light Detection and Ranging)기술이 각광을 받고 있다. 이러한 데이터들은 양이 방대하기 때문에 자료를 효율적으로 저장해야 한다. 따라서 국내에서는 균등분할 쿼드트리(Quad Tree)[1]에 의하여 3차원 정보의 효율적 표현을 위한 자료구조 연구가 수행되었으며, 이와 유사하게 비균등분할 방식에[2] 의하여 LiDAR자료로부터 쿼드트리를 구성하는 방법이 발표되었었다. 균등분할쿼드트리 방식은 일정영역을 설정 기준을 만족할 때까지 계속하여 정확하게 4분할하여 나가는 방식을 의미한다. 이에 반하여 비균등분할 쿼드트리 방식은 동일한 공간정보를 표현하지만 영역을 균등하게 분할하지 않고 수집되는 자료를 기준으로 비균등하게 영역을 분할하여 나가는 방식이다. 이 두 가지 방법의 효율과 장단점을 비교분석할 필요가 있으나, 아직까지 특별한 비교방법이 이루어지지 않았다. 본 논문에서는 두 가지 방법의 구동 원리를 대조 설명하였다. 또한 각 알고리즘을 비교 분석하고 각 알고리즘의 성능을 정량적으로 고찰하였다.

2. 표현 방법 및 압축 알고리즘

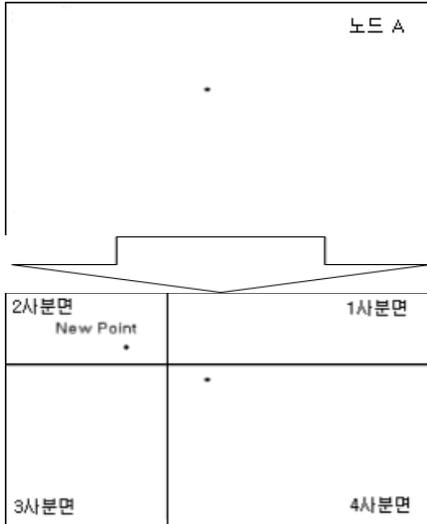
원시 LiDAR 데이터는 평면좌표를 표현하는 X, Y와 높이 Z, 반사율 정보 I가 텍스트 형식으로 주어진다. 이 데이터들의 순서는 규칙적이지 않아 데이터 검색 및 처리에 용이하지 않다. 따라서 이를 처리하기 쉬운 구조로 변환해야 한다. 두 논문에서는 공통적으로 자료를 쿼드트리로 표현한다. 쿼드트리는 구축하는 방법에 따라 약간씩 차이가 있을 수 있다.

예를 들어 비균등분할 쿼드트리 방법에서는 다음과 같은 조건을 만족하도록 트리를 구축하였다.

1. 각 노드들은 영역을 지정한다.
2. 실제 점 정보들은 리프 노드에 저장이 된다. 따라서 리프 노드는 영역 정보와 점 정보를 모두 갖는다. 이 점의 위치는 지정하는 영역 안에 포함된다.
3. 자식노드들의 지정하는 영역은 그 부모노드가 지정하는 영역에 포함되며 4개의 자식노드들이 지정하는 영역들의 합집합은 그 부모노드가 지정하는 영역과 같다.

위와 같은 조건을 만족시키기 위해 루트 노드는 지리정보의 전 영역을 지정하게 된다. 이후에는 텍스트 형식의 점 데이터가 하나씩 순차적으로 트리에 삽입된다. 따라서 첫 번째 점은 루트노드가 저장하게 된다. 이후 점 데이터들은 트리 구조를 따라 내려가면서 삽입되어야 할 위치를

찾는다. 삽입될 점 위치를 포함하는 종단노드에 데이터를 삽입하게 되는데, 이 때 그 노드가 이미 점 정보를 저장하고 있다면 자식 노드들을 생성하여 영역 분할을 한다. 분할 점은 노드가 원래 저장하고 있던 점과 새로이 삽입될 점을 연결한 선분의 중점이다. 분할 점을 원점으로 삼고 좌표평면을 그려서 1사분면, 2사분면, 3사분면과 4사분면에 해당하는 영역을 각각 0, 1, 2, 3번째 자식 노드가 지정하게 된다. 그림 1은 영역 분할 과정을 도식화한 것이다.



(그림 1) 비균등분할 쿼드트리의 영역분할 방법

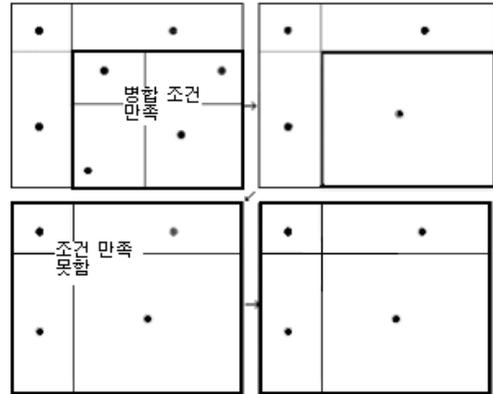
노드 A가 그림처럼 분할되고 나면 원래 영역과 점 정보를 저장하던 노드 A는 점 정보를 자식 노드 (그림에서는 3번 자식노드)에게 넘겨주고 자신은 영역만을 지정하게 되고 삽입된 점은 위치에 따라 저장할 곳이 정해진다. 이렇게 만들어진 트리구조는 데이터 삽입 순서에 따라 균형적일 수도 있고 매우 불균형적일 수도 있다. 이에 데이터 삽입 순서를 조정하여 균형을 높이는 작업이 도움이 된다. 이와 같이 구조화된 데이터를 압축할 수 있는 방법은 다음과 같이 설명할 수 있다.

1. 비균등분할 쿼드트리 방법

압축대상은 쿼드트리로 표현된 지리정보이다. 압축알고리즘은 한 번에 두 노드만이 합쳐질 수 있는 방법에서 부모노드가 같은(i.e. 형제 노드) 노드들이 한꺼번에 합쳐질 수 있도록 수정을 하였다.

이 노드들 중 최고점과 최저점의 차이가 threshold보다 작다면 이 노드들은 병합되어 부모노드들에게 흡수된다. 따라서 이 노드들은 삭제되며 부모노드가 점 데이터를 갖

게 되는데 X, Y, Z, I 정보는 병합된 노드의 평균값을 갖게 된다.



(그림 2) 비균등분할 쿼드트리의 압축 과정

(그림 2)는 비균등분할 트리의 압축 과정을 표현한 것이다. 위 과정이 리프노드부터 반복 수행된다.

2. 균등분할 쿼드트리 방법

비균등분할 쿼드트리의 방법과는 반대로 쿼드트리를 만들어가는 방법이다. 한 영역은 높이 값을 가지며 이 값은 그 영역 안에 속하는 모든 점의 높이 값의 평균이다. 우선 영역 내의 모든 점의 평균높이와 표준편차를 구한다. 표준편차가 정해진 임계값을 초과한다면 영역을 4등분한다. 이때에도 나누어진 영역은 높이 값으로 그 영역 안에 속하는 높이의 평균값을 가진다. 이렇게 표준편차가 임계값을 초과하는 곳이 없어질 때까지 반복한다.



(그림 3) 균등분할 쿼드트리의 구축방법

3. 두 방법의 비교 분석

3.1 수행시간

수행시간을 보자. 우선 원 데이터(압축률 0%)를 생성하는 연산 횟수를 비교하자. 오버헤드나 포인터로 연결된 노드를 참조하는 등의 연산은 제외하였으며 단순히 비교, 산술연산 횟수만을 구하였다.

1) 비균등분할 퀴드트리 방법

퀴드트리의 구축과정과 압축과정을 구분하여 살펴볼 필요가 있다. 우선 퀴드트리를 구축하는 과정은 데이터 점을 차례대로 트리에 삽입함으로써 구현할 수 있다. 한 점을 트리에 삽입하는 작업을 생각해보자. 우선 루트부터 비교를 하면서 리프노드까지 내려간다. 내려가면서 X, Y좌표를 분할 점과 비교를 하여 경로를 찾는다. 2개의 비교연산이 트리의 깊이(0부터 시작)만큼 수행되는 것이다. 트리의 평균 깊이는 $\log_4[\text{트리의 리프노드 수}]$ 이다. 따라서 총 점의 개수가 n이라고 하면 수행되는 비교연산의 횟수는 다음과 같이 표현할 수 있다.

$$2[\lceil \log_4 1 \rceil + \lceil \log_4 2 \rceil + \lceil \log_4 3 \rceil + \dots + \lceil \log_4 (n-1) \rceil] = 2[0 + 4 \times \frac{3}{4} \log_4 4 + 4^2 \times \frac{3}{4} \log_4 16 + \dots + (n-1) \times \frac{3}{4} \log_4 (n-1)] = \frac{6}{4} \times [4 \times 1 + 4^2 \times 2 + \dots + (n-1) \log_4 (n-1)]$$

이는 $2 \times n \log_4 n$ 보다 작다. 영역분할이 일어날 경우의 오버헤드를 더하여 $2 \times n \log_4 n$ 라고 할 수 있다. 압축을 하지 않을 경우 연산은 여기서 끝나게 된다. 즉, 이 방법은 압축을 전혀 하지 않을 경우 수행시간이 제일 짧다.

2) 균등분할 퀴드트리 방법

균등분할 퀴드트리 방법은 노드의 생성을 재귀적으로 구현한다.

```

트리생성(노드)
{
    영역안의 모든 점 높이 평균 계산
    영역안의 모든 점 표준편차 계산
    if(표준편차가 threshold보다 크면) {
        영역분할
        트리생성(영역 0 노드)
        트리생성(영역 1 노드)
        트리생성(영역 2 노드)
        트리생성(영역 3 노드)
    }
    노드의 높이 = 평균 값
    return 0
}
    
```

트리가 균형적이라고 가정하자. 루트에서부터 레벨별로 평균과 표준편차를 구해나간다. 처리하는 레벨이 l이라고 한다면 처리해야하는 (분할된) 영역은 4^l 개이며, 영역별로 처리해야하는 점의 수는 $\frac{n}{4^l}$ 이다. 따라서 l번째 레

벨에서 평균값을 구하기 위하여 덧셈 $\frac{n}{4^l} \times 4^l = n$ 번,

나눗셈 4^l 번의 연산이 필요하다. 표준편차는

$$\sqrt{\frac{\sum_{i=1}^n (z_i - \mu)^2}{n}}$$

이므로 표준편차를 구하기 위해서는

덧셈 $\frac{n}{4^l} \times 4^l = n$ 번, 뺄셈 $\frac{n}{4^l} \times 4^l = n$ 번, 제곱연

산 $\frac{n}{4^l} \times 4^l = n$ 번, 나눗셈 4^l 번, 제곱근 연산이 4^l

번 필요하다. 또한 표준편차를 임계값과 비교하기 위해

4^l 번의 비교연산이 필요하다. 이를 정리하면 l번째 레벨

에서 산술연산이 $4n + 3 \times 4^l$ 번, 비교연산이 4^l 번 수행되는 것이다. 이러한 연산이 전 레벨에 걸쳐 수행된다. 따라

서 종합적으로 산술연산이 $\sum_{l=1}^{depth} [4n + 3 \times 4^l]$ 번, 비교

연산이 $\sum_{l=1}^{depth} 4^l$ 번 수행된다.

3) 비교

앞서 설명했듯이 비균등분할 퀴드트리 방법에서는 압축을 전혀 하지 않을 때 수행시간이 가장 짧다. 비교 연산이

약 $2 \times n \log_4 n$ 만큼 수행되고 반대로 균등분할 퀴드트리방법에서는 압축을 전혀 하지 않을 때 수행시간이 가장

길다. 이때를 비교해 보면 균등분할 퀴드트리방법의 수행 시간이 더 길다. 다음 표는 데이터의 개수 n에 따른 두 가지 방법의 연산횟수를 비교한 것이다.

<표 1> 연산횟수의 비교

n	비균등분할 퀴드트리	균등분할 퀴드트리
100000	1660964	3721924
1000000	19931569	43863133
10000000	232534967	5050699929

<표 1>에서 보듯이 균등분할 퀴드트리 방법의 경우 데이터의 개수가 늘어날 때마다 필요한 연산의 횟수가 급격히 증가한다. 따라서 데이터 정보가 많을 때에는 부적절할 수 있으나 데이터의 수가 작고 어느 정도 압축된 결과를 얻을 경우에는 비균등분할 퀴드트리방법보다 빠를 수가 있다.

3.2 저장 공간

저장 공간으로써의 자료의 효율성을 비교해보면 데이터 정보만을 저장하는 균등분할 퀴드트리와는 달리 비균등분할 퀴드트리 방법에서는 노드마다 영역을 지정하기 때문

에 자료의 크기가 상당히 크다. 즉, 균등분할 쿼드트리 방법이 효율적이다.

4. 실험 및 결과

<표 2>는 비균등분할 쿼드트리 방법으로 특정 지역을 처리한 실험 결과이고 <표 3>은 논문 [장영운, 김두용, 최연웅, 조기성, 2008, 3차원 정보의 효율적 표현을 위한 자료구조 연구]의 내용을 인용한 것이다.

<표 2> 비균등분할 쿼드트리 방법의 결과

threshold(m)	압축률	오차(cm)
0.1	23%	1.1
0.2	42%	2.7
0.3	53%	4.3
0.4	59%	5.9
0.5	63%	7.5
0.6	65%	9.2
0.7	67%	11.0
0.8	69%	12.8
0.9	70%	14.5
1.0	71%	16.3

<표 3> 균등분할 쿼드트리 방법의 결과

threshold(m)	압축률	오차(cm)
0.1	9%	0.5
0.2	18%	1.3
0.3	23%	2.1
0.4	26%	3.1
0.5	30%	3.3
0.6	33%	5.8
0.7	35%	7.1
0.8	37%	8.2
0.9	39%	10.0
1.0	40%	10.9

<표 2>와 <표 3>은 두 방법에 따른 결과를 나타내고 있다. 압축률은 다음과 같이 정의할 수 있다.

$$\text{압축률} = \left(1 - \frac{\text{압축된 데이터}}{\text{원본 데이터}}\right) \times 100(\%)$$

오차를 구한 방법은 두 가지 방법이 서로 다르다. 비균등분할 쿼드트리 방법에서는 다음의 식을 적용하였다.

$$\text{오차} = \sqrt{\frac{\sum_{i=1}^n (\text{원래의 높이} - \text{압축된 후의 높이})^2}{n}}$$

균등분할 쿼드트리 방법에서는 단순히 압축 전과 압축 후 표고변동의 평균을 구한 값이다.

5. 결론

지리정보를 가장 일반적으로 표현하는 비균등분할 쿼드트리와 균등분할 쿼드트리의 표현 및 압축 방법에 대해 비교하였다. 실험결과 비균등분할 쿼드트리방법에서는 트리의 균형성이 상당히 큰 영향을 미쳤다. 최초 데이터의 순서는 많은 구간에서 한쪽으로 증가하는 경향을 보여 깊이가 수백 개에 달하는 매우 불균형적인 트리를 생성하였다. 이에 우리는 균형적인 트리를 구축하기 위하여 데이터의 순서를 무작위로 섞는 간단한 루틴을 이용하여 데이터의 순서를 바꾸었다. 이 작업으로 트리의 균형성이 상당히 향상되었으며, 이 처리된 데이터로 트리를 구축한 결과 트리의 깊이(루트노드의 레벨은 1로 정의)는 고작 24였다. <표 2>의 결과는 처리 후의 결과이며 압축률이 매우 우수함을 알 수 있다. 불균형적인 트리에서는 균형적인 트리의 압축률에 절반도 미치지 못했다.

비록 비균등분할 쿼드트리 방법의 압축률이 균등분할 쿼드트리 방법의 압축률보다 높게 나타나나 실험 지역이 다르고 데이터의 효율성이 균등분할 쿼드트리가 높기 때문에 어느 방법이 특별히 우수하다고 할 수 없다. 오차 또한 계산한 방법이 다르기 때문에 수치만을 가지고 비교하기에는 무리가 있다. 두 방법 모두 높은 압축률에 비교적 작은 오차를 보여주고 있다.

하지만 3절에서 살펴보았듯이 수행시간 면에서는 데이터가 크면 클수록 연산 횟수가 상대적으로 적은 비균등분할 쿼드트리 방법이 유리하다고 할 수 있으며 데이터의 효율성 면에서는 균등분할 쿼드트리 방법이 우수하다고 할 수 있다. 그리고 자료 성질에 따라 여러 가지 방법으로 이 기본 알고리즘을 수정하는 시도가 가능하며 좀 더 나은 결과를 얻을 수도 있을 것이다.

이 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(R01-2007-000-20577-0)

참고문헌

- [1] 장영운, 김두용, 최연웅, 조기성, 2008, 3차원 정보의 효율적 표현을 위한 자료구조 연구
- [2] 이효중, 우승용, 조기성, 2008, LiDAR 데이터의 Quad Tree 구조 표현과 압축에 관한 연구, 대한전자공학회 학술대회, pp. 753-754
- [3] 김대식, 항공레이저 매핑시스템 LiDAR와 GPS 기술의 응용, 한국관개배수, 제 10권 1호, pp 103-111
- [4] Hanan Samet, Data Structures for Quadtree Approximation and Compression, Communication of ACM, Vol 28, No 9, pp 973-993, 1985