

# 최근점 쌍을 이용한 벡터 맵 디지털 워터마킹

## (A Blind Vector Digital Watermarking for GIS using the Closest Pair of Points)

김 정 업 <sup>†</sup>                      박 수 흥 <sup>††</sup>  
(Jungyeop Kim)                      (SooHong Park)

**요 약** 본 연구에서는 벡터 데이터의 저작권 보호를 위한 디지털 워터마킹 기법을 제안하였다. 제안한 방법은 벡터 데이터를 구성하고 있는 포인트들의 최단 거리를 구하고 그 거리를 이용하여 워터마크를 삽입하도록 하였다. 워터마크를 삽입한 실험 데이터에 다양한 공격을 한 후 워터마킹 방법의 강인성을 테스트하였다. 실험 결과 제안한 방법은 기존 방법보다 강인함을 나타내었다. 그리고 워터마크를 삽입한 후에도 위상이 변하지 않아 제안한 방법은 GIS환경에서의 벡터 데이터를 고려한 방법이 될 수 있음을 알 수 있었다.

**키워드** : 디지털 워터마킹, GIS, 벡터 맵 데이터

**Abstract** This paper proposed a novel vector digital watermarking method to protect copyright. The proposed method embeds watermarks after finding the closest pair of points and calculating the distance of the points. We tested the robustness of the method through several attacks on watermarked data. The experimental results show that the proposed method has more robustness than previous methods. And the new method doesn't change the topology of the vector data. Therefore, this method can be the vector digital watermarking for GIS.

**Key words** : Digital Watermarking, GIS, Vector Map data

### 1. 서 론

급격한 기술 발전과 멀티미디어와 네트워크의 넓은 사용과 더불어, 디지털 데이터 보호에 관한 관심도 매우 증대되고 있다. 보통 디지털 데이터의 제작은 많은 시간과 비용이 필요하나 복제는 짧은 시간에 비용이 거의 들지 않는다[1]. 따라서 저작권을 보호하기 위한 장치가 반드시 필요하다[2]. 이를 위해 여러 연구들이 있어왔으며, 그 중에서 디지털 워터마킹은 불법적인 복제와 조작으로부터 보호할 수 있는 기술로 많은 관심을 받고 있

다[3]. 디지털 워터마킹이란 디지털 멀티미디어 데이터의 저작권 보호를 위해 특별한 형태의 워터마크를 디지털 데이터의 변화 없이 삽입하고 추출하는 모든 기술적 방법을 지칭한다. 이러한 디지털 워터마킹은 삽입한 워터마크를 추출할 때, 원본의 사용 여부에 따라 non-blind 방식과 blind 방식으로 구분할 수 있다. 워터마크를 추출할 때 원본이 필요한 방법은 non-blind 방식이라고 칭하며, 원본이 필요 없는 경우는 blind 방식이라고 한다. 그림 1과 2는 원본이 필요한 non-blind 방식의 디지털 워터마킹을 보여주고 있다.

디지털 워터마킹의 특징으로는 워터마크를 삽입한 이후에 워터마크를 검출할 수 있는 능력인 입력 효과성(Embedding Effectiveness), 워터마크를 삽입한 이후에 원본 데이터 질의 훼손 여부를 보는 충실도(Fidelity), 데이터에 다양한 처리를 한 후에도 워터마크를 검출할 수 있는 능력인 강인성(Robustness), 워터마크를 삽입하지 않은 데이터에서 워터마크가 검출되는 긍정적 오류율(False Positive Rate)의 특징을 가지고 있다[3].

디지털 워터마킹의 특징들은 서로 절충되며 상반되는 성격을 나타내기도 한다. 즉, 하나의 특징을 강조하게

<sup>†</sup> 학생회원 : 인하대학교 지리정보공학과  
jyfloo@empal.com

<sup>††</sup> 정 회원 : 인하대학교 지리정보공학과 교수  
shpark@inha.ac.kr  
논문접수 : 2009년 1월 14일  
심사완료 : 2009년 9월 23일

Copyright©2009 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 정보통신 제36권 제6호(2009.12)

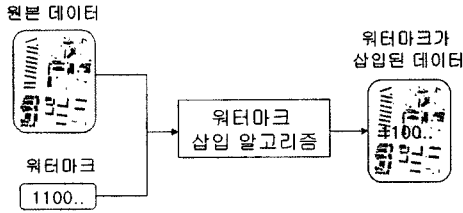


그림 1 워터마크 삽입 과정

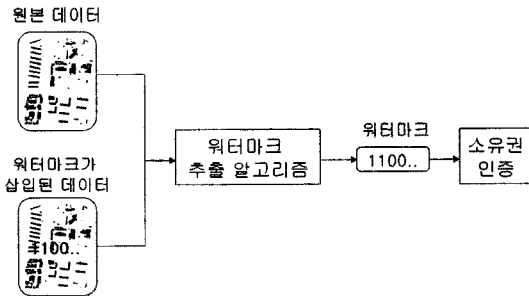


그림 2 워터마크 추출 과정(원본을 이용한 경우)

되면 다른 특징이 감소하는 것이다. 따라서 모든 가능한 공격에 대해 강인하면서 많은 정보를 담고 있는 워터마크를 삽입하거나 모든 공격에 대해서 워터마크를 검출할 수 있는 방법은 현실적으로 어려울 수밖에 없다[1, 2]. 이러한 어려운 점들을 극복하기 위해 디지털 워터마킹 연구가 많이 이루어지고 있다.

하지만, 다양한 데이터에 대한 디지털 워터마킹 연구가 있는 반면에 벡터 맵 데이터를 위한 연구는 많이 부족한 상태이다. 벡터 맵 데이터는 GIS에서 많이 사용되고 있는 데이터 형태로 벡터 맵 데이터를 생산하고 유지하는 것은 매우 많은 시간과 비용이 소요된다[4-6]. GIS는 하드웨어, 소프트웨어, 데이터, 전문 인력으로 구성되는데, 이 중에서 데이터 획득은 GIS 구성 요소 중에서 많은 부분을 차지하고 있으며, 때로는 전체 예산의 80%를 차지하기도 한다[7]. 특히, GIS 데이터에서 포인트 좌표의 획득은 높은 정밀도를 지니는 기구를 통한 측량이나 항공사진 분석, 벡터라이징 등과 같은 작업에 의해 얻어지는 것이기 때문에 매우 많은 비용과 노력이 들며, 다른 데이터에 비해 더 높은 가치를 지니고 있다.

이렇듯 벡터 맵 데이터는 높은 가치를 지니고 있으며 데이터 생성과 유지에 있어 비용이 많이 소요되므로 벡터 맵 데이터를 무료로 사용하기에는 무리가 있다. 따라서 벡터 맵 데이터에 대한 소유권 보호를 할 수 있는 벡터 맵 데이터의 디지털 워터마킹이 절실히 필요하다[4-6,8,9].

이를 위해 본 연구에서는 벡터 맵 데이터의 저작권을 보호하기 위한 새로운 디지털 워터마킹 방법을 제안하

고자 한다. 본 연구의 방법은 벡터 데이터를 사용할 때 쉽게 변하지 않는 값을 이용하였다. 쉽게 변하지 않는 값을 상대적인 값이라고 할 수 있으며, 벡터 데이터에서 이러한 값은 두 포인트의 거리가 대표적이라 할 수 있다. 벡터 맵 데이터는 좌표의 위치 정확도를 중요시 여기는 데이터이므로 좌표의 위치를 변화시키는 일은 거의 이루어지지 않는다. 따라서 두 좌표의 거리를 이용하는 것은 벡터 맵 데이터를 위한 디지털 워터마킹에 있어서 매우 좋은 정보가 될 수 있다. 본 연구에서는 이러한 정보를 이용하여 새로운 워터마킹 알고리즘을 개발하였다. 제안하는 방법의 효율성을 검증하기 위하여 여러 가지 공격에서도 워터마크가 살아남는지 실험적으로 평가하였으며, 워터마크를 삽입한 이후의 왜곡 현상에 대해서도 평가하였다.

## 2. 관련 연구

벡터 데이터를 위한 기존의 디지털 워터마킹에 관한 연구는 크게 두 가지로 분류를 할 수 있다. 그 중 주파수 영역에서의 방법은 데이터를 주파수 영역으로 변환하여 워터마크를 삽입하고 검출을 하는 방법이다. 주파수 영역에서 기존의 디지털 워터마킹 방법은 주로 DFT, DWT, DCT의 변환을 이용하였다. 이러한 주파수 영역에서의 방법은 일반적으로 공격에 강인하다고 알려져 있지만, 워터마크 삽입으로 인해 데이터의 정확도가 떨어지는 경우 이를 수정하기가 어려운 점이 있다. 주파수 영역의 디지털 워터마킹 연구 방법 중에서 [4,10]은 좌표를 새로운 값(complex value =  $x+iy$ )로 바꾸고 그 바꾼 값을 이용하여 주파수 영역에서 워터마크를 삽입하는 방법을 제안하였다. 이들의 방법들은 워터마크를 삽입하면 원본 데이터와 많은 차이가 나며, 특정한 공격에서만 워터마크를 검출할 수 있다는 단점이 있었다. [11]은 이와 달리 complex value를 DWT로 주파수 변환을 하고, 그 후에 변환 계수의 변화를 통해 새로운 값의 좌표를 가진 벡터 맵 데이터를 만들었다. 이들 방법은 일부 공격에서는 워터마크 검출이 가능하였으나 모든 공격에서 워터마크를 검출하지 못하는 문제점을 지니고 있었다. [6]은 원본이 필요 없는 blind 방식의 워터마킹 기법으로 DCT 방법을 이용해 워터마크를 삽입하였다. 가까이 있는 좌표들은 유사한 특성을 갖는다는 점을 이용해 계수의 상관관계를 비교하여 워터마크를 삽입하는 방법을 소개하였다. 이 방법 또한 특정 공격에서만 워터마크를 검출하는 문제점을 지니고 있었으며, 워터마크를 삽입하였을 때, 원본 데이터와 많은 왜곡현상을 나타내었다.

이와는 달리 공간 영역에서 워터마크를 삽입하는 방법에 대한 연구가 있었다. 공간 영역에서의 워터마킹은

주파수 영역에서의 워터마킹보다 다양한 공격에 약하다고 알려져 있으나 왜곡 현상을 통계하기가 쉽다는 장점을 지니고 있다. [8,12]는 벡터 맵 데이터를 일정한 구역으로 나누고, 그 나뉜 구역의 좌표에 PRNS(Pseudo Random Number Sequence)를 이용하여 워터마크를 삽입하도록 하였다. 이렇게 좌표에 직접적으로 변화를 시킴으로써 워터마크가 삽입된 데이터를 생성할 수 있었다. 그리고 유사하게 [5]는 벡터 맵 데이터를 일정한 구역으로 나누지만, 구역 내에 있는 좌표들의 통계적인 값들을 이용하여 워터마크를 삽입하고 추출할 수 있도록 하였다. 이들 방법들은 모두 특정한 공격에서만 워터마크가 검출이 되었으며, 워터마크를 삽입하면 원본 데이터에 손상이 나타났다. 이는 GIS에서 다양한 데이터 조작을 한 경우 워터마크를 검출할 수 없는 단점과 함께 위상관계 변화라는 단점을 지니게 되는 것이다. [13]은 Douglas-Peucker 알고리즘에 의해 특정 좌표들을 추출하고, 그 좌표들의 상관관계에 따라 좌표에 워터마크를 삽입하는 방법을 이용하였다. 이 방법은 Douglas-Peucker 방법이 아닌 일반화 알고리즘에 굉장히 약하며, 회전이나 원점이동 등의 공격에도 매우 약한 모습을 보여주었다.

기존의 연구들을 살펴보면 모든 연구들이 특정 공격에서만 워터마크를 검출할 수 있다는 단점을 지니고 있었다. 게다가, GIS에서 벡터 맵 데이터를 사용하는데 중요한 정보인 위상 관계를 고려하지 않았다. 즉, 워터마크를 삽입한 후에 위상 관계가 변하거나 원본과의 차이가 심한 경우가 많이 발생하였다. 따라서 본 연구에서는 다양한 공격에서 강인함을 가지며, 충실도를 보장할 수 있는 벡터 맵 데이터 디지털 워터마킹 방법을 제안하고자 한다.

### 3. 워터마크 삽입 방법

본 연구에서 제안하는 방법은 워터마크를 삽입하기 위해 최근점 쌍을 구하고 그 쌍이 이루는 거리와 일방함수를 이용하여 워터마크를 생성하도록 하였다. 그리고 생성된 워터마크는 두 최근점 쌍에 반영을 하여 워터마크가 반영된 새로운 데이터를 생성하도록 하였다.

#### 3.1 최근점 쌍 찾기

벡터 맵 데이터는 좌표로 구성되어 있다. 따라서 좌표들을 이용하여 두 좌표간의 거리를 계산할 수 있다. 벡터 맵 데이터는 좌표의 위치 정확도가 매우 중요하기 때문에, 좌표를 변화시키는 경우는 그리 많지 않다. 따라서 두 좌표간의 거리는 벡터 맵 데이터에서 일반적인 데이터 조작으로는 쉽게 변하지 않는 값이 된다. 이러한 이유로 두 좌표간의 거리를 활용하면 워터마크 삽입과 검출에서 매우 효과적일 수 있다.

최근점 쌍 찾기는 점집합 원소들 중에서 가장 거리가 가까운 한 쌍의 점을 찾아내는 문제이다. 최근점 쌍을

찾기 위해 본 연구에서는 분할 점령(Divide and Conquer) 방법을 이용하여 최근점 쌍을 찾으려 하였다[14].

이러한 방법을 최근점 쌍 찾기 문제에 적용하는 과정은 다음과 같다.

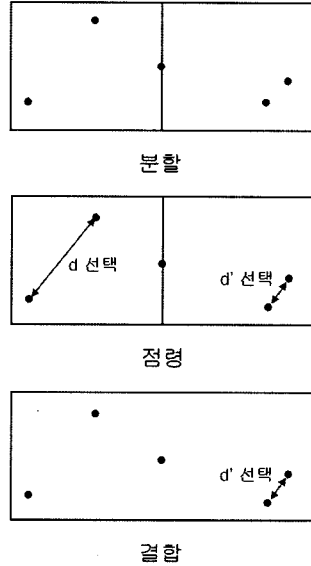
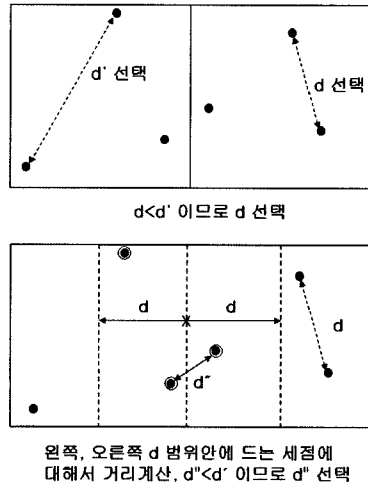


그림 3 분할 점령 방법



왼쪽, 오른쪽 d 범위안에 드는 세점에 대해서 거리계산.  $d < d'$  이므로  $d$  선택

그림 4 두 분할의 경계를 걸치는 최근점 쌍

우선, 점집합 P를 x 좌표를 기준으로 하여 정렬을 한다. 그리고 그 점집합의 중앙을 기준으로 왼쪽과 오른쪽의 두 문제로 분할한다. 그림 3에서의 분할처럼 나누어진 부분에 점이 2개 혹은 3개가 남을 때까지 재귀적으로 반복한다. 최종적으로 각 분할에는 점이 2개 혹은 3개가 포함되게 되고, 이 점들에 대해서 가장 가까운 한

쌍인 최근점을 찾아낸다. 마지막으로 나누어졌던 두 분할에서 제시하는 최근점을 서로 비교하여 더 작은 거리의 최근점을 선택하는 과정을 반복하여 전체적으로 최근점을 구할 수 있다.

하지만, 만일 최근점이 최소 분할 내의 두 점에 있지 않고, 분할과 분할 사이에 존재하는 경우도 있다. 따라서 이러한 경우에는 그림 4에서처럼 두 최소 분할에서 결합되어 제시된 최소의 거리  $d$ 를 이용하여 중앙  $m$ 에서  $d$  간격 안에 드는 점, 즉  $x$ 좌표가  $m-d \leq x \leq m+d$ 인 점집합  $S$ 의 원소들끼리만 가장 가까운 최근점 쌍을 찾아낸다. 이 최근점 쌍의 거리를  $d''$ 라 할때,  $d$ 와  $d''$ 중 작은 것을 되돌려주면 최근점 쌍을 구하게 된다.

**3.2 일방함수**

대부분의 수학 함수는 한 숫자를 다른 숫자로 변화시키기도 쉽고, 이를 역으로 하여 얻은 숫자를 처음 숫자로 되돌리기도 쉽다. 이러한 성질의 함수를 양방향 함수라고 한다. 예를 들면,  $y=3x$ 라는 함수가 있다면,  $x$ 의 값을 이용하여  $y$ 의 값이 쉽게 계산이 되며, 반대로  $y$ 의 값을 알게 되면,  $x$ 의 값도 쉽게 구할 수가 있다. 하지만 해쉬 함수처럼 한쪽 방향으로만 값을 구할 수 있으며, 결과 값을 가지고 입력 값을 구하기가 어려운 함수를 일방 함수라고 한다. 모듈러 수학 분야는 이러한 일방 함수를 다루며, 암호학 분야에서도 이러한 이유로 많이 사용되고 있다.

본 연구에서는 일방 함수 중 나머지 함수를 이용하였다. 나머지 함수는 피제수를 제수로 나누어 나오는 나머지를 반환하는 함수로 나머지 값을 이용하여 제수를 구하는 것이 어려운 함수이다. 또한 나머지로 나오는 값은 제수보다 작기 때문에 일정한 범위 안으로 제한할 수 있다. 이러한 특징을 이용하여 본 연구에서는 워터마크를 생성할 때 나머지 함수를 이용하였다.

**3.3 워터마크 삽입 절차**

워터마크를 생성하고 삽입하기 위해 우선 본 연구에서는 랜덤 테이블을 생성한다. 랜덤 테이블은 워터마크 삽입과 검출시에 워터마크 키로 활용을 할 수 있다. 랜덤 테이블을 워터마크 키로 활용을 하게 되면, 워터마킹 알고리즘이 알려진다고 하더라도 삽입한 워터마크를 역으로 추적하여 훼손하려는 시도를 방지할 수 있다.

랜덤 테이블을 작성하여 적용하는 방법은 다음과 같다. 예를 들어, 소유자가 13이라는 값으로 랜덤 변수 범위를 지정하면, 난수는 0부터 13까지 발생할 수 있다. 이 때, 0부터 9의 값은 0부터 13의 범위를 갖는 난수로 대체하게 된다(표 1). 즉, 실제 최근점 쌍의 거리가 13.1649라면, 67.1649라는 값으로 바꿀 수 있다. 이렇게 바뀐 새로운 값은 워터마크 삽입을 위한 계산에서 사용된다.

표 1 랜덤 테이블

실제 값	0	1	2	3	4	5	6	7	8	9
변환 값	5	6	11	7	0	2	12	5	8	3

이렇게 랜덤 테이블을 이용하여 최근점 쌍의 거리를 변화시킬 때에 한 가지 고려해야할 사항이 있다. 본 연구에서는 두 포인트 간의 실제 거리를 이용한 것이 아니라 가상의 거리를 계산하였다. 만약 실제 두 포인트 간의 거리를 계산하게 되면 굉장히 작은 수로 나타나기도 하며, 대부분의 경우에 있어서는 거리가 무한소수로 나타나게 된다. 예를 들어, 두 좌표가 (3.3, 5.2), (6.1, 7.5)라면, 두 좌표의 거리는 8.01560977...과 같이 무한소수로 나타나게 된다. 따라서 만약 두 점  $A(x', y')$ ,  $B(x'', y'')$ 가 있다면 본 연구에서는 거리 계산을 할 때 루트를 씌우지 않고, 아래의 식처럼 가상의 거리계산을 하도록 하였다.

$$\text{두점간의거리} = (x'' - x')^2 + (y'' - y')^2 \tag{1}$$

위와 같이 두 좌표의 가상의 거리를 변화시키는 과정은 벡터 맵 데이터의 모든 최근점 쌍에 대해서 이루어진다. 즉, 벡터 맵 데이터에서 가장 가까운 두 점의 좌표를 선택하고, 그들 간의 가상의 거리를 랜덤 테이블을 적용시켜 새로운 값으로 변화시킨다. 그리고 그 두 점을 제외하고 나머지 좌표들에서 가장 가까운 두 좌표를 선택하여 가상의 거리를 계산한 후, 다시 랜덤 테이블을 적용한다. 이렇게 새로 변경된 값에서 정수자리의 값은  $P(x)$ 로 취하여 워터마크를 생성하기 위한 값으로 사용한다.

워터마크 생성을 위한 다음 단계는 데이터 사용자의 고유값을 받아들이는 것이다. 본 연구에서는 워터마킹 알고리즘의 보안성을 위해 데이터 사용자의 고유 값을 받아들이 워터마크 삽입과정과 검출과정에서 활용을 할 수 있도록 하였다. 이렇게 사용자의 고유 값을 이용하면 향후에 워터마크를 검출하여 저작권을 증명하려 할 때, 어떤 사용자의 실수로 인해 불법적인 배포가 되었는지 증명할 수 있는 수단이 된다. 사용자가 지정한 고유의 값은  $G(x)$ 로 활용을 하여 워터마크 생성을 위해 계산을 하게 된다. 즉,  $P(x)$ 와  $G(x)$ 를 정하고, 아래 식을 이용하면 워터마크를 생성할 수 있다.

$$G(x) - P(x) \% G(x) = \text{워터마크} \tag{2}$$

워터마크 삽입은 생성된 워터마크를 두 좌표의 가상의 거리에서 소수점 이하를 생성된 워터마크로 대체함으로써 이루어진다. 즉, 실제 두 좌표의 가상의 거리가 37.2105이며, 랜덤 테이블과 식 (1), 식 (2)를 통해 생성된 워터마크가 102라면, 이를 두 포인트의 가상의 거리에 삽입한다. 즉, 워터마크 삽입은 실제 두 포인트 간의

가상의 거리인 37.2105를 37.102로 변경하면서 이루어진다. 이렇게 두 좌표의 거리가 변경이 되면, 결국 두 좌표도 그 가상의 거리 차(0.1085)만큼 이동을 하게 된다. 거리 차만큼의 이동은 두 좌표 모두에게 반영이 되어야 하므로, 거리 차를 반으로 나누어서 각각의 좌표에 적용을 하게 된다. 즉, 0.1085의 절반인 0.05425 만큼 양 좌표를 이동시키게 되는 것이다. 하지만 여기서 실제로 좌표를 이동할 때에는 0.05425라는 값이 루트를 이용하여 계산한 실제 거리 값이 아니기 때문에, 실제로 두 좌표에 적용을 할 때에는 루트를 이용하여 실제 거리를 계산한 후 좌표의 이동이 이루어진다.

위터마크 생성과 삽입 과정은 이와 같이 이루어지며 구체적인 구현 과정은 다음과 같다.

- Step 1. 최근점 쌍의 거리와 랜덤 테이블을 이용하여  $P(x)$ 를 계산한다.
- Step 2. 사용자 고유값을 이용하여  $G(x)$ 를 계산한다.
- Step 3.  $G(x) - P(x)\%G(x)$  연산을 통해 위터마크를 계산한다.
- Step 4. 최근점 쌍의 거리에서 소수점 이하를 위터마크로 변환한다.
- Step 5. 직선의 방정식과 원의 방정식을 이용하여 위터마크가 반영된 좌표를 계산한다.

### 3.4 위터마크 추출 절차

본 연구의 방법은 위터마크 추출을 위해 원본 데이터가 필요로 하지 않는 blind 방법의 위터마킹이다. 그리고 위터마크 삽입시에 사용하였던 랜덤 테이블을 위터마크 키로 활용을 하여 위터마크 검출시에 사용을 하게 된다.

본 연구의 방법에서 위터마크 검출은 기본적으로 위터마크 삽입 과정과 유사하다. 위터마크를 삽입한 데이터에서 가장 가까운 두 좌표를 구하여 가상의 거리를 계산하고, 위터마크 삽입과정에서 사용한 랜덤 테이블을 이용하여 새로운 값으로 변환한다. 그리고 사용자의 고유값을 통해  $G(x)$ 와  $P(x)$ 를 구하고, 식 (2)와 같이 위터마크를 계산한다. 마지막으로 계산된 위터마크가 두 좌표간의 가상의 거리에 반영이 됐는지를 검사한다. 검사 결과 계산된 위터마크가 거리에 반영이 되었다면, 위터마크가 삽입된 것으로 간주할 수 있다.

위터마크 검출의 구현 과정은 다음과 같다.

- Step 1. 위터마크가 삽입된 데이터에서 최근점 쌍을 구한다.
- Step 2. 위터마크 삽입시 사용하였던 랜덤 테이블과 사용자 고유값을 이용하여  $P(x)$ 와  $G(x)$ 를 계산한다.
- Step 3.  $G(x) - P(x)\%G(x)$  연산을 통해 위터마크를 계산한다.

Step 4. 계산된 위터마크가 데이터에 반영이 되었는지를 판단한다.

본 연구에서는 CR(Correspondence Ratio)을 통해 위터마크 검출율을 확인할 수 있도록 하였다. 위터마크 삽입을 위해 기본적으로 데이터 내의 모든 좌표에서 최근점쌍을 구하고 거리를 계산하였다. 즉, 거리를 계산하기 위해서는 두 좌표가 필요하게 된다. 따라서 위터마크 검출시에 사용된 공격당한 데이터의 좌표 개수를 2로 나누는 수가 분모가 된다. 그리고 검출과정을 통해 위터마크 삽입이 됐음을 인식한 거리의 개수가 분자가 되어 CR을 계산할 수 있다.

$$CR = \frac{\text{위터마크가 검출된 거리의 개수}}{\text{공격당한 데이터의 좌표의 개수}/2}$$

## 4. 실험 결과와 분석

### 4.1 위터마크 삽입

본 연구에서는 실험을 위해 벡터 데이터 형태를 가진 shp 파일을 이용하였다. 실험에서 사용한 shp 파일은 1:1,000의 축척을 가진 데이터로 서울 강동구 명일역 근방의 건물 데이터이다. 랜덤 테이블의 범위는 19로 하여 실험을 하였으며, 실험에서 사용한 랜덤 테이블은 표 2와 같다. 그리고 실험에서 사용한 데이터 사용자의 고유값 BGS로 하였고, 이를 아스키코드 값으로 바꿔 합한 값인 220(B(66), G(71), S(83))을  $G(x)$ 로 하여 실험을 하였다.  $P(x)$ 는 최근점 쌍을 구하여 계산된 가상의 거리를 랜덤 테이블을 이용하여 변경하였다.

표 2 실험에서 사용한 랜덤 테이블

실제 값	0	1	2	3	4	5	6	7	8	9
변환 값	16	3	17	15	6	4	8	12	18	10

이와 같이  $P(x)$ 와  $G(x)$ 를 구하고, 식 (2)를 통해 위터마크를 계산할 수 있다. 계산된 위터마크는 원본 데이터에서 최단 거리 쌍의 거리에 반영이 된다. 하지만 위터마크가 삽입된 새로운 거리는 실제 거리가 아니라 루트를 씌우지 않은 가상의 거리이다. 따라서 위터마크가 삽입된 것을 실제 좌표에 반영하기 위해 가상의 거리에 루트를 씌워 실제 거리로 변환을 한다. 예를 들어, 위터마크가 삽입되기 전의 가상 거리가 4.6212이고 실제 거리가 2.149697653 일 때, 위터마크를 삽입하면 가상의 거리가 4.113이고, 실제 거리는 2.028053254가 된다. 결국 두 포인트의 변환 거리만큼을 각 포인트에 반영하여 좌표의 이동이 일어나게 된다. 이와 같이 위터마크가 반영되면 데이터의 품질에 영향을 끼치게 된다. 본 실험에서 사용한 데이터(축척 1:1,000)의 좌표에 대한 위치 정확도의 허용오차는 0.7m이므로, 위터마크가 삽입이 되

어 그 이하에서 변하는 것은 맵의 유효성을 떨어뜨리지 않는다고 할 수 있다. 제안한 방법에서는 워터마크를 삽입하여도 거리의 변화가 최대 1m를 넘지 않는다. 따라서 변화 거리를 각각의 좌표에 반영을 하기 위해 2로 나누게 되면, 최대 0.5m가 되고, 이는 맵 데이터의 위치 허용오차를 넘지 않기 때문에 제안한 방법은 벡터 맵 데이터의 유효성 또한 유지할 수 있다.

워터마크를 삽입한 후 충실도를 살펴보기 위하여 본 연구에서는 RMSE(Root Mean Squared Error)와 좌표의 변화량을 살펴보았다. 실험 결과 RMSE는 0.0855m로 나타났으며, 워터마크를 삽입한 이후의 좌표의 최대 이동 거리는 0.2660m, 최소 이동거리는 0.0000m, 평균 이동거리는 0.0544m로 나타났다(표 3). 따라서 본 연구의 방법은 수치지도의 위치 허용오차를 벗어나지 않고 있음을 확인할 수 있으며, 좌표의 정확도를 유지하고 있다는 것을 알 수 있었다.

위상관계 변화 여부는 교차테스트를 통하여 실험하였다. 교차테스트는 두 선분이 만나는 경우와 교차하는 경우, 일치하는 경우로 구분을 하였다. 워터마크를 삽입하기 전과 삽입한 이후에 두 선분의 교차테스트 결과가 달라지는 것은 워터마크를 삽입하여 위상관계가 변하는 경우이다. 실험 결과 본 연구에서는 위상관계가 변하는 경우는 하나도 없었다. 따라서 거리를 이용한 방법이 위치정확도를 유지시키는 것뿐만 아니라 맵 데이터의 위상관계도 유지하고 있음을 알 수 있었다. 그림 5는 폴리곤 데이터에 워터마크를 삽입하기 전·후의 그림이다.

표 3 워터마크 삽입 이후의 변화량

RMSE	0.0855m
평균 좌표 변화거리	0.0544m
최대 좌표 변화거리	0.2660m
최소 좌표 변화거리	0.0000m
위상 관계 변화경우	없음

4.2 워터마크 추출

제안한 알고리즘에 의해 워터마크를 삽입한 후, 워터마크 검출이 제대로 이루어지는지 실험을 하였다. 워터마크 검출은 아무런 공격을 하지 않고 워터마크 검출을 시도하였을 때 CR 값이 96.38%가 나왔다. 삽입한 워터마크가 100%가 나오지 않는 이유는 워터마크 삽입으로 인해 좌표의 작은 변화가 나타나기 때문이다. 즉, 좌표의 변화로 인해 워터마크를 삽입할 때 찾아낸 최근점 쌍이 아닌 다른 최근점 쌍을 찾아서 검출 과정을 진행하기 때문이다.

실험에서는 삽입한 워터마크가 검출이 이루어지는 지를 알아보는 것과 별도로 디지털 워터마킹의 특징인 긍정적 오류율을 테스트하였다. 긍정적 오류율은 합법적인 데이터 사용자를 불법적인 사용자로 오인하거나 정당한 방법으로 데이터를 구입하였으나 그 데이터를 사용하지 못하게 할 수 있다. 따라서 이러한 긍정적 오류율은 디지털 워터마킹에서 중요한 문제가 되기도 한다. 본 연구에서 제안한 방법에 의한 긍정적 오류율은 0.40%로 나왔다. 이는 실질적인 워터마크 검출율에 비해 매우 적은 수치로서 제안한 방법에서는 긍정적 오류율이 크게 문

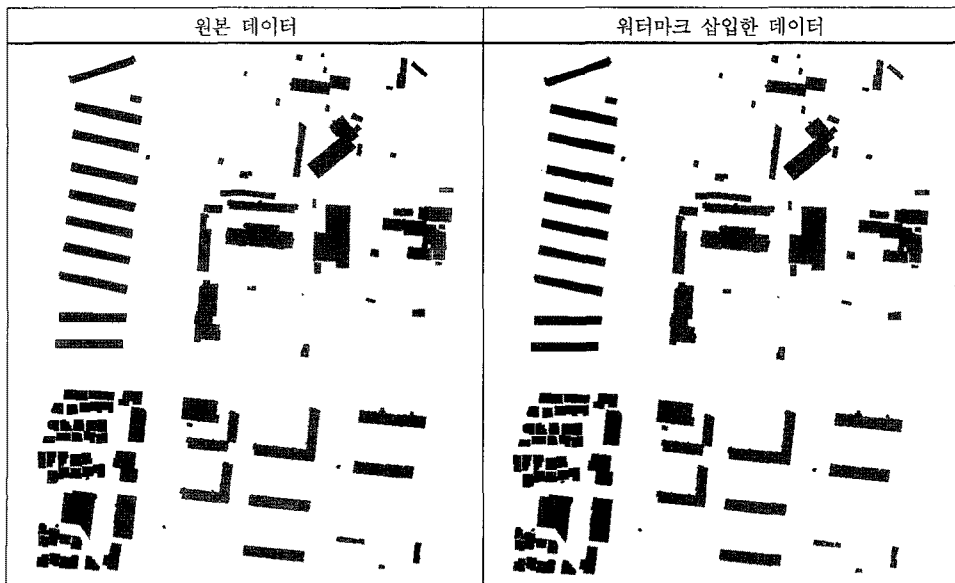


그림 5 원본 데이터와 워터마크를 삽입한 데이터

제가 되지 않는다고 할 수 있다.

**4.3 다양한 공격에 대한 평가**

위터마킹에서 공격이란 데이터를 사용할 때 고의적이든 고의적이지 않던 발생하게 되는 데이터의 조작 행위를 말한다. GIS에서 벡터 데이터에 대한 공격은 geometrical 공격, vertex 공격, noise 공격으로 분류를 할 수 있다[15]. 본 연구에서는 이에 대해 위터마크 검출 여부를 살펴보았다. Geometrical 공격은 회전, 원점 이동, 좌표계 변환 등의 데이터의 포인트 개수에 변화를 주지 않으면서 이루어지는 공격이다. 이러한 공격에 대한 검출율을 살펴보면 표 4와 같다.

표 4 Geometrical 공격에 대한 검출율

공격	공격 범위	검출율
회전	1°	96.38%
	10°	96.38%
	100°	96.18%
원점 이동	1m	96.38%
	10m	96.38%
	100m	96.38%
좌표계 변환	TM↔경위도	96.38%

표 4를 통해 실험 결과를 살펴보면 회전이나 원점 이동에 대한 공격에 대해서는 삽입한 위터마크를 검출하는데 큰 문제가 없음을 알 수 있었다. 즉, 거리는 절대적 수치인 좌표와 달리 계산에 의한 상대적인 값이기 때문에 좌표가 변하더라도 거리의 값이 변하지 않는다면 위터마크를 검출할 수 있다.

다음으로 vertex 공격에 대해 위터마크 검출 여부를 살펴보았다. vertex 공격은 벡터 데이터를 사용할 때 일반화, 객체 추가, 잘라내기, 맵 매칭 등 데이터의 포인트 개수에 변화가 이루어지는 조작이라고 할 수 있다. 본 연구에서 일반화는 Douglas-Peucker 알고리즘을 이용하였으며, 객체 추가는 위터마크가 삽입된 데이터에 새로운 객체들을 추가하였다. 잘라내기는 특정한 지역을 잘라내고 위터마크를 검출할 수 있는지 여부를 살펴보았다. 맵 매칭은 GIS 분야에서 자주 사용되는 조작으로 두 맵을 붙여 하나의 맵으로 사용하는 조작이다. 본 연구에서는 이와 같은 맵 매칭에 대해서도 실험을 하였다.

표 5를 통해 실험 결과를 살펴보면, vertex 공격에 대해서 삽입한 위터마크를 검출하는데 있어서도 큰 문제가 없음을 알 수 있었다. 하지만 이처럼 공격에 따라 검출율이 달라지는 이유는 좌표가 사라지거나 추가되는 경우에는 추가/삭제된 좌표들로 인해 최근점 쌍의 변화가 나타나기 때문이다. 하지만 vertex 추가나 삭제로 인해 최근점 쌍이 연속적으로 변하는 것이 아니라 최근점 쌍을 찾는 과정이 반복되면 결국 원래의 최근점 좌표

표 5 Vertex 공격에 대한 검출율

공격	공격 범위	검출율
잘라내기	1번째	94.67%
	2번째	97.26%
	3번째	91.89%
일반화	1m	84.64%
	2m	83.23%
	3m	83.50%
	4m	86.96%
	5m	85.53%
객체 추가		94.46%
맵 매칭		91.55%

쌍으로 돌아가서 검출이 이루어지게 되므로 검출율이 많이 떨어지는 않는다.

Noise 공격으로는 좌표에 약간의 변화를 주는 공격과 데이터 포맷의 변화를 주는 공격을 하였다[8]. 즉, 실험에서는 원점 이동과 회전에서 맵 데이터의 위치 허용오차를 벗어나지 않는 범위에서의 좌표의 움직임을 noise 공격으로 간주하여 실험을 하였다. 그리고 데이터의 포맷을 변환하는 공격과 데이터의 특정 구역에 있는 객체들마다 임의의 값들로 좌표에 약간의 변화를 주는 공격으로 실험을 하였다.

실험 결과를 보면 기존 연구에서 사용하였던 같은 방식으로 맵 데이터 전체에 약간의 변화를 주는 공격에서는 높은 검출율을 나타내고 있음을 알 수 있었다. 이러한 noise 첨가는 결과적으로 보았을 때 geometrical 공격과 같은 것으로 보이지만, geometrical 공격과는 다른 의미를 지니고 있다. 즉, 이 공격은 삽입된 위터마크를 삭제하기 위하여 좌표에 미세한 변화를 주는 악의적인 공격이라고 할 수 있다. 표 6은 이러한 noise 공격이 거리를 이용한 방법에는 별 영향을 끼치고 있지 않다는 것을 보여주고 있다.

이와는 별도로 데이터의 일부 지역마다 다른 값으로 좌표의 변화를 주는 noise 공격도 실험을 하였다. 즉, 실험 데이터의 전체 영역을 무작위로 나누어 각 영역마다 좌표에 각각 다른 미세한 변화를 주도록 한 것으로 두 번에 걸쳐 실험을 하였다. 실험 결과를 보면 검출율이 약간 떨어지는 것으로 나타나지만 검출율이 매우 높게 나타남을 알 수 있다. 이는 본 연구의 방법이 최근점 쌍을 이용하는 것이기 때문이다. 최근점 쌍은 멀리 떨어진 좌표가 아니라 가까운 좌표와 쌍을 이루도록 한 것이기 때문에 특정 영역마다 다른 변화를 주는 것은 위터마크 검출에 크게 영향을 끼치지 않는다.

벡터 맵 데이터를 사용하는 응용 프로그램마다 다른 형식으로 저장이 되기 때문에 파일 포맷 변환은 삽입된 위터마크에 영향을 끼칠 수 있다. 이러한 이유로 본 연





하는 방법은 거리를 이용한 방법이므로, 만약, 워터마킹 알고리즘이 알려져 모든 최근점 쌍의 거리에 변화를 주는 공격이 이루어진다면, 워터마크 검출율이 낮아질 수 있다. 따라서 한 가지 방법이 아닌 복합적인 방법으로 워터마크를 삽입하고 검출을 할 수 있는 연구가 필요하다. 이러한 보완이 이루어진다면 보다 좋은 벡터 맵 데이터 워터마킹 방법이 될 수 있으며, 벡터 맵 데이터의 소유권을 확실하게 보장할 수 있는 훌륭한 방법이 될 수 있을 것이다.

### 참 고 문 헌

- [1] 고윤호, 2002, "모양적용 이산변화에 기반한 객체 지향 워터마킹 기법과 이계층 워터마킹 기법", 박사학위논문, 한국과학기술원.
- [2] 송학현, 2004, "이산 웨이블릿 변환과 퍼지 추론을 이용한 강인한 영상 워터마킹 알고리즘", 박사학위논문, 목원대학교 대학원.
- [3] 김현승, 디지털 워터마킹, 도서출판 그린, 2005.
- [4] Kitamura, Itaru., Kanai, Satoshi. and Kishinami, Takeshi., "Copyright Protection of Vector map using Digital Watermarking Method based on Discrete Fourier Transform," *Proceedings of the IEEE 2001 International Symposium on Geoscience and Remote Sensing Symposium*, vol.3, pp.1191-1193, 2001.
- [5] Voigt, Michael. and Busch, Christoph., "Feature-based Watermarking of 2D Vector Data," *Proceedings of the SPIE-Security and Watermarking of Multimedia Content*, vol.5020, pp.359-366, 2003.
- [6] Voigt, Michael., Yang, Bian. and Busch, Christoph., "Reversible Watermarking of 2D-Vector Data," *Proceedings of the 2004 Multimedia and Security Workshop on Multimedia and security*, pp.160-165, 2004.
- [7] Carlos, López., "Watermarking of digital geospatial datasets: a review of technical, legal and copyright issues," *International Journal of Geographical Information Science*, vol.16, no.6, pp.589-607, 2002.
- [8] Ohbuchi, Ryutarou., Ueda, Hiroo. and Endoh, Shuh., "Robust Watermarking of Vector Digital Maps," *Proceedings of IEEE International Conference on Multimedia and Expo*, vol.1, pp.577-580, 2002.
- [9] Ohbuchi, Ryutarou., Ueda, Hiroo. and Endoh, Shuh., "Watermarking 2D Vector Maps in the Mesh-Spectral Domain," *Proceedings of International Conference on Shape Modeling and Applications*, pp.216-225, 2003.
- [10] Solachidis, Vassilios., Nikolaidis, Nikos., and Pitas, Ioannis., "Fourier descriptors watermarking of vector graphics images," *Proceedings of the International Conference on Image Processing*, vol.3, pp.9-12, 2000.
- [11] Li, Yuan-Yuan. and Xu, Lu-Ping., "A Blind Watermarking of Vector Graphics Images," *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications*, pp.424-429, 2003.
- [12] Kang, Hwan-Il., Kim, Kab-Il. and Choi, Jong-Uk., "A vector watermarking using the generalized square mask," *Proceedings of International Conference on Information Technology: Coding and Computing*, pp.234-236, 2001.
- [13] Shao, Cheng-Yong., Wang, Hai-Long., Niu, Xia-Mu. and Wang, Xiao-Tong., "A Shape-preserving Method for Watermarking 2D Vector Maps based on Statistic Detection," *Proceedings of IEICE-Transactions on Information and Systems*, vol. E89 -D, no.3, pp.1290-1293, 2006.
- [14] 이재규, C로 배우는 알고리즘, 도서출판 세화, 2000.
- [15] Niu, Xia-Mu., "A Survey of Digital Vector Map Watermarking," *International Journal of Innovative Computing, Information and Control*, vol.2, no.6, pp.1301-1316, 2006.



김 정 업

2002년 인하대학교 지리정보공학과 졸업(공학사). 2004년 인하대학교 지리정보공학과 졸업(공학석사). 2008년 인하대학교 지리정보공학과 졸업(공학박사). 2009년~현재 University of Missouri/Post-doctor 관심분야는 Digital Watermarking, Spatial analysis, Spatial statistics



박 수 흥

1989년 서울대학교 지리학과 졸업(학사) 1991년 서울대학교 지리학과 졸업(석사) 1996년 Univ. of South Carolina at Columbia 졸업(지리학박사). 1996년 7월~1997년 8월 Indiana University/ Reserch Associate. 1997년 9월~1997년 12월 서울대학교/연수연구원. 1998년 1월~2000년 2월 서울시정개발연구원/연구위원. 2000년~현재 인하대학교 지리정보공학과 부교수. 관심분야는 u-GIS service model, spatial databases, spatial data models