

LDI 표현방법을 이용한 3D GIS 구현

송상훈*, 정영기**

LDI(Layered Depth Image) Representation Method using 3D GIS Implementation

Sang Hun Song* , Young Kee Jung**

요약

지리정보시스템(Geographic Information System : GIS)은 지리적으로 참조 가능한 소프트웨어 시스템을 말한다. 이러한 지리정보시스템의 주요 특징 중 지리정보의 표현이라는 문제를 중심으로 살펴볼 때 기존의 2차원적인 표현에서 3차원적인 표현으로의 연구개발이 활발하게 진행되고 있으나, 방대한 양의 지리정보를 빠르고 효율적으로 처리하는데 많은 문제가 있어, 본 논문에서는 GIS의 효율적인 장면 렌더링을 위해서 영상기반 모델링과 렌더링을 위해 제안된 LDI(Layered Depth Images)를 이용하여 3차원의 GIS를 렌더링하고자 한다. 실측기반 방식으로부터 3차원의 지형데이터를 획득하고, 이렇게 획득된 3차원의 지형 데이터는 깊이 정보를 가지고 있는데, 이러한 깊이 정보를 기반으로 하여 LDI를 생성하였다. 또한 전통적인 모델링 방식인 3DS-Max를 이용하여 LDI를 생성하였다. 이렇게 획득된 LDI 정보를 이용하여 보다 효율적인 3차원의 GIS 렌더링을 할 수 있었다.

Abstract

Geographic information system (GIS) geography reference it talks the software system which is possible. When like this geographic information system in key feature trying to observe the problem which is an expression of geography information in the center, the research and development with 3 dimension expressions is active from 2 dimension expressions of existing and it is advanced, double meaning geography information which is huge to be quick, the place where it controls efficiently there is a many problem, the ring from the dissertation which it sees and 3 dimensions and efficient scene of the GIS rendering compared to the ring from hazard image base modeling and rendering compared to hazard proposal LDI (Layered Depth Images) it uses GIS rendering compared to the ring to sleep it does. It acquired the terrain data of 3 dimensions from thread side base method, terrain data of 3 dimensions which are acquired like this the place where it has depth information like this depth information in base and the LDI, it did it created. Also it was a traditional modeling method and 3DS-Max it used and it created the LDI. It used LDI information which is acquired like this and the GIS of more efficient 3 dimensions rendering compared to the possibility of ring it was.

▶ Keyword : LDI(Layered Depth Images), 3D GIS(Geographic Information system), IBR(Image Based Rendering), CG/VR

1. 서론

최근 GIS의 발달로 인해 GIS가 다양한 분야에 적용되고 있으며, 그 규모 역시 확대되고 있다. 최근 GIS시장의 흐름을 보면 클라이언트와 서버(C/S)방식에서 웹(Web) GIS로, 2차원에서 3차원 방식으로 변화해 가고 있다. 또한 Desktop GIS방식에서 Mobile GIS방식으로 변화하고 있다. 뿐만 아니라 공간정보 획득 및 분석방법에 있어서 다양한 자료에 대한 취득 및 분석이 이루어지고 있다.

현재 GIS시장은 과거 GIS분석을 위한 C/S개발 위주의 GIS에서 인터넷 기술의 발달로 인터넷 기술과 GIS가 접목된 Web GIS형태로 발전되고 있다. 이를 통해 인터넷으로 연결된 수많은 서버에 분산되어 있는 데이터와 기능의 객체들이 사용자의 요구에 따라 결합, 통합되어 사용자가 GIS 프로그램을 소유하지 않아도 웹을 통하여 보다 손쉽게 GIS를 활용할 수 있게 되었다. 현재의 Web GIS는 처리방식에 있어서 전송부하를 최소화하는 동시에 클라이언트와 서버의 컴퓨팅 자원을 최대한 활용할 수 있는 혼합방식이 주류를 이루며, 데이터 포맷에 있어서는 전송속도에서 유리하고 객체선택이 가능한 벡터방식이 주를 이루고 있다. 또한 plug-in, Active X, Java 등 인터넷 GIS의 구현방식에 있어 각각의 장단점을 결합한 방식을 채택하는 경우가 많아지고 있다.

기존의 2차원 GIS는 실제세계의 지리요소를 점, 선, 면의 2차원 객체로 일반화함으로써 정보의 손실을 가져온다는 한계점이 있다. 이와 같이 현실세계 반영의 측면에서 2차원 GIS가 가지고 있는 한계점을 극복하고 실제 현실세계보다 유사하게 반영하고 다양한 분석을 수행할 수 있는 3차원 GIS로 발전되고 있다. 3차원 GIS는 현실세계와 유사한 입체 공간 정보를 제공함으로써 사용자로 하여금 현실세계의 공간현상에 대한 이해를 돕고, 보다 정확하고 다양한 분석을 수행할 수 있도록 도와준다.

3차원 GIS 관련 기술은 1980년대부터 현재까지 3차원 지형분석의 2차원적 표현에서부터 3차원 지형의 가시화 및 분석 시스템을 거쳐 최근 3차원 가상도시 단계까지 발전해오고 있지만, 현재의 3차원 기술은 3차원 공간분석, 질의처리 등 필수적인 공간분석 기능들은 제한적으로 사용되고 있는 실정이다.

【표 1】 3차원 GIS 관련 기술발전 추세

연도	개요	내용
1980년대	3차원 지형분석의 2차원적 표현	3차원 지형분석 알고리즘은 개발 되었으나 컴퓨터 그래픽 기술의 부족으로 인해 분석의 결과를 3차원적으로 표현하지 못하고 2차원 평면에서 표현함
	3차원 지형 가시화 (Flight Simulation)	컴퓨터 그래픽 기술의 발달과 하드웨어의 성능 향상으로 지형의 3차원 가시화 및 애니메이션이 가능해 졌으며 이와 관련된 상용 제품(ERDAS IMAGINE 등)이 출시되었고 3차원 GIS의 정형처럼 인식됨
1990년대	3차원 지형 분석	3차원 지형의 단순한 가시화뿐만 아니라 3차원 지형에 대한 다양한 분석의 결과를 3차원적으로 나타내어 사용자의 이해력을 높이고 요구를 충족시킴. 모의군사작전 시스템 등에 응용됨

2000년대	3차원 가상도시 (Browsing)	웹이 널리 사용되고 컴퓨터 그래픽과 가상현실 기술의 발전으로 인해 다양한 3차원 가상도시가 구축되고 있으나, 이는 위상학적 관계가 전혀 없는 단순한 점들의 집합에 불과하므로 브라우징 기능만 제공되고 검색, 분석, 편집 등의 유용한 지리적 정보를 얻을 수 없음
	능동적 3차원 가상세계 (Analysis)	3차원 지형과 사물물이 연동 관리되고 3차원 공간 지리정보가 위상학적인 자료구조에 의해 유동적이고 능동적으로 관리되므로 3차원 공간 정보의 검색, 편집, 분석이 가능하고 사용자 질의처리 및 물입적 상호작용을 지원하여 사용자로 하여금 현실 같은 가상세계에서 더욱 빠르고 쉽게 이해하고 의사결정 하도록 지원하는 시스템

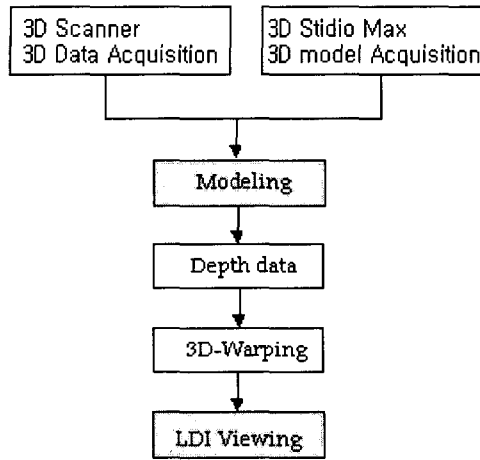
과거의 연구 성과는 대부분 모델링, 혹은 지리분석 시스템을 위한 공간데이터의 처리 분야로 이루어져 있다. 최근 들어서 이러한 공간 자료에 3차원 가시화 기술을 접목시키기 위한 노력들이 제시되고 있다. [8] 인터넷상에서 3차원의 공간정보를 제공하는 데는 부피를 갖는 하나의 Object로 취급하는 3D Solid 모델(3차원 데이터 모델)을 사용하고 있다. 3D Solid 모델은 CAD/CAM 및 컴퓨터 그래픽 분야에서 먼저 사용하고 있으며, X, Y 좌표 지점에 대해 복수개의 Z 값을 갖고 있다. 데이터조작 및 공간분석 등 각종 연산 과정이 복잡하므로 3차원의 Surface 모델링 기법을 보편적으로 사용하고 있다. Surface 모델링 기법은 연속적인 2차원 X, Y 좌표에 고도 값을 마치 속성처럼 할당하여 처리하는 방식이다. Surface 모델링은 연속적인 면을 대상으로 하지만 현실적으로 면상에 연속적으로 존재하는 모든 지점의 고도 값을 저장하기 어렵기 때문에 대표 지점만의 고도 값으로 지형을 표현하는 Sampling Method를 사용한다. Sampling Method에는 2차원의 등고선이나 3차원 GIS에서 활용하는 DEM, TIN 등 보편적으로 사용된다. DEM과 같은 고도 데이터를 취득하는 방법은 지상측량, GPS측량, 항공사진측량, 수치지상측량, 위성영상 등 여러 가지가 사용되고 있으나 항공사진이나 위성영상 등에 의한 방법을 많이 사용하고 있다.

그러나 2D에서 3D로 이동하고 있는 현재의 GIS는 자료 크기가 기하급수적으로 커지고 이로 인해 처리속도가 느려지고 있으며 사용자의 실시간 렌더링(Rendering) 요구는 커지고 있다. 대용량의 공간자료에 대한 처리속도, 3차원 처리기술, 가상현실 처리기술 등의 제약조건과 3차원 GIS를 가시화하기 위해서는 방대한 양의 데이터 처리 문제와 데이터를 처리하기 위한 시간과 비용이 많이 과다하게 발생하는 문제점을 가지고 있어 본 논문에서는 영상기반 모델링과 렌더링 방식에서 사용되는 LDI(Layered Depth Images)를 이용하여 3차원의 GIS를 렌더링 하고자 한다.

II. 전체 시스템 개요

3차원 스캐너를 이용한 실측기반의 방식으로부터 3차원의 지형데이터를 생성하고 인터그래이션과 레지스트레이션 과정을 통해 3차원의 깊이 정보(Depth Data)를 획득하였다. 이렇게 획득된 깊이 정보를 기반으로 LDI를 생성하였다.

전통적인 모델링 방식인 3D Studio Max 모델링 툴을 이용하여 보여 질 장면을 모델링하고 모델링 데이터를 기반으로 하여 각각의 카메라의 위치를 설정하였다. 이렇게 설정된 카메라에 대한 정보를 얻고, 카메라 정보를 기반으로 하여 Camera Matrix를 획득하게 된다. 획득된 Camera Matrix를 이용하여 각각의 카메라 위치에 대한 3차원의 워핑 작업으로 통하여 LDI를 생성하였다. 이렇게 생성된 LDI를 이용하여 3차원의 GIS정보를 렌더링 할 수 있었다.



[그림1] 3차원 스캐너와 3D-Max를 이용한 LDI생성

2.1. 입력 데이터 생성

2.1.1. 3차원 스캐너를 이용한 LDI생성

3차원 스캐너를 이용하여 측정된 데이터는 스캐너가 측정 가능한 영역만큼 획득할 수 있다. 보통 측정 대상물은 스캐너의 영역보다 크기 때문에 3차원 측정은 측정 대상물의 전체 형상을 고려하여 각 부분을 빠짐없이 측정하여야 한다. 이렇게 측정된 측정 대상물의 각 부분에 해당하는 측정데이터는 서로 다른 좌표계에 있게 된다. 이러한 측정데이터의 좌표계를 하나의 기준 좌표계로 정렬시키는 과정을 레지스트레이션이라고 한다. 레지스트레이션 단계에서 정렬된 데이터는 어느 정도의 오차가 존재하게 된다. 따라서 기준 좌표계에서 하나의 모델에 대한 전체적인오차를 최소화하는 과정이 필요하고, 또한 데이터들의 중첩 부분을 적절히 처리하여 데이터 수를 줄이고 누락된 부분에 대한 보간을 수행해야 하는데 이러한 과정을 인티그레이션이라고 한다. [1],[2]

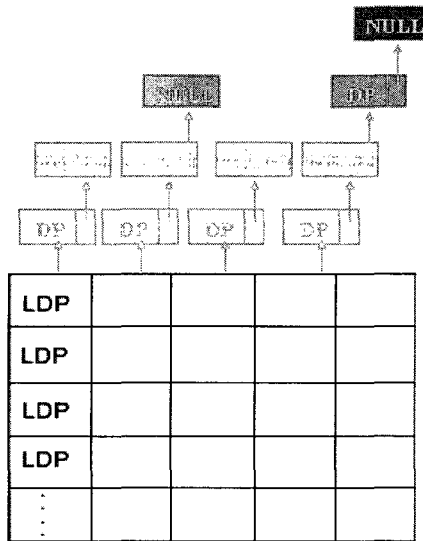
위와 같이 3차원 스캐너에 의해 3차원의 지형 데이터를 획득하고 레지스트레이션과 인티그레이션 과정을 거쳐서 3차원 모델의 구조를 복원 한다. 이렇게 복원된 3차원의 모델 구조는 깊이정보(Depth-Data)를 포함하고 있는데, 포함된 깊이 정보를 기반으로 하여 3차원의 워핑 작업(3D-Warping)을 수행하여 LDI를 생성하였다.

2.1.2. 3DS-Max를 이용한 LDI생성방법

우선 3DS-Max의 모델링 Tool을 이용하여 보여 질 장면에 대한 모델링을 하게 된다. 모델링 된 데이터를 기반으로 하여 각각의 카메라의 위치를 설정하고, 설정된 카메라에 대한 정보와 위치 값을 획득하게 된다. 이렇게 획득된 카메라에 정보와 위치 값을 기반으로 하여 Camera Matrix를 구하고, 각각의 카메라 위치에 대한 3차원 워핑 작업을 통하여 LDI를 생성하였다.

2.2. LDI의 구조

LDI는 여러 시점에서 생성된 깊이 영상을 합성하여 하나의 데이터 구조로 만드는 것이다.



[그림 2] LDI 구조

LDI는 카메라 정보를 저장하는 구조체와 LDP(Layered Depth Pixel) 2차원 배열로 구성되어 있고, 깊이 픽셀의 개수를 저장하는 변수와 깊이 픽셀을 연결 리스트를 이용하여 저장하는 구조로 되어 있다. 깊이 픽셀은 각 픽셀에 대한 색상 정보와 깊이 값을 저장할 수 있는 구조로 되어있다.

LDI는 여러 시점에서 얻은 다수의 깊이 영상 정보를 포함하므로, 각각의 LDI 화소는 색상 정보, 깊이 정보 및 렌더링을 지원하는 특성 정보를 담고 있으며, 각 화소의 위치마다 다수의 계층을 가지고 있다.

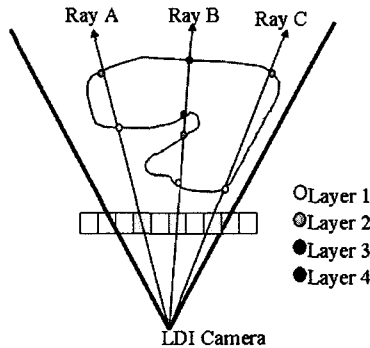
```

Depth Pixel =
{ ColorRGBA: 32 bit integer
  Z: 20 bit integer
  SplatIndex: 11 bit integer
}

LayeredDepthPixel =
{ NumLayers : integer
  Layers[0..numlayers-1]:
  array of DepthPixel
}

LayeredDepthImage =
{ Camera: camera
  Pixels[0..xres-1, 0..yres-1]:
  array of Layered DepthPixel
}
    
```

[그림 3] LDI 저장 구조



【그림 4】 Layered Depth Image.

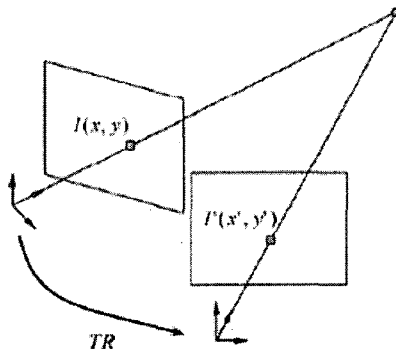
III. LDI 표현 방법

생성된 LDI 를 표현하기 위해서는 픽셀기반 3 차원 워핑 기술이 사용된다. 깊이 픽셀의 3 차원 워핑 연산은 텍스처 매핑 에서 사용되는 연산과 유사하며, McMillian의 Ordering Algorithm을 사용하여 분석하였다. [3]

3.1. 3차원 워핑 수행

3 차원 영상 워핑 방법에는 입력 영상을 메쉬로 구성한 다음 워핑 하는 방법과, 각각의 픽셀 단위로 워핑하는 방법이 있다. 일반적으로 메쉬를 구성하는 방법은 사진과 같은 영상이 시작영상과 끝 영상으로 주어졌을 때 이 두 영상의 특징점들의 대응 관계를 정해주어 그 사이의 변화를 자연스럽게 연결시켜 주기 위해 사용되는 방법이다. 정확한 3 차원 정보가 있을 경우에는 특징점을 찾을 필요 없이 깊이 정보를 이용하여 입력 영상을 삼각 메쉬로 구성하여 줄 수 있다. 삼각 메쉬 구성은 입력 영상이 적은 수의 폴리곤으로 렌더링 된 경우에 워핑에 드는 시간을 단축시킬 수 있다는 장점을 가진다. 그러나 워핑 이전에 입력 영상에 메쉬를 구성해주어야 하는 번거로움이 있다. 픽셀단위로 워핑 함수를 적용하는 것은 메쉬 구성과 같은 번거로운 작업이 필요 없이 간단하게 워핑을 수행한다.

먼저 입력 영상에 역투사 변환을 적용하여 원래의 3 차원 값을 복원한 뒤, 다시 새로운 시점에서의 투사 변환을 적용하여 결과 영상을 얻는다. LDI 의 표현은 기본적으로 픽셀 기반 3D 워핑(Warping)에 의해서 표현이 된다. [9]



【그림 5】 3차원 워핑

$R = [\gamma_v]$ 을 회전 행렬이라고 하고, $T = (\Delta_x, \Delta_y, \Delta_z)^T$ 을 카메라의 이동 벡터라고 했을 때, 새로운 위치에서는 깊이 픽셀은 다음과 같이 표현 될 수 있다. [3]

$$x' = \frac{(\gamma_{11}x + \gamma_{12}y + \gamma_{13})Z(x,y) + \Delta_x}{(\gamma_{31}x + \gamma_{32}y + \gamma_{33})Z(x,y) + \Delta_z}$$

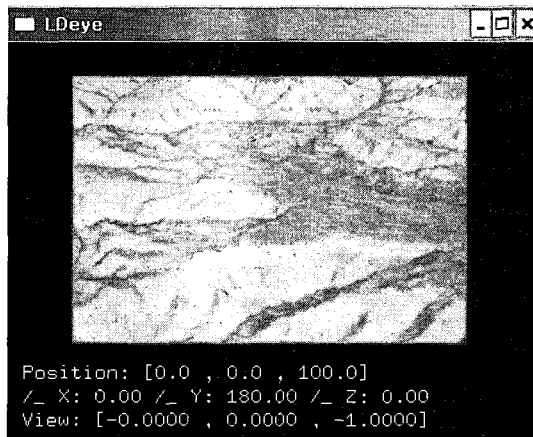
$$y' = \frac{(\gamma_{21}x + \gamma_{22}y + \gamma_{23})Z(x,y) + \Delta_y}{(\gamma_{31}x + \gamma_{32}y + \gamma_{33})Z(x,y) + \Delta_z}$$

3.2. LDI Viewer

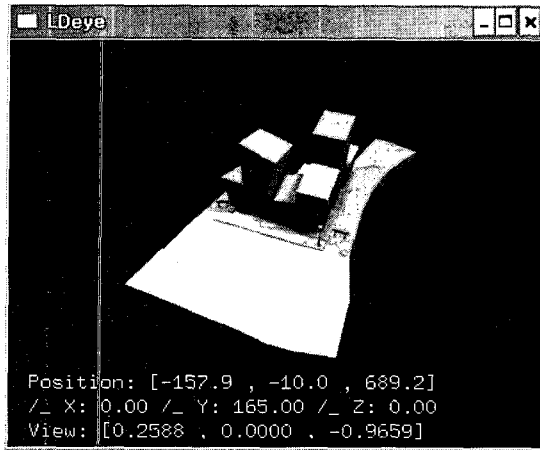
3차원 스캐너와 3D Studio Max를 이용하여 생성한 깊이 정보를 통하여 폭과 높이, 카메라 정보를 얻게 된다. 획득된 카메라에 대한 정보는 X, Y, Z에 대한 위치 값을 가지고 있다. 카메라에 대한 위치 값과 이미지 플랜 사이의 거리 값을 통하여 깊이 정보를 생성 하였다. 생성된 깊이 정보를 기반으로 3차원 워핑 작업을 통해 새로운 Layered depth image를 생성하고, 생성된 LDI는 폭, 높이 값과 Layered-depth pixel의 정보를 포함하게 된다.

IV. 실험결과

그림6은 3차원 모델 데이터 획득을 위하여 공간부호화 광학방식의 EZSCAN2002 3차원 스캐너를 사용하여 3차원의 지형데이터를 생성하고 인터그레이션과 레지스트레이션 과정을 거쳐서 3차원의 깊이 정보(Depth Data)를 획득하고 획득된 깊이 정보를 기반으로 LDI를 생성 하였다. 또한 그림7은 3D모델링 툴을 통하여 LDI를 생성 할 수 있었다.



【그림 6】 3차원 스캐너를 이용한 LDI생성



[그림 7] 3D-Max를 이용한 LDI 생성



[그림 8] 생성된 LDI 정보

V. 결론 및 향후과제

3차원 GIS는 인터넷 응용기술, 대용량 데이터베이스 응용기술, 실시간 대용량 자료 처리기술, 3차원 그래픽 처리기술, 가상현실 기술 등의 제반 요소 기술들이 종합적으로 적용되는 분야로써, 기존의 2차원 GIS가 갖는 한계를 극복하고 다양한 응용 분야에서 3차원 지리정보의 처리 및 활용이 활발히 모색되고 있다.

본 논문에서는 3차원의 GIS 데이터를 렌더링하기 위해서 영상기반 모델링과 렌더링에서 사용되는 LDI(Layered Depth Images)를 이용하여 3차원으로 가시화 하는 방법을 제안하였다. 추후 이미지와 이미지가 겹쳤을 때 나타나는 이미지 폴딩 문제와 홀과 갭 문제를 해결할 것이다.

참고문헌

- [1] 김동규, 윤인도, 현웅근, 정영기, "하이브리드 모델링 방식을 이용한 3차원 모델의 생성", HCI2005 학술대회 논문집, 2005
- [2] 권대현, 최이백, 이의택, "ETRI-모아레 스캐너를 이용한 3차원 모델의 자동생성", 한국방송학회 학술대회, 1999권, 47-52쪽, 1999
- [3] Jonathan Shade, Steven Gortler, Li-wei He, Richard Szeliski. "Layered Depth Images", SIGGRAPH98 Conference proceedings. 1998
- [4] Steven J. Gortler, Li-wei He, Michael F. Cohen. "Rendering Layered Depth Image", Microsoft Research Technical Report(MSTR-TR-97-09). 1997
- [5] Alan Watt, "Computer Graphics Third Edition", ADDISON-WESLEY PUBLISH, pp.443-472. 2000
- [6] Marc Pollefeys, "Tutorial on 3D Modeling from Images", Dublin, Ireland In conjunction with ECCV, 2000
- [8] Samet,H. The Quadtree and Related Hierarchical Data Structure. ACM Computing Survey16(2), June 1984, pp.187-260.
- [9] "Three-Dimensional Computer Vision - A Geometric Viewpoint(4th edition)", OLIVER FAUGERAS PUBLISH, pp.165-243. 2001

저자소개



송상훈

2005년 : 호남대학교 컴퓨터공학과 졸업

2005년 ~ 현재 : 호남대학교 컴퓨터공학과 대학원 재학 중

관심분야 : 3D GIS, VR, LDI, 3D렌더링