

4S-Van 카메라 데이터를 이용한 비디오 지리정보 서비스

김성수, 김경호, 이종훈
한국전자통신연구원

e-mail: {sungsoo, kkh, jong}@etri.re.kr

Video Geographical Information Services using 4S-Van Camera Data

Sung-Soo Kim, Kyong-Ho Kim, Jong-Hun Lee
Spatial Information Technology Center, ETRI

요 약

본 논문에서는 3 차원 데이터베이스와 4S-Van 에서 획득된 GPS 관련 데이터를 이용하여 사용자 에게 비디오내의 지리 객체들의 정보를 제공해 줄 수 있는 미디어 GIS 시스템 구조를 제안한다. 비디오내의 지리객체에 대한 지리정보 서비스를 위해 실세계 비디오와 3 차원 그래픽 세계를 매핑할 수 있는 가상 세계 매핑(VWM; Virtual World Mapping) 이라는 새로운 상호 운용적인 지리데이터 서비스 개념을 소개한다.

제안한 기법은 임의의 비디오 프레임에 대한 3 차원 가상공간을 재구성하여 지리정보 및 속 성정보를 쉽게 검색해 낼 수 있다. 또한, 2 차원, 3 차원, 미디어 데이터간의 상호연계 및 운용기능을 제공함으로써 기존의 지리정보시스템 분야의 새로운 패러다임을 제시한다.

가상 세계 매핑기법은 기존의 비디오에서 얻어진 이미지에 대한 별도의 이미지 프로세싱 과 정 없이 지리정보를 서비스 할 수 있다는 장점을 가진다. 제안된 방법을 이용하여 비디오 영상을 스트리밍하면서 클라이언트가 지리 정보서비스 받을 수 있는 GeoVideo 서버 및 웹을 통해 서비스 를 제공하는 GeoVideo 클라이언트 시스템을 구현하였다.

1. 서론

최근 실세계 동영상을 이용하여 사용자에게 지리정 보를 제공하거나 의사결정을 할 수 있는 비디오 GIS (Video GIS)에 관한 연구들이 연구 시작단계에 있다.

본 논문에서 논의되는 미디어 GIS (Media GIS)란 동영상을 실시간으로 전송 받으면서 2 차원 수치지도, 3D GIS 를 상호 연계하여 지리정보 서비스를 제공하 는 기술을 말한다.

현재 미디어 GIS 에서 가장 문제점으로 대두되는 것은 서로 다른 형태의 데이터 형식으로 인한 데이터 상호 운영 및 연계가 어렵다는 문제점이 있다. 데이터 상호운용성(interoperability)에 관한 해결책으로 마이크로소프트에서 제공하는 OLE DB 기술을 이용하여 시스템을 구축하는 것이다. 하지만, 각각 서로 다른 데이터간의 단방향 혹은 양방향 데이터 연계하는 문 제는 통합된 데이터 구축을 통한 방법 이외에는 별다 른 해결책이 제시되지 않고 있다.

현재까지 지리정보 시스템에서 주요하게 사용되 고 있는 데이터 형식으로는 2 차원 수치지도, 3 차원

데이터베이스, 4S-Van 차량으로 획득된 데이터 등이 대표적이다.

본 연구에서의 궁극적인 목적은 이러한 서로 다 른 공간데이터 형식 및 속성을 가진 데이터들을 별도 의 데이터 수정 및 통합 작업이 없이도 시스템 상에 서 상호 연계하여 지리정보 서비스를 제공하는 것이 다.

2. 관련연구

역사적으로 볼 때 비디오와 지리정보를 결합한 최 초의 연구는 1978 년에 MIT 에서 개발한 Aspen Movie Map 프로젝트가 있다. 이 시스템은 두개의 스크린을 이용하여 비디오와 지도를 볼 수 있는 단순한 시스템이었다[3]. 이후의 관련 연구에서도 대부분 비 디오 클립(clip)을 사용하였으며 비디오 클립을 통해 직접적으로 사용자 입력을 처리해 줄 수 있는 시스템 이 아닌, 임의의 비디오 클립에 대해 지도와 단순하게 연계하는 시스템이 소개되었다.

최근 Navarrete[2]는 비디오와 지리정보를 통합하기

위한 방법으로 비디오 내의 지리객체를 이미지 프로세싱 과정을 통해 비디오 세그멘테이션(segmentation) 하는 기법을 제시했다. 하지만, 이 방법은 비디오 세그멘테이션을 위한 이미지 프로세싱의 처리에 소요되는 비용 및 정확도 등의 문제점을 가지고 있다.

3 차원 GIS 분야에서는 데이터 재사용성 및 상호 운용성을 높이기 위해 이미 구축된 2 차원 GIS 데이터와 최소한의 3 차원 속성을 기반으로 건물 및 도로를 3 차원 모델링 할 수 있는 장면 모델러 컴포넌트가 제시되었다[5, 7]. 또한, OGC (OpenGIS Consortium) 표준[9]을 준수하는 2 차원 개방형 GIS 컴포넌트인 Mapbase 에서 3 차원 모델링 방법을 연계함으로써 기존 2 차원에 대한 3 차원 확장을 용이하게 할 수 있는 기법이 제시되었다[8].

4S 기술이란, 요즘 광범위하게 활용되고 있는 지리정보시스템(GIS)와 위성측위시스템(GNSS), 공간영상정보시스템(SIS), 지능형교통체계(ITS)의 네 가지 시스템을 통칭하는 말로, 공간정보라는 키워드를 공통적으로 가지는 네 분야를 통합 연계하는 첨단기술을 말한다[4, 1].

본 논문에서 현재 소개된 여러 공간정보 처리 기술들을 이용하여 비디오 GIS 시스템 구축을 위한 가능한 접근 방법을 세 가지로 분류해 보았다. 첫 번째 접근방법으로는 4S-Van 을 통해 획득한 두 CCD 영상들(오른쪽, 왼쪽)을 스테레오 이미지 처리 통해 3 차원 공간정보를 구축하는 방법이며, 이 접근방법은 3 차원 공간정보 및 속성데이터 구축을 위해 수작업으로 해당 이미지들을 처리해야 하는 단점이 있다[10]. 두 번째 접근방법은 비디오내의 공간객체를 MPEG4 표준에 맞추어 비디오의 프레임별로 저장하는 방법이다. 이 방법 또한 MPEG4 포맷으로 저장하는데 많은 수작업이 요구된다는 단점이 있다.

세 번째 접근방법은 기존의 구축되어진 3 차원 데이터베이스를 이용하는 방법으로 본 논문에서 제안하는 접근 방법이다.

본 논문에서 제시하는 기법을 이용한 시스템은 2 차원 수치지도 및 속성, 2 차원에서 확장된 3 차원 데이터베이스, 멀티미디어(비디오, 오디오) 등을 상호 연계할 수 있다는 의미로 "미디어 GIS (Media GIS)"라고 명명하였다.

현재까지 비디오(멀티미디어) 상에서 사용자가 상호작용이 이루어져 지리정보를 질의 및 검색이 가능한 시스템은 소개된 바가 없었다. 따라서 본 연구에서 제시하는 미디어 GIS 시스템은 데이터 상호연계 및 운용기능을 제공함으로써 기존의 지리정보시스템 분야의 새로운 패러다임을 제시한다는 데 큰 의미가 있다.

3. 미디어 GIS 시스템

3.1. 시스템 구조 (System Architecture)

본 연구에서 제안하는 미디어 GIS 시스템은 크게 3 차원 GIS 컴포넌트들로 구성되는 GeoVideo 서버와 비디오 동영상 브라우징을 위한 GeoVideo 클라이언트로 구성된다. 그림 1 은 미디어 GIS 시스템을 위한

구성 컴포넌트를 보여주고 있다.

GeoVideoServer 는 공간정보의 상호운용을 지원하기 위해 서로 다른 데이터 소스로부터 동일한 인터페이스를 통해 데이터 접근을 제공하는 데이터 제공자(WorldProvider, WorldDAO) 컴포넌트와 데이터 제공자로부터 전달 받은 데이터를 내부 시스템 구조에 따라 관리하는 데이터관리자 (WorldManager) 컴포넌트, 3 차원 장면을 표현하고 모델링하기 위한 장면그래프를 관리하는 장면관리자 (Scene Manager) 컴포넌트로 구성된다.

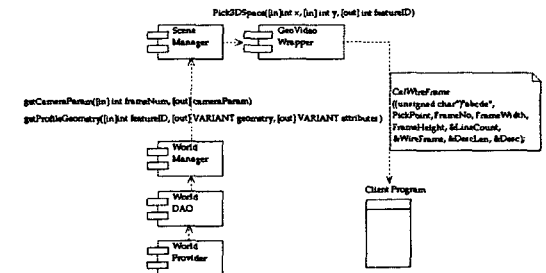


그림 1. 미디어 GIS 시스템 컴포넌트 다이어그램

그리고, 서버와 클라이언트 통신을 위해 RPC (Remote Process Communication) 인터페이스를 정의하였다.

표 1 의 RequestFeatureInfo 메시지를 사용하여 서버는 클라이언트에서 전달받은 입력을 이용하여 건물 또는 시설물의 속성정보를 검색하여 클라이언트에 전달한다.

```
void RequestFeatureInfo(
/* [string][in] */ unsigned char __RPC_FAR *pszVideoFilename,
// video file 이름
// 사용자가 화면에서 마우스 클릭한 좌표
/* [in] */ PointInt ptPickPoint,
/* [in] */ long lFrameNo, // 프레임 번호
/* [in] */ long lFrameWidth, // 프레임의 폭
/* [in] */ long lFrameHeight, // 프레임의 높이
// 계산된 시설물 외관에 대한 선(line)의 갯수
/* [out] */ long __RPC_FAR *lLineCount,
/* [size_is][size_is][out] */ LineInt __RPC_FAR * __RPC_FAR *ptDeviceLine,
// 외관의 선 데이터들
// 시설물에 대한 설명의 길이
/* [out] */ long __RPC_FAR *lDescStringLen,
/* [size_is][size_is][out] */ unsigned char __RPC_FAR * __RPC_FAR *Desc
// 시설물에 대한 설명
/* [out] */ long __RPC_FAR *lSelectedID // 시설물 ID
)
```

표 1. 미디어 GIS RPC 인터페이스 함수

3.2. 데이터 구축

본 논문에서 제안하는 미디어 GIS 시스템에서 사용되는 데이터는 2 차원 수치지도에 3 차원 부가속성을 구축한 3 차원 데이터베이스와 4S Van 을 통해 획득한 데이터가 있다.

4S-Van 의 하드웨어 구조는 크게 GPS (Global Positioning System)와 IMU (Inertial Measurement Unit), 컬러 CCD 카메라, 흑백 CCD 카메라, 1 대의 적외선 카메라 등의 센서부와 데이터 저장부로 나뉘어진다. 소프트웨어 부분으로는 각각의 센서들로부터

입력된 데이터들을 통합, 분석 처리할 수 있는 부분으로 GPS/IMU 통합 알고리즘, 좌표변환과 렌즈/카메라 자세에 대한 보정, CCD 영상별 외부 표정요소 산출 및 보정 등을 통하여 각각의 센서들로부터 입력된 데이터들을 통합, 분석 처리할 수 있다.

그런데, 본 연구에서 비디오 스트리밍을 위해 제안하는 시스템은 4S Van 에서 제공되는 CCD 촬영 이외에 일반 캠코더를 이용하여 동영상을 획득하였으며 초기 동영상과 CCD 이미지간의 보정을 통해 동영상 및 카메라데이터 등을 미디어 데이터베이스로 구축하였다. 미디어 데이터베이스는 각 미디어 파일에 대한 카메라 위치 (x,y,z) 및 자세정보 (omega, phi, kappa)를 저장하고 있는 테이블과 미디어 파일에 대한 테이블 그리고 두 테이블을 맵핑해 주는 맵핑 테이블로 구성된다.

3 차원 GIS 데이터베이스 구축은 기존의 구축된 2 차원 수치지도의 기하정보와 3 차원 부가정보 (건물의 DEM 상 밀면높이, 건물의 높이)를 연계한 형태로 구축하였다. 시설물의 속성데이터 구축은 해당 건축물 대장과 현지조사를 통해 구축하였다. 구축된 3 차원 DB 를 이용하여 3 차원 시설물을 모델링 해 낼 수 있다. 시설물의 속성데이터 구축은 해당 건축물 대장과 현지조사를 통해 구축하였다.

3.3. 가상세계 매핑 (VWM; Virtual World Mapping)

3 차원 그래픽 매핑을 위해 서버 (GeoVideoServer) 측은 해당 시설물에 대한 3 차원 모델링을 수행한 뒤 사용자 입력을 기다린다.

사용자측 (GeoVideoClient)에서 특정 건물에 대한 속성질의를 위해 마우스 클릭한 경우 처리 흐름을 그림 2 를 통해 알아보자.

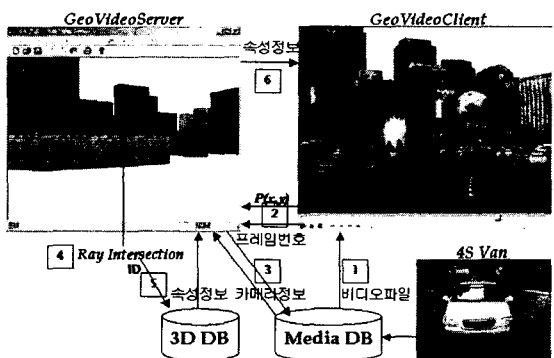


그림 2. 가상 세계 매핑 기법 (VWM)의 처리 흐름.

먼저 사용자가 선택한 비디오 파일, 해당 프레임번호, 마우스 클릭위치 P(x,y)를 서버측에 전달한다. 서버는 전달 받은 정보를 이용하여 미디어 DB 에서 해당되는 카메라정보를 얻어와 3 차원 가상 카메라를 위치시킨 뒤 광선(ray)을 투사하여 충돌되는 시설물의 ID를 얻는다. 해당 ID 에 대한 속성정보를 3D 데이터베이스에서 검색하여 클라이언트에게 되돌려 주게 된다.

다. 여기서, 픽킹(picking) 연산을 구현하기 위해서 광선과 모델링된 건물의 바운딩 박스(bounding box)의 교차 여부를 계산하게 된다. 교차된 박스들의 깊이(depth, z-value)를 비교하여 광선과 가장 가까운 박스에 대한 건물을 구하게 된다. 그림 3 에서 t_{near} 의 큰 값과 t_{far} 의 작은 값을 아래와 같이 비교함으로써 교차여부를 알 수 있다[6].

$$t_{near} (larger) > t_{far} (smaller) : \text{교차하지 않음}$$

$$t_{near} (larger) < t_{far} (smaller) : \text{교차함}$$

교차되는 건물의 속성을 얻기 위해서 소요되는 시간 복잡도(time complexity)는 전체 n 개의 건물에 대한 바운딩 박스와 광선의 교차 판정을 수행하여야 하므로, O(n)의 시간복잡도가 요구된다.

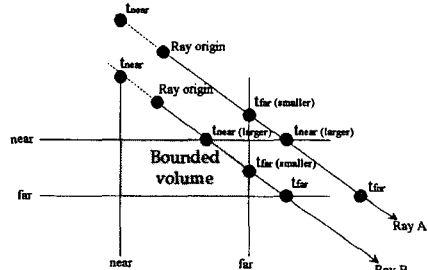


그림 3. 광선과 박스의 교차 (Ray-box intersection)

```
void RequestFeatureInfo
(unsigned char __RPC_FAR *pszVideoFilename,.....)
{
    CGeoVideoServerView* pView = GetServerView();
    pView->resizeView(lFrameWidth, lFrameHeight);
    pView->SetCamera(lFrameNo);
    pView->Pick3DSpace(ptPickPoint.x, ptPickPoint.y);
    pView->getLinesAttributes(lFrameNo, lLineCount,
        ptDeviceLine, lDescStringLen, Desc);
}
```

알고리즘 1. GeoVideoServer 수행 알고리즘

알고리즘 1 은 클라이언트측의 입력을 받아 처리되는 서버(GeoVideoServer)측 수행절차에 대한 알고리즘을 기술하고 있으며, 여기서 광선과 박스의 교차를 계산하는 부분이 Pick3DSpace 멤버함수이다.

이와 같이 비디오 공간과 3 차원 가상공간을 주어진 미디어 DB 의 카메라 정보를 통해 연계할 수 있으므로, 역으로, 3 차원 가상 세계에서 미디어 데이터베이스의 카메라 정보를 이용하여 공간을 네비게이션하다가 임의의 카메라 위치에 해당하는 비디오 영상도 쉽게 검색해 낼 수 있다.

4. 시스템 구현 및 실험 결과

본 논문의 실험을 위해 서울시 중구 지역 4200 여개의 시설물에 대한 3D 데이터베이스를 이용하였으며, 4S Van 을 통해 세종로 일대 10 개의 비디오에 대한 미디어 데이터베이스를 사용하였다.

초기 4S Van 을 통해 구축된 데이터는 획득 후 일반 캠코더의 카메라 내부 파라미터(focal length, FOV;

field of view) 및 외부파라미터(omega, phi, kappa)등을 적용하여 카메라 정보를 보정하여 사용하였다.

가상세계 매핑기법을 구현한 미디어 GIS 의 운영 환경은 아래와 같다.

■ 클라이언트 (GeoVideoClient)

: Pentium4-1.7GHz (1Gbytes RAM) PC, Windows 2000 Professional, C++로 ActiveX Control 형태로 구현.

■ 서버 (GeoVideoServer)

: Pentium4 (2.4GHz, 1Gbytes RAM) PC, Windows 2000 Server, IIS (Internet Information Server), MMS (Microsoft Media Server) , C++/OpenGL library 로 구현.



그림 4. 서울특별시 시청 지역

정확도 실험을 위해 클라이언트에서 이미지상에서 70 회 이상 일정간격으로 픽킹연산을 수행하였다. 표 2 는 그림 4 에서 보여주는 비디오에서 수행한 픽킹연산에 대한 에러 행렬 (error matrix)이며, 에러행렬을 이용하여 공급자 정확도 (producer's accuracy)를 계산하였다. 표 2 에서 1 의 경우에 공급자 정확도는 $(80/84)*100=95.23\%$ 가 된다.

입력 결과 \	1	2	3	4	5	row total	공급자 정확도
1	80	24	0	0	0	104	95.23 %
2	4	60	0	0	0	64	71.42 %
3	0	0	50	0	2	52	67.56 %
4	0	0	0	62	4	66	79.48 %
5	0	0	24	16	64	104	91.42 %
column total	84	84	74	78	70	390	

표 2. 가상세계매핑기법의 정확도 실험결과

3, 4 번 건물에 대해 정확도가 떨어짐을 알 수 있는데, 이는 3 차원 데이터베이스 데이터 자체는 2 차원 단면에 높이 속성을 이용하여 모델링 되기 때문이다. 그리고, 3 차원 가상세계의 시설물의 기하정보를 비디오 상에 정합하는 절차는 앞서 언급한 오차 문제로 정확한 정합이 어렵다는 문제점이 있었다.

5. 결론 및 향후과제

본 논문에서는 카메라정보를 가진 비디오 데이터

에서 사용자 입력을 3 차원 가상세계 및 데이터베이스를 통해 속성 정보를 연계할 수 있는 가상세계 매핑 (VWM; Virtual World Mapping)이라는 새로운 기법을 제안했다. 제안한 기법은 비디오에 대한 이미지 프로세싱 작업이 요구되지 않는다는 장점이 있지만, 3 차원 데이터베이스가 요구되고, 4S Van 데이터의 정확도에 의존적이며 정확도를 높이기 위해 초기 보정 작업이 요구된다.

향후 연구과제로는 카메라 오차 보정을 위한 초기 보정 작업을 자동화 해 줄 수 있는 알고리즘과 3 차원 가상세계와 비디오 영상간의 오차를 계산할 수 있는 연구가 이루어져야 할 것이다. 그리고 초기 데이터 오차보정 및 정확한 3 차원 가상세계 기하정보를 이용하여 비디오 이미지상에 정합 기술에 관한 연구가 필요하다.

참고문헌

[1] Seung-Young Lee, Byoung-Woo Oh, Eun-Young Han, A Study on Application for 4S-Van, In *Proceedings of ISRS(International Symposium on Remote Sensing) 2001*, pp. 124-127, Oct. 2001.

[2] Toni Navarrete, Josep Blat, VideoGIS: Combining Video and Geographical Information, *Research Report, Dept. of Computer Science, University of Pompeu Fabra*, 2001.

[3] Lippman, A., Movie Maps: An Application of the Optical Videodisc to Computer Graphics, In *Proceedings of SIGGRAPH'80*, pp. 32-43, July 1980.

[4] In-Hak Joo, Min-Soo Kim, Byoung-Woo Oh, Development of Disaster Control System Based on 4-S VAN and Mobile Environment, In *Proceedings of ISRS(International Symposium on Remote Sensing) 2001*, pp. 137-142, Oct. 2001..

[5] Sung-Soo Kim, Seung-Keol Choe, Jong-Hun Lee, Young-Kyu Yang, Rule-Based Modeler Component Design for 3D GIS Software, In *Proceedings of ISRS(International Symposium on Remote Sensing) 2001*, pp. 89-94, Oct. 2001.

[6] Alan Watt, Fabio Policarpo, 3D Games : Real-time Rendering and Software Technology, *Addison-Wesley*, pp. 14-22, 2001.

[7] 김성수, 최승걸, 이종훈, 양영규, 3 차원 GIS 소프트웨어를 위한 장면 모델러 컴포넌트 설계, *제 16 회 한국정보처리학회 추계 학술발표 논문집*, pp. 81-84, Oct. 2001.

[8] 김성수, 김광수, 이성호, 최승걸, 김경호, 이종훈, 양영규, 규칙기반 모델링을 이용한 Mapbase 컴포넌트 3 차원 확장, *한국멀티미디어학회 추계 학술발표논문집*, pp. 171-176, Nov, 2001.

[9] Open GIS Consortium, OpenGIS Simple Feature Specification for OLE/COM, OGC, Revision 1.1, 1999.

[10] 변기중, 이동춘, 김주완, 박찬용, 장병태, 증강현실을 이용한 차량항법 시스템에 관한 연구, *한국정보과학회 가을 학술발표논문집*, 제 28 권, 2 호, pp. 493-495, Oct. 2001.