韓國컴퓨터情報學會 論文誌
第15卷 第12號, 2010. 12.

2010-15-12-4-7

구글 어스를 이용한 비행 상황인식을 위한 3차원 시각화

박 석 규*, 박 명 철**

3D Visualization for Flight Situational Awareness using Google Earth

Seok-Gyu Park*, Myeong-Chul Park**

요 약

본 논문은 수집된 항공기의 상황 자료를 이용하여 웹 기반의 구글 어스와 OpenGL을 이용한 3차원 시각화을 통하여 항공기의 올바른 상황인식과 상태정보를 지원하는 시스템을 제안한다. 기존에 제안된 시스템은 방대한 지형 자료와 지도 및 위성사진을 연동하여 표현함으로 속도 저하 및 한정된 지도 정보만을 가시화하는 문제점을 가진다. 제안하는 시스템은 개방된 구글 어스의 Open API을 이용하여 최소한의 비행 정보만으로 실시간 상황인식을 위한 경제적이고 전역적인 시각화 도구를 지원한다. 또한 사용자의 시각적 편리성을 위하여 다중 위치 정보를 볼 수 있는 확장 뷰를 제공한다. 본 연구결과는 웹 환경에서 항공기의 상황인식을 지원하는 시스템으로 이용될 수 있는 가능성을 보인다.

Abstract

This paper proposes 3D visualization systems for the real-time situation awareness and a state information of the aircraft. This system was embodied with OpenGL and the Google Earth of web base using situation data of the aircraft. The existing system has problem which speed decrease and visible restricted map because massive data of terrain and satellite photo. This system is supports the visualization tool which is economic and entire area for a real-time situation awareness with minimum flight information using Open-API of the Google Earth. Also provides a visible convenience to expansion-view using multiple location information. This research result could be used to system for the situation awareness of the aircraft from web environment.

▸ Keyword : 상황인식(Situational Awareness), 구글어스(Google Earth), 시각화시스템(Visualization System), 오픈지엘(OpenGL), 모니터링 시스템(Monitoring System)

• 제1저자 : 박석규
• 투고일 : 2010. 10. 04, 심사일 : 2010. 10. 12, 게재확정일 : 2010. 10. 15.
* 강원도립대학 컴퓨터인터넷과 ** 송호대학 컴퓨터정보과

Ⅰ. 서 론

항공기의 공중 상황인식은 3차원 공간상에서 이루어지는 정보를 이용한 복잡한 패턴뿐만 아니라 방대한 지형정보를 이용하여 실시간으로 시각화하기 때문에 많은 어려움이 있다. 공중감시를 위한 일반적인 시스템은 도플러 방식의 레이더를 통해 전송된 아날로그 항적 정보를 디지털로 가공하여 모니터에 표시하고 지형 정보는 별도의 데이터베이스에서 관리되어 동기화하는 방식으로 운영되고 있다. 그러나 이러한 기존의 시스템은 방대한 3차원 지형 자료와 지도 및 위성사진을 연동하기 때문에 신속하게 상황정보를 전달하여 의사결정을 내리는데 제한을 가지고 있다. 또한 별도의 지형 데이터베이스 및 시스템을 이용하는 고가의 도구로서 경제적인 측면에서 운용상의 문제점도 가진다.[1,2]

본 논문에서는 실시간으로 수집된 비행정보를 이용하여 별도의 지형데이터를 사용하지 않고 공개된 구글 어스의 API를 이용하여 비행정보에 따른 상세한 지형정보를 표시하고 여러 항공기의 상황정보를 위한 확장 뷰를 제공하는 시스템을 제안한다.[3,4] 대부분의 시스템들은 공중 상황을 인식하기 위한 요소 자원보다는 지형정보나 표시정보에 많은 자원을 소비함으로서 전체적인 시스템의 사용성을 저해하고 있다. 또한 항공기의 항적정보의 모니터링 정보를 보여주지 못하기 때문에 시각적으로 제한된 화면만으로 상황을 인식해야 한다는 제안점을 갖는다. 이러한 문제를 극복하기 위하여 제안하는 시스템에서는 별도의 지형정보를 이용하지 않고 OpenGL을 이용한 단위 지도정보에 위치를 표시하고 웹 환경의 구글 어스을 통한 가시화을 보인다.[5,6] 그리고 경로전이정보를 식별할 수 있게 별도의 모니터링 화면을 제공하여 사용자가 보다 쉽고 빠르게 공중 상황을 인식할 수 있게 한다. 논문의 구성은 다음과 같다. 2장에서는 항공기의 공중 상황인식과 관련된 가시화 기법에 대해 살펴보고 3장에서는 제안하는 3차원 가시화 시스템에 대해 기술한다. 4장에서는 구현 환경과 결과를 보이고 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 관련연구

1. 지형 시각화

지형정보를 시각화하는 가장 일반적인 방법은 LOD(Level of Detail) 기법이다. LOD 기법은 지형을 정확하게 보이기 위해 지형의 특정 부분을 자세히 보이게 하기 위한 결정요소로 경험치의 집합을 사용한다. 지형 전체를 동일한 레벨로 화면에 표시하는 것이 아니고 사용자의 식별유무에 따라 여러 레벨을 나누어 식별성이 높은 영역은 많은 폴리곤의 집합으로 지형을 생성하고 식별성이 거의 없는 부분은 아주 적은 폴리곤만으로 지형을 표시하는 기법으로 방대한 폴리곤 생성으로 인한 시간적 동기화 문제와 사실성 저하 문제를 해결할 수 있는 기법으로 널리 사용되고 있다. 대표적인 LOD 기법에는 Hoppe[7], Lindstrom[8], Duchaineau[9]가 소개한 기법이 있는데 Hoppe는 성세한 지형을 요구하는 영역에 대해 임의의 메쉬에 삼각형들을 추가하는 방법으로 적응적 메쉬에 기반한 알고리즘을 사용하지만 다소 복잡하고 많은 메모리 자원을 요구하는 단점이 있다. Lindstrom은 지형의 단위 조각을 표시하기 위하여 쿼드트리라는 구조체를 제시한 기법으로 지형을 재귀적으로 분할하여 높이 맵에 대한 근사값을 생성한다. Duchaineau 는 이진 삼각형 트리에 기반한 알고리즘을 소개하였는데, 지형의 각 조각은 하나의 이등변 삼각형으로 구성되며 삼각형의 정점에서 빗변의 중점을 분할하여 두 개의 이등변 삼각형을 생성한다. 이는 상세한 수준에 도달할 때 까지 자식 노드들에 대해 재귀적 방법을 반복하여 표현하는 기법으로 대부분의 게임 상의 지형 가시화에 이용되고 있다.

이러한 LOD 기법은 표현하고자 하는 영역의 높이값을 모두 메모리에 로드한 후에 적용하는 기법인 반면, Lindstrom[10]은 Out-of-core 기법은 대용량을 지형 데이터를 메모리에 모두 로드하지 않고 필요한 특정 데이터만을 메모리에 로드하여 가시화하는 방법을 제안하였다. 이 방법은 대용량 모델을 균일한 격자 셀로 나눈 후, 셀 안의 꼭지점에 대해 이차 형식을 구하고, 각 셀 내의 이차 형식을 합하여, 선형 시스템에 의해 새로운 점의 위치를 구한다. 그러나 이러한 방법도 해싱 테이블을 구성하기 위해 많은 메모리가 사용된다. 그래서 Lindstrom[11]은 메모리의 비중을 좀 더 낮추면서 하드 디스크를 효율적으로 활용할 수 있고, 경계 부분을 고려하여 효율적으로 메모리를 이용하는 기법을 제안하였다. 또한 경계 부분의 보존을 위해 에지 대한 이차식을 유도함으로써 모델의 품질을 향상시켰다. Shaffer와Garland[12]는 대용량 메쉬를 효과적으로 간략화 하는 다른 방법을 제안하였다. 입력 메쉬에 대해 2가지 패스를 수행한다. 첫 번째 패스는 표면을 분석하고 그리고 공간을 적응적으로 분할한다. 즉 표면을 분석하여 BSP-트리를 구성한다. 마지막 패스는 BSP-트리를 사용하여 원래 꼭지점을 모으게 되고 원래 메쉬에 대해 근사화를 하는 것이다. 이 방법은 BSP-트리를 구성하고 이차 형식 정보를 위한 추가적인 메모리가 필요하다. 그러나 이 방법은 Lindstrom의 방법보다 더 나은 모델의 품질을 보여준다.

2. 상황인식 시각화

상황인식 시각화는 상업적인 항공관제, 지리정보, 게임 뿐만 아니라 해저의 어군이나 어구의 불규칙적인 영향에 따른 인식을 위한 다양한 분야에서 활용되고 있다. 항공기의 비행 상황인식을 위한 논문으로 김성남 등[1]은 Lindstrom의 Out-of-core 기법을 이용하여 [그림 1]과 같이 공중작전 상황인식을 위한 3차원 가시화 도구를 소개하였다.

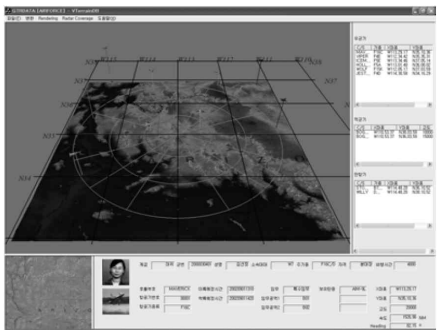

그림 1. 공중작전 상황인식 가시화
Fig. 1 Visualization for Situational Awareness of Air Force Operations

송진오 등[2]은 [그림 2]와 같이 별도의 지형 정보 없이 항공기의 상태전이정보를 이용한 비행경로를 시각화한 논문을 소개 하였다.

그림 2. 비행경로의 시각적 정보
Fig. 2 Visual Information of Flight Path

그리고 박명철 등[13,14,15]은 [그림 3]과 같이 OpenGL을 이용하여 해저의 어군과 어구의 진행 상황정보를 인식할 수 있는 시뮬레이션 도구를 소개하였다.

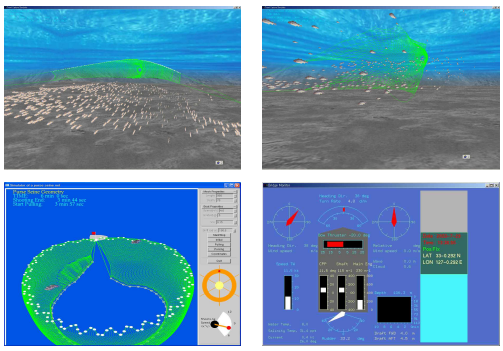

그림 3. 해저의 어군과 어구의 상황인식 가시화
Fig. 3 Visualization for Situational Awareness of Fish and Fishing Gear

그러나 김성남 등의 제안 시스템에서는 방대한 지형정보를 가시화하기 때문에 폴리곤의 생성에 소요되는 시간이 많이 소요되고 지도 정보의 대용량 텍스쳐를 매핑하지 못하는 제안점을 가진다. 송진오 등의 시스템은 비행경로만을 가시화하기 때문에 사실성 있는 가시화는 표현하지 못하고 있다. 그리고 박명철 등이 구현한 시각화 시스템은 별도의 해저의 다양한 상태정보와 어군의 유영을 혼돈이론으로 모델링하여 사실성 있는 시각화 시스템을 구현하는데 큰 기여를 했지만 별도의 지형정보를 이용하지 못하여 사용자에게 위치 정보등을 텍스트 형태로 제공하는 제안점을 가진다.

3. 구글 어스의 COM API

구글 어스는 위성 영상을 바탕으로 전 세계의 상세한 지형정보를 무료로 제공하고 있다. 또한 다양한 운영체제에서 동작하며 구글 어스 COM API을 이용하여 자유롭게 구글 어스를 제어할 수 있다. 응용프로그램의 질의 정보를 구글 어스에 명령으로 전달하여 사용자 뷰 위치를 바꿀 수 있고 경도, 위도 정보 등을 변경하면 자연스러운 비행정보 시각화 도구를 구현할 수 있다. 구글 어스는 태그 기반의 KML(Keyhole Markup Language) 파일 형식에 지리정보를 저장 및 표현하여 구글 어스에 표시할 수 있다. 그러나 본 논문에서는 KML 파일을 사용하지 않고 API를 통하여 직접 표시 정보를 전달한다. 구글 어스의 카메라 위치 이동과 애니메이션을 위해 ApplicationGE 클래스를 인스턴스화 하고 구글 어스 COM API의 주 영역에 해당하는 EARTHLib 라이브러리의 IApplicationGE를 사용하고 카메라의 뷰를 표현하기 위하여 ICameraInfoGE 클래스 포인터를 이용한다. 본 논문에서 사용하는 IApplicationGE 인터페이스의 주요 멤버 함수는 표 2에 보인다[16].

표 1. IApplicationGE 인터페이스의 주요 멤버함수
Table 1. Member Functions of IApplicationGE Interface

멤버함수	기능
GetCamera	실행중인 현재 카메라 뷰을 반환
GetMainHwnd	구글 어스의 메인 윈도우를 위한 핸들러 반환
SetCameraParams	사용자 정보를 실제 뷰 영역에 설정하는 함수

III. 3차원 가시화 시스템

1. 전체 시스템 구조

논문에서 제안하는 전체 시스템의 구조는 그림 4와 같다. 항공기의 비행정보는 특수성과 보안성 문제로 실제 데이터를 수집하기에는 어려움이 있으므로 가상의 시나리오를 작성하여 X-Plane 시뮬레이터에서 항적정보를 취득하여 데이터베이스를 구축한다[17]. 수집되는 항적정보는 항공기의 위치를 가시화하기 위하여 경도, 위도, 고도 등을 수집하고 구글 어스와 맵상에 표현되는 지형을 조종사의 시각에서 사실적으로 가시화하기 위하여 Roll, Pitch, Yew, 각 축 방향의 속도(Heding Velocity(VX), Drift Velocity(VY), Vertical Velocity(VZ)) 등의 정보들을 부가적으로 수집한다. 또한 비행제어시스템에서 데이터를 수집할 수 있다는 가정 하에 각 요소 장치들의 상태정보도 수집하여 경로와 상태전이 시각화 시에 항공기 기체의 상태를 인식할 수 있게 한다.

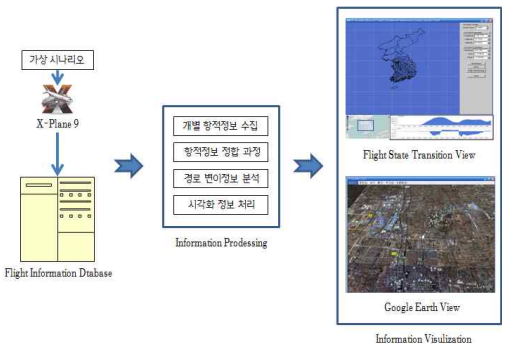

그림 4. 전체 시스템의 구조도
Fig. 4 System Architecture

기체의 상태정보는 현재 연료 유입량(FF)과 두 개의 엔진 RPM(N1, N2), 받음각 지시계(AOA)와 자세방향지시계(ADI), 수직속도계(VVI), 유류 온도(OT), 유류 압력(OP) 등을 수집하고 단위 항공기별로 그림 5와 같이 테이블을 구축한다. 항적정보의 정합 과정은 수집한 정보에 대해 시간적 동기화를 위한 처리 부분으로서 변이상태가 비정상적인 부정확한 데이터를 삭제하는 것이 주 목적이다. 그리고 경로 변이정보 분석 모듈에서는 비행상태 전이와 경로 추적을 위하여 해당 정보를 별도의 임시 테이블에 기록하는 역할을 담당한다. 시각화 정보처리 모듈에서는 각 시각화 뷰에 표시될 데이터를 분류하고 동시성을 위하여 각 뷰에 전달하는 역할을 담당한다.

Sheet1

ELAPSE	LAT	LON	ALT	HEADING	ROLL	PITCH	N1
2096.700195	37.570213	126.778526	65.506753	142.366623	0.129638	-0.269907	33.5023
2096.812744	37.570213	126.778526	65.506753	142.366623	0.129772	-0.269913	33.502327
2096.923584	37.570213	126.778526	65.506759	142.366623	0.129877	-0.269917	33.502327
2097.047607	37.570213	126.778526	65.506753	142.366623	0.129976	-0.269924	33.502327
2097.162109	37.570213	126.778526	65.506747	142.366623	0.130051	-0.269922	33.502327
2097.272949	37.570213	126.778526	65.506728	142.366623	0.130121	-0.269924	33.502327
2097.383545	37.570213	126.778526	65.506728	142.366623	0.13018	-0.26993	33.502327
2097.504395	37.570213	126.778526	65.506734	142.366623	0.130234	-0.269928	33.502327
2097.625732	37.570213	126.778526	65.506734	142.366623	0.130273	-0.269942	33.502327
2097.747803	37.570213	126.778526	65.506734	142.366623	0.130289	-0.269949	33.502327
2097.867676	37.570213	126.778526	65.506734	142.366623	0.130275	-0.269949	33.502327
2097.987061	37.570213	126.778526	65.506747	142.366623	0.130235	-0.269969	33.5023
2098.097656	37.570213	126.778526	65.506759	142.366623	0.130187	-0.269988	33.502289
2098.210449	37.570213	126.778526	65.506778	142.366623	0.130124	-0.269994	33.502281
2098.320801	37.570213	126.778526	65.506784	142.366623	0.130048	-0.270004	33.502281
2098.443359	37.570213	126.778526	65.506784	142.366623	0.129958	-0.270016	33.502274
2098.56543	37.570213	126.778526	65.506778	142.366623	0.129863	-0.270026	33.502258
2098.688965	37.570213	126.778526	65.506778	142.366623	0.129778	-0.270041	33.502258
2098.799316	37.570213	126.778526	65.506778	142.366623	0.129714	-0.270041	33.502258
2098.919189	37.570213	126.778526	65.50679	142.366623	0.129663	-0.270048	33.502258
2099.040771	37.570213	126.778526	65.506809	142.366623	0.129625	-0.270049	33.502258
2099.164063	37.570213	126.778526	65.50684	142.366623	0.129597	-0.270061	33.502258
2099.281982	37.570213	126.778526	65.506834	142.366623	0.129578	-0.270072	33.502258
2099.4021	37.570213	126.778526	65.506815	142.366623	0.129562	-0.270059	33.502258

그림 5. 항공기의 항적정보 테이블
Fig. 5 Flight Information Table of Aircraft

2. 시각화 정보처리 모듈

시각화는 크게 두 영역으로 나뉘어 지는데 첫 번째 영역은 비행 경로와 다중 위치의 표시 및 상태 값의 변이를 보이는 창이고 두 번째 영역은 구글 어스에 사실적 화면을 보이는 곳이다. 첫 영역에 OpenSteetMap을 보이기 위하여 웹 질의를 통하여 지도 정보을 획득하게 된다. 질의의 인수는 DB로부터 전달된 경도와 위도을 넘겨주면 해당 영역을 지도를 png파일 형태로 이미지를 얻을 수 있다. 예를 들어, http://www.openstreetmap.org/?mlat=37.570213&mlon=126.778526&zoom=13 로 질의를 넘기면 [그림 6]과 같이 서울김포국제공항 할주로 부근의 지도를 확대 수준 13(1~17)에 해당하는 지도를 얻는다.

그리고 구글 어스와의 데이터 교류는 공유 메모리를 이용하여 항적 정보를 교류한다. MMF(Memory Mapped File)을 이용하여 프로그램 간에 데이터를 주고 받을 수 있는데 파일과 메모리 객체를 연결시킨 후 그 메모리 객체에 이름을 지정한다. 그 이름을 이용하여 메모리에 데이터를 읽고 쓰는 작업을 하게 되면 그 내용이 메모리와 연결되어 있는 파일에 똑같은 효과가 적용되는 구조이다.

그림 6. OpenSteetMap을 통한 지도 획득
Fig. 6 The map acquisition using OpenSteetMap

본 시스템에서는 파일을 사용하지 않고 특정 메모리 객체를 생성하여 통신한다. 파일명을 적는 곳에 INVALID_HANDLE_VALUE 라고 해주면 메모리 객체만 생성이 된다. 그리고 이 메모리 객체에 이름을 설정해 주면 공유 메모리의 생성이 완료된다. 이렇게 하면 메모리 상에 메모리 영역이 확보되고 각각의 프로세스들은 이 영역을 설정해 둔 이름으로 찾아서 접근(데이터 읽기, 쓰기)하여 작업을 하게 된다.

Ⅳ. 구현 환경과 결과

1. 구현 환경

본 논문에서 제안하는 시각화 시스템의 구현 환경은 [표 2]와 같다.

표 2. 시스템의 구현 환경
Table 2. Implementation Environment of System

구성요소	설명
CPU	Intel(R) Core(TM)2 Duo CPU 3.16GHz
MEMORY	2GB
Graphic Card	NVIDIA GeForce 9300 GS
OS	MS Windows XP
Language	Visual C++ 6.0
Graphic Library	OpenGL 1.4, GLUT, GLUI
etc.	Google Earth. OpenSteetMap

제안하는 시각화 시스템을 구현하기 위한 환경은 윈도우즈 기반에서 C++언어로 구현되었고 OpenGL을 이용하여 그래픽을 표현하였다. OpenGL은 산업 표준화된 그래픽 라이브러리로서 향후 내장형 시스템등에 개발할 때 사용가능성이 유리하다는 장점을 가진다. 그리고 윈도우 기능 및 입출력 제어을 위하여 GLUT(OpenGL Utility Toolkit) 라이브러리를 이용하였다[18]. GLUT는 OpenGL 과 드라이브 사이에서 인터페이스 역할을 담당하는데 필요한 콜백함수를 등록하고 해당 콜백함수에 원하는 내용을 채워 넣기만 하면 이에 대한 호출은 GLUT가 알아서 처리하게 된다. 그리고 GLUI(OpenGL User Interface Library)는 GLUT에 기반한 C++ 사용자 인터페이스 라이브러리로 버튼, 체크박스, 라디오 버튼등의 고급 사용자 인터페이스를 제공한다[19]. 메인 화면의 입출력 요소에 대한 인터페이스 설계에 이용되었다. 지형 모델은 별도의 데이터베이스가 필요 없는 세가지 방법을 이용하였다. 먼저, 각 항공기가 비행되는 모습을 보이기 위하여 구글 어스의 실시간 맵을 이용하고 해당 비행 지역의 다양한 축척을 통한 사용자 대화 뷰를 위해서 OpenStrrtMap 정보를 온라인 쿼리에 의해서 획득하여 가시화한다[20]. 그리고 항공기의 전체적인 위치 표시를 위한 가시화는 텍스트 맵 정보를 이용하여 OpenGL을 통하여 직접 드로잉한다.

2. 구현 결과

구현된 전체 시스템의 화면 구조는 [그림 7]과 같이 항공기의 위치 정보를 알 수 있는 화면(A)과 특정 항공기의 지도 정보를 볼 수 있는 화면(B), 그리고 항적정보의 변이를 모니터링 할 수 있는 화면(C)으로 구성되고 좌측의 GLUI을 통한 인터페이스로 항공기의 선택과 확대 정도 및 경로 모니터링 여부를 조정할 수 있다.

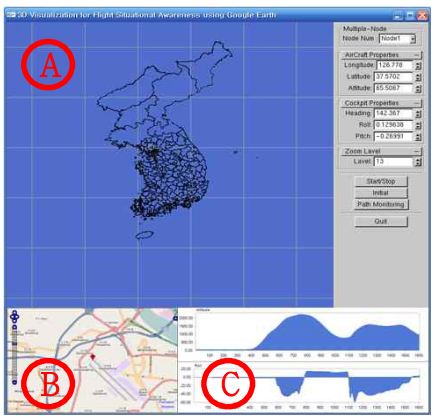

그림 7. 시스템의 전체 화면
Fig. 7 Screen of Visualization System

그리고 사용자 인터페이스의 Multiple-Node에서 특정 항공기 노드를 선택하면 [그림 8]와 같이 경도, 위도, 고도, 비행 자세정보를 이용하여 구글 어스에 연동된 별도의 가시화 화면이 동작하게 된다.

그림 8. 구글 어스 이용한 시각화
Fig. 8 Visualization Using Google Earth

항공기의 위치는 [그림 7]의 (A) 영역에 별도의 폴리곤을 이용한 형상 대신에 원형기호의 색상 정보를 이용하여 표시한다.

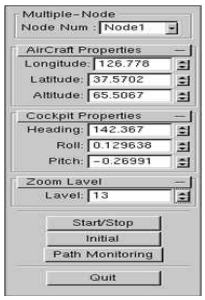

그림 9. 사용자 인터페이스 메뉴
Fig. 9 User Interface Menu

[그림 9]의 사용자 인터페이스 화면에서는 Multiple Node에서 항공기을 선택할 수 있고 각 항공기의 경도, 위도, 고도 정보는 Aircraft Properties 에서 실시간으로 확인 할 수 있고 계기판의 상태정보는 Cockpit Properties에서 확인 할 수 있다. 그리고 [그림 7]의 (B) 영역의 OpenSteetMap 확대 정도를 Zoom Lavel에서 조정할 수 있는데 조정 계수는 1부터 17사이의 정수 값으로 지정할 수 있다. [그림 10]은 서울 김포국제공항(Lat:37.570213, Lon:126.778526)의 Zoom Lavel을 10과 14로 지정했을 때의 표시 정보를 보이고 있다.

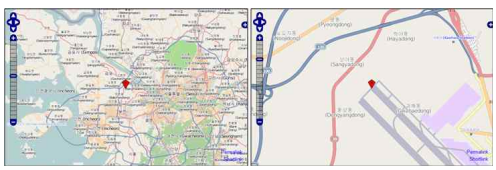

그림 10. OpenSteetMap의 레벨 조정(Zoom Lavel:10,14)
Fig. 10 Lavel Control of OpenSteetMap

항공기의 경로 궤적은 사용자 인터페이스 메뉴의 Path Monitoring 버턴을 선택하면 [그림 11]과 같이 [그림 7]의 (A)영역에 표시되고 각 항공기의 상태값에 대한 변화는 [그림 12]와 같이 [그림 7]의 (C) 영역에 시간의 변화에 따라 그래프 형태로 표시됨으로서 특정 시점에서 상태값의 변화가 급격히 변이하는 정도를 사용자가 직시할 수 있게 된다.

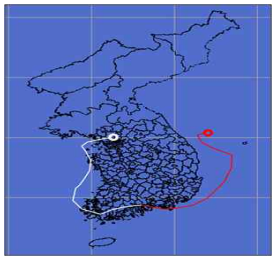

그림 11. 경로추적
Fig. 11 Path Tracking

이 화면을 통하여 불규칙적인 고도의 변화나 자세 변화를 확인하여 비행을 평가할 수 있고 향후 동일 영역비행에 대해 별도의 차트를 구성하여 비교 분석 작업을 거쳐 비행 개선에도 활용할 수 있다. 상태값 모니터링은 환경파일에 설정된 정보에 의해 나오게 되는데 기본적인 환경파일과 이에 해당하는 모니터링 뷰는 [그림 12]와 같다.

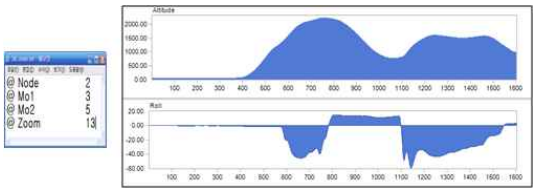

그림 12. 환경파일과 모니터링 뷰
Fig. 12 State File & Monitoring View

@Node 변수는 최대 항공기의 개수이고 DB에 저장된 순번에 의해 지정된다. @Mo1과 @Mo2는 모니터링 하고자 하는 항적 정보를 의미하며 요소에 따른 변수의 값은 [표 3]과 같다. [그림 12]에서는 Altitude와 Roll Angle정보를 표시한 예이다.요소 별 변수의 값은 사용자 요구에 따라 확장하여 표시할 수 도 있다. 그리고 @Zoom은 OpenSteetMap의 표시될 확대 레벨의 기본값을 의미합니다. 이 값은 사용자 인터페이스 메뉴에서 변경 가능합니다.

표 3. 모니터링을 위한 항적 정보
Table 3. Flight Information for Monitoring

요소명	변수 값
Latitude	1
Longitude	2
Altitude	3
Azimuth angle (Heading)	4
Roll Angle	5
Pitch Angle	6
Yaw Rate	7
RPM Engine	8

실제 비행 시각화를 위한 구글 어스의 동작은 별도의 화면에서 이루어지는데 구글 어스의 카메라 위치는 공유 메모리로 전달된 정보를 이용하여 IApplicationGE 인터페이스의 SetCameraParams() 메소드를 호출하여 구현하였다. 공유 메모리를 위한 응용 프로그램간의 통신은 호스트 프로그램에서 CreateFileMapping (INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE, 0, 1024, "FLIGHT_INFO") 을 이용하여 "FLIGHT_INFO"라는 공유메모리를 만든다. 그리고 구글 어스를 제어하는 프로그램에서 OpenFileMapping (FILE_MAP_ALL_ACCESS, FALSE, "FLIGHT_INFO")을 이용하여 데이터를 참조한다. 데이터의 읽고 쓰기는 MapViewOfFile을 이용하여 포인터를 생성하여 접근하면 된다. [그림 13]은 최종적으로 완성된 시각화 시스템의 통합 화면으로 서울 인근을 비행하는 과정을 시간 간격으로 캡쳐한 결과물로서 각 화면의 결과가 실시간으로 변화하는 것을 확인할 수 있다.

V. 결론

본 논문은 항공기의 비행 상황인식을 위한 효과적인 3차원 시각화 도구를 제안하였다. 복잡한 지형정보를 별도의 DB나 영상처리 기법없이 구글 어스와 OpenSteetMap을 이용하여 제공함으로서 경제적이고 활용성 높은 가시화 도구를 구현하였다. 또한 OpenGL모델을 이용하여 사용자에게 식별성 높은 화면을 제공한다. 또한 항공기의 비행경로를 추적하기 위한 별도의 궤적화면을 제공함으로서 항공기의 상황인식을 돕는다. 본 연구결과는 상황 인식을 위한 비행평가 및 개선에 이용될 수 있고 웹 기반의 개방형 도구를 사용함으로 적용성이나 활용도가 매우 높다고 할 수 있다. 향후 연구는 실제 항적정보를 실시간으로 전송을 받아 제안 시스템에 적용함으로서 실용적 가치를 높이는 연구 활동을 지속할 예정이다.

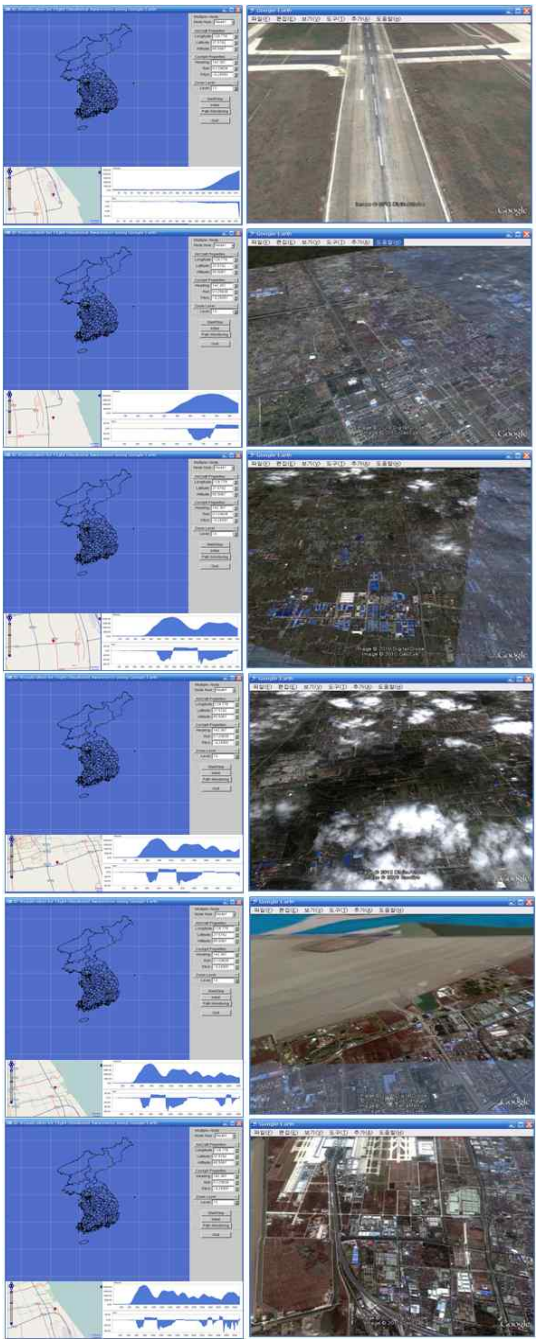

그림 13. 3차원 시각화 시스템의 통합
Fig. 13 Integration of 3D Visualization System

참고문헌

[1] 김성남, 최종인, 김창헌, 임철수, “공중작전 상황인식을 위한 3차원 가시화,” 정보과학회논문지:시스템 및 이론, 제32권, 제6호, 314-322쪽, 2005년 6월.

[2] 송진오, 박태진, 김종석, 최윤철, “항공기 상태전이정보를 이용한 비행경로 시각화 기법 연구,” 한국정보과학회 학술발표논문집, 제34권, 제2호(B), 172-177쪽, 2007년 10월.

[3] Google Earth COM API, http://earth.google.com/comapi

[4] 최진우, 양영규, “Google Earth를 이용한 택시 텔레매틱스 운행 이력 데이터 가시화 시스템의 설계 및 구현,” 대한원격탐사학회지, 제25권, 제1호, 61-69쪽, 2009년 2월.

[5] Shreiner, Dave, “OpenGL Programming Guide,” Addison-Wesley, 2009.

[6] Edward Angel, “Interactive Computer Graphics,” PEARSON, 2009.

[7] Hoppe, H. "Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering," IEEE Visualization'98 Conference, 35-42. October 1998.

[8] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L., Faust, N., Turner, G., "Real-Time, Continuous Level of Detail Rendering of Height Fields," ACM SIGGRAPH 96, p109-118, August 1996.

[9] Duchaineau, M., Wolinski, M., Sigeti, D., Miller, M., Aldrich, C., and Mineev-Weinstein, M. "ROAMing Terrain: Real-time Optimally Adapting Meshes," IEEE Visualization'97 Conference, 81-88, October 1997.

[10] P. Lindstrom. "Out-of-core simplification of large polygonal models," In SIGGRAPH 00,p256-262, July 2000.

[11] P. Lindstrom and C. Silva. "A Memory Insensitive Technique for Large Model Simplification," IEEE Visualization'01, October 2001.

[12] E. Shaffer and M. Garland. "Efficient adaptive simplification of massive meshes," IEEE Visualization'01, October 2001.

[13] 박명철, 김용해, 하석운, “혼돈이론을 응용한 예망어구에 대한 어류반응 행동모델의 수중현상 시각화,” 한국해양정보통신학회논문지, 제8권, 제3호, 645-653쪽, 2004년 6월.

[14] 박명철, 박석규, “OpenMP 병렬프로그램을 이용한 그물의 수중형상 시뮬레이션 구현,” 한국컴퓨터정보학회논문지, 제13권, 제2호, 11-17쪽, 2008년 3월.

[15] 박명철, 박석규 “혼돈이론을 이용한 선망어구에 대한 어군유영 시각화,” 한국컴퓨터게임학회논문지, 제17호, 145-149쪽, 2009년 6월.

[16] Martin C. Brown, “Hacking Google Maps And Google Earth,” John Wiley & Sons Inc, 2006.

[17] X-Plane SDK Documentation, *http://www.xsquawkbox.net/xpsdk/docs/*

[18] GLUT - The OpenGL Utility Toolkit, *http://www.opengl.org/resources/libraries/glut/*

[19] GLUI User Interface Library, *http://glui.sourceforge.net/*

[20] OpenSteetMap, http://www.openstreetmap.org/

저 자 소 개

박 석 규

2005년 : 경상대학교 컴퓨터과학과 공학박사

2001년 ~ 현재 : 강원도립대학 부교수

관심분야 : 소프트웨어 신뢰성, 시스템분석, 멀티미디어

박 명 철

2007년 : 경상대학교 컴퓨터과학과 공학박사

2007년 ~ 현재 : 송호대학 전임강사

관심분야 : 시뮬레이션, 임베디드 소프트웨어, 병렬프로그래밍 및 디버깅