공중작전 상황인식을 위한 3차원 가시화
(3D Visualization for Situational Awareness of Air Force Operations)

김 성 남 † 최 종 인 † 김 창 헌 †† 임 철 수 †††
(Seong-Nam Kim) (Jong-In Choi) (Chang-Hun Kim) (Cheol-Su Lim)

요 약 본 논문은 전장지역에서 벌어지는 공중작전에 대한 상황의 자료를 통합하여 3차원으로 가시화함으로써 공중작전의 최고결정권자가 올바른 상황인식 및 지휘결정을 할 수 있게 지원하는 시스템을 제안한다. 공중작전 상황판단을 위한 다양한 자료들, 레이더 항적정보, 비행계획 데이타베이스, 지도 및 위성사진 정보 등을 하나의 자료로 통합하여 3차원으로 가시화 시킴으로써, 작전의 최고 결정권자가 한 순간에 종합적인 작전 결정을 할 수 있는 정보를 지원하는 시스템이다. 대량의 데이타를 종합적이고 체계적으로 가시화 하기 위하여 Out-of-Core 방법을 사용함으로써 휴대용 노트북에서도 운용이 가능함을 실험을 통하여 보여 주고 있으며, 공군의 공중작전 수행 시 상황인식을 지원하는 시스템으로 활용될 수 있는 가능성을 보이고 있다.

키워드 : 상황인식, Out-of-core 방법, 지형 렌더링

Abstract This paper proposes a real-time 3D visualization system for situational awareness of Air force operations. This 3D system of situational awareness supports a high-level commander of Air force during the war game operations. These situation aware supporting data such as the aircraft track data of radar, aircraft schedule database, map and satellite image data are integrated into one structured data and those are visualized as 3D structure. By using an Out-of-Core method, we can visualize a 3D huge data in real-time in mobile notebook environment. The experiment shows several examples of 3D visualization supporting situation awareness for Air force operation.

Key words : Situational Awareness, Out-of-core Method, Terrain Rendering

1. 서 론

전장의 상황인식에 대한 필요성은 인류의 전쟁 역사와 그 맥을 같이 한다. 전장에 대한 상황인식의 중요성은 수 많은 전쟁 사에서 나타나듯이 최고 결정권자의 정확한 상황인식 여부에 따라 전쟁의 승패가 좌우되었기 때문이다. 고대에는 모래 위에 막대기로 적군과 아군을 표시함으로써 상황인식에 대한 가시화를 하여, 중요한 전략 및 전술을 계획하고 실행하였다. 중세를 거쳐 20세기 말까지는 전장 관찰자들로부터 원거리에 있는 지휘본부까지 각 시대에 가장 빠른 통신 수단-봉화, 파발, 전신, 전화, 위성통신 등-을 이용하여 전장 상황을 최고 결정권자가 볼 수 있도록 지도나 상황판에 그리는 수준이었다.

그림 1은 현재 공군에서 운용하고 있는 전장 상황을 표시하는 장비들이다. 공중감시를 위한 도플러 방식의 레이더를 통해 항적 정보를 아날로그 레이더 스코프에 시현하고, 이를 디지털로 가공하여 2차원으로 모니터에 표현하는 시스템, 지도·위성사진과 모든 비행 계획에 관련된 자료를 관리하는 작전 데이타 베이스를 운영하고 있으며, 조종사에 대한 훈련 및 자격을 관리하기 위한 조종사 인사 데이타 베이스를 운영하고 있다. 또한 통신장비를 이용하여 조종사와의 직접적인 대화를 통해 전장에 대한 상황을 인식하고 의사 결정을 하고 있다.

그러나 이러한 다양한 시스템들은 각각 자신의 목적에 맞게 개발되어 있어 이를 사용하는 최고 결정권자는 각각의 자료들을 파악하여 순간순간 발생하는 전장상황을 종합적으로 인식하기 어려울 뿐 아니라, 신속하고 정확하게 의사 결정을 내릴 수 없다.

† 비 회 원 : 고려대학교 컴퓨터학과
camsam@korea.com
gmcg2001@korea.ac.kr
†† 종신회원 : 고려대학교 컴퓨터학과 교수
chkim@korea.ac.kr
††† 비 회 원 : 서경대학교 컴퓨터공학과 교수
cslim@skuniv.ac.kr
논문접수 : 2002년 12월 30일
심사완료 : 2005년 4월 1일

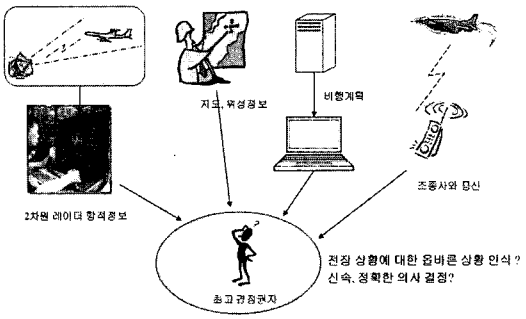


그림 1 현재 공군의 상황 인식 시스템 요소

이에 공군 최고 결정권자에게 전장에 대한 신속, 정확한 상황을 전달하여 보다 신속한 의사결정을 내릴 수 있도록 그림 2와 같이 전장에 대한 3차원 지형 자료와 공중 항적에 관한 데이타 베이스 자료, 그리고 지도 및 위성사진을 연동하여 하나의 통합된 가시화 인터페이스를 제안하고자 한다.

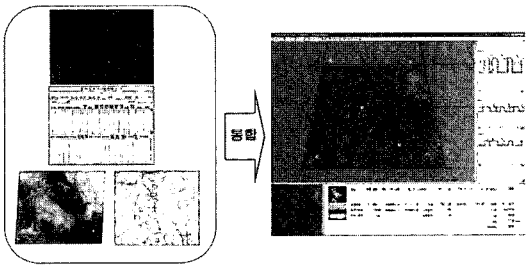


그림 2 전장 3차원 가시화와 공중작전 데이타 베이스와의 통합

2. 관련 연구

2.1 지형모델 가시화

실시간에 대규모의 지형정보를 최적으로 가시화하는 방법에는 시점에서 보이는 부분만을 선택하는 컬링 기법, 시점에 따라 가까운 곳 및 중요한 부분 등의 기준에 따라 레벨을 조절하는 LOD 기법, 대용량 자료를 처리하는 Out-of-core 기법, 지형 처리와 이미지 처리를 혼합한 기법, 장면 복잡도와 무관한 렌더링 성능을 위한 이미지 기반 렌더링 기법 등 여러 가지 방법들이 연구되고 있다.

LOD 기법은 전체 지형 중 뷰 프러스텀 내의 지형 중에서 인간의 시각인지에 가능한 충실한 영상이 표현되도록 선택 기준에 따라 필요한 만큼의 폴리곤으로 장면 복잡도를 조절하여 다양한 레벨로 전체 장면을 표현하는 기법이다. 즉, 인간의 시각이 인지할 수 없는 먼 거리에 있는 지형은 적은 수의 폴리곤으로, 인간의 시각이 인지할 수 있는 가까운 거리에 있는 지형은 많은 수의 폴리곤으로 표현하여 실제감을 떨어뜨리지 않으면서 렌더링 속도를 향상시키는 기법이다. 이러한 LOD 기법은 수 많은 연구가 이루어졌다.

Hoppe는 TIN을 이용한 시점 종속적 점진적 메쉬 방법으로 지형을 LOD로 표현하였는데, 이는 각 LOD 레벨을 그림 3과 같이 에지 축약과 정점 분할을 이용하여 최소의 폴리곤 개수로 최적의 간략화를 수행하였다[1]. 그러나 렌더링 속도가 최종 결과 메쉬의 크기에 비례함으로 메쉬 크기가 크면 클수록 수행 속도는 느려지며, 매 프레임마다 최적의 간략화를 찾아야 하는 단점이 있다.

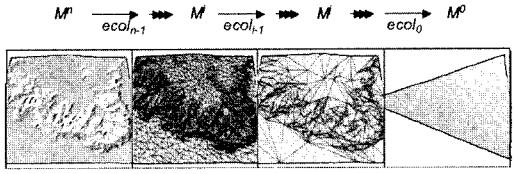


그림 3 TIN 구조의 점진적 메쉬 방법을 이용한 지형 모델 표현

Lindstrom 등은 quadtree 구조로서 그림 4와 같은 블록 간략화와 정점 간략화의 두 단계에 걸친 간략화 방법을 이용하여 전체 장면의 해상도를 조절하였다[8]. 그리고 화면 공간 오차 척도를 이용하여 화면 공간 임계 값 설정 값에 따라 다양한 삼각형 개수로 전체 장면을 표현할 수 있도록 하였다.

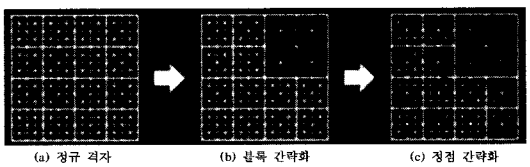


그림 4 블록 간략화 및 정점 간략화

그러나 상향식 간략화를 사용하므로 이미지의 영상품질은 좋아지지만, 하향식 품질 개선에 비해 계산시간이 많이 소요되는 단점이 있다.

Duchaineau 등은 이진 트리 구조로서 삼각형 다이아몬드 기반의 삼각형 분리, 병합을 수행하였다. 하향식 품질 개선 방식으로 단순화 효율성을 높였고, 정점 보간 오차를 우선순위 큐로 처리하여 메쉬를 점진적으로 개선하였다[2]. 삼각형 개수와 시점 방향에 의한 시스템 성능을 조절하며, 대부분의 게임 분야에서 적용하는 효율적인 알고리즘이다. Blow는 Duchaineau에 의한 결과가 시점에 가까운 지형과 실루엣에 집중되어 전체적인 이미지의 영상품질을 저하시키고, 병합과 분리 우선순위

큐에서 삼각형의 우선순위 계산 시간이 너무 많아 렌더링 성능이 저하되므로, 각 시점에서 모든 버텍스의 오차를 계산하는 대신, 각 버텍스 별로 등가면을 두어 시점이 포함되는 등가면의 버텍스를 중첩하여 활성화시키는 방법을 사용하였다[3].

그러나 위에서 언급된 LOD 기법들은 전체 장면을 표현하기 위해 해당 지역의 height field 값을 모두 메모리에 올린 상태, 즉 In-core에서 각각의 제안된 알고리즘을 적용하는 기법들이다[1-4]. 그러나 수백 기가 이상 대용량의 데이타를 처리해야 하는 3차원 지형모델의 특성상 대용량의 데이타는 하드 디스크에 저장하고 필요에 따라 특정 지역의 데이타를 메모리에 올려 도시하는 Out-of-core 가시화 방법이 요구된다. 따라서 In-core 방법들을 Out-of-core에 직접 적용하기에는 많은 제약이 있다. 예를 들어, LOD에서 모서리 축약이나 꼭지점 제거를 하기 위해서는 지역 연결성을 이용해야 하는데 이 부분이 메모리에 없을 경우에는 적용할 수 없다. 따라서 Out-of-core 기법은 전체 지형 데이타를 적절히 분할한 후 사용자의 시점에서 필요한 부분을 빠르게 메모리에 올리고 불필요한 부분을 제거하는 효율적인 페이징, 캐싱 기법을 적용해야 한다. Lindstrom 등은 데이타 접근 순서를 유지하는 효율적인 데이타 배치와 이에 대한 인덱스 기법을 개발하여 메모리 지역성을 높이고 페이징 되는 양을 줄이는 알고리즘을 제안하였다[5].

2.2 상황인식 가시화

컴퓨터의 급속한 기술발전에 힘입어 전장 상황인식을 위한 가시화에 대한 연구도 컴퓨터 그래픽스의 주요한 연구 주제가 되었다. 현재까지는 미국의 국방성이 중심이 되어 연구가 되고 있으나 상황인식 가시화라는 연구주제는 군 뿐만 아니라 상업적인 항공교통 관제, 지리정보 시스템, 3차원 게임 등 여러 분야에서 사용될 수 있다.

공중작전을 위한 전장에 대한 가시화는 광대한 지형과 함께 적기, 아군기, 민항기 등 여러 항적에 대한 내·외적인 속성자료와 같은 다양한 정보를 표시해야 한다.

Durbin과 Rosenblum은 responsive workbench를 이용하여 전장에 대한 상황인식 가시화 시스템인 드래곤을 개발하였다. 드래곤 시스템은 육, 해, 공군 전력을 모두 가상 이미지로 표시하고[6,7], 지형 모델의 경우에는 LOD-Sprite 기법을 이용하였다[8]. 또한 사용자의 편의성을 위해 pinchglove, 조이스틱, 음성 인식 장치 등과 같은 인터페이스를 사용하였다.

Hix 등은 드래곤 시스템을 개발하면서 요구된 실시간 전장 상황 인식 가시화를 위한 사용자 중심의 개발 및 평가 방법에 대해 기술하였다[9].

Feibush 등은 지구 전역에 대한 지형모델을 소개하고 [10], 관찰자의 시점 중심에 따라 가장 관심 지역을 플레이 박스로 구성, 약 1000,000Km2를 LOD기법을 적용하여 실시간으로 도시할 수 있는 JOVE(Joint Operations Visualization Environment) 시스템을 개발하였다.

그러나 위에서 언급한 이전의 상황인식 가시화 시스템은 가상 Workbench 라든지, SGI Infinite Reality 시스템과 같은 고가의 장비에서 구현될 수 있었으며, 비효율적인 Out-of-core 방법을 이용하여 메모리 사용량이 많았다.

3. 공중작전 상황인식 3차원 가시화

3.1 시스템 구조

그림 5는 본 논문에서 제안하는 공중작전 상황인식 가시화 시스템의 구조이다. 먼저 중앙에 공중작전 상황인식 서버를 두고, 여기에 가시화 모듈과 정보획득 모듈을 구성한다. 정보획득 모듈에서는 레이더에 포착된 각 항적에 대한 적아 식별 및 위치, 고도, 속도 등의 정보를 획득하고, 비행계획 작전 데이타 베이스와 조종사 인사 데이타 베이스 등 현재 운용되고 있는 공중 작전 데이타 베이스 자료 중 항적과 일치하는 자료를 추출한다. 가시화 모듈에서는 기하 공간 데이타를 이용하여 3차원 지형모델을 가시화하고, 현실감 및 사실감을 증가시키기 위해 또는 사용자의 목적에 따라 지도, 위성사진 등의 이미지를 텍스쳐 매핑한다. 또한 정보 획득 모듈에서 포착된 항적을 사용자가 쉽게 알 수 있도록 약속된 기호 또는 간략화된 3차원 항공기로 항적을 가시화한다. 이렇게 추출 및 가시화된 상황인식 자료는 사용자의 권한에 따라 지휘부용과 비행단용으로 나눠 지휘부의 경우 영공 전역에 대한 항적을 확인 할 수 있도록 하여 공중작전 상황을 정확히 인식 할 수 있도록 한다. 비행단의 경우에는 자신의 공역에 대한 항적만을 관리할 수 있다.

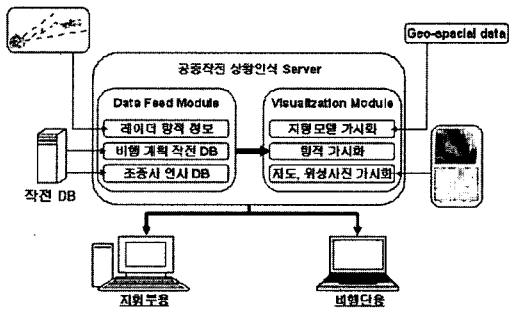


그림 5 시스템 구조도

3.2 항적 정보 획득 모듈

본 논문에서 항적정보를 획득하기 위한 모듈에는 크

게 세가지 서브모듈로 구성되어있다. 첫째, 도플러 레이더를 통한 항적정보 획득 모듈이다. 공군에서 현재 사용하는 레이더를 이용, 아날로그 항적정보를 컴퓨터에서 사용할 수 있도록 디지털화하여 레이더와 항적과의 상대적 3차원 위치를 계산, 3차원 지형모델에서 정확한 위치를 도시하는 모듈이다.

둘째, 각 항적에 대한 속성자료라 할 수 있는 비행계획 작전 데이타 베이스 모듈이다. 레이더에 포착된 항적자료는 항적에 대한 외적인 속성-적아 식별, 위치, 고도, 속도 등-만을 가지고 있다. 공중작전에 대한 정확한 상황인식을 위해서는 이러한 외적인 특성뿐만 아니라 내적인 속성 -비행임무, 임무공역, 보유 항공 탄 종류, 이·착륙 시간 등-도 매우 중요하다. 이를 작전 데이타 베이스를 통해 생성 및 추출한다. 마지막으로 조종사 인사 데이타 베이스 모듈이다. 내적인 속성 중 가장 중요한 항공기 탑승 조종사의 자료를 작전 데이타 베이스를 통해 생성 및 추출함으로써 어떠한 자격의 조종사가 어떤 임무로 어떤 항공기를 타고 있는가를 시스템을 통해 바로 알 수 있다.

이러한 정보획득 모듈을 통해 얻어진 자료는 가시화 모듈 중 항적을 가시화 하는데 사용된다. 실험을 위해 항적은 아군기 6대, 민항기 2대 적기 2대 등 총 10대를 생성하였으며, 각 항적이 계속적으로 3차원 지형모델에서 레이더에 포착된 것 같이 움직일 수 있도록 항적마다 3000 프레임을 생성하였다. 정보획득 모듈의 자료들은 모두 비밀 자료이므로 실험을 위해 그림 6과 같은 임의의 테이블을 생성하였으며, 실제 공군에 시스템을 적용 시에는 사용자 요구에 따라 테이블이 달라 질 수 있다.

3.3 가시화 모듈

가시화 모듈은 지형모델 가시화, 항적 가시화, 지도·위성사진 가시화 서브모듈로 구성되어 있다. 첫째, 지형모델 가시화는 기하 공간 데이타를 이용하여 전장지역에 대한 3차원 지형모델을 가시화하는 모듈이다. 이 부분은 대용량의 디지털 지형 자료를 실시간으로 렌더링 하기 위해 전장 지역을 사용자의 시각이 인지 할 수 없는 범위에서 여러 레벨로 간략화 하는 LOD 방법과 대용량 자료 처리를 위한 Out-of-core 방법을 사용한다. 둘째, 항적 가시화는 정보획득 모듈에서 추출된 항적 자료를 가시화하는 모듈이다. 마지막으로 지도·위성사진 가시화는 최고 결정권자의 전장 지역에 대한 현장감과 실제감을 높이기 위해, 지도 또는 위성사진을 3차원 지형모델 위에 텍스쳐 매핑 하는 모듈이다.

3.3.1 지형모델 가시화

지형모델 가시화는 PC 기반의 CPU, 메모리 성능으로 대용량의 디지털 지형 데이타를 실시간으로 가시화하는 것이다. 기존의 지형모델 가시화에 대한 논문들에서는 주로 In-core에서의 지형모델 가시화 방법들을 제안하였다[1-4,8].

그러나 수백 기가 이상 대용량의 데이타를 처리해야 하는 3차원 지형모델의 특성상 대용량의 데이타는 하드디스크에 저장하고 필요에 따라 특정 지역의 데이타를 메모리에 올려 도시하는 Out-of-core 가시화 방법이 요구된다. 따라서 In-core 방법들을 Out-of-core에 직접 적용하기에는 많은 제약이 있다. 예를 들어, LOD에서 모서리 축약이나 꼭지점 제거를 하기 위해서는 지역 연결성을 이용해야 하는데 이 부분이 메모리에 없을 경우에는 적용할 수 없기 때문이다.

본 논문에서는 3차원 지형모델 가시화를 위해 Lindstrom의 Out-of-core 방법을 이용하였다. 이 방법은 하향식 품질 개선 방식으로 어플리케이션에서 원하는 지형 데이타의 접근 순서에 근접한 순서로 데이타를 구조화하는 방법과 인덱스 계산 방법을 적용하여 실제 요구하는 페이징은 운영 체제가 담당하면서 메모리의 응집도를 높이는 방법이다[1]. 즉, 페이징은 논리적인 메모리 주소에 하드 디스크의 특정 파일을 할당해 주는 윈도우 시스템 함수인 MapViewofFile 함수를 사용하고, 효율적인 데이타 구조와 인덱스 계산을 통해서 메모리와 하드 디스크간의 페이징 이벤트를 줄여 렌더링 속도를 향상시키는 것이다. 또한 품질 개선 조건 기준을 만족하기

RADAR TRACK // RADAR TRACK 테이블

CS_NO	VARCHAR(10) // 호출부호
TR_NO	VARCHAR(4) // 트랙번호
FRAME_NO	VARCHAR(7) // 프레임번호
TR_X	VARCHAR(15) // X좌표
TR_Y	VARCHAR(15) // Y좌표
ALT	VARCHAR(6) // 고도
FOE	VARCHAR(1) // 피아식별

AIRCRAFT // AIRCRAFT 테이블

SQ	VARCHAR(5) // 대대
AC_NO	VARCHAR(15) // 항공기번호
AC_DT	VARCHAR(10) // 항공기
ETA	VARCHAR(16) // 이륙예정시간
ETD	VARCHAR(16) // 착륙예정시간
TO	VARCHAR(16) // 이륙시간
LD	VARCHAR(16) // 착륙시간
MSN	VARCHAR(20) // 임무
MSN_AREA1	VARCHAR(15) // 임무공역1
ORD1	VARCHAR(10) // 탑재무장1
CS_NO	VARCHAR(10) // 호출부호

PILOT // PILOT 테이블

RANK	VARCHAR(10) // 계급
SN	VARCHAR(10) // 군번
PILOT_NAME	VARCHAR(30) // 이름
SQ	VARCHAR(5) // 대대
QUALIFIED_AC	VARCHAR(10) // 주기종
QUALIFY	VARCHAR(20) // 자격
FLIGHT_TIME	VARCHAR(4) // 비행시간

그림 6 항적 정보 획득 모듈 임시 테이블

위해 기존의 알고리즘들이 모든 정점들을 방문하여 렌더링 속도를 떨어뜨렸던 단점을 극복하고자, Blow의 중첩된 구 알고리즘과 유사한 방법을 이용하여 정점들의 방문을 최소화한다[3]. 즉, 품질 개선 과정에서 모든 정점들을 방문하여 품질이 개선 될 조건, 즉 i번째 정점의 위치 Pi를 중심으로 하는 구를 하나 생성하고, 이 구는 i번째 화면 공간 오차 $\rho i = \rho(\varepsilon i, Pi, e)$의 iso-contour를 나타낸다. 여기서 εi는 물체 공간 오차며, e는 시점이다.

이러한 구(ρi) 값이 사용자가 정의한 오차 허용오차(τ)보다 작다면 자식 노드의 구로 품질 개선 계산을 반복한다. 만약 사용자가 정의한 오차 허용오차보다 크다면 품질 개선 과정은 종료된다.

- $\rho i < \tau$: refinement
- $\rho i > \tau$: termination

그러나 부모 노드의 화면 공간 오차(ρi) 값이 자식 노드의 화면 공간 오차보다 작다는 보장을 해 주어야 하므로 다음과 같은 속성을 만족해야 한다. 여기서 i는 부모 노드, j는 4개의 quadtree상의 자식 노드, Ci는 DAG(Directed Acyclic Graph)에서 자식 노드의 집합을 가리킨다[5].

$$\rho(\varepsilon_i, P_i, e) \geq \rho(\varepsilon_j, P_j, e) \qquad \forall_j \in C_i$$

그리고 효율적인 데이타 구조와 인덱스 계산을 위해서 전체 지형정보는 그림 7과 같이 quadtree 두 개를 중첩한 형태인 interleaved quad-tree(white(Qw), black(Qb))로 구성한다[5].

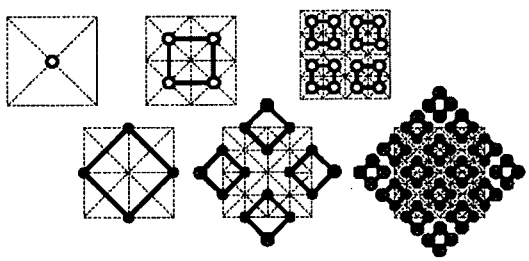


그림 7 Interleaved quad-tree(white and black quad-tree)

지리적으로 가까운 정점들은 메모리 내에 저장될 때도 가깝게 위치 되도록 하기 위해서 식 (1)과 같은 식으로 부모(p)와 자식(c), 그리고 자식간의 인덱스를 유도할 수 있다.

$$c(p,k) = 4p + k + m \quad with \quad k = 0,1,2,3 \qquad (1)$$

여기서 m은 연속적인 레벨간의 인덱스 거리 및 root 노드의 인덱스에 의존하는 상수이다.

그림 8은 interleaved quadtree를 이용하여 구성된 white quadtree와 black quadtree 각각 2 레벨씩 진행된 품질 개선 인덱스들을 나타낸다[5].

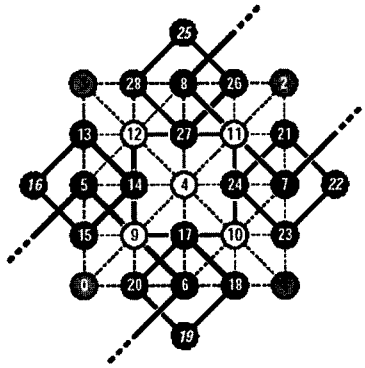
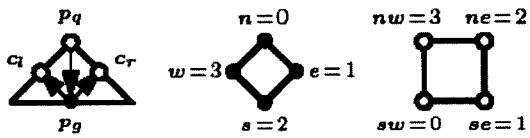


그림 8 Interleaved quadtree에 대한 인덱스

그리고 품질 개선 과정에서 추가적인 오버헤드를 막기 위해, 정점을 방문하는 순회과정에 필요한 효율적인 인덱스 계산 방법이 필요하다. 그래서 그림 9와 같은 품질 개선 구조에서 각 노드에 대한 명명 규칙을 이용하여 식 (1)의 인덱스 계산 방법을 개선한다[5].



(a) 품질 개선 (b) Black quadtree (c) White quadtree

그림 9 품질 개선 구조에서 각 노드에 대한 명명 규칙

그림 10(a)는 interleaved quadtree를 사용하였을 때 인덱스 계산을 위한 전이 테이블이다. 여기서 각각의 레코드의 필드들은 그림 9(a)를 참조한다. 즉, white quadtree Pq의 인덱스(kq)가 n일 때, black quadtree Pg의 인덱스(kg)는 sw, se, ne, nw와 같은 4개의 경우가 발생하며, 각각의 경우에 따라 Pq의 자식 노드인 cl, cr의 인덱스(kl,kr)가 테이블과 같이 나타나게 된다.

그림 10(b)는 (a)와 같이 2개의 quadtree(white quadtree, black quadtree) 각각에 대해 다른 인덱스 전이 테이블을 사용하던 방식을 하나의 전이 테이블로 단순화하고, 여기에 그림 9(b), 10(c)의 인덱스를 사용한 것이다. Interleaved quadtree에서 black에서 white로의 전이 테이블(그림 10(a) 위쪽)과 black에서 white로의 전이 테이블(그림 10(a) 아래쪽)의 인덱스를 뽑아서 비교하면 보면 그림 10(b)와 같은 하나의 테이블로 만들 수 있다.

이러한 단순화된 전이 테이블을 이용하여 식 (1)의 인덱스 계산 방법을 다음과 같이 개선함으로써 렌더링 성능을 개선한다.

K_l, K_r		Kg			
		n	*e*	*s*	*w*
Kq	*sw*	*se,ne*	*ne,nw*	*nw,sw*	*sw,se*
	se	*nw,sw*	*sw,se*	*se,ne*	*ne,nw*
	ne	*se,ne*	*ne,nw*	*nw,sw*	*sw,se*
	nw	*nw,sw*	*sw,se*	*se,ne*	*ne,nw*

K_l, K_r		Kg			
		sw	*se*	*ne*	*nw*
Kq	*n*	*e, s*	*s, w*	*w, n*	*n, e*
	e	*w, n*	*n, e*	*e, s*	*s, w*
	s	*e, s*	*s, w*	*w, n*	*n, e*
	w	*w, n*	*n, e*	*e, s*	*s, w*

(a) interleaved quadtree 전이 테이블

K_l, K_r		Kg			
		0	1	2	3
Kq	0	1, 2	2, 3	3, 0	0, 1
	1	3, 0	0, 1	1, 2	2, 3
	2	1, 2	2, 3	3, 0	0, 1
	3	3, 0	0, 1	1, 2	2, 3

(b) 단순화된 전이 테이블

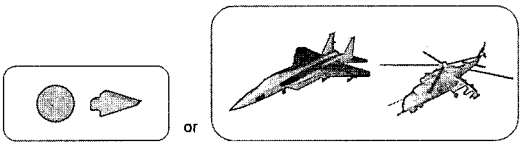
그림 10 전이 테이블

$$C_l(P_q, P_g) = 4P_q + ((2P_q + P_g + m + 1) \bmod 4) + m$$
$$C_r(P_q, P_g) = 4P_q + ((2P_q + P_g + m + 2) \bmod 4) + m$$

그리고 본 논문의 지형모델 가시화 모듈에서는 사용자가 지형의 고저를 쉽게 파악할 수 있도록 고도의 차이에 따라 다른 색을 표시하기 위해 해면고도부터 10,000m까지 칼라 테이블을 사용한다.

3.3.2 항적 가시화

항적에 대한 가시화는 일반적으로 그림 11과 같이 약속된 기호로 표현하는 방법과 실제 항공기와 같은 3차원 폴리곤 모델로 표시하는 방법을 사용할 수 있다. 그리고 항적을 약속된 기호로 표현하는 방법은 미 국방성 인터페이스 교범 중 일반 전투 기호 교범(MIL_STD_2525B)을 기초로 하여 표현할 수 있다. 본 논문의 실험에서는 가시성을 높이기 위해 간략화된 실제 항공기 메쉬를 사용하여 표현한다.



(a) 기호 표현 (b) 간략화된 3차원 실제 항공기 폴리곤 모델

그림 11 항적 가시화

3.3.3 지도, 위성사진 가시화

3차원 지형 모델에 사실감과 현장감을 높이기 위해 그림 12와 같이 지도나 위성사진을 텍스쳐 매핑한다. 각

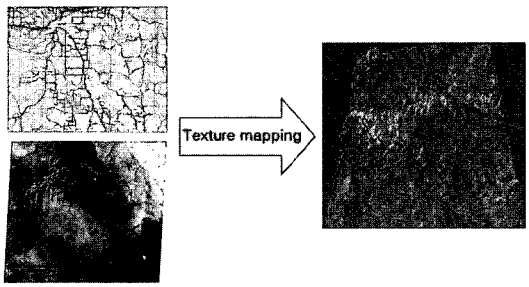


그림 12 지도, 위성사진 가시화

용도에 따라 다양한 지도와 위성사진을 사용하여 지휘관에게 정확한 위치파악을 지원할 수 있다. 지도와 위성사진의 경우에는 이미지이므로 전처리 단계에서 밉맵으로 구성하여 LOD의 레벨에 따라 다른 밉맵을 적용한다.

4. 실험 결과

공중 작전 상황 인식 가시화를 위한 실험 환경은 다음과 같다.

- CPU, 메모리, 그래픽 카드 : Pentium 4 1.4GHz, 640MB, GeForce2 pro 64MB
- 언어, 라이브러리, DBMS : Visual C++ 6.0, OpenGL 1.2, InterBase 5
- Height Field : 8193 * 8193
- 버텍스 용량, 데이타 용량 : 20Byte, 1.34GB

디지털 지형 모델은 http://www.usgs.gov에서 제공되는 DEM(Digital Elevation Model) 1:250,000 축척의

자료를 사용하였다. 각 DEM 자료는 1201 * 1201의 높이 데이타로 구성되어 있어, 실험 환경과 같은 x, y축 모두 2n + 1 자료로 만들기 위해 전처리 단계에서 8193 * 8183로 재구성하였다.

그림 13에서 보는 것과 같이 전체 시스템 화면을 4개의 윈도우로 분할하여 전장에 대한 3차원 가시화 및 항적에 대한 내·외적인 속성자료를 표현할 수 있도록 구현하였다.

3차원 지형 렌더링의 경우, 사용자가 정의한 허용오차(τ)를 2 픽셀로 하였을 경우 탐색 경로에 따라 삼각형의 개수는 대략 30,000~300,000개 렌더링 되었으며, 평균적인 렌더링 속도는 초당 20 프레임이었다. 그리고 항적을 가시화하는데 본 논문에서는 약속된 기호 대신 사실감을 증가시키기 위해 항공기당 2000개의 폴리곤으로 구성된, 간략화 된 실제 항공기 메쉬를 사용하였다.

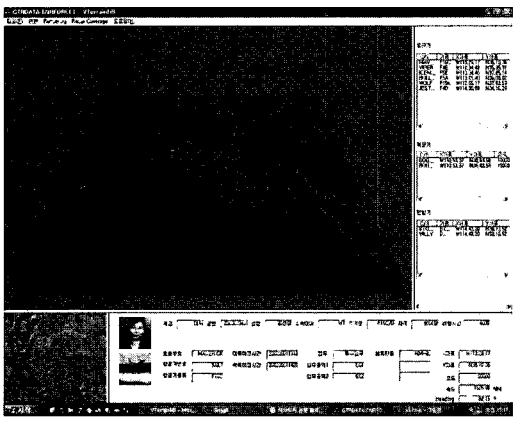


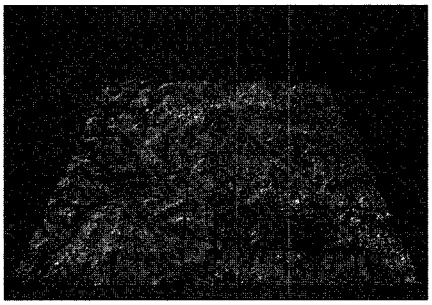
그림 13 공중작전 상황인식 가시화 전체화면

각각의 분할된 윈도우는 메인 윈도우, 아군기, 적기, 민항기 리스트 윈도우, 전체 지형 윈도우, 그리고 상세 데이타 베이스 윈도우로 구성되어 있다.

메인 윈도우는 그림 14에서 보는 것과 같이 전장 지역의 3차원 지형모델 가시화, 항적 가시화, 지도·위성사진 가시화, 지형 및 항적의 위치를 판단하기 쉽도록 위·경도 가시화, 그리고 레이더 범위 등과 같은 가시화할 수 있는 모든 자료들을 표현하는 윈도우다.

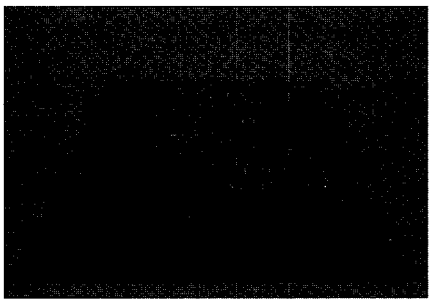
또한 지형의 고도에 따라 다른 색깔로 표현되도록 칼라 테이블을 적용하여 3차원 지형모델을 시각적으로 보다 보기 쉽도록 가시화하였다.

아군기, 적기, 민항기 리스트 윈도우는 그림 15와 같이 항적에 대한 리스트 박스 형태의 간단한 자료를 표현하였다. 리스트 박스는 아군기, 적기, 민항기로 구분하여 전장 지역에서 각각의 대수, 위치, 고도, 속도, 콜 사인 등의 정보를 표현하며, 최고 지휘결정권자에게 전장의 공중 항적에 대한 대강의 정보를 제공해 준다. 이 자



(a) 3D 지형 + 지도 텍스쳐

(b) (a) + 색상 테이블



(c) (b) + 항적

(d) (c) + 위·경도 + 레이더 범위

그림 14 메인 윈도우

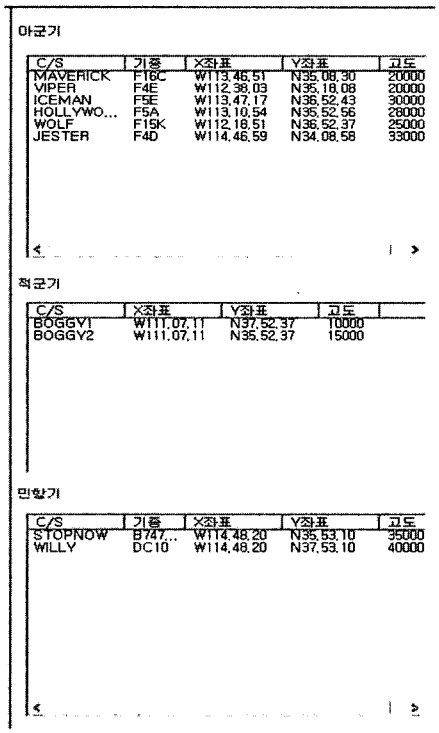


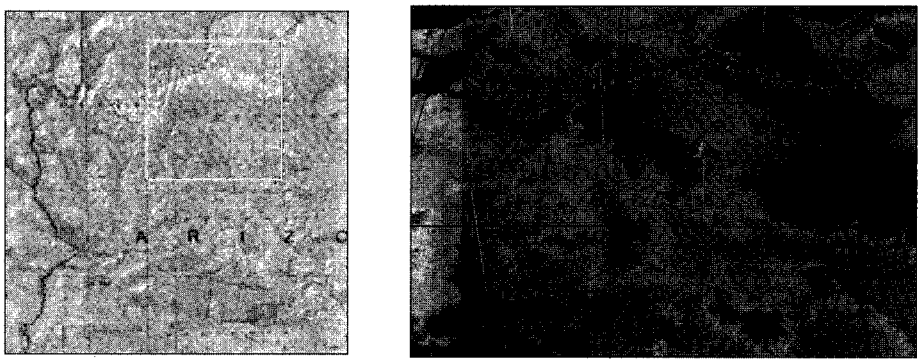
그림 15 아군기, 적기, 민항기 리스트 윈도우

료는 사용자의 요구에 따라 다른 정보를 표현할 수 있도록 하였다.

전체 지형 윈도우는 전처리 과정 중에 Top 윈도우 시점에서 고 해상도의 전체 지형 모델을 렌더링 후 이를 이미지로 저장하여 그림 14와 같은 전체화면 로드 시에 텍스쳐 매핑한 윈도우다. 전체 지형 윈도우는 그림 16(a)와 같이 전장 지역 전반이 아닌 특정 지역을 보다 자세하게 확대하여 볼 수 있도록 하기 위한 목적과 특정 지역을 선택하여 확대할 경우 전장 전체 지역에서 특정 지역이 어디인지를 쉽게 구분할 수 있도록 하기 위한 윈도우다.

사용자가 특정 지역을 상세히 보기 위한 작업은 단순히 전체 지형 윈도우에서 마우스를 이용하여 특정지역을 선택하기만 하면 되고, 이 작업의 결과는 그림 16(b)와 같이 메인 윈도우에 특정 지역이 확대되어 표현된다.

마지막으로, 상세 데이타 베이스 윈도우는 그림 17과 같이 각 항적에 대한 자세한 데이타 베이스 자료를 표현하는 윈도우다. 첫 레코드에는 조종사 이름, 자격, 주기종, 비행시간과 같은 항적의 내적인 속성 중 조종사에 관련된 항목을 자세히 나타내고 있으며, 둘째 레코드에는 항공기 종류, 임무, 탑재무장, 공역과 같은 항적의 내적인 속성 중 비행계획과 관련된 항목을 자세히 나타내



(a) 전체지형 윈도우 영역선택 (b) 특정지역 선택 후 메인 윈도우 결과 화면

그림 16 전체 지형 윈도우

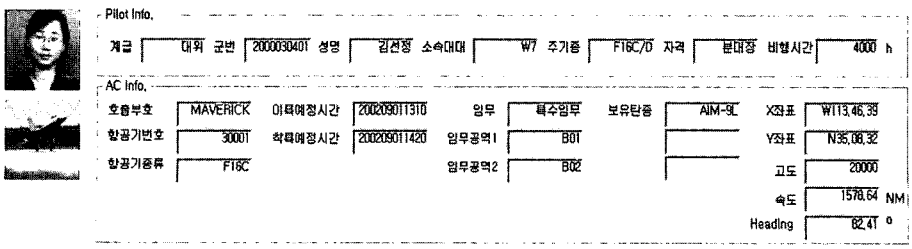


그림 17 상세 데이타 베이스 윈도우

고 있다. 또한 아군기, 적기, 민항기 리스트 윈도우에서 보여진 것과 같이 위치, 고도, 속도, 방향 등과 같은 항적에 대한 외적인 속성도 표현하고 있다.

또한 메인 윈도우에서 가시화된 특정 항공기를 선택할 경우에는 그 항공기에 대한 내·외적인 속성자료가 상세 데이타 베이스 윈도우에 가시화된다.

5. 결론 및 향후 과제

본 논문에서는 전장에서 벌어지는 상황을 원거리의 지휘관에게 신속, 정확하게 전달하여, 올바른 지휘결심을 하도록 지원하는 공중작전 상황인식 3차원 가시화 시스템을 구현하였다.

이를 위해 디지털 지형 자료를 이용하여 전장 지역을 3차원으로 가시화하였고, 2차원 레이더 항적 정보만 표시되던 기존 항적 정보 시스템을 간략화된 3차원 실제 항공기 폴리곤 모델로 가시화하였으며, 항적의 위치, 고도, 속도 등 외적인 속성뿐 아니라 공중 작전 데이타 베이스에서 비행계획 데이타 베이스 자료와 조종사 인사 데이타 베이스 자료를 생성 및 추출하여 항적에 대한 내적인 속성 자료를 연동하였다.

향후 과제로는 각각의 항적 시점 중심의 3차원 지형을 표시할 수 있도록 프로그램을 개선하고, 간략화된 3차원 실제 항공기 폴리곤 모델을 보다 빠른 속도로 구현할 수 있도록 연구할 예정이다. 또한 지도, 위성사진 가시화를 좀 더 보완하여, 고화질, 대용량 텍스쳐를 매핑하는 방법을 연구할 예정이다. 마지막으로 기하에 관련된 Out-of-core LOD 방법과 이미지와 관련된 이미지 기반 렌더링을 접목하여 랜더링 속도는 LOD와 비슷하면서 이미지의 영상 품질을 향상시킬 수 있는 방법을 연구할 예정이다.

참 고 문 헌

[1] H. Hoppe, "Smooth View-dependent Level-of-Detail Control and its Application to Terrain Rendering," Proc. of IEEE Visualization, pp. 35-42, Oct. 1998.

[2] M. A. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein, "ROAMing Terrain: Real-time Optimally Adapting Meshes," IEEE Visualization 97, 81-88, Nov. 1997.

[3] J.Blow, "Terrain Rendering at High Levels of Detail," GDC 2000.

[4] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. Turner, "Real-Time, Continuous Level of Detail Rendering of Height Fields," Siggraph 96 Conf. Proceedings, Holly Rushmeier, ed., Ann. Conf. Series, ACM Press, New York, Aug. 1996, pp. 109-118.

[5] P. Lindstrom, V. Pascucci, "Visualization of large terrains made easy," in IEEE Visualization 2001, Kenneth I. Joy, Amitabh Varshney, and Thomas Ertl, Eds., San Diego, California, Oct. 2001, pp. 363-370.

[6] J. Durbin, J. Edward Swan, Tony King, "Battlefield Visualization on the Responsive Workbench," *Proc. IEEE Visualization 98*, ACM Press, New York, Oct. 1998, pp. 463-466.

[7] L. J. Rosenblum, J. Durbin, R. Doyle, R. King and D. Tate, "Situational Awareness Using the VR Responsive workbench," IEEE Computer Graphics and Applications, Vol.17, No.4, July/Aug 97, pp.12-13.

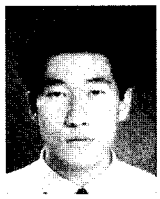
[8] Baoquan Chen, J. Edward Swan, Eddy Kuo, and Arie Kaufman, "LOD-Sprite Technique for Accelerated Terrain Rendering," Proc. of IEEE Visualization, pp.291-298, 1999.

[9] D. Hix, Mike McGee, J. Edward Swan, "User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment," Proc. IEEE Virtual Reality 99, IEEE Computer Society Press, Los Alamitos, Calif., 1999, pp. 96-103.

[10] Eliot Feibush, Nikhil Gagvani, Daniel Williams, "Visualization for Situational Awareness," IEEE Computer Graphics and Applications, Vol. 20, No. 5, Sep/Oct 2000, pp 38-45.

김 성 남

2003년 고려대학교 컴퓨터학과 대학원(석사). 관심분야는 컴퓨터 그래픽스



최 종 인

2003년 고려대학교 컴퓨터학과 대학원(석사). 2003년~현재 고려대학교 컴퓨터학과 대학원(박사과정). 관심분야는 컴퓨터 그래픽스

김 창 현

1979년 고려대학교 경제학과(학사). 1993년 Tsukuba 대학 전자정보학과(박사) 1979년~1987년 한국과학기술연구소(KIST) 연구원. 1987년~1995년 한국과학기술원 시스템공학연구소 책임연구원. 1989년~1990년 일본 동경공업대학 객원교수. 2003년~2004년 UCLA 객원교수. 1995년~현재 고려대학교 정보통신대학 컴퓨터학과(교수). 1996년~현재 한국컴퓨터그래픽스학회 부회장. 관심분야는 컴퓨터 그래픽스

임 철 수

1985년 서울대학교 계산통계학과(학사) 1988년 인디애나주립대 전산과 대학원(석사). 1995년 서강대학교 전산과 대학원(공학박사). 1985년~1986년 ㈜데이콤 정보통신연구소 연구원. 1988년~1994년 ㈜아시아나항공 시스템 센터 선임연구원 1994년~1997년 ㈜신세기통신 기획실 차장. 1997년~현재 서경대학교 컴퓨터공학과 교수. 관심분야는 멀티미디어 영상처리 및 신호처리