

공개형 3D 그래픽 엔진을 활용한 선박 운항 시뮬레이터용 실시간 3D 가시화 시스템 개발

황태현* · 오재용** · 황호진*** · 김병철**** · †문두환

*** 한국해양연구원 해양안전방재기술연구부 선임연구원 선임연구원
*** 한국해양연구원 해양안전방재기술연구부 연구원
**** 삼성중공업 산업기술연구소 책임연구원
† 경북대학교 기계자동차공학부 조교수

Development of a 3D real-time visualization system for ship handling simulators using an open source 3D graphics engine

Tae Hyun Fang* · Jaeyong Oh** · Ho-Jin Hwang*** · Byung Chul Kim**** · †Duhwan Mun

*, **, ***Maritime and Ocean Engineering Research Institute, KORDI, Daejeon 305-343, Korea
****Institute of Industrial Technology, Samsung Heavy Industries, Co., Ltd., Daejeon 305-380, Korea
†School of Mechanical and Automotive Engineering, Kyungpook National University, Gyeongsangbuk-do 742-711, Korea

요 약 : 선박운항자의 인적사고 방지 및 운항 훈련을 위한 방안으로 선박 운항 시뮬레이터가 사용된다. 선박운항 시뮬레이터에서 실시간 3D 가시화 기술은 현실감 있는 직관적인 영상을 제공하여 피교육자에 인지력을 향상시켜 시뮬레이터의 교육 효과를 높이는 중요한 요소이다. 본 연구에서는 공개형 3D 그래픽 엔진을 기반으로 선박 운항 시뮬레이터용 실시간 3D 가시화 시스템을 설계하고 구현한 결과에 대해서 설명한다. 실시간 3D 가시화 시스템은 요소 기능, 기존 그래픽 데이터 활용, 타 시스템과의 연동 측면에서 도출된 운용 요구사항들을 만족하며 추가적인 기능의 확장이 용이한 구조로 설계되었다.

핵심용어 : 공개형 그래픽 엔진, 선박 운항 시뮬레이터, 실시간 3D 가시화

Abstract : A ship handling simulator is popular means of preventing marine accidents caused by human error. It can also be used to train navigators. A real-time 3D visualization system, a component of a ship handling simulator, is an important component, as realistic and intuitive image generation play an essential role in improving the effects of education using ship handling simulators. This paper discusses the design of a new real-time 3D visualization system based on an open source 3D graphics engine as well as its implementation. The developed real-time 3D visualization system satisfies the operational requirements derived in terms of visualization functionalities, reuse of legacy graphic data, and interoperability with other systems constituting a ship handling simulator. This system has an architecture in which new functionalities are easily added.

Key words : Open source 3D graphics engine, Ship handling simulator, Real-time 3D visualization

1. 서 론

선박, 항공기, 자동차와 같이 사고 발생 시 막대한 인적 및 물적 피해가 발생하는 운송 시스템의 경우, 사고 예방을 위해서 운항자(조종사, 운전자)의 교육이 중요하다. 그러나 이 운송 시스템들은 그 구조가 복잡하여 전문적인 교육이 필요하고 장비의 가격이 고가여서 실제 장비를 가지고 교육을 하

기가 어렵다. 따라서 운송 시스템에 대한 인적사고 방지 및 운전 훈련을 위한 방안으로 다양한 시뮬레이터 기술(Magee, 1997; Murai et al., 2004; Wallace and Northam, 1997; Kuhl et al., 1995)이 적용된다.

시뮬레이션 가시화는, 응용 목적에 맞는 추상적인 모델을 정의한 후, 정의된 모델과 외부 객체 및 환경과의 상호작용을 전산도구를 이용하여 수치 연산을 수행하여 계산한 결과

† 교신저자 : 연희원, dhmun@knu.ac.kr 054)530-1271

* 대표저자 : 연희원, thfang@moeri.re.kr 042)866-3625

** 연희원, oiyong@moeri.re.kr 042)866-3648

*** 연희원, hjhwang@moeri.re.kr 042)866-3645

**** 연희원, bc7871.kim@samsung.com 042)865-4368

(주) 이 논문 내용의 일부는 “선박 운항 시뮬레이터를 위한 실시간 3D 가시화 시스템의 설계”란 제목으로 “2010공동학술대회 한국항해항만학회논문집(북포해양대학교, 2010.4.4.22-24)”에 발표되었음.

를 사용자에서 시각적으로 모사하는 것이다. 시각 정보는 사용자와의 상호작용을 위한 대표적인 인터페이스로서 현실감 있는 직관적인 시뮬레이션 가시화는 피교육자에 인지력을 향상시켜 시뮬레이터의 교육 효과를 높이는 중요한 요소이다. 이와 같은 가시화 기술은 비단 시뮬레이터 뿐만 아니라 제조 시뮬레이션(Mujber et al., 2004)이나 유지보수 시뮬레이션(Bian et al., 2006)에도 중요한 기술이다.

조선해양 시뮬레이션 가시화를 위해서는 해양과의 모사(Tessendorf, 1999; Lee and Han, 2010), 해수면에서의 광원 처리(Jensen and Goliias, 2001), 해양 환경의 가시화, 해양과 동적 객체(예를 들어 선박)와의 상호작용(Mitchell, 2005; Glassner, 2002) 등의 특수한 가시화 기능들이 필요하나, 상업용 3D 그래픽 엔진을 활용한 가시화 방법으로는 대응에 한계가 있다(황·문, 2009). 또한 상업용 3D 그래픽 엔진은 시스템 내부의 폐쇄성으로 인해서 수정 또는 확장이 제한돼 최신 그래픽 기법들의 적용이 어렵고 조선해양 분야에 특화된 기능 구현에 제약을 받는다.

한국해양연구원에서 개발한 선박 운항 시뮬레이터(Gong et al., 2004)의 실시간 3D 가시화 시스템도 상업용 그래픽 엔진인 VEGA Prime(Presagis, 2009)을 기반으로 개발되었기 때문에 조선해양 시뮬레이션 고유의 가시화 요구에 대응하기 위해서 시스템의 기능을 확장하거나 최신 그래픽 기술을 적용하는데 제약이 있다.

상업용 3D 그래픽 엔진과 다르게 공개형 3D 그래픽 엔진은 소스가 공개되어 있어 개발자가 새로운 가시화 요구에 능동적으로 대처할 수 있는 장점이 있다. 또한 시스템의 확장을 통해 최신 그래픽 기술의 적용이 가능하다. 조선해양(차등, 2009)뿐만 아니라 타 분야에서도 실시간 3D 가시화 목적으로 공개형 3D 그래픽 엔진을 적용한 다수의 사례(Burns and Osfield, 2004; Friese et al., 2008; Roth et al., 2004)를 찾을 수 있다.

이 논문에서는 공개형 그래픽 엔진인 Ogre3D(Ogre3D, 2010)를 기반으로 선박 운항 시뮬레이터용 실시간 3D 가시화 시스템을 개발한 결과에 대해서 설명한다. 개발된 실시간 3D 가시화 시스템을 한국해양연구원의 선박 운항 시뮬레이터와 연동을 하여 검증 실험을 수행하였다.

2. 선박 운항 시뮬레이터의 3D 가시화 요구

2.1. 선박 운항 시뮬레이터의 구성

선박 운항 시스템의 주요 구성은 Fig. 1과 같다. 선박 운항 시스템은 크게 선교와 통제실로 나뉜다. 선교에는 사용자 조작부(조타기 및 엔진), 항해지원 디스플레이, 실시간 3D 가시화 시스템이 있다. 통제실에는 Motion Solver와 IOS(instructor operating station)가 있다.

해상 상태에 따른 각 선박의 거동은 선박의 운동 모델에 따라 Motion Solver에서 계산된다. Motion Solver는 계산을

하기 위해 필요한 정보를 IOS를 통해 받은 후 계산 결과를 다시 IOS로 보낸다. IOS는 Motion Solver의 계산 결과를 바탕으로 갱신된 시뮬레이션 데이터를 시뮬레이터의 사용자 인터페이스(항해지원 디스플레이, 실시간 3D 가시화 시스템)에 전송한다.

시뮬레이터를 구성하는 시스템들 간의 통신을 위해 TCP/IP(Transmission Control Protocol/Internet Protocol)나 UDP/IP(User Datagram Protocol) 방식이 사용된다. IOS는 브로드캐스트(broadcast) 방식의 UDP 통신을 통해 항해지원 디스플레이나 실시간 3D 가시화 시스템에 갱신된 시뮬레이션 데이터를 전송한다.

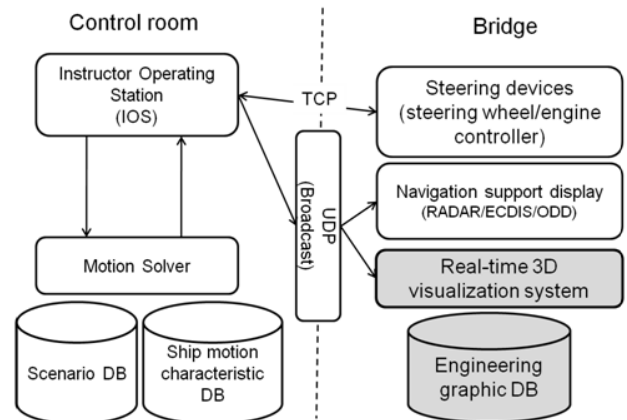


Fig. 1 System configuration of a ship handling simulator

2.2. 실시간 3D 가시화 시스템의 요구사항

실시간 3D 가시화 시스템의 설계를 위해서 선박 운항 시뮬레이터의 가시화 요구를 분석하였다(문 등, 2010). 실시간 3D 가시화 시스템의 입력, 출력, 사용 기법, 참조 데이터의 4가지 측면에서 시스템의 요구사항을 분석하였다. 분석 결과 실시간 3D 가시화 시스템의 운용 요구는 다음과 같이 3가지 종류로 분류가 되었다.

- 기능: 조선해양 시뮬레이션을 위한 공개형 그래픽 엔진 응용 가시화
- 통합: 선박 운항 시뮬레이터의 타 시스템들과의 연동
- 데이터: 항구, 선박, 부표, 등대에 대한 기존 3D 그래픽 데이터의 변환

본 절의 이후에는 실시간 3D 가시화 시스템의 요구사항에 대해서 살펴본다.

1) 실시간 3D 가시화 시스템의 주요 기능

실시간 3D 가시화 시스템은 Table 1에 나열된 3D 그래픽 기능을 제공하면서 실시간 가시화가 가능해야 한다.

바다 모사의 사실감에 대한 대표적인 평가 척도는 황천(약천후) 시의 백파(white cap, white breaker cap)의 모사이다. 동적 개체 애니메이션은 별도의 저작 도구를 사용하여

그래픽 모델에 삽입한 애니메이션을 가시화 시스템에서 선택/실행/중지를 하는 기능을 말한다.

실시간 3D 가시화 시스템의 GUI로는 문자열 출력, 2D 오버레이 (overlay)가 있고 데이터 로깅 기능을 지원해야 한다. 그리고 해양 및 항만 환경 데이터와 선박 데이터는 크기가 크면서 종류가 다양하기 때문에 대용량 데이터 처리 기법이 적용되어야 한다.

Table 1 Graphic capabilities required for maritime simulation

Item	Sub item	Details
Ocean waves	Ocean wave simulation	Spectral model-based ocean wave generation
	Stern/bow wakes	2D or 3D-based wakes
	Infinite grid for ocean wave	Projective, rectangular, circular type grid generation
	Sea surface optics	Reflection, refraction, godray, caustics
Dynamic objects	Ship search lights	
	Ship/buoy floating effect	Simulation of traffic ships' or buoys' motion affected by ocean waves
	Dynamic object animation	Ship navigation lights On/OFF, lighthouse On/OFF, Ship's sub system animation (e.g. radar rotation)
Environments	Weather	Clear/fog/cloud/rain/snow
	Time-elapse	Day/night/sunrise/sunset/dust
Scene	Scene graph management	
	Scene LOD and culling	

실시간 3D 가시화 시스템은 단일 PC로도 구동할 수 있어야 하며, 그래픽 클러스터 (멀티 채널)(Kim et al., 2006; 차등, 2010) 기능을 지원해야 한다. 그래픽 클러스터 구현 시 요구되는 에지 블렌딩 (edge blending)을 위해서, 하드웨어적으로 처리하는 방법과 가시화 시스템 내부에서 처리하는 방법 중에서, 하드웨어를 이용하여 처리하는 방법을 적용한다.

2) 선박 운항 시뮬레이터의 타 시스템과의 연동

실시간 3D 가시화 시스템은 선박 운항 시뮬레이터의 IOS 부터 브로드캐스트 방식의 UDP 통신을 통해 가시화에 필요한 데이터를 받는다. 따라서 선박 운용 시뮬레이터를 구성하는 타 시스템들과 상호운용성을 보장하기 위해서 실시간 3D 가시화 시스템은 UDP 통신 패킷 (packet)에 대한 처리가 가능해야 한다.

3) 기 구축된 3D 그래픽 데이터의 활용

실시간 3D 가시화 시스템에서 사용되는 그래픽 데이터로는 항구 3D 데이터, 자선 3D 데이터, 타선 3D 데이터, 부표 3D 데이터, 등대 3D 데이터가 있다. 동일 함선이라도, 요구

되는 상세도의 차이로 인해서, 자선일 때와 타선일 때 사용하는 3D 그래픽 데이터 파일이 다르다. 항구별 부표와 등대는 별도의 리스트 파일로 관리한다. 그리고 해양 환경 설정을 위한 파일과 선박 설정 파일이 있다.

조선해양 분야의 시뮬레이션 시 주로 사용되는 상업용 그래픽 엔진인 VEGA Prime의 경우 입력 데이터는 Multigen Creator (Presagis 2009)의 FLT 형식이다. 그러나 공개형 그래픽 엔진인 Ogre3D(Ogre3D, 2010)는 고유의 형상 및 재질 데이터 형식을 가진다. 따라서 실시간 3D 가시화 시스템에서 사용하는 그래픽 데이터 형식이 기존의 형식과 다르기 때문에 기존 3D 그래픽 데이터의 변환 절차에 대한 정립이 필요하다.

3. 선박 운항 시뮬레이터용 실시간 3D 가시화 시스템의 구조

3.1. 실시간 3D 가시화 시스템의 유즈 케이스 (use case)

선박 운항 시뮬레이터용 실시간 3D 가시화 시스템의 개발을 위해서 먼저 시스템의 사용 시나리오를 Fig. 2와 같이 정의하였다.

통제실의 관리자가 IOS와 Motion Solver를 실행하고 선교의 사용자가 가시화 시스템을 실행한다. 가시화 시스템을 시작할 때 필요한 설정 정보는 별도의 파일에 저장하고 사용자가 설정 값을 변경할 수 있다. 선박 운항 시뮬레이션 과정에서 사용자는 필요할 경우 카메라의 시점을 변경하거나 시뮬레이션 화면을 복사하여 이미지로 저장한다.

IOS는 사용자의 조타기 조작 정보를 받아 선박 운항 시뮬레이션을 수행한 후 그 결과를 가시화 시스템에 전달한다. 관리자가 선박 운항 시뮬레이션의 종료 명령을 내리면 IOS는 가시화 시스템을 종료한다.

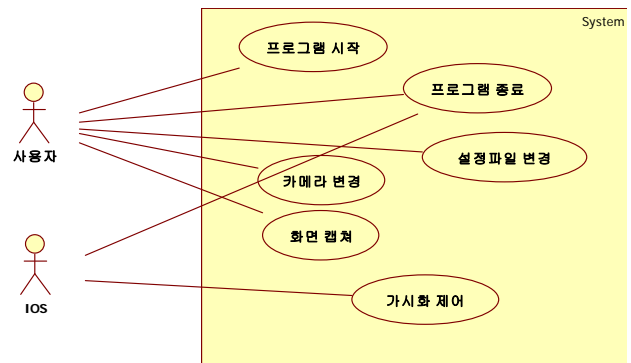


Fig. 2 Use case of a real-time 3D visualization system for a ship handling simulator

3.2. 실시간 3D 가시화 시스템의 구조

1) 데이터 동기화

선박 운항 시뮬레이터에서 실시간 3D 가시화 시스템의 역

할은 선박 운항 시뮬레이터의 IOS와 연동을 통해 IOS로부터 받은 시뮬레이션 결과를, 그래픽 클러스터 기술을 활용하여, 하나 또는 그 이상의 화면 (채널)에서 가시화하는 것이다. 그래픽 클러스터 기법을 적용하면 가시화 시스템은 하드웨어적으로 여러 대의 컴퓨터로 구현이 된다. 이에 따라 가시화 시스템 내의 컴퓨터들간에 데이터 동기화를 어떻게 하느냐에 따라 가시화 시스템의 전체적인 구조가 영향을 받는다.

다채널 가시화를 지원하는 선박 운항 시뮬레이터용 가시화 시스템에서 데이터 동기화는 1) IOS로부터 받은 시뮬레이션 결과 데이터에 대한 동기화, 2) 가시화 시스템에서 사용하는 3D 그래픽 데이터 및 다른 설정 데이터에 대한 동기화로 세분화된다. 이 연구에서는 동기화를 위해 Fig. 3과 같은 시스템 아키텍처를 구성하였다.

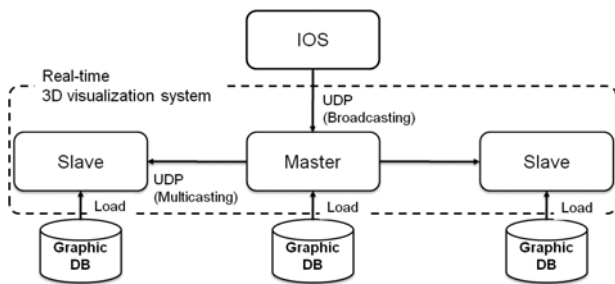


Fig. 3 Data synchronization of a real-time 3D visualization system which supports the graphic-cluster capability

IOS로부터 받은 시뮬레이션 결과 데이터의 동기화를 위해서, 가시화 시스템을 구성하는 컴퓨터 중 하나를 마스터 (master)로 지정하고 다른 컴퓨터들은 슬레이브 (slave)로 지정하는 방법을 사용하였다. 마스터는 IOS로부터 시뮬레이션 결과 데이터를 받아 가시화에 필요한 연산을 수행한 후, scene 그래프를 구성하는데 필요한 객체 정보와 카메라의 갱신 정보를 슬레이브에 전달한다. 마스터에서 수행하는 연산 작업이란 scene 그래프 상의 객체 위치 및 방향 정보를 IOS의 시뮬레이션 주기와 가시화 시스템의 프레임 주기의 차이를 반영하여 보간하는 것이다. 슬레이브는 IOS와 연동을 하지 않고 마스터로부터 받은 정보를 사용하여 보간 없이 scene 그래프를 갱신한다.

2) 실시간 3D 가시화 시스템의 구성 모듈

선박 운항 시뮬레이터용 실시간 3D 가시화 시스템의 운용 요구사항으로부터 가시화 시스템의 구성 모듈을 Fig. 4와 같이 정의하였다.

그래픽 클러스터 방식의 다채널 가시화를 위해서 가시화 시스템은 설정 정보에 따라 마스터 또는 슬레이브의 역할을 수행해야 한다. 이를 위해서 Application Role 모듈을 정의한다.

Multichannel Synchronization Manager는 그래픽 클러스

터를 구성하는 컴퓨터 사이에서 시뮬레이션 데이터의 동기화를 관리한다. 그래픽 클러스터의 마스터는 Multichannel Synchronization Manager를 이용하여 화면 갱신에 필요한 데이터를 멀티캐스팅 방식으로 슬레이브로 보내고, 슬레이브는 Multichannel Synchronization Manager를 이용하여 마스터가 보낸 데이터를 받는다. IOS Packet Receiver는 선박 운항 시뮬레이터의 IOS가 브로드캐스팅 방식으로 송신하는 시뮬레이션 결과 데이터를 받아 미리 정의한 C 언어의 구조체에 저장한다. 구조체에 저장된 시뮬레이션 결과 데이터는 시스템의 다른 모듈들이 사용한다.

Maritime Scene Manager는 실시간 3D 가시화 시스템의 scene 그래프를 관리하는 모듈이다. Scene 그래프 생성 시 필요한 3D 그래픽 데이터의 로딩 및 설정 데이터의 로딩을 위해서 각각 Graphic Data Loader와 ConfigManager가 사용된다. 그래픽 데이터나 설정 데이터가 XML 형식으로 인코딩 되어 있을 경우 이 데이터의 처리를 위해서 XML Parser를 사용한다. 그리고 시스템의 작동 상태를 외부 파일에 저장하는 Logger가 필요하다.

가시화 요소 기능과 관련해서는 해양과 자체를 모사하는 모듈인 Ocean Wave, 선박과 해양과의 상호작용으로 발생하는 선수 및 선미 궤적을 모사하는 모듈인 Stern Wake Effector, Bow Wake Effector가 있다. 그리고 주변 환경 (하늘과 시간의 흐름)을 모사하기 위해서 Environment 모듈과 각종 기상상태를 모사하기 위한 Weather Effector가 필요하다. 이 외에도 선박 및 부표 등의 동적 객체에 대한 단순화된 운동 모델을 사용하여 해양파에 의한 선박 및 부표의 운동 (종동요, 횡동요, 상하동요)을 모사하는 모듈인 Motion Solver가 필요하다.

3) 실시간 3D 가시화 시스템의 기본 구조

조선해양 시뮬레이션 가시화 요구사항을 바탕으로 도출된 실시간 3D 가시화 시스템 구성 모듈들을 참조하여 실시간 3D 가시화 시스템의 기본 구조를 설계하였다. 실시간 3D 가시화 시스템은, Fig. 5와 같이, functional element layer, system management layer, network communication layer, application layer의 4 개의 계층으로 구성되는 시스템 구조를 가진다.

Functional element layer는 조선 해양 시뮬레이션 가시화 필요한 요소 기능을 제공하는 계층으로 실시간 3D 그래픽 엔진을 기반으로 구현된다. System management layer는 scene 그래프 관리, 데이터 로딩, 시스템 설정, 데이터 로깅 기능 등의 시스템 관리 기능을 제공하는 계층이다. Network communication layer는 선박 운항 시뮬레이터의 타 시스템과의 연동이나 그래픽 클러스터 내 컴퓨터들 간의 데이터 동기화를 위해 필요한 통신 기능을 제공하는 계층이다. Application layer는, 그래픽 클러스터 내 각 컴퓨터의 역할에 따라, 하위 계층의 기능을 조합하여 역할에 맞는 시스템 수준의 응용 기능을 제공하는 계층이다.

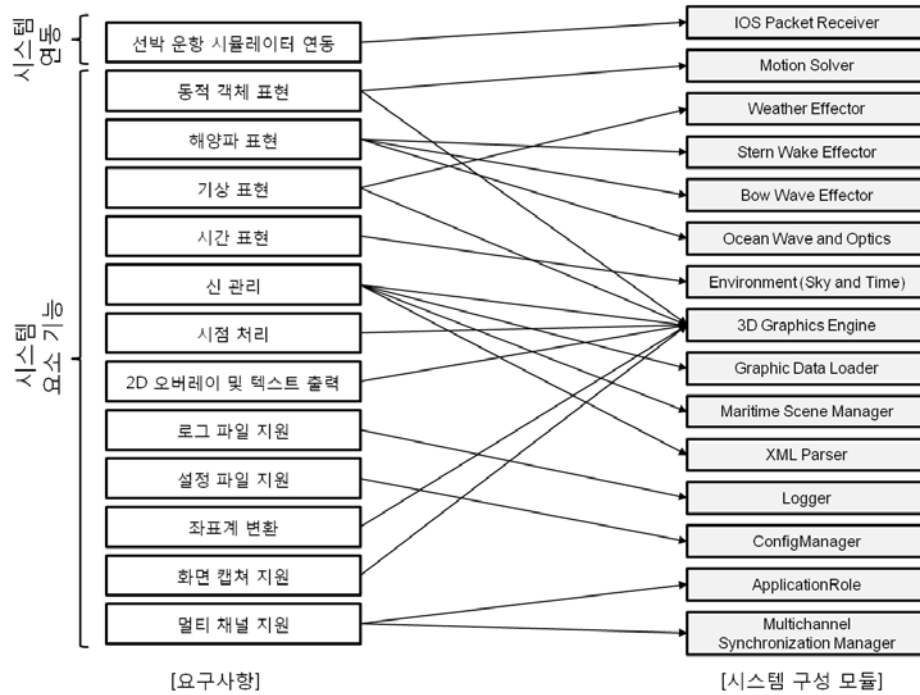


Fig. 4 Derivation of real-time 3D visualization system's components from operation requirements

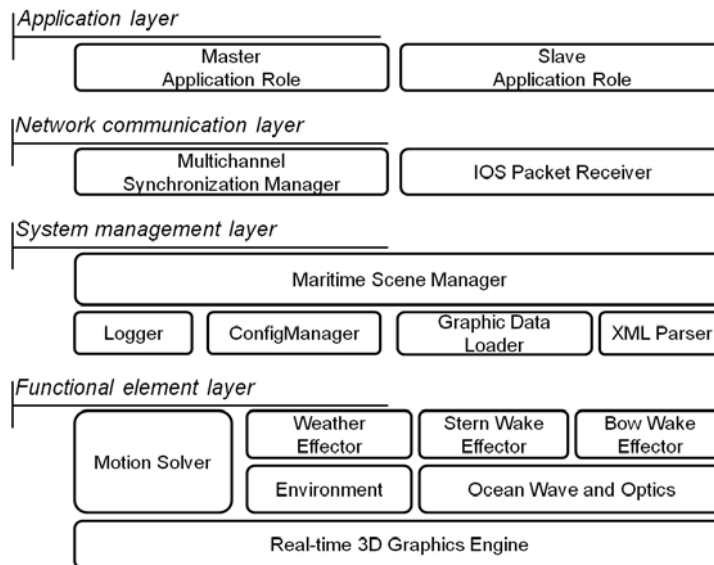


Fig. 5 System architecture of a real-time 3D visualization system based on an open source 3D graphics engine

4. 실시간 3D 가시화 시스템의 상세 설계

3절에서 정의 한 시스템의 기본 구조를 바탕으로 UML 모델링 방법(Fouler and Scott, 1999)을 사용하여 실시간 3D 가시화 시스템의 상세 구조를 설계 하였다.

실시간 3D 가시화 시스템의 최상위 클래스는, Fig. 6과 같이, RevisysApplication과 ApplicationRole이다. RevisysApplication 클래스는 프로그램의 엔트리 포인트(entry point) 역할을 한다. ApplicationRole 클래스는, 선박 운항 시뮬레이션 과정에

서, IOS로부터 받은 시뮬레이션 결과 데이터를 하는 가시화 하는 과정 전체를 관리하는 클래스이다. 3D 가시화 시스템이 설정에 따라 그래픽 클러스터의 마스터 또는 슬레이브 역할을 하기 때문에, ApplicationRole 클래스를 상속받아 MasterApplicationRole과 SlaveApplicationRole 클래스를 정의하여 역할에 맞는 함수와 로직을 구현한다.

3D 가시화 시스템의 초기화, 실행, 종료 절차를 시퀀스 다이어그램을 이용하여 설계한 결과가 Fig. 7이다. 사용자가 3D 가시화 시스템을 실행하면 RevisysApplication 클래스는

GlobalConfigManager 클래스를 통해 시스템 초기설정을 한 후, 그래픽 클러스터의 설정 정보에 따라 ApplicationRole 클래스의 하위 클래스의 객체 (MasterApplicationRole 또는 SlaveApplicationRole)를 생성하여 초기화한다. 초기화 과정이 끝나면, 운항 시뮬레이션 과정 동안 IOS로부터 시뮬레이션 결과 데이터 패킷을 받아 화면에 가시화를 한다. 사용자의 종료 명령 또는 IOS로부터 종료 패킷을 받으면 RevisysApplication 클래스가 3D 가시화 시스템을 종료한다.

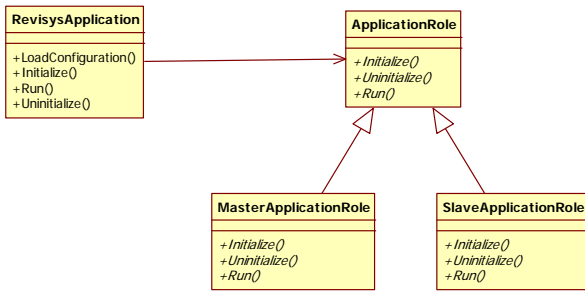


Fig. 6 Design of a real-time 3D visualization system: class diagram of RevisysApplication and ApplicationRole

그래픽 클러스터의 마스터와 슬레이브 간의 데이터 동기화는 Fig. 8과 같은 과정으로 이뤄진다. 마스터는 IOS로부터 시뮬레이션 결과 데이터를 UDP 패킷으로 받은 후 MaritimeSceneManager의 UpdateScene() 함수를 호출하여 scene 그래프를 갱신한다. 마스터는 갱신된 신 그래프 정보를 동기화 패킷을 통해 슬레이브로 전달한다. 동기화 패킷에는 시뮬레이션 진행 시간 (이전 화면의 갱신 이후 경과된 시간), 선박이나 부표와 같은 동적 객체에 관한 정보, 등대와

같이 상태가 변화되는 정적 객체에 관한 정보, 카메라 시점 정보, 파고나 풍향 등의 환경 설정 정보가 기록된다. 슬레이브는 마스터로부터 받은 동기화 패킷 데이터에 따라 MaritimeSceneManager의 UpdateSceneFromSyncPacket() 함수를 호출하여 scene 그래프를 갱신한다.

3D 가시화 시스템은 Fig. 9와 같은 구조로 scene 그래프를 생성하고 관리한다. Scene 그래프는 1) 주변 환경, 바다, 기상 상태 관련 노드, 2) 항구, 자선, 타선 관련 노드, 3) 부표와 등대 관련 노드로 구성된다. 3D 가시화 시스템에서 카메라는 자선과 함께 이동하기 때문에, 카메라의 위치 및 방향 정보는 자선의 좌표계를 기준으로 표현된다. 따라서 시스템 설정 데이터에 저장되어 있는 카메라 위치 및 방향 정보들을 읽어, 카메라 노드들을 자선 노드 OwnShip의 하위 노드로 정의한다.

5. 실시간 3D 가시화 시스템의 구현 및 실험

3절과 4절에서 설명한 실시간 3D 가시화 시스템의 설계에 따라 3 채널 그래픽 클러스터 방식의 실시간 3D 가시화 시스템을 구현하여 한국해양연구원의 선박 운항 시뮬레이터에 적용하였다.

가시화 기능 구현을 위해 필요한 3D 그래픽 엔진으로 공개형 3D 그래픽 엔진인 Ogre3D 1.6.3을 사용하였다. 그래픽 데이터 로딩, 해양과 가시화, 환경 가시화 모듈 구현을 위해서 Ogre3D를 기반으로 개발된 공개형 라이브러리인 Hydrax (Hydrax, 2010), SkyX (SkyX, 2010)를 사용하였다.

시스템 관리 계층에서 데이터 로깅 모듈은 Apache log4cxx (Apache, 2010)를 사용하여 구현하였고, 그래픽 데이터 로딩 모듈은 Ogre3D 기반으로 개발된 공개형 라이브러리인 DotSceneLoader (DotSceneLoader, 2010)를 사용하여

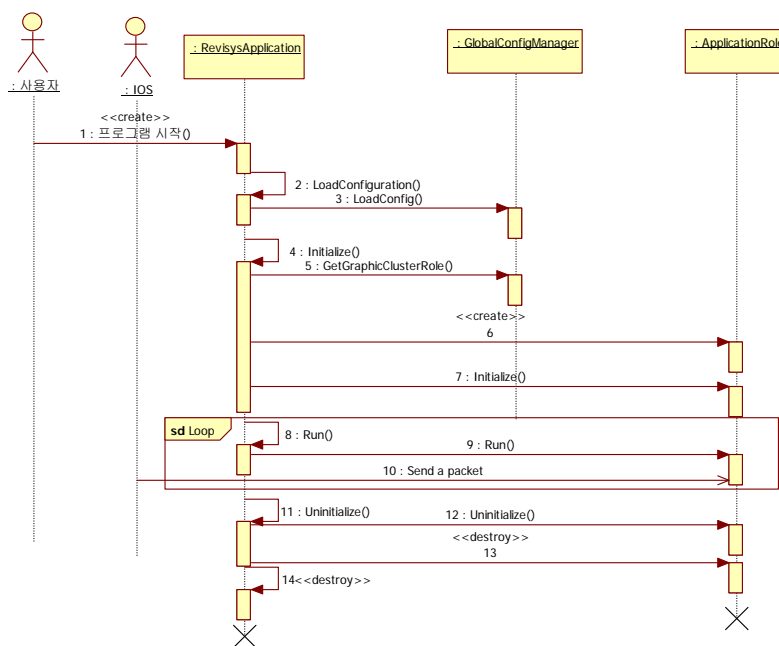


Fig. 7 Design of a real-time 3D visualization system: sequence diagram of RevisysApplication and ApplicationRole

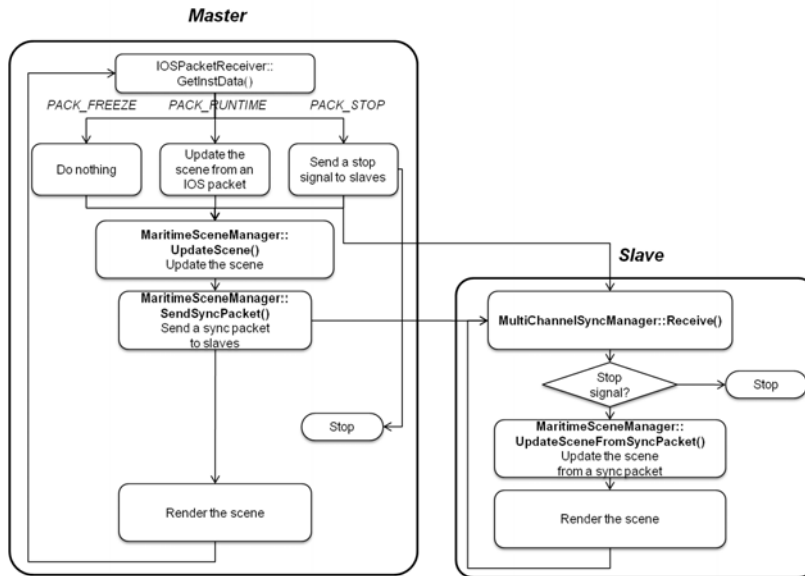


Fig. 8 Design of a real-time 3D visualization system: Data synchronization process between master and slaves

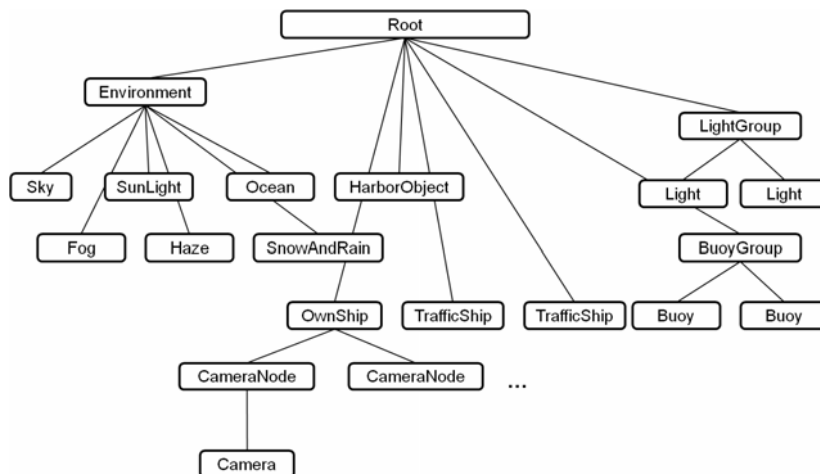


Fig. 9 Design of a real-time 3D visualization system: class diagram for scene objects

구현하였다. 그리고 XML 데이터 파싱을 위해서 TinyXML (TinyXML, 2010)를 사용하였다.

네트워크 통신 계층의 Multichannel SyncMgr과 IOS Packet Receiver의 구현을 위해서 네트워크 통신 라이브러리인 Windows Socket(WinSock) 2.2를 사용하였다.

실시간 3D 가시화 시스템과 한국해양연구원의 선박 운항 시뮬레이터와 연동하여 가시화 실험을 수행한 결과가 Fig. 10이다. 연동 실험을 통해, 개발한 실시간 3D 가시화 시스템이 기존의 상업용 3D 그래픽 엔진 기반의 시스템과 비교하여 전반적으로 대등한 성능을 보임을 확인하였다.

그래픽 기능의 경우, 해수면의 광원 효과는 기존 시스템에 비해서 우수하나 선수/선미 캐적 모사 기능은 개선이 필요하다. 멀티채널 가시화의 경우, 타일 디스플레이(tiled display)와 같은, 다양한 채널 구성이 가능한 장점이 있으나 편의 기능이 부족한 것이 단점이다.



Fig. 10 Visualization of ship handling simulation using the developed system

6. 결 론

선박 운항 시뮬레이터용 실시간 3D 가시화 시스템은 조선 해양 시뮬레이션에서 요구되는 특수한 가시화 요구를 만족시키고 시뮬레이터를 구성하는 타 시스템과의 연동이 용이하도록 개발이 되어야 한다. 이 연구에서는 소스 코드가 공개되어 기능의 확장이나 최신 기술의 적용 측면에서 장점이 있는 공개형 3D 그래픽 엔진을 기반으로 선박 운항 시뮬레이터용 실시간 3D 가시화 시스템을 개발하였다.

가시화 시스템 개발을 위해 요소 기능, 기존 그래픽 데이터 활용, 타 시스템과의 연동 측면에서 운용 요구사항을 도출하였다. 그리고 도출된 운용 요구사항을 바탕으로 시스템에 필요한 구성 모듈을 정의한 후 계층화 과정을 통해 실시간 3D 가시화 시스템의 일반 구조를 정의하고 설계를 수행하였다. 실시간 3D 가시화 시스템의 설계 결과를 바탕으로 공개형 그래픽 엔진인 Ogre3D와 관련 공개형 라이브러리를 사용하여 가시화 시스템을 구현한 후 한국해양연구원의 선박 운항 시뮬레이터와의 연동 실험을 통해 구현된 시스템을 검증하였다.

후 기

이 논문은 한국해양연구원의 ‘기관목적사업’인 ‘예부선의 사고분석 및 예방기술 개발’ 과제의 지원으로 수행된 연구 결과임을 밝힙니다.

참고문헌

[1] 문두환, 박세길, 오재용, 황호진 (2010), 선박 운항 시뮬레이터를 위한 실시간 3D 가시화 시스템의 운용 요구, 2010 한국CAD/CAM학회 학술대회

[2] 차무현, 이재경, 황진상 (2010), 오픈소스 그래픽스 엔진을 이용한 가상현실 타일 디스플레이 시스템의 개발, 2010 한국CAD/CAM학회 학술대회

[3] 차주환, 노명일, 이규열 (2009), 범용 동역학 모듈과 가시화 모듈을 이용한 조선 블록 탑재 시뮬레이션, 한국 CAD/CAM학회논문집, 14권, pp.69-75

[4] 황호진, 문두환 (2010), 선박 운항 시뮬레이션 가시화를 위한 그래픽 렌더링 엔진 기술의 개선 방향, 한국항해항만학회지, 34권, pp. 153-160

[5] Apache (2010), <http://logging.apache.org/log4cxx/index.html>

[6] Bian, Z., Ishii, H., Izumi, M., Shimoda, H., Yoshikawa, H., Morishita, Y., Kanehira, Y. (2006), "Development of a Tracking Method for Augmented Reality Applied to NPP Maintenance Work and its Experimental Evaluation", in the Proceedings of the ACM Symposium on Virtual Reality Software and

Technology

[7] Burns, D., Osfield, R. (2004), "Open Scene Graph A: Introduction, B: Examples and Applications ", in the Proceedings of the IEEE Virtual Reality Conference 2004

[8] DotSceneLoader (2010), <http://www.ogre3d.org/tikiwiki/New+DotScene+Loader&comzone=show>

[9] Fouler, M., Scott, K. (1999), "UML Distilled: a Brief Guide to the Standard Object Modeling Language", Second Edition, Addison Wesley Longman, inc

[10] Friese, K. I., Herrlich, M., Wolter, F. E. (2008), "Using Game Engines for Visualization in Scientific Applications", IFIP International Federation for Information Processing, Vol. 279, pp. 11-22

[11] Glassner, A. (2002), "Duck!", IEEE Computer Graphics and Applications, Vol. 22, pp. 88-97

[12] Gong, I. Y., Lee, C., Lee, K. J., Lee, H. J., Kim, S. Y. (2004), "Development of Ship Handling Simulator System for Korean Navy and Its Mathematical Model", in the Proceedings of 31st IMSF AGM

[13] Hydrax (2010), <http://www.ogre3d.org/tikiwiki/Hydrax>

[14] Jensen, L., Goliass, R. (2001), "Deep-Water Animation and Rendering", in the Proceedings of the 2001 Game Developer's Conference

[15] Kim, Y. S., Yang, J., Han, S. (2006), "A Multichannel Visualization Module for Virtual Manufacturing", Computers in Industry, Vol. 57, pp. 653 - 662

[16] Kuhl, J., Evans, D., Papelis, Y., Romano, R., Watson, G. (1995), "The Iowa Driving Simulator: An Immersive Research Environment", IEEE Computer, Vol. 28, pp. 35 - 41

[17] Lee, H., Han, S., (2010), "Solving the Shallow Water Equations Using 2D SPH Particles for Interactive Applications", Visual Computer, Vol. 26, pp. 856-872

[18] Magee, L. E. (1997), "Virtual Reality Simulator (VRS) for Training Ship Handling Skills", In Seidel, R. J. and Chatelier, P. R. (Eds), Virtual Reality, Training's Future? , Defence Research Series, Vol. 6, Plenum Press: New York, pp. 19-29

[19] Murai, K., Hayashi, Y., Miyoshi, Y., Inokuchi, S. (2004), "A Basic study on Navigators' Visual Observation Area and Stress Level for Ship Handling by Actual Ships and Simulators", IEEJ Transactions on Electronics, Information and Systems, Vol. 123, pp. 1311-1318

[20] Mitchell, J. L. (2005), "Real-Time Synthesis and Rendering of Ocean Water," ATI Research Technical Report

- [21] Mujber, T. S., Szecsi, T., Hashmi, M. S. J. (2004), "Virtual reality applications in manufacturing process simulation", Journal of Materials Processing Technology, Vol. 155-156, pp.1834-1838
- [22] Ogre3D (2010), Open Source 3D Graphics Engine, <http://www.ogre3d.org>
- [23] OgreMax (2010), <http://www.ogremax.com>
- [24] Presagis (2009), <http://www.presagis.com/>
- [25] Roth, M., Voss, G., Reiners D. (2004), "Multi-threading and clustering for scene graph systems", Vol. 28, pp. 63-66
- [26] SkyX (2010), <http://www.ogre3d.org/tikiwiki/SkyX>.
- [27] Tessendorf, J. (1999), "Simulating Ocean Water," In SIGGRAPH Course Notes, Addison-Wesley
- [28] TinyXML (2010), <http://www.grinninglizard.com/tinyxml/>
- [29] Wallace, P. R., Northam G. A. (1997), "Training Task Analysis Methodology for Operational Flight Trainers", in the Proceedings of the 19th Interservice/Industry Training, Simulation and Education Conference. Orlando, Florida

원고접수일 : 2010년 10월 8일
심사완료일 : 2010년 12월 9일
원고채택일 : 2011년 3월 23일