Proceedings of KFIS Fall Conference, 2003. 5. 3

지능형 선박 자율운항제어시스템을 위한 2D 디스플레이 구현

Implementation of 2D Display for Autonomous Ship Control System using Intelligence Techniques

정현도, 정민우, 김창민, 김용기
경상대학교 컴퓨터과학과

Hyeon-Do Jeong, Min-Woo Jung, Chang-Min Kim, Yong-Gi Kim
Department of Computer Science, Gyeongsang National University
E-mail : hdjeong@ailab.gsnu.ac.kr

요 약

최근 증가하고 있는 해상충돌사고는 대부분 선박을 조종하는 인간의 습관이나 실수에 기인하고 있다. 이를 해결하기 위한 방안으로 선박의 자율화 및 지능화의 연구가 활발히 진행되고 있는데 대표적인 연구로 지능형 자율운항제어시스템이 있다. 지능형 자유운항제어시스템은 항로계획, 항로감시 및 충돌회피와 같은 전문가의 지식을 전산화한 것으로써 실제 선박에 장착되거나 시뮬레이션이 이루어질 때 시스템정보의 효과적인 전달을 위한 시각적 표현이 필요하다. 본 연구에서는 지능형 자율운항제어시스템의 2D기반 그래픽환경 구축을 위한 디스플레이를 개발한다. 디스플레이는 가상세계로부터 입력되는 데이터를 가상이미지로 변환한 후 사용자가 원하는 정보를 표현한다. 지능형 자율운항제어시스템에 근거한 디스플레이는 선박의 항해장비와 직접 연동되지 않기 때문에 구현, 수정 및 확장이 용이하다는 장점이 있다. 또한 본 연구에서 개발된 디스플레이는 가상이미지라는 개념을 도입하여 편리한 출력 수정과 선택이 가능하다. 제안된 연구는 가상세계의 데이터와 디스플레이에 표현된 정보를 비교·평가하여 성능을 검증한다.

Key word : 디스플레이, 지능형 자율운항제어시스템, 가상이미지, 가상세계

1. 소개

첨단 과학기술의 발전과 더불어 선박의 대형화, 고속화 추세에 따라 각종 해난사고가 발생하고 있다. 이에 따른 막대한 재산상의 손실은 물론 환경파괴, 인명피해 등의 그 심각성이 대두되고 있다. 선박의 충돌 및 전복사고를 미리 예측, 방지 할 수 있는 방법으로 선박의 지능화, 자율화를 이루어 해난사고를 최소화하는 것이다.

선박의 지능화란 선박 내 하드웨어적인 각종 센스로부터 입력되는 데이터를 디지털화하여 인간이 선박을 실제 조작하는 것처럼 일관성 있는 행동을 유발하는 시스템을 의미한다. 여기서 지능형 자율운항제어 시스템(ASCSIT : Autonomous Ship Control System using Intelligence Techniques)은 숙련된 항해사의 능력을 전산화하여 이를 선박에 적용하여 자율적 운항이 가능한 전문가 시스템이다. 자율운항시스템은 항해사와 거의 유사한 지식을 가지고 스스로 항해, 계획, 감시, 충돌회피를 함으로써 해난사고를 미연에 방지할 수 있다.

ASCSIT는 실제 선박에 구축되거나 시뮬레이터로 구현 및 테스트가 먼저 이루어진다[1][2][3]. 이때 ASCSIT의 시뮬레이터 결과를 시각적으로 표현하여 직관적인 이해와 운항시 운항정보를 전달하기 위해 디스플레이의 구현이 필요하다. 본 연구에서는 ASCSIT의 가상세계를 통하여 실세계의 데이터를 고수준의 정보로 변환한 정보를 이용하여 디스플레이를 구현한다.

세계각국에서 활발히 진행되고 있는 VAIS, Radar, GPS, GYRO, Speed Log, Echo-Sounder등의 장비와 연동하여 수집된 정보를 ECDIS(Electronic Chart Display and Information System)상에서 도시하여 여러 가지 정보를 알 수 있고, 국내에서 개발된 NARU 2000은 선박운항을 계획, 수정 및 감시하여 각종 항해정보를 하나의 화면에 표시하는 기능을 가진 시스템을 이용하고 있다[4][5][6][7][8]. 이러한 시스템은 단지 항해 센스로부터 입력되는 정보만 관리하는 시스템으로써 지능화 및 자율시스템이라 할 수 없으며, 선박의 항해 장비와 직접 연동되어 있기 때문에 구현과 수정, 확장성에 많은 어려움이 있다.

본 논문에서 제안하는 ASCSIT에 근거한 디스플레

이는 지능화가 이루어진 가상세계의 고수준 정보를 이용한다. 또한 선박의 항해 장비와 독립되어 있는 시스템으로써 구현의 간단성과 수정, 확장성이 용이하다.

제 2절에서는 ASCSIT의 기능과 역할에 관하여 살펴보고, 제 3절에서는 본 논문에서 제안하는 디스플레이의 구조 및 설계 대해 살펴본다. 그리고 제 4절에서는 디스플레이 구현과 실행결과를 평가하며 마지막으로 제 5절에서는 결론 및 향후 연구과제에 대해 논의한다.

2. 지능형 자율운항제어 시스템

ASCSIT는 인지, 사고, 행위와 같은 인간의 지적능력을 내포한 인공지능시스템으로써 실세계의 정보를 받아들여서 고수준의 정보로 산출하여 적절한 제어를 행하는 시스템이다. 본 절에서는 ASCSIT의 기본 아키텍처인 RVC모형과 구성시스템에 대해서 알아본다.

2-1. 통합 아키텍처 RVC 지능시스템 모형

ASCSIT는 인공지능의 다양한 영역 중에서 기호기반 인공지능기법과 행위기반 인공지능기법의 효율적 통합과 구성시스템간의 구조적 및 기능적 독립성의 극대화를 이룬 RVC(Reactive Layer-Virtual world-Considerative layer) 지능시스템 모형을 제안한다. 행위기반 인공지능은 행위기반 시스템으로써 실세계의 불확실한 문제의 표현과 해결에 적합하므로 실세계의 정보를 기호화된 형태의 정보로 제공한다. 이 정보는 기호기반 인공지능인 숙고형 시스템에서 다시 가공, 처리하여 적절한 정보를 다시 만들어낸다. 이때 행위기반 시스템과 숙고형 시스템의 중간층에서 정보의 경유지인 공유 데이터 저장소가 존재한다. 공유 데이터 저장소를 '가상세계'라 일컫는다. 본 논문에서는 '가상세계'의 정보를 이용해서 디스플레이를 구현하게 된다. 그림 1은 RVC 지능시스템 모형의 개념도이다[9].

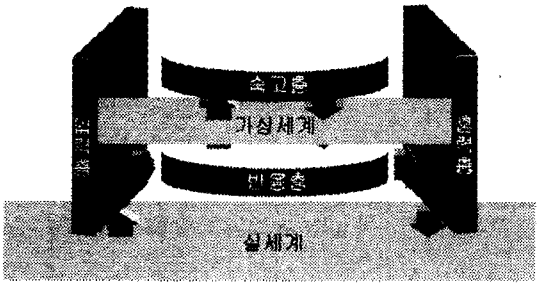


그림 1. RVC 모형도

2-2. 지능형 자율운항제어 시스템 구성

RVC를 기반한 ASCSIT는 구성시스템간의 독립성을 유지하는 충돌회피시스템, 항해시스템, 자료융합시스템, 운동제어시스템으로 이루어진 하부 시스템을 가진다. 자료융합시스템은 반응층에 배치되어 운항센서나 장애물감지센스로부터 입력되는 데이터를 가공하여 보다 고 신뢰도의 정보를 가상세계에 제공한다. 타 시스템은 숙고층에 배치되어 가상세계의 정보를 이용하여 항해시스템은 최적의 항로를 산출해서 항로계획, 감시를 하며, 충돌회피시스템은 외부시스템과 연동되어 다음 이동경로를 산출하고, 운동제어시스템은 선박의 움직임을 제어한다. 이런 모든 데이터의 이동은 가상세계를 경유해서 정보를 요구하고 전송한다. 그림 2는 ASCSIT의 구성도이다[9][10].

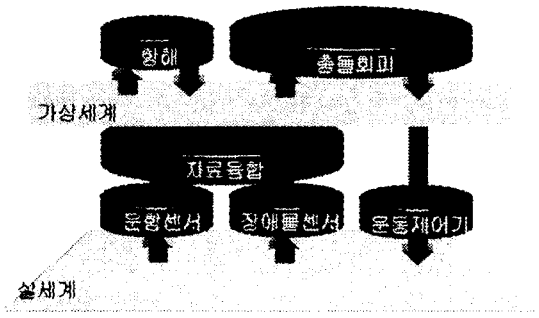


그림 2. 지능형 자율운항제어 시스템 구성도

3. 지능형 자율운항제어 시스템에 근거한 디스플레이 설계

ASCSIT에 근거한 디스플레이는 가상세계에 저장된 정보를 이용하여 선박의 현재상태를 실시간으로 2D 그래픽을 사용해서 보여준다. 본 절에서는 가상세계 데이터의 흐름과 디스플레이에서 입력된 정보가 2D그래픽으로 변환과정을 설계한다.

가) 디스플레이의 데이터 흐름

디스플레이 설계는 그림 3과 같이 가상세계로부터 정보를 받아들이는 입력층, 데이터를 가공 처리하는 정보처리층, 화면으로 보여주는 출력층으로 크게 3부분으로 나누어지고 데이터는 단방향으로만 전송된다.

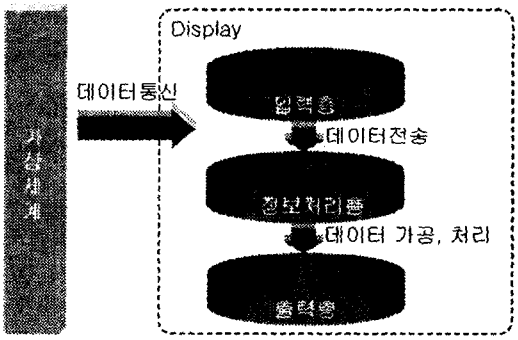


그림 3. 디스플레이 데이터 흐름도

나) 데이터 처리 및 출력

입력층에서 선박의 환경 정보, 외부정보를 각종 센스로부터 직접 가져오는 것이 아니라 가상세계라는 공유 데이터 영역에서 약속된 규칙에 의해서 데이터를

요구하고 전송 받는다. 본 논문에서의 디스플레이는 ASCSIT에서 처리, 가공된 정보를 2D 그래픽을 통한 시각적 표현이므로 정보의 요구만 이루어진다.

정보처리층에서는 가상세계로부터 입력된 선박의 위치, 속도, 헤딩각 등의 선박환경정보와 선박외부환경정보를 분류해서 2D그래픽으로 이미지화 시킨다. 가상세계에서 넘어오는 모든 데이터를 이미지로 변환하여 메모리에 먼저 저장하게 된다. 본 논문에서는 이를 '가상이미지'라 한다. 가상 이미지 중에서 필요한 부분만 선택해서 2D로 출력하게 되는데, 사용자는 정보의 취사선택이 쉽고, 이미지 확대, 축소, 등 조작이 편리해진다. 정보처리층의 구성도는 그림 4와 같다.

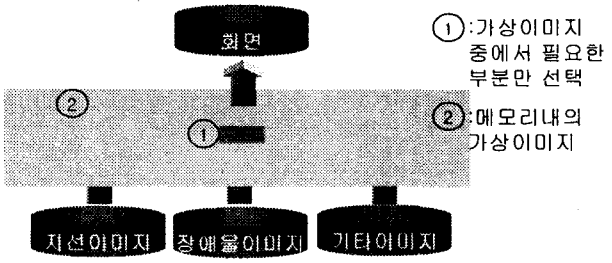


그림 4. 정보처리층의 구성도

출력층에서는 기본적으로 자선을 화면에 출력하는 창(window)의 중심에 있게 한다. 여기서 자선의 좌표를 이용하여 창(window)의 좌측 상단의 좌표(BeginPiont)를 구할 수 있다. 이 좌표가 변함으로써 가상이미지에서 원하는 부분을 2D로 보여준다. 그림 5는 출력층의 개요도이다.

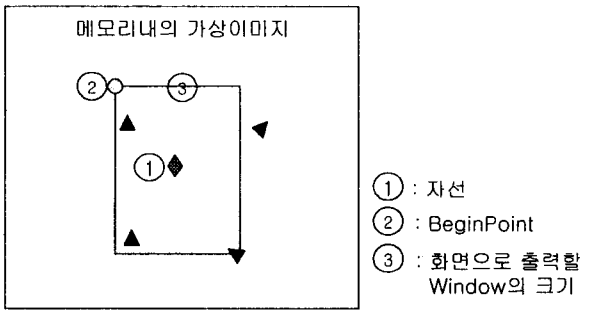


그림 5. 출력층의 개요도

다) 사용자 편의 인터페이스

자선의 이동 유무에 따라 항해모드와 충돌회피모드 두 개의 window를 볼 수 있다. 장애물이 거의 없거나 기타 위험요인이 적은 대양에서 항해할 경우에는 항해모드를 적용해서 자선은 window의 중심에 항상 위치하게 하고 상대적으로 장애물과 해도가 움직이게 한다. 연안에서 고정장애물과 이동장애물의 잦은 접촉이 우려되는 곳에서 충돌회피모드로 해도는 고정되어 있고 자선과 이동장애물의 이동을 보여준다. 그리고 보다 더 크고, 자세하게 혹은 넓은 지역을 한눈에 볼 수 있게 축소, 확대를 통해서 쉽게 조작할 수 있다.

4. 디스플레이 구현 및 테스트

디스플레이의 구현 환경은 충동회피시스템, 자료융합시스템, 운동제어시스템과 시뮬레이터를 구현한 ASCSIT에 기초를 두고 구현하였다.

가) 데이터 통신 및 처리

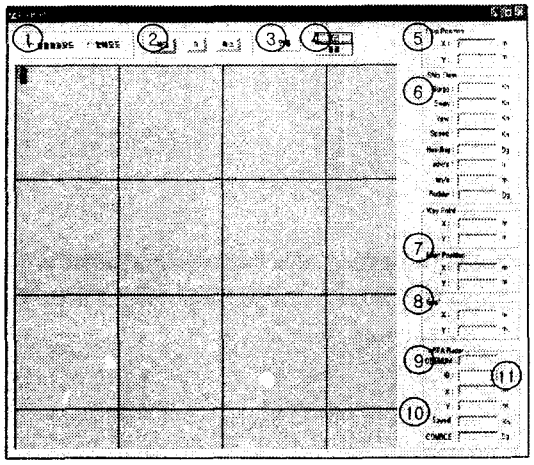
디스플레이는 필요한 모든 데이터를 가상세계로부터 입력받는다. 가상세계와 데이터 교환을 위해서 일정한 통신 규약을 만들어서 원활한 데이터 전송을 이루게 한다. 그림 6은 미리 약정된 통신 규약을 이용하여 가상세계와 통신을 하고있는 구조를 보여주고 있다. 이런 데이터 통신을 통해서 전송 받은 데이터는 내부함수 ViewMapBeginPoint(), SelectShip(), OnGeneral(), OnStretch(), OnReduce(), OnCollisionescape(), OnNavigation()에 의해서 window의 위치를 구하고, 자선과 장애물의 헤딩에 따른 이미지변경, 축소·확대와 충돌회피 모드, 항해모드 설정의 처리가 이루어진다.

```
1: m_pVWClient->InitVWClient(...);
2: //system Initailization
3:     RegisterNumber(...);
4:         :
5:     RegisterString(...);
6:         :
7: m_pVWClient->SynchronizeScheduleSession();
8: VWClient->ProbeVWStatus(iStatusOfSchedule,...);
9: while (iStatusOfSchedule == SCHEDULE_SESSION)
10: {     m_pVWClient->SynchronizeSessionHead();
11:         :
12:       m_pVWClient->SynchronizeSessionRead();
13:         :
14:     DoOwnJob();     }
```

그림 6. 통신 프로토콜 구조

나) 인터페이스 구현 및 테스트

그림 7은 가상세계로부터 들어온 데이터를 이미지로 변환하여 가상이미지로부터 사용자가 원하는 정보를 출력하는 인터페이스를 보여주고 있다. windowd에 일정한 간격의 격자는 한 격자당 4Km를 나타내고 있는데, 이것은 장애물과 자선의 거리를 직관적으로 쉽게 알아보기 위해 표현했다. 그리고 자선의 위치, 속도, 헤딩각 등 주요정보와 각 장애물마다 특정 ID를 부여해서 ID를 선택하면 그 ID에 해당하는 장애물의 정보를 쉽게 알아 볼 수 있다.



① : 항해 · 충돌모드선택　② : 확대 · 축소 선택
③ : 가상세계와 통신　④ : 종료
⑤ : 자선위치 정보　⑥ :자선의 기타정보
⑦ : Way Point 정보　⑧ : Next Pont 정보
⑨ : Goal 정보　⑩ : 장애물 정보
⑪ : 장애물 ID 입력

그림 7. 디스플레이 사용자 인터페이스

디스플레이의 결과는 ASCSIT와 연동하여 자선의 움직임, 그에 상응하는 장애물의 운동, 자선의 정보와 가상세계의 데이터 값과 비교해서 알 수 있다. 그림 8은 가상세계와 통신을 통해서 실행되는 모습을 보여주고 있다.

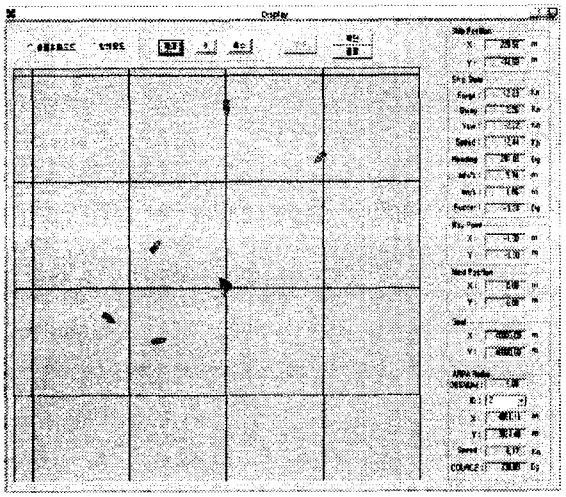


그림 8. 디스플레이 실행결과

5.결론 및 향후과제

본 논문에서 제안하는 디스플레이는 ASCSIT의 가상세계와 통신을 통해 현재 선박의 상태를 입력 값으로 받아 2D로 출력한다. 2D 이미지는 먼저 가상이미지로 먼저 생성된 후 사용자는 가상이미지에서 필요한 정보를 볼 수 있다. ASCSIT에 근거한 디스플레이는 확장과 수정 및 개발이 쉽고, 개발비용이 낮다. 가상이미지를 이용해서 차후에 출력부분의 확장성과 정보의 취사선택이 사용자의 편이에 맞게 수정이 용이하다.

디스플레이의 실행결과는 가상세계의 데이터와 비교, 확인을 통해서 올바른 수행이 이루어진다는 것을 입증하였다.

본 연구에 이어 향후에 개선해야 할 과제는 다음과 같다. 첫째, 주변환경의 더욱 자세한 이해를 위해 전자해도를 이용하여 디스플레이를 구현한다. 둘째, 보다 미려한 그래픽을 위해 3D로 구현해서 사용자가 손쉽게 이용가능한 사용자 인터페이스 구성이 남아 있다.

6. 참고 문헌

[1] Gordon, G., "System simulation (second edition)," Prentice-Hall, 1978.

[2] Law, A. M. and Kelton, W. D., "Simulation modeling and analysis," McGraw-Hill Book Company, 1982.

[3] 이원호, 김창민, 김용기, "지능형 자율운항제어 시스템을 위한 시뮬레이터 구현," 한국퍼지 및 지능 시스템 학회 추계학술대회 학술발표 논문집, Vol. 11, No. 2, pp.169-172, 2001.

[4] Depeng, Zhao Wenli, Sun Yuanhui, "The Research and Prospects of Electronic Display and Information System(ECDIS) in China," International Symposium on VTS & IBS, Vol. 1, No. 1, 1995.

[5] Smeaton. G. P., Dineley. W. O., Tucker. S. M., "Display Requirements for ECDIS/ARPA Overlay Systems," The Journal of navigation, Vol .48, No. 1, 1995.

[6] Baziw, E. J., "ECDIS Steering Algorithm for Vessel Autopilot," POSITION LOCATION AND NAVIGATION SYMPOSIUM, 1996.

[7] Davis. D., "PinPoint Systems discusses the politics behind ECDIS implementation," Shipping world & shipbuilder, Vol. 197 No. 4124, 1996.

[8] Riches. B. A. E., "ECDIS - Its Impact Today," The Journal of navigation, Vol. 48, No. 2, 1995.

[9] 김창민, 김용기, "무인자율항체를 위한 지능제어 아키텍처에 관한 연구," 한국군사과학기술학회지, Vol. 4 No. 2, pp246-255, 2001.12.

[10] Scerri. P. and Reed N., "Requirements for a General Agent Architecture for Agent-Based Simulation Environments," Agents'99 Workshop on Autonomy Control Software, 1999.