

협소 해역에서 자율운항선박의 충돌 감지에 대한 기초 연구

김경훈, 차주형, 이재용(동의대)

A preliminary study on the collision detection for autonomous ships in narrow seas

Gyeonghoon Kim, Joo Hyoung Cha, Jaeyong Lee (Dong-eui University)

Abstract

One of the most difficult areas for autonomous ships to operate is in narrow waters near quay walls or with complex waterways. A lot of research has been conducted on the possibility of collision with ships approaching from a long distance, but a 3D model should be considered in a narrow sea area where the distance between ships is small. In this study, we present a 3D model-based collision detection technique required for operational scenario verification for autonomous ships. Types of boundary volumes are investigated and applied to the ship model. The current state of technology is reviewed, and the conditions for application to simulation for performance verification are explained.

Keywords : Bounding volume(경계체적), Collision detection(충돌감지), Autonomous ship(자율운항선박), Automatic berthing(자동 접안)

1. 서론

자율운항선박의 운용에 있어 충돌을 감지하고 이를 회피하는 것은 가장 기본적인 요소 중 하나이다. 정상 운항 중인 선박간의 충돌을 감지하고 회피하는 기동에 관해서는 많은 연구가 진행되어 왔다. 무인선이나 자율운항선박의 안전한 운항과 자동 이-접안을 위해서는 정밀한 위치 유지가 필요하다. 항만 또는 협소한 해역에서 접-이안 또는 근접 운항 자체에 대한 연구는 진행되었지만 그 과정에서 일어날 수 있는 충돌을 고려한 연구는 많지 않은 상황이다. 이는 가까운 거리에서의 충돌을 감지하기 위해서는 선박의 형태를 정확하게 표현할 수 있는 정교한 모델을 사용해야하기 때문이다. 연구를 위해서는 정밀한 위치와 자세를 반영하는 것이 필요한 데, 현재의 시뮬레이터에서 이를 적용하여 구현한 예는 그리 많지 않다. 대부분의 시뮬레이터는 선박의 움직임을 유체역학적으로 해석하거나, 환경에 의한 하중을 단순화 시킨 후 제어 알고리즘을 적용하는 것에 초점을 맞추었기 때문이다. 따라서 좁은 해역에서 다양한 유체력을 고려하면서도, 경로계획 및 제어 알고리즘을 적용하여 성능을 평가할 수 있는 기법을 개발하는 것은 어렵다. 따라서 선박과 선박, 선박과 안벽과의 충돌을 고려한 알고리즘은 구현하기 어렵다.

3D 그래픽 분야에서는 일찍부터 두 도형의 겹침 문제에 대해 연구를 진행해 왔다. 게임 분야에서는 이를 활용하여 3D 개체들 간의

충돌을 감지하고 나아가 물리엔진을 적용하여 충돌에 대한 반응이 이루어지도록 하고 있다. 일반적으로 충돌 처리는 물체들끼리의 충돌 여부를 찾아내는 충돌 탐지 단계와 충돌에 대한 반응으로 구성된다. 충돌 해석이 필요한 여러 개의 기하학적 모델에 대해 충돌 감지 알고리즘을 거쳐 이에 대한 반응이 이루어지도록 하는 것이다.

본 연구에서는 게임 분야에서 널리 활용되고 있는 충돌 감지 기법들을 활용하여 협소 해역에서 선박과 선박의 충돌을 감지하는 방법과 접이안시 선박과 안벽과의 접촉을 감지하는 방법에 대한 기본적인 접근 방법을 제시하고자 한다.

2. 충돌 감지를 위한 방법

선박 사이의 충돌을 감지하기 위해 3D 그래픽에서 가장 기본적으로 활용되는 방법은 경계체적(bounding volume)을 이용하는 방식이다. 경계체적은 원하는 3D 객체를 모두 포함하는 다면체를 말한다. 가장 직관적으로 사용되는 형태는 직육면체이며 경계상자(bounding box)라고 불린다. 경계상자에는 좌표축에 평행하도록 구성하는 AABB (axis-aligned bounding box) 방식과 좌표축에 관계없이 해당 객체를 감싸는 최소한의 직육면체를 구성하는 OBB (oriented bounding box)가 있다. 구 (sphere)을 이용한 방식도 활용 가능하며, 구와 원통형이 결합된 캡슐 (capsule) 형태의 경계 체적

도 사용할 수 있다. 객체를 조금 더 효율적으로 감싸기 위하여 AABB와 구의 장단점을 적절히 절충한 형태인 k-dop (k-discrete orientation polytopes)도 많이 활용된다. k-dop에 비해 보다 많은 자유도를 주는 방식이 컨벡스 헐 (convex hull)로 감싸는 것이다. 물체를 감싸는 다면체를 구성하되 볼록한 형태가 유지되도록 구성하는 것이다. 이와 같이 여러 종류의 경계 체적을 선박 모델에 적용한 예가 Fig. 1에 표현되어 있다.

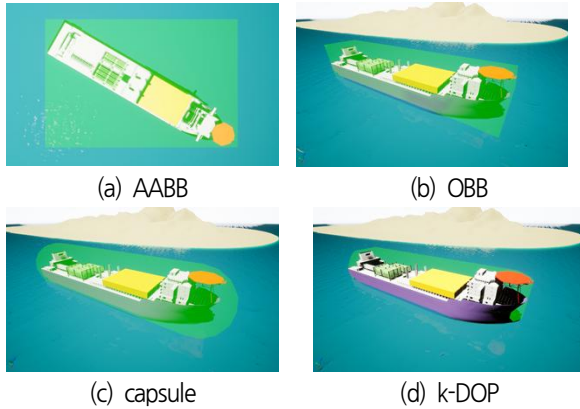


Fig. 1. 다양한 종류의 경계 체적

선박 모델의 가장 큰 특징은 상부 구조물의 형태가 복잡하고, 하부 선체는 유선형을 띄고 있다는 것이다. 이와 같이, 객체의 모양이 복잡하고 곡선이 많이 포함된 경우 하나의 경계 체적이 아니라 여러 개로 구성하여 트리를 형성하는 기법이 사용될 수 있다. 트리 (BVG, Bounding Volume Hierarchy)의 형태로 여러 개의 경계체적을 사용하여 전체 선박을 감싸는 구조를 만드는 것이다. Fig. 2는 오픈소스로 공개되어 널리 사용되고 있는 VTK (Visualization Toolkit)의 라이브러리 중 하나인 vtkOBBDTree 클래스를 이용하여 OBB로 구성된 트리 체계를 나타내고 있다. Level 0의 경우 전체 선체를 하나의 OBB로 감싸는 경우이고, 점차 그 수를 늘어가면서 level 15의 경우 전체적인 모양을 표현할 수 있음을 보여준다. 다른 구, AABB, 그리고 k-dop에 대해서도 이런 방식으로 구조체를 구성할 수 있다.

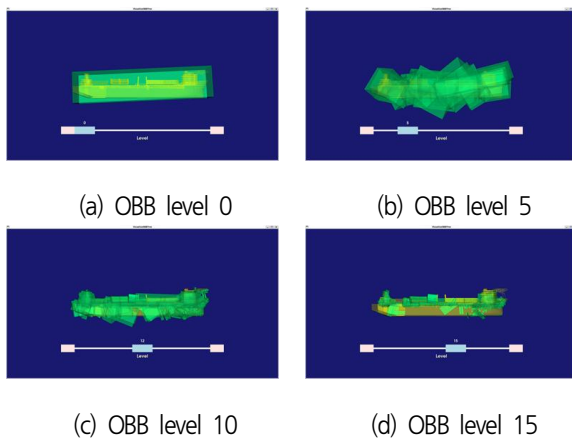


Fig. 2 OBB들로 구성된 hierarchy 생성

컨벡스 헐 방식과 관련하여 현재 가장 진보된 형태는 V-HACD (volumetric hierarchical approximate convex decomposition)

방식이다 (Thul, et al., 2018). Fig. 3은 V-HACD 알고리즘을 이용하여 컨벡스 헐을 이용하여 트리를 구성한 예이다. 컨벡스 수가 늘어날수록 전체적인 선박의 형태를 더 잘 표현할 수 있다. 특히 직육면체로 구성되는 OBB 트리 방식에 비해 곡선이 많이 들어가는 선박의 특성상 선박의 외형을 보다 자연스럽게 구현할 수 있고, 아주 상세하게 표현할 경우 선체의 3D 모델에 버금가는 정확도를 나타낼 수 있다.



(a) Convex hull 50개 (b) Convex hull 1683개
Fig. 3 선박모델에 V-HACD를 적용한 예

보다 정밀한 충돌 검사를 위해 사용되는 대표적인 알고리즘은 GJK (Gilbert-Johnson-Keerthi) 알고리즘이다 (Montanari, 2017). 2개의 컨벡스 헐 사이의 최소 거리를 계산하는 방법으로 1988년 발표된 이후 널리 활용되고 있다. 언리얼 등의 게임엔진에서도 GJK 알고리즘을 적용한 충돌 감지 기능을 제공하고 있다.

3. 상용 게임 엔진을 이용한 충돌 감지 시뮬레이션

현재 널리 사용되고 있는 게임 엔진은 언리얼(Unreal)과 유니티(Unity)이다. 이들 엔진에서 3D 캐드 모델로 생성한 메쉬 모델간의 충돌을 감지하는 기능은 아직 구현되어 있지 않다. 다만 메쉬 모델과 경계체적간의 충돌 위치 추출은 가능하다. NVIDIA사에서 디지털 트윈용으로 개발하고 있는 Omniverse 플랫폼의 경우 주 물리엔진인 PhysX5에서 mesh와 mesh간의 충돌도 감지할 수 있도록 지원한다 (Hölzer, et al., 2022). 하지만 이러한 mesh-mesh 충돌 계산은 많은 연산을 필요로 한다. Fig. 3에서 예를 든 선박 모델의 vertex의 수는 약 24만개, 삼각형의 수는 약 12만개 정도로 구성되어 있다. 선박 외형이 복잡해질수록 그 수는 더욱 많아지게 된다. 이럴 경우, 선박 간의 충돌을 계산하는 데는 많은 시간이 소요될 가능성이 있다. 고성능 NVIDIA의 GPU를 활용하여야 하며 개체 수가 많아지면 느려질 가능성도 있다. 기존 게임엔진들의 경우 연산을 가속화하기 위하여 NVIDIA의 GPU를 활용하거나 인텔의 ISPC (Intel SPMD program compiler) 등과 같은 CPU 가속화 기능을 활용하고 있다.

Unreal의 경우 자체적으로는 V-HACD에서 컨벡스헐의 최대 개수를 64개까지 생성하도록 지원하고 있다. 그 이상의 컨벡스 헐을 생성하기 위해서는 블렌더(Blender)등과 같은 3D 그래픽 툴에 V-HACD 알고리즘을 적용하여 컨벡스 헐을 생성하고, 이 결과물을 게임엔진에서 활용되는 파일 (Unreal의 경우 SDF) 포맷으로 변경한 후 시뮬레이션에 사용하게 된다. 현재의 게임엔진들은 블렌더를 애드온 하에 활용할 수 있도록 지원하기도 한다 (Plowman, 2016).



(a) 충돌 전 (b) 충돌 후

Fig. 4 Unreal에서 컨벡스 혈을 이용한 선박과 안벽과의 충돌검사

Fig. 4는 언리얼의 물리 엔진을 이용하여 충돌 감지에 활용하는 예를 보여준다. 선박 캐드 모델에 대해 Blender를 이용하여 V-HACD 기법을 적용하였다. 총 1683개의 컨벡스 혈로 구성된 선박이 안벽으로 접근하면서 충돌이 일어날 때 해당 컨벡스 혈의 색상이 붉은색으로 변경된다. 선박이 안벽 쪽으로 천천히 이동함에 따라 최초로 접촉한 컨벡스 혈의 색상이 변화도록 구성한 것이다. 물리엔진에서 제공되는 충돌 감지는 hit로 표현되고, 위 예와 같이 경계체적을 사용하여 겹치는 부분을 찾는 것은 overlap으로 표현하는 것이 일반적이다.

이와 같이 경계 체적을 이용하여 충돌 지점을 특정한 후 해양 시뮬레이터를 이용하여 동역학을 고려한 반응이 이루어지도록 활용 가능하다. 원본 메쉬 모델이 약 24만개의 꼭짓점과 12만개의 삼각형으로 이루어진 것에 비해 충돌 검사에 필요한 도형의 수를 크게 줄임으로써 연산에 필요한 시간을 줄일 수 있다. 시뮬레이션을 수행하는 강력한 하드웨어의 지원이 가능한 경우 메쉬 모델을 사용할 수 있으나, 그렇지 못할 경우에도 컨벡스 혈의 트리 구조를 이용하여 빠른 충돌 연산이 이루어질 수 있도록 구성이 가능한 것이다.

4. 결 론

자율운항선박 시뮬레이션에 필수적인 충돌 감지 방법에 대한 기초연구를 수행하였다. 전통적인 방식은 경계체적을 이용하여 연산시간을 줄이는 것이다. 컴퓨터 3D 모델을 직접적으로 사용하기에는 연산시간이 과도하게 소요되므로 충돌 감지 단계에서 광역과 협역의 2단계로 작업을 진행한다. 공간 분할이나 경계체적 방식 또는 두 방식을 혼합하여 정밀한 감지가 필요한 개체들을 선정한 후 GJK 알고리즘 등을 이용하여 겹침 여부를 판정하는 것이다. 현대에는 컴퓨터 그래픽을 위한 하드웨어가 급속도로 발전함에 따라 3D 모델 자체를 충돌 여부 판정에 사용하는 경우도 생기고 있다. 하드웨어의 지원이 필요한 부분이므로 선박 해양 분야 시뮬레이션에 활용하기 위해서는 그 목적에 따라 다양한 방법을 염두에 두고 가장 적합한 방식을 채택하는 것이 필요하다. 추가 연구를 통하여 선박의 이접안과 협업 작업 진행 시 접촉 구역을 정확하게 판별하는 데 적합한 방식을 선정하여 시뮬레이션에 적용할 예정이다.

후 기

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성 (Grand ICT연구센터) 사업의 연구결과로 수행되었음 (IITP-2023-2020-0-01791)

참 고 문 헌

- Hölzer, C., Schmidt-Vollus, R., Gödrich, A., Helldörfer, B., Jehle, G. and Riedlbauer, D.R., 2022. Comparison of Commercial Physics-Based Simulation Environments with a Collision-Rich Benchmark. *10th International Conference on Systems and Control (ICSC)*, pp. 516-521.
- Montanari, M., Petrinic, N. and Barbieri, E., 2017. Improving the GJK algorithm for faster and more reliable distance queries between convex objects. *ACM Transactions on Graphics*, 36(3), pp.1-17.
- Plowman, J., 2016. *3D Game Design with Unreal Engine 4 and Blender*. Packt Publishing Ltd.
- Thul, D., Ladicky, L., Jeong, S. and Pollefeys, M., 2018. Approximate convex decomposition and transfer for animated meshes. *ACM Transaction on Graphics*, 37(6), pp.226-1.