

자율운항 선박의 접안 시뮬레이션에 사용되는 경계 체적의 생성

이재용
동의대학교 조선해양공학과

Generation of Bounding Volumes for Berthing Simulation of Autonomous Ships

Jaeyong Lee
Department of Naval Architecture and Ocean Engineering, Dong-eui University

요약 자율운항 선박의 접안을 시뮬레이션하기 위해서는 선박과 안벽 사이의 접촉을 감지하는 것이 필수적이다. 현재 자율운항 선박의 알고리즘 검증에 위한 시뮬레이션은 대부분 2차원 평면에서 이루어지고 있다. 본 논문에서는 3차원 시뮬레이션에서 개체 간의 접촉 여부뿐 아니라 접촉 지점까지 추출하기 위한 경계 체적의 생성 기법을 제시한다. 복잡한 형태의 선체를 4개로 분할하고, V-HACD 방법을 분할된 선체에 순차적으로 적용한다. 평평한 부분이 대부분인 선체의 중앙부에 대해서는 V-HACD 알고리즘의 한계로 인해 세분화 작업을 별도로 진행한다. 이후 충돌 감지용 소프트웨어를 활용하여 계층화된 구조체에 대하여 안벽과의 접촉과 관련한 설정을 진행한다. 시뮬레이션을 통해 사등분된 선체에 대해 각각 수행된 세분화 결과물들을 하나의 선체로 통합하는 과정이 정상적으로 이루어지는 것을 확인하였다. 접안 시뮬레이션에서 선체 측면의 접촉 위치가 1m의 정밀도로 성공적으로 검출되었다. 제시된 방법은 선박의 접안 제어 알고리즘의 성능을 검증하는 시뮬레이션 기법을 고도화하는 데 사용될 수 있을 것이다.

Abstract In order to simulate the berthing of an autonomous vessel, it is essential to detect the contact between the vessel and the quay walls. Currently, simulations to verify algorithms for autonomous ships are mostly performed on a two-dimensional plane. This paper presents a process to generate bounding volumes to extract not only contact between objects in a 3D simulation, but also the point of contact. A complex hull shape is divided into four parts, and the V-HACD method is sequentially applied to the divided hull. For the middle section of the hull, which is mostly flat, subdivision is performed separately. Next, the software for collision detection proceeds with settings related to contact with the quay for a hierarchical structure. The simulation confirmed that the process of integrating the decomposition work performed on each quartered hull portion into one hull was successful. In the berthing simulation, the contact position of the hull side was successfully detected with a precision of 1 m. The proposed method could be used to improve simulation techniques to verify the performance of a ship's berthing control algorithm.

Keywords : Berthing, Bounding Volume, Collision Check, Convex Hull, V-HACD

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임. (RS-2023-00247537)
본 연구는 과학기술정보통신부 및 IITP의 지역지능화혁신인재양성 사업의 연구결과로 수행되었음. (IITP-2023-2020-0-01791)
이 논문은 2023학년도 동의대학교 교내연구비에 의해 연구되었음. (202301370001)
*Corresponding Author : Jaeyong Lee(Dong-eui University)

email: jlee@deu.ac.kr
Received August 8, 2023
Accepted October 6, 2023

Revised September 7, 2023
Published October 31, 2023

1. 서론

자율 시스템에 대한 관심이 증가하면서, 자율운항 선박에 대한 연구 또한 활발히 이루어 있다. 자율운항 선박의 운영과 관련하여 어려운 문제 중의 하나는 접안과 관련한 것이다. 따라서 자율운항 선박의 접안에 관한 연구를 위해서는 안벽과 접촉하는 정보를 추출하여 제어 알고리즘에 반영하는 것이 필요하다. 또한 항만에서의 사이버물리시스템을 구현하기 위해서도 접안 때 접촉 지점에 대한 정보가 필요하다[1]. 하지만 현재 수행되는 접안 관련 시뮬레이션에서 알고리즘의 성공 여부는 2차원 평면에서 이루어지는 경우가 많다[2]. 이는 3차원 개체들에 대하여 접촉/충돌을 감지하고 정확한 위치를 산출하는 것이 어렵기 때문이다.

자율운항 선박의 접촉/충돌 여부를 검사하는 과정은 크게 두 단계로 이루어진다. 검사 과정이 많은 수의 개체를 대상으로 할 때는 효율성을 높이기 위하여 물체의 정보를 이용하여 충돌 감지 알고리즘을 적용할 개체를 선별하는 작업을 먼저 진행한다. 이를 광역 충돌 탐지(broad phase collision detection)라고 한다. 광역 충돌 탐지에서 선별된 쌍들에 대해 접촉 여부를 판정하는 것이 협역 충돌 탐지(narrow phase collision detection) 단계에서 이루어진다.

광역 충돌 탐지는 경계 체적(bounding volume) 등과 같이 개체를 간략하게 표시하는 방법을 주로 사용한다. 경계 체적은 개체를 완전히 포함하기 위하여 사용되는 닫힌 체적이다. 충돌 가능성이 있는 쌍을 추려내고 나머지는 제외하는 작업이 수행된다. 비교적 빠르게 검사하는 것이 가능하며, 여러 가지 가속화 기법이 사용된다.

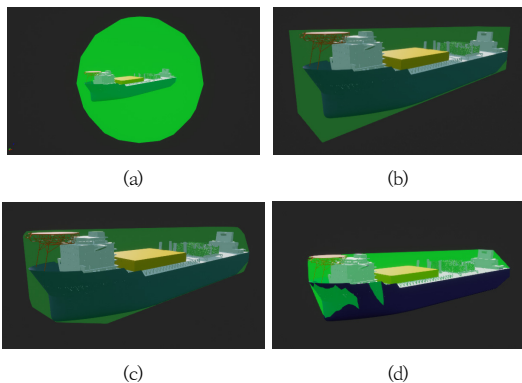


Fig. 1. Types of bounding volumes for a ship
(a) Sphere (b) AABB/OBB (c) k-DOP (d) Convex hull

협역 단계에서는 광역 단계에서 선정된 쌍에 대해 접촉 여부 검사를 수행하는 단계이다. 광역 단계에서 수행된 검사보다 정밀한 충돌 검사가 이루어진다. 협역 단계 내에서도 세부 단계를 두어 검사를 수행할 수도 있다. 삼각형과 삼각형 또는 삼각형과 선의 접촉에 대한 검사가 주로 이루어진다. 광역 단계에 사용되는 방식에 비해 많은 시간이 소요될 수 있다. 개체의 수가 n 개일 때 최악의 경우 $O(n^2)$ 의 연산이 필요할 수도 있다.

충돌검사를 위한 경계 체적으로는 보통 구, AABB (axis aligned bounding box), OBB (oriented bounding box), k -DOP (discrete oriented polytope), 그리고 볼록 껍질(convex hull) 등이 사용된다. Fig. 1은 선박의 접촉 감지에 사용되는 경계 체적의 예를 보여주고 있으며, 광역 충돌 탐지 단계에서 활용될 수 있다. Nguyen et al.은 시추선 상부의 이동형 장비들에 대해 OBB를 이용하여 장비 사이의 충돌을 방지하는 로직을 개발하였다 [3]. 경계 체적을 이용하는 경우 기본적으로 접촉이 일어났음에도 접촉이 없는 것으로 판정하는 거짓 음성(false negative)이 발생하지 않아야 한다.

선박과 같이 복잡한 물체의 경우 하나의 경계 체적으로는 형태를 정확하게 표현하는 데 한계가 있다. 이런 경우 경계 체적들의 조합으로 물체를 표현하면 더욱 정밀한 표현이 가능하다. 이러한 개념은 1990년대부터 제시되었으며 기본 경계 체적인 구, AABB/OBB, k -DOP 및 convex hull을 사용하는 방식들이 사용되고 있다[4-7]. 예를 들어, 경계 체적을 생성하고자 하는 개체에 여러 개의 구를 사용하여 목표 개체를 감싸는 식이다. Varela와 Soares는 계층적 구조를 활용하여 접촉 지점을 추출하는 방식을 제안하였다[8,9]. 현재 많이 활용되는 방법은 V-HACD (Voxelized-Hierarchical Approximate Convex Decomposition)이며, 해당 개체를 여러 개의 볼록 껍질로 구성하는 데 사용되는 최적화 기법이다[10]. 대형 선박의 경우 복잡한 형태의 상부 구조물, 유선형 형태를 지니는 선수와 선미, 그리고 비교적 평평한 부분이 많은 선측과 선저를 가지고 있는 경우가 많다. V-HACD를 적용하는 경우 최적화 기법이 적용되기 때문에 평평한 부분은 적은 수의 볼록 껍질로 분해된다. 선박에 있어 이러한 점은 정교한 접촉 지점을 추출하는 데 제약사항으로 작용한다. 또한 선박 운영의 시나리오에 따라 접촉이 일어날 수 있는 영역을 제한하면서 계산상의 효율을 높일 필요가 있다. 하지만, 단일한 알고리즘을 적용할 때 V-HACD 알고리즘에 의해 선수, 선미 그리고 상부 구조물은 선측이나 선저에 비하여 많은 수의 볼록 껍질로 이

루어지게 된다. 접안 시뮬레이션에서는 상부 구조물은 상세한 분할이 불필요하고 하부 선체(선수, 선미 그리고 선측)에 대한 상세한 분할이 필요하다. 만일 선측을 기준으로 하여 상세한 구분을 하면 상부와 선수 그리고 선미에 지나치게 상세한 구획이 이루어진다. 반면에 상부 구조물을 기준으로 단순한 구분을 한다면, 선측 부분이 너무 단순하게 구획된다. 따라서 하나의 알고리즘으로는 원하는 수준의 정밀도를 유지하기가 어렵다.

본 논문에서는 선체를 V-HACD와 세분화 기법으로 분할하여 많은 수의 경계 체적으로 구성하고, 이를 안벽 접안 시뮬레이션에 활용하는 방법을 제시한다. 3D 시뮬레이션에서 개체 사이의 접촉 여부를 검사하는 일반적인 사항을 고려한 후, 선박 접안 시뮬레이션에 고려되어야 하는 점을 도출한다.

2. 충돌 감지와 경계 체적

2.1 충돌 감지 알고리즘

일반적으로 경계 체적으로 AABB나 OBB가 사용되었을 경우, 초평면 분리 정리(hyperplane separation theorem)에 이론적 기반을 두는 분리 축 이론(separating axis theorem)이 사용된다. 복수의 다면체를 특정 축에 투영시켰을 때, 서로 겹치지 않는다면 그 축을 분리 축이라고 한다. 분리 축의 존재 여부로 다면체가 겹치는지에 대한 여부를 판단하는 것이다. k -DOP나 컨벡스 헐 등의 복잡한 도형에 대해서는 민코프스키 덧셈(Minkowski addition)의 특수한 형태인 민코프스키 차분(Minkowski difference)을 이용하게 된다. 두 개체에 대하여 민코프스키 차분을 수행한 결과가 원점을 포함하고 있으면, 두 개체는 겹치는 부분이 있다는 것을 의미한다. 만일 그 결과가 원점을 포함하지 않다면, 원점과 민코프스키 차분의 결과물과 원점 사이의 최소거리가 두 개체 사이의 최단 거리를 나타낸다.

GJK (Gilbert-Johnson-Keerthi) 알고리즘을 사용하면, 볼록 겹질 전체에 대해 민코프스키 차분을 구하지 않고, 일부분만 사용해서 두 개체 사이의 중첩을 판정할 수 있다[11]. 중첩 후 얼마나 안쪽까지 들어왔는지에 대한 깊이 값은 확장 폴리토프 알고리즘(EPA, expanding polytope algorithm)을 추가로 적용함으로써 알 수 있다. 즉 민코프스키 차분으로 접촉 여부를 판정하고, EPA 알고리즘으로 접촉 깊이를 지정하게 된다.

2.2 경계 체적의 특성 설정

컴퓨터 그래픽 분야에서는 두 개체가 같은 공간을 점유하고 있을 때 상대방에 대해 어떻게 처리할 것인지에 대한 여러 가지 설정이 가능하다. 용어는 사용하는 프로그램에 따라 여러 가지가 사용된다. 접촉이 일어난 이후 처리와 관련해서는 보통 겹침(overlap), 차단(block), 또는 무시(ignore) 등이 사용되고, 응답에 대한 정보 생성과 관련해서는 충돌(hit)과 겹침(overlap) 등이 사용된다. 차단은 두 개의 물리 개체가 서로 중첩되지 않도록 하는 것이고, 겹침과 무시는 중첩이 일어나는 것을 허용하는 것이다. 겹침과 무시의 차이는 충돌 응답의 생성 여부이다. 겹침에서는 충돌 응답을 생성하는 것이 가능하지만, 무시에서는 그렇지 않다. 선박 운영과 관련한 시뮬레이션에서는 때에 따라 겹침과 차단을 혼용해서 사용하는 것이 필요하다. 선박 접안의 경우 안벽에 보호용 방현재(fender)가 설치되어 있는 경우가 많고, 정상적인 운영을 위해서는 선박과 방현재 사이에 차단 개념을 적용하는 것이 필요하다. 또한 선박 간의 충돌이나, 선박과 안벽과의 충돌 또는 기계학습에 적용하기 위해서는 겹침을 적용하는 것이 필요할 수도 있다. 시뮬레이션을 위해서는 겹침을 허용할 것인지, 그리고 접촉이 발생하는 순간 알림을 설정할 것인지를 결정하는 것이 필요하다.

Table 1. Objects' settings and corresponding events that can be generated for objects

		O_1^2	
		Block	Overlap
O_2^1	Block	Hit event	Overlap event
	Overlap	Overlap event	Overlap event

Table 1은 2개의 개체가 상대방을 어떻게 처리할 것인가에 대한 설정과 이에 따른 응답의 종류를 나타낸다. 겹침과 무시는 응답(response) 생성 여부에 대해서만 다르고 특정 자체는 같으므로 응답이 있는 경우만을 고려해야 하는 본 논문에서는 사용하지 않기로 한다. O_1^2 는 개체 1이 개체 2를 어떻게 처리하는가에 대한 것이고, O_2^1 는 개체 2가 개체 1을 처리하는 방식이다. 각 개체의 차단과 겹침은 상대방을 차단 처리하거나 상대방의 설정과 관계없이 겹침이 되도록 한다는 것을 의미한다. 선박 운영에 적용할 때는 상황에 따라 상대방에 대해 차단 또

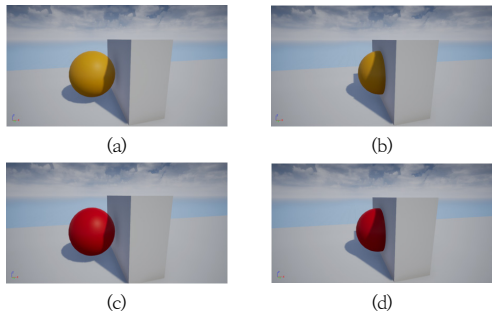


Fig. 2. Types of responses depending on the settings as two objects approach each other
 (a) Block-Block (b) Overlap (c) Hit response
 (d) Overlap response

는 겹침 설정을 하게 된다. 예를 들어 자동 접안 시뮬레이션에서는 안벽과 선박이 서로 차단하도록 설정하여 hit 이벤트를 발생시킴으로써 선박과 안벽의 접촉 지점을 추출하게 된다. 선박 운용 시뮬레이션에서는 주어진 입력에 따라 어느 정도의 겹침이 발생하는지를 정량적으로 파악하기 위하여 겹침으로 설정하는 것이 필요할 수도 있다.

Fig. 2는 두 개의 물리 개체(구와 직육면체)에 대하여 실시간 3D 제작 소프트웨어로 널리 사용되고 있는 언리얼 엔진(Unreal engine)에서 겹침을 처리하는 방식을 보여주고 있다. 구를 개체 1, 직육면체를 개체 2라고 설정한다. (a)는 두 개체가 모두 차단으로 설정되었을 때, 개체 1을 개체 2의 방향으로 이동시켜도 겹침 현상이 없이 두 개체가 충돌하는 모습을 보여주고 있다. (b)는 두 개체 중 하나의 설정이 겹침일 경우 다른 개체의 설정과 관계없이 겹침이 발생하는 모습을 나타낸다. 즉 두 개체 사이에 겹침-차단, 차단-겹침 및 겹침-겹침 등 3가지 조합 모두 같은 현상을 보여주게 된다. (c)에서는 두 개체 모두 차단으로 설정된 (a)의 경우에 대하여 개체 1에 hit event를 발생시켰을 경우를 보여주고 있다. 즉 개체 1이 다른 개체와 차단-차단으로 접촉할 때 적색으로 변하는

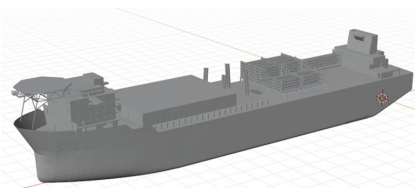


Fig. 3. 3D mesh model for the simulation (drilling vessel without derrick and associated equipment)

hit event를 설정한 것이다. 위 예는 색상이 변하는 것을 표현한 것이나 충돌이 발행한 개체의 위치 정보 등을 함께 생성시킬 수 있다. 개체 2의 경우 hit response를 설정하지 않아 아무런 정보도 생성시키지 않는 상황을 나타낸 것이다. (d)는 개체 1에 대하여 overlap event를 발생시키고 있는 경우이다. 즉 (b)에서와 같이 개체 1이 개체 2로 접근하다가 겹침 현상이 발생하였을 경우 색상이 적색으로 변하도록 설정한 결과를 나타낸다. Overlap event는 원하는 결과를 얻을 수 있도록 다양한 설정이 가능하다. 개체 2는 앞선 경우와 마찬가지로, overlap event 설정을 하지 않음으로써 아무 변화가 없는 것을 확인할 수 있다.

접안 시뮬레이션에 적합한 설정은 선박을 개체 1로 안벽을 개체 2로 가정하는 것이다. 선박이 접안할 때 정상적인 운영이 이루어지면 3D 개체 간의 중첩은 거의 일어나지 않는다. 따라서 개체들은 서로를 차단으로 설정한다. 이때, 개체 1은 선박으로서 접촉 지점에 대한 정보를 추출하는 것이 필요하다. 따라서 (c)에서와 같이 개체 1에 hit response가 일어나도록 설정하여 접촉 지점에 관한 정보가 생성되도록 한다. 선박과 선박 사이 또는 선박과 안벽 사이의 중첩을 허용해야 할 수도 있다. 선박을 구성하는 경계 체적에 대해 다른 개체와 중첩이 일어나도록 설정하고, (d)에서와 같이 overlap response가 일어나도록 구성하면 관련된 정보를 생성할 수 있다.

Table 2. Parameter names and their usage in V-HACD 4.0

parameter name	usage	values
n	Maximum number of output convex hulls	1,000,000
voxelresolution	Total number of voxels to use	10,000,000
volumeErrorPercent	Volume error allowed as a percentage	0.001
maxRecursionDepth	Maximum recursion depth	64
maxHullVertCount	Maximum number of vertices in the output convex hull	64
minEdgeLength	Minimum size of a voxel edge	variable

3. 선박용 경계 체적의 설정

Fig. 3은 시뮬레이션에 사용될 전장 215m의 시추선 모델이다. 접안 시뮬레이션에 활용하는 목적이므로 상부 구조물에 탑재되는 시추탑(derrick) 구조물과 그 부속 기 자재는 제외하였다. 접안 시뮬레이션에서 접촉 부위를 추출하기 위해서는 충돌 검사에 사용되는 경계 체적의 크기가 작아야 한다. 본 논문에서는 기본 경계 체적으로 볼록 껍질을 선정하고, 여러 개의 볼록 껍질로 대상 선박의 외형을 구현하는 계층적 분해 방법을 적용한다. 3D 시뮬레이션에 많이 활용되는 언리얼 엔진 (Unreal Engine)의 경우 총 64개의 Convex hull을 생성할 수 있어 세분화에 한계가 있다. 따라서 V-HACD 4.0 라이브러리를 이용하여 선박의 부위별로 볼록 껍질의 수를 조절하는 방법을 적용하였다. V-HACD 4.0은 총 12개의 파라미터들을 조절하면서 개체에 대한 분할을 수행할 수 있다. 파라미터들 중 볼록 껍질의 크기 설정과 직접적으로 관련된 6개의 파라미터들에 대한 설명과 시뮬레이션에서 사용한 설정값들을 Table 2에 정리하였고, 그 결과들은 Fig. 4에 나타내었다. 다른 파라미터들은 같은 값을 유지하면서 minEdgeLenth를 20, 10, 5, 1로 점차 감소시켜 감에 따라 생성된 볼록 껍질의 수가 201개, 901개, 3,510개, 27,626개로 증가한 것을 확인할 수 있

다. 이는 같은 선박을 더 많은 수의 볼록 껍질로 구성하였다는 것을 의미하므로 더 세분된 결과라고 할 수 있다. 선수와 선미의 곡선 부분 그리고 상부 구조물의 복잡한 형태는 예상했던 바와 같이 아주 자세한 분할이 이루어진 것을 볼 수 있다. 하지만 선체 측면부의 경우 miniEdgeLength를 감소시키더라도 큰 변화가 없는 것을 확인할 수 있다. 이는 선박 중앙부의 측면은 평평한 부분의 면적이 넓어 최적화 알고리즘이 적용되는 V-HACD 알고리즘으로는 한계가 있기 때문이다. 분할된 볼록 껍질의 크기가 크면 다른 구조물과 겹침이 발생하였을 때, 정확한 위치 정보를 추출하기가 어렵다. 따라서 선박 전체에 대해 V-HACD를 적용하여 분할하는 것은 충돌 위치 추정에 어려움이 발생한다.

4. 시뮬레이션

Fig. 5는 3D 그래픽 제작 소프트웨어로 널리 사용되는 Blender를 이용하여 해당 선박에 대해 계층적 구조를 구성한 예이다. (a)에서 전체적으로는 선박을 상부 구조물, 선수, 선미 및 선체 중앙부로 사 등분 하였다. 앞서 그림에서 확인한 바와 같이 선체 중앙부에 대한 세밀한 묘사를 위해서는 별도의 경계 체적 설정이 필요하다. 본

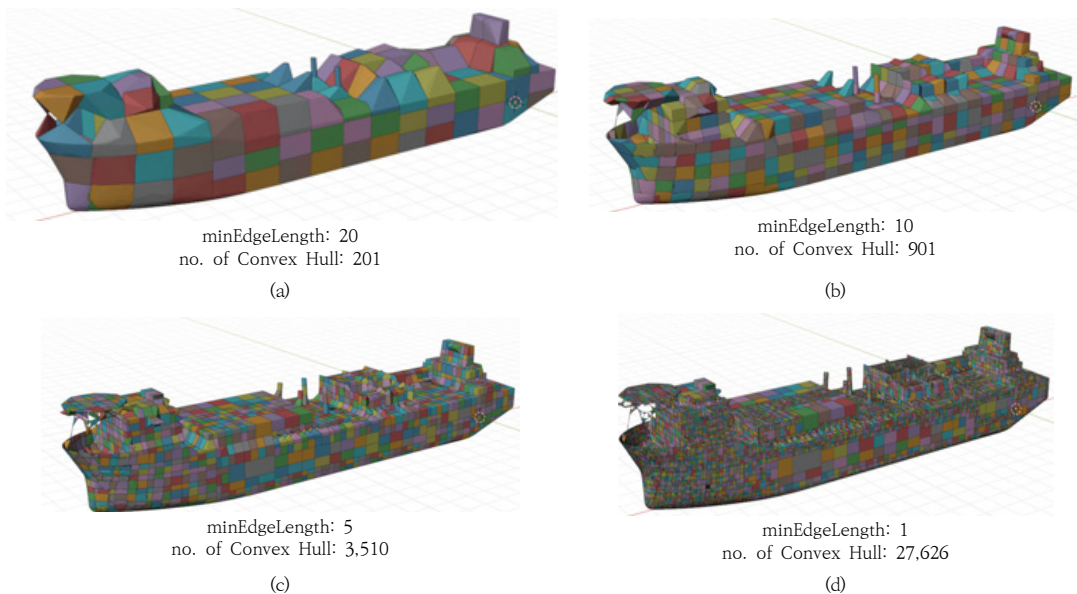


Fig. 4. Results of V-HACD algorithm for the target vessel

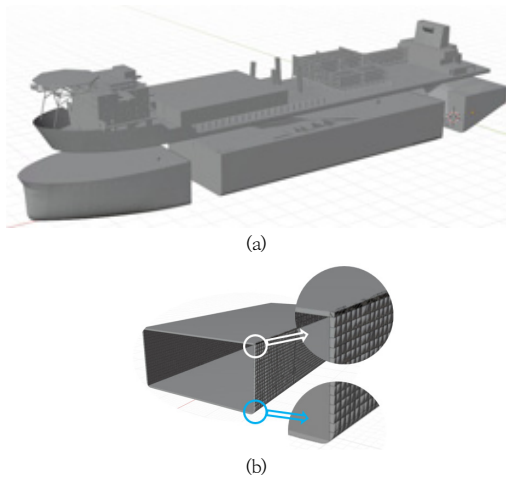


Fig. 5. Formation of a hierarchical structure for the hull

(a) The hull is divided into 4 parts: bow, stern, super structure and mid section (b) The corners of the mid section are decomposed into smaller bounding volumes

논문에서는 선체 측면을 가로와 세로의 길이를 각 1m로 세분화하였다. 길이 90m에 대하여 전체 3,864개의 개체로 구성된 새로운 선체 중앙부 세분화 작업이 진행되었다. 세분화 작업은 Blender에서 길이 1m의 직육면체와 해당 선체 사이에 불(Boolean) 연산을 통하여 진행하였다. (b)에서 선체 분할된 경계 체적들이 선체의 외곽 면을 잘 표현하고 있는 것을 확인할 수 있다.

Fig. 6은 사등분된 선체에 대해 수행된 세분화의 결과들을 모아 다시 하나의 개체로 결합한 것이다. 이를 fbx로 변환하여 언리얼 엔진에서 시뮬레이션을 수행하였다. 시뮬레이션은 크게 접촉 상황에서 hit event를 발생시키는 것과 겹침 상황에서 overlap event를 발생시키는 것으로 하였다.

시뮬레이션은 접촉을 감지해야 하는 상황과 중첩을 허용하는 상황으로 구분하여 진행하였다. 외부 크레인이 없는 상황을 설정하였고, 안벽에는 방현재를 설치하여 선박이 접촉해야 할 지점을 지정하였다. Blender와 V-HACD 알고리즘을 적용하여 각 부분에 필요한 정도의 경계 체적들로 분해하였다. 각각의 경계 체적들은 고

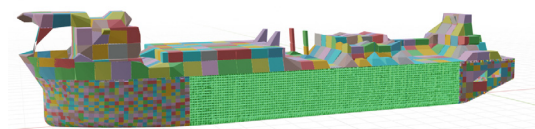


Fig. 6. Reconstructed hull form

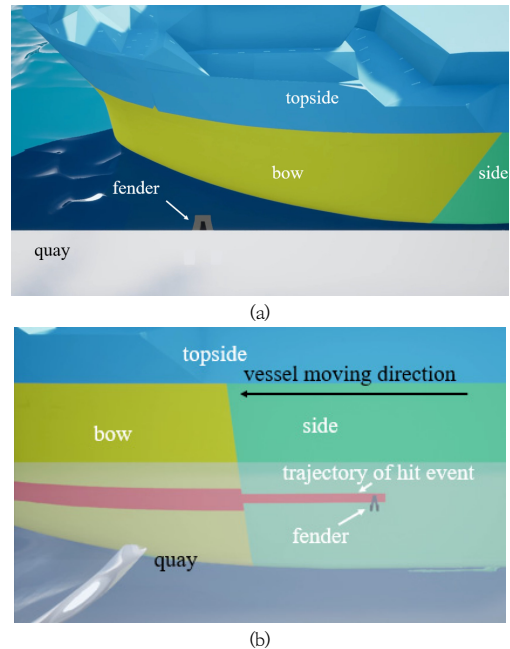


Fig. 7. Generation of hit event representing the collision between the ship and the fender

(a) General configuration for the simulation (b) The color of the bounding volumes that collided with the fender is sequentially changing.

유 아이디를 가지는 별개의 경계 체적이 된다. 시뮬레이션은 Unreal 4.27 버전으로 수행하였다.

Fig. 7은 선박이 접안을 위해 안벽으로 접근하면서 세분화된 경계 상자들이 정상적으로 안벽의 방현재와 접촉하여 hit event를 발생시키는 시뮬레이션 결과이다. 그림 (a)는 전체적인 구조를 나타내고 있다. 선박의 계층적 구조를 다른 색상으로 표시하였다. 안벽에는 보통 선박과의 직접적인 접촉을 피하고 완충작용을 위해 그림에서와 같이 방현재가 설치되어 있다. 방현재와 선박이 각각 상대방을 차단하도록 설정하여 hit event가 발생할 수 있도록 하였다. 계층적 구조에 의해 선박을 구성하고 있는 경계 체적들도 모두 안벽과 그 부속물에 차단 설정이 이루어진다.

선박이 안벽에 접근하면서 접촉이 일어난 경계 체적이 hit event가 발생하고, 해당 경계 체적의 색상이 변하는 것을 그림 (b)에서 확인할 수 있다. 그림에서 선박이 방현재를 연속으로 접촉하면서 지나가므로 순차적으로 해당 경계 체적의 색상이 변하는 것을 볼 수 있다. 선수에서 선미 방향으로 선체 외부가 방현재와 연속적인 접촉할 때, 경계 상자에 대한 hit event가 연속적으로 진행되는 상황을 나타내고 있다. 선수 부분은 V-HACD 알고리

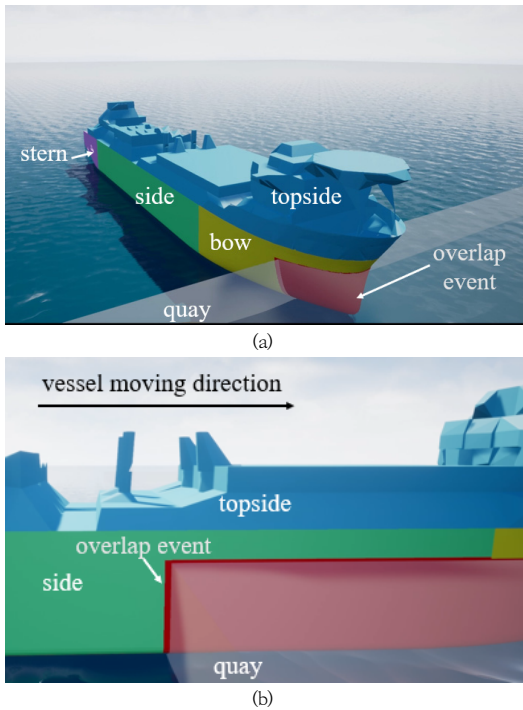


Fig. 8. Generation of hit event representing the collision between the ship and the fender (a) General configuration for the simulation (b) The color of the bounding volumes that collided with the fender is sequentially changing.

즘이 적용되었으므로, 정형적이지 못한 모양을 나타낸다. 선측 부분은 앞서 선체 중앙부에 대한 세분화 작업의 결과로 가로와 세로가 각 1m인 정사각형 모양이 연속적으로 나타나고 있다. 그림에서 접촉이 끝난 구역에 hit event의 효과가 지속되고 있는 것은 충돌 이력을 관리하고 있다는 것을 보이기 위해서이다.

Fig. 8은 선박과 안벽이 중첩이 일어나도록 설정한 후, overlap event를 발생시킴으로써 중첩이 되는 경계 체적의 색상이 변하도록 한 결과를 나타내고 있다. 일반적인 3D 객체의 중첩과 같은 것으로 보일 수도 있으나 실제로는 다른 프로세스를 거치게 된다. Overlap event가 발생하기 위해서는 중첩이 일어나는지에 대한 검사가 필요하다. 선박이 안벽 쪽으로 계속 진입해 들어감에 따라 새로이 중첩되는 경계 체적들의 색상이 변하는 것을 확인할 수 있다. Overlap event가 발생한 경계 체적들의 아이디를 저장하도록 하였고, 이를 통해 해당 경계 체적을 구성하는 꼭짓점들과 변들의 정보를 이용하여 추가적인 작업을 진행할 수 있을 것이다.

5. 결론

본 논문에서는 선박의 접안 시뮬레이션에 활용되는 접촉 지점 산출에 대한 기초 연구를 수행하였으며 주요 결과는 다음과 같이 정리할 수 있다.

첫째, 선박과 안벽의 접촉을 감지하기 위하여 선박을 여러 개의 경계 체적으로 분해하는 작업을 진행하였다. 연산량이 과도해지는 것을 방지하고, 분해에 사용되는 알고리즘의 한계를 보완하기 위하여 선박을 4개의 구역으로 분할한 다음 각각의 구역에 대해 적절한 개수의 경계 체적이 생성되도록 하였다. 선체 중앙부의 경우 볼록 꺾질의 수가 기존 알고리즘의 경우 572개였으나, 5,402개로 크게 증가하였다. 또한 제안된 방식은 볼록 꺾질의 가로 세로 길이도 기존의 11.5m에서 1m로 정밀도를 대폭 향상시켰다.

둘째, 개별 아이디를 가지는 경계 체적들로 계층화된 구조를 생성한 후 접촉 및 중첩 검사를 수행하였다. 선박의 접촉과 관련하여 상대적인 대응을 block과 overlap으로 구분하고, 이에 대한 응답으로는 hit과 overlap 이벤트가 발생하도록 하였으며, 정상적으로 이벤트가 발생하는 것을 확인하였다.

셋째, hit 이벤트 발생 시 해당 볼록 꺾질의 색상을 변화시켜 접촉 지점이 나타나도록 하였고, 변경된 색상이 유지되도록 하여 이력 관리가 정상적으로 이루어지는 것을 확인하였다. 또한 Overlap 이벤트가 발생할 때는 Overlap이 발생한 볼록 꺾질들 전체가 색상이 변하도록 설정하였고, 시뮬레이션에서 정상적으로 작동함을 확인하였다.

추후 관련 연구를 진행하여 자율운항 선박의 접안과 관련한 시뮬레이션과 알고리즘 개발에 활용할 예정이다. 또한 Overlap과 관련해서 현재는 중첩이 일어난 볼록 꺾질들에 대한 구분만 가능하다. 하지만 실제 응용에 활용하기 위해서는 정확한 중첩 면적 및 공간의 분리가 필요하다. 이를 위해서는 중첩된 볼록 꺾질 정보를 이용하여 절단면을 생성하는 알고리즘이 필요하다. 이를 통해 선박 운용과 관련한 기계학습 알고리즘에 적용할 수 있을 것으로 기대된다.

References

- [1] S. Ahn, J.-B. Lee, J. Kim, Y.-H. Sohn, H.-M. Koo, "A Study on the Design of the Import/Export Ports

Cyber-Physical System for Intelligent Vehicle”, *Journal of the Korea Academia-Industrial cooperation Society*, Vol.22, No.12, pp.25-31, 2021.

DOI: <https://doi.org/10.5762/KAIS.2021.22.12.25>

- [2] S. Hong, J. Kim, “Reinforcement Learning Based Tugboats Control for Autonomous Ship Berthing”, *Journal of the KNST*, Vol.4, No.1, pp.72-77, 2021.
DOI: <https://doi.org/10.31818/JKNST.2021.03.4.1.72>
- [3] D. T. Nguyen, K. H. Jung, K. Y. Kwon, N. Ku, J. Lee, “A kinematic collision box algorithm applied for the anti-collision system of offshore drilling vessels”, *Journal of Marine Science and Engineering*, Vol.8, No.6, 2020.
DOI: <https://doi.org/10.3390/jmse8060420>
- [4] P. M. Hubbard, “Approximating polyhedral with spheres for time-critical collision detection”, *ACM Transactions on Graphics*, Vol.15, No.3, pp.1779-210, 1996.
DOI: <https://doi.org/10.1145/231731.231732>
- [5] G. V. D. Bergen, “Efficient collision detection of complex deformable models using AABB trees”, *Journal of Graphics Tools*, Vol.2, No.4, pp.1-13, 1997.
DOI: <http://dx.doi.org/10.1080/10867651.1997.10487480>
- [6] S. Gottscha, M. C. Lin, D. Manocha, “OBBTree: A hierarchical structure for rapid interference detection”, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp.171-180, August 1996.
DOI: <https://doi.org/10.1145/237170.237244>
- [7] J. T. Klosowski, M. Held, J. S. Mitchell, H. Sowizral, K. S. Zikan, “Efficient collision detection using bounding volume hierarchies of k-DOPs”, *IEEE transactions on Visualization and Computer Graphics*, Vol.4, No.1, pp.21-36, 1998.
DOI: <http://dx.doi.org/10.1109/2945.675649>
- [8] J. M. Varela, C. Guedes Soares, “Geometry and visual realism of ship models for digital ship bridge simulators”, *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, Vol.231, No.1, pp.329-341, 2017.
DOI: <http://dx.doi.org/10.1177/1475090216642470>
- [9] J. M. Varela, C. Guedes Soares, “Software architecture of an interface for three-dimensional collision handling in maritime Virtual Environments”, *Simulation*, Vol.91, No.8, pp.735-749, 2015.
DOI: <http://dx.doi.org/10.1177/0037549715598008>
- [10] K. Mammou, V-HACD library version 4.1, Available from: <https://github.com/kmammou/v-hacd> (accessed Aug. 06, 2023)
- [11] E. G. Gilbert, D. W. Johnson, S. S. Keerthi, “A Fast procedure for computing the distance between complex objects in three-dimensional space”, *IEEE Journal on Robotics and Automation*, Vol.4, No.1, pp.193-203, 1988.
DOI: <http://dx.doi.org/10.1109/56.2083>

이 재 용(Jaeyong Lee)

[정회원]



- 2002년 8월 : Texas A&M Univ., 기계공학과 (공학석사)
- 2007년 8월 : Texas A&M Univ., 기계공학과 (공학박사)
- 2007년 8월 ~ 2012년 8월 : 삼성중공업 연구소 책임연구원
- 2012년 9월 ~ 현재 : 동의대학교 조선해양공학과 교수

<관심분야>

해양시스템 동역학 및 제어, 자율운항시스템 설계, 동적위 치유지 시스템 설계 및 해석