

# 지표면 모델링 및 폴리곤 검색기법에 관한 연구

공지영<sup>\*</sup>, 강현주<sup>\*</sup>, 윤석준<sup>\*\*</sup>

e-mail : jykong@lycos.co.kr tel : 461-7444 fax : 461-7444

<sup>\*</sup> (주)시스코프

<sup>\*</sup> 세종대학교 대학원 항공우주공학과

<sup>\*\*</sup> 세종대학교 기계항공우주공학부

## A Study on Terrain Surface Modeling and Polygon-Searching Algorithms

JiYoung Kong, Hyunjoo Kang, Sugjoon Yoon

### Abstract

Terrain surfaces have to be modeled in very detail and wheel-surface contacting geometry must be well defined in order to obtain proper ground-reaction and friction forces for realistic simulation of off-road vehicles. Delaunay triangulation is one of the most widely used methods in modeling 3-dimensional terrain surfaces, and T-search is a relevant algorithm for searching resulting triangular polygons. The T-search method searches polygons in successive order and may not allow real-time computation of off-road vehicle dynamics if the terrain is modeled with many polygons, depending on the computer performance used in the simulation. In order to accelerate the searching speed of T-search, a terrain database of triangular polygons is modeled in multi-levels by adopting the LOD (Level of Detail) method used in realtime computer graphics. Simulation results show that the new LOD search is effective in shortening the required computing time. The LOD search can be even further accelerated by introducing an NN (Neural Network) algorithm, in the cases where a appropriate range of moving paths can be predicted by cultural information of the simulated terrain, such as lakes, houses, etc.. Numerical tests show that LOD-NN search almost double the speed of the original T-search.

**Key Words:** Delaunay Triangulation, Level of Detail, Neural Network

## 1. 서론

자동차나 군용 전차와 같이 지표면 위를 움직이는 동체의 운동을 발생시키는 근본적인 에너지는 연료와 그 연료를 이용한 엔진장치이다. 하지만, 차량이 이동할 수 있는 최종적인 근거는 차륜과 지면 사이에 이루어지는 작용 및 반작용이다.

즉, 차량의 운동을 엄밀하게 시뮬레이션하기 위해서는 지표 반발력을 정확히 산출할 수 있어야 하는데, 이를 위해서는 지형과 지표면의 정밀한 모델링이 차량 자체의 모델링 못지 않게 중요하게 대두된다. 특히, 차량이 비포장 도로나 들판과 산악 지형을 주행할 경우, 지표면은 평평한 형태를 갖고 있지 않기 때문에 지표면과 동체가 맞닿은 면의 정보 즉, 경사도와 그에 따른 지표반발력이 동체의 움직임이나 반동을 정확히 표현하기 위해 중요하다 할 것이다.

보통의 지형 데이터는 직교형 데이터로 한 점을 데이터에 추가하고자 할 때, 부수적으로 수많은 점들이 자동적으로 데이터에 추가된다. 따라서 정밀한 데이터를 표현하기 위해서는 데이터 테이블의 크기가 커지는 것을 피할 수 없다. 따라서 일반적인 데이터 테이블을 이용해서 지형을 사실적으로 정확히 표현하기 위해서는 그에 비례한 거대한 양의 데이터가 요구된다. 예를 들어, 그림 1에서 보면 단순한 수직, 수평분할 map에서 한 점을 추가하면 수많은 점들이 추가된다. 그러나 일반적인 컴퓨터 환경에서 실시간으로 시뮬레이션을 실행하기 위해서는 RAM의 용량이나 CPU의 처리속도 등 하드웨어적 기능이 고려되어야 한다. 따라서 다량의 데이터를 가지고 실시간 시뮬레이션에서 요구되는 검색 및 연산을 수행하기 위해서는 경제적 부담이 증가하거나 주어진 하드웨어로 실시간 연산처리가 불가능해질 수 있다. 그러므로 3차원 지표면에서 차량의 운동의 실시간 시뮬레이션은 빠른 속도의 검색기법이 요구된다.

Delaunay triangulation[1-3]과 Voronoi diagram[4-5]

은 적은 양의 데이터로 3차원 데이터를 정확하게 표현하기 위해 일반적으로 사용되는 분할방법이다. Voronoi diagram은 다각형으로 지표면을 모델링하고, 사실적인 모델링에 효과적이다. 그러나 평지는 일반적으로 다양한 다각형으로 되지 않는다. 그러므로 데이터를 동일하게 삼각형 형태로 분할하는 Delaunay triangulation이 검색 등 다른 알고리즘과 함께 사용하기에 보다 적합하다. Delaunay 방법은 GIS(Geographical Information System)에 광범위하게 사용되고[1-5], triangular 모델링은 정보 처리 상호운용을 위해 HLA(High Level Architecture)[6]의 대부분의 시뮬레이션에 적용된다.

Delaunay triangulation 분할방법으로 형성된 map에서 일반적으로 사용하는 검색 기법은 T-search 방법[1]이다. 그러나 T-search 방법은 순차검색을 하므로 실시간에 충분한 빠른 속도를 얻을 수 없다. 이 논문에서는 T-search 방법의 검색속도를 향상시키기 위해 두 가지 새로운 검색알고리즘을 개발하였다. 첫 번째 방법이 렌더링 속도를 가속시키기 위하여 실시간 그래픽에서 사용하여 데이터를 구성하는 LOD 기법[7-9]을 도입하였다. 수치시험 결과, 검색에 소요되는 시간이 일반적인 T-search 방법보다 빠르게 나타나 검색시간을 단축시키는데 효과적인 방법임을 알 수 있었다. 두 번째 검색 방법은 시뮬레이션의 대상이 되는 동체의 경로를 미리 예측할 수 있는 경우에서 얻은 것이다. 예를 들어, 동체는 깊은 호수, 높은 산 또는 숲으로 이동하지 않는다. 이러한 지형들은 검색하는 데이터 지형에는 필요 없다. 만약 동체의 경로를 미리 예측할 수 있다면, NN (Neural Network) 기법[10]은 LOD 방법보다 빠른 검색속도를 얻을 수 있다. 수치시험에서 검색시간을 단축시킴을 확인할 수 있었다.

## 2. 3D 지형데이터 모델링

### 2.1 Delaunay Triangulation

지형을 표현하는 방법 중 가장 평범하게 쓰이는 방법은 그림 1에서 보듯이 수평 좌표축과 수직 좌표축을 분할하여 그 교차지점의 데이터를 이용하여 표현하는 방법이다. 나타내고자 하는 지형을 얼마나 정밀하게 표현하고자 하는지에 따라 지형을 표현하기 위한 데이터의 개수가 결정되게 된다.

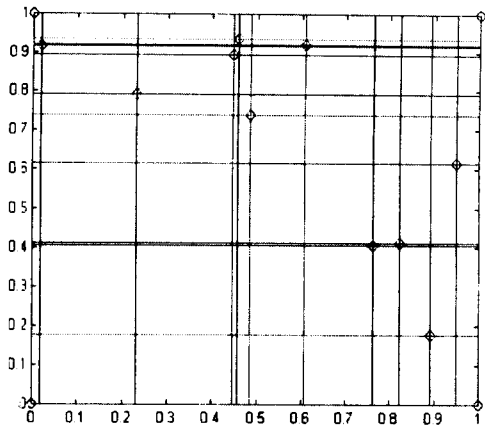


그림 1. 수직 수평 분할 map

한편, 그림 2를 보면 평면을 구성할 수 있는 최소 단위인 삼각형으로 지형을 표현할 수 있는 방법인 **Delaunay triangulation**을 고려할 수 있다. 좌표축을 분할하여 데이터를 형성하는 방법은 원하는 데이터를 추가하기 위해 많은 양의 부수적인 데이터들이 추가되게 되지만, **Delaunay triangulation** 방법을 사용하게 되면 지형의 특징을 나타내거나 혹은 원하는 데이터만으로 지형을 표현할 수 있다.

**Delaunay triangulation** 방법은 데이터가 입력될 때마다 기존의 데이터 점들과 연결하여, 그림 3과 같이 각 삼각형의 내접이 다른 점들이 포함되지 않도록 삼각형을 형성한다. 이렇게 삼각형을 형성함으로써 지형의 본래 형태가 그대로 유지된다.

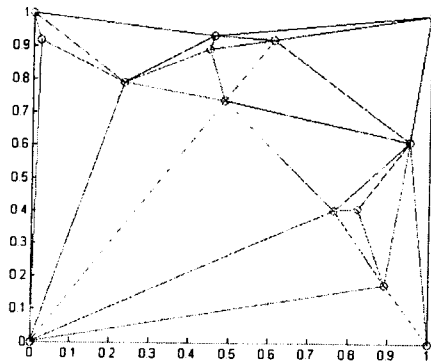


그림 2. Delaunay triangulation map

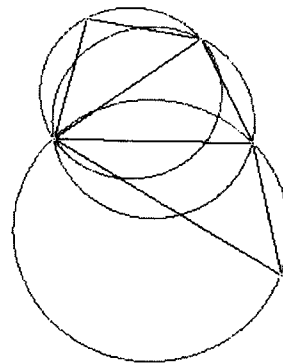


그림 3. Delaunay triangulation 과 내접원

## 2.1 Voronoi diagram

Voronoi diagram은 Delaunay 방법과는 달리 3차원 평면을 내각이  $180^\circ$ 이하인 다각형(convex polygon)들로 나타내는 방법이다. 데이터가 입력될 때마다 각각의 점들과 연결했을 때, 그 선을 수직으로 이등분하는 직선들로 다각형들을 형성하게 된다.

앞서 말한 Delaunay triangulation과 Voronoi diagram의 관계는 Voronoi diagram의 다각형 중심을 Delaunay triangulation의 외접원이 포함하고 있다. 그러므로 Delaunay triangulation은 지역을 모두 같은 모양(삼각형)으로 분할하기 때문에 검색 등과 같은 다른 알고리즘을 접합시키기 용이하다. 따라서 본 논문에서는 지형 위를 움직이는 동체의

시물레이션에서 요구되는 실시간 검색을 위해 Delaunay triangulation을 사용하였다.

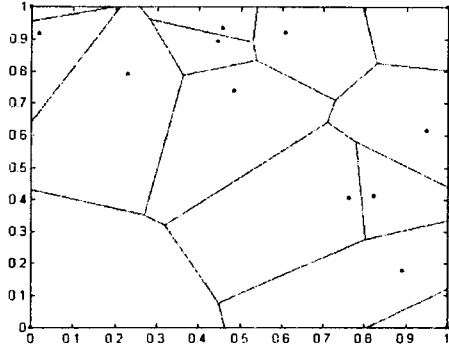


그림 4. Voronoi diagram

### 3. 데이터 검색기법

#### 3.1 T-search 기법

Delaunay triangulation으로 형성된 지형 데이터 베이스에서 원하는 삼각형(동체의 바퀴가 위치한 지역을 포함하는 삼각형)을 찾기 위해서 일반적으로 사용하는 검색방법이 T-search(Triangle search)이다. T-search는 Delaunay 방법으로 삼각형을 형성하는 과정에서 결정된 초기 삼각형에서부터 검색하고자 하는 삼각형의 방향으로 순차적으로 삼각형을 이동하면서 원하는 삼각형을 검색하는 알고리즘이다.

그림 5는 Delaunay triangulation으로 형성된 데이터 테이블에서 T-search 방법을 이용하여 검색했을 때의 검색 경로를 나타내는 것이다. Delaunay triangulation으로 형성된 지형에서 동체가 움직인 경로를 고려하여 보자. 그림 5에서 1번 삼각형이 삼각형을 형성하는 과정에서 결정된 초기 삼각형이고 표시가 된 삼각형은 동체가 맞닿는 면이다. 그림 5처럼 초기 삼각형에서 5번 삼각형으로 순차적으로 이동하여 검색하고자 하는 삼각형을 찾게 된다. 따라서 검색과정에서 초기 삼각형의

위치와 검색 대상이 되는 삼각형 사이에 거치게 되는 삼각형들의 개수가 검색에 소요되는 시간을 결정하게 된다. 규칙적으로 입력과 출력이 요구되는 실시간 시물레이션은 항상 고정된 타임 프레임에서 검색을 완료해야만 한다. 그러므로 실시간이 적용되어 반복되는 검색과정에 worst case가 고려되어야 한다. T-search의 worst case는 검색 대상이 되는 데이터 베이스의 크기로 결정된다. 그러므로 지형도에서 전체 삼각형의 개수가 실시간에 적용된 검색속도를 결정한다.

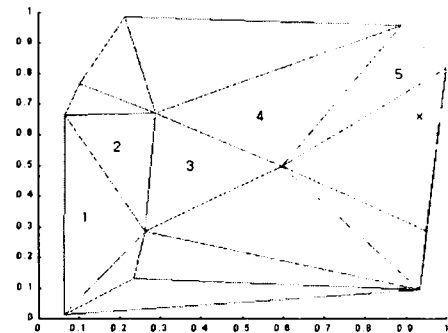


그림 5. Delaunay triangulation에서의 검색 경로

T-search 알고리즘은 다음과 같다:

#### T-search 알고리즘

1. 검색대상이 되는 점점이 초기 삼각형 내부에 존재하는지 체크한다.
2. 초기검색 삼각형에 접해있는 삼각형들과 검색대상이 되는 점점 사이의 거리를 계산한다.  
(※ 삼각형의 무게중심과 점점의 거리)
3. 계산된 거리가 가장 짧은 삼각형을 초기 삼각형으로 수정한다.
4. 1~ 3 과정을 반복한다.

#### 3.2 Level of Detail을 이용한 검색 기법

LOD (Level Of Detail) 기법은 실시간 컴퓨터 그래픽스 또는 가상현실에서 화면의 갱신을 일정하게 유지하기 위하여 주로 사용하는 기법이다. 3

차원의 동일한 형태를 다단계로 구성하여 각각의 상황에 적합한 형태를 보여주기 위한 방법이다. 다시 말해서 동일한 물체를 표현하고자 하더라도 거리 혹은 필요에 따라 세밀하게 표현하는 정도를 달리하여 한 순간에 보여주는 데이터의 양을 조절하는 것이다. 이러한 기법을 이용하여 컴퓨터 그래픽스 분야에서는 한 순간에 보여주는 데이터의 양을 적절히 조절하여, 보다 세밀히 표현할 부분을 포함하면서도 화면이 갱신되는 속도를 일정하게 유지한다.

LOD-search가 적용된 LOD기법은 컴퓨터 그래픽스의 방법과 비슷하지만 약간 다르다. 적용된 LOD기법은 일상생활에서도 흔히 사용하고 있던 방법이라고 할 수 있다. 예를 들어, 우리가 지도책에서 국내 농 혹은 구 단위의 어떤 지명을 찾을 때, 큰 지역의 지도로부터 점점 보다 상세한 정보가 있는 작은 지도로 찾게 된다(그림 6). 만약 지역의 주소를 모를 경우 전국 지도에서 찾아야만 한다.

위에서도 설명했듯이 T-search와 같이 순차적으로 검색하는 알고리즘에서는 데이터 베이스의 크기가 검색속도를 결정하게 된다. 예를 들어, 두 개의 다른 level로 된 지형 데이터 베이스를 고려해보자. 검색대상이 되는 데이터 베이스를 적당한 크기로 분할하여 하위 level map들을 형성하고, 하위 level map의 대표성을 지닌 삼각형들로 상위 level map을 형성하여 검색에 활용할 수 있다. 두 개의 다른 level을 두 번 검색하여도 실제적으로 T-search로 검색 대상이 되는 삼각형의 개수는 충분히 줄일 수 있다. 실시간이 적용된 T-search에서 worst case의 검색속도를 향상시킬 수 있게 된다. 예를 들어, 만약 150개의 삼각형으로 구성된 데이터 베이스를 각각 10개의 삼각형씩 분할한다면, 10개의 삼각형으로 구성된 15개이 하위 level map과 각 하위 level map을 대표하는 15개의 삼각형으로 구성된 상위 level map이 형성된다. 각 level map을 형성하기 전의 데이터 베이스를 대상

으로 검색하였을 때의 검색속도를 150이라고 한다면, level map에서의 검색속도는  $15+10=25$ 가 되어 속도가 향상될 것을 예상할 수 있다. 상위 level map의 크기와 하위 level map의 크기의 분할에 따라 검색속도에 영향을 미친다. 상위 level map의 크기와 하위 level map의 크기가 비슷하도록 분할하는 경우에 가장 빠른 검색속도를 기대할 수 있다는 것은 수학적으로 쉽게 증명된다. 그러나 지형 map의 지리적인 제약이 이론적인 최적조건을 허용하지 않는다. 만약 지형 데이터 베이스가 매우 많은 삼각형의 단위 표면들로 구성되어 있다면, 검색 속도를 가속시키기 위해 좀 더 많은 level을 만들면 된다.

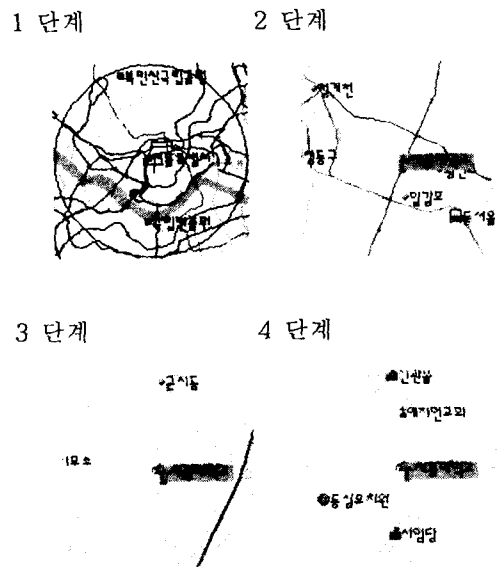


그림 6. 지도 검색 예

컴퓨터 그래픽에서 사용되는 LOD 기법과 이 방식의 가장 큰 차이점은 상위 level map 모양이 실제 데이터와 무관하게 형성되어도 된다는 점이다. 여기서 적용한 LOD 기법은 단지 검색 대상이 되는 map의 크기를 줄여 검색속도를 향상시키는 역할만 하게 되므로 실제 지형 데이터의 형태와 달라도 문제가 되지 않는다.

### 3.2 NN(Neural Network)을 이용한 검색 기법

Neural Network(NN) 기법은 경험적 데이터를 이용한 데이터 분석 기법으로 다량의 불규칙적인 데이터를 분석하고 예측하는 데 이용되는 방법이다. 앞에서 설명한 LOD 기법을 이용한 검색 기법에 NN 기법을 적용시켜 데이터의 검색 속도를 향상시키고자 한다.

시뮬레이션 대상이 되는 동적 시스템의 움직임이 과거의 경험에 대해 무관하지 않은 경우, 이 정보를 이용하여 검색에 소요되는 시간을 줄일 수 있다. 일상생활에서도 교통량에 따라 도로의 폭이나 신호의 길이 등을 정하는 것과 같이, 과거에 동적 시스템의 경로를 바탕으로 동적 시스템의 움직임을 예상하여 하위 level map의 크기를 다르게 결정할 수 있다. 다시 말해서, 검색 대상이 될 것으로 예상되는 하위 level map의 크기를 작게 하여 검색 속도를 보다 향상시키는 것이다. 이러한 정보들 외에 문화적 정보를 이용할 수도 있다. 예를 들어 강이나, 건물들의 위치 같은 문화적 정보를 이용하여 하위 level map의 크기를 결정할 수도 있다.

### 4. 수치시험

수치시험은 T-search를 포함한 새로운 LOD기법과 NN기법을 비교하였다. 수치시험을 위한 데이터는 다음의 그림처럼 각각 50개의 난수(random number)를 X와 Y, Z 좌표로 생성하고, Delaunay triangulation 방법을 이용하여 전체 map(그림 7)을 형성하였다. 이 map은 두개의 level로 구성하였다. 상위 level map은 전체 map의 상하를 횡단하는 직선을 이용하여 형성하였으며, 상위 level map의 각 삼각형과 공유하는 면을 가진 모든 삼각형들의 집합으로 그 사각형의 하위 level map을 구성하였다. 그림 8은 전체 map과 상위 Level map

을 겹쳐 그린 그림이다. 이렇게 map을 2단계로 구분하고 검색을 실행시켰을 경우에는, 먼저 상위 level map에서 검색 대상의 위치를 파악하고, 검색 결과와 대응되는 하위 level map에서 다시 한번 검색을 하게 된다.

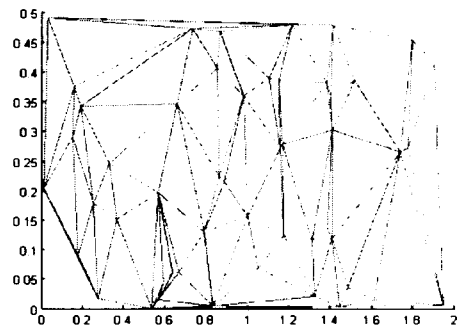


그림 7. 시험데이터의 전체 map

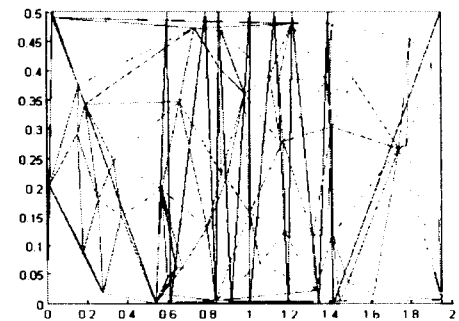


그림 8. 전체 map과 상위 level map

적용된 NN 기법은 과거의 경험이나 지형의 특징적 정보를 이용하여 각각의 하위 Level map의 크기에 차이를 두는 방법으로, 검색대상이 될 확률이 높은 지역에 대해서는 하위 Level map의 크기를 작게 하여 검색속도의 향상을 도모하는 방법이다. 과거의 경험이나 지형적 특징 등 하위 level map을 형성하는 기준이 정확할수록 검색속도에 안정성을 기할 수 있다. 그림 9, 그림 10은 ○로 표현한 동체의 운동 경로를 가정하고 그에 따라 형성되는 상위 level map을 나타내는 그림이다. 그림에서 볼 수 있듯이 NN 기법을 이용할 경우 경로를 포함하지 않는 많은 삼각형들을 가능한 하나의 큰

상위 level map에 묶고, 경로를 포함하는 나머지 삼각형들은 비슷한 개수씩 묶어서 상위 level map을 형성하였다.

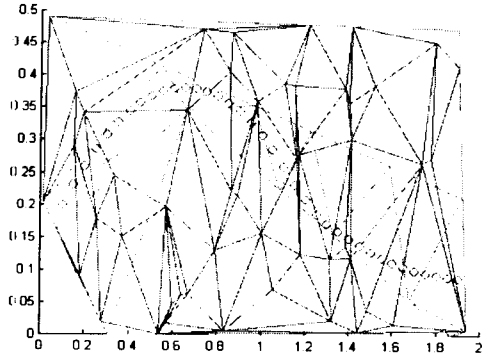


그림 9. 경로 1과 그에 따른 lod map

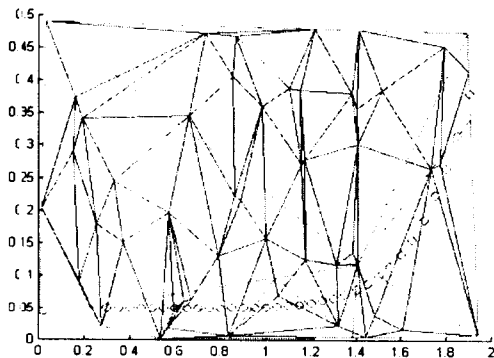


그림 10. 경로 2와 그에 따른 lod map

다음의 표 1은 세 개의 다른 조건에 결과를 비교한 것이다. 일반적인 T-search 방법과 LOD 기법을 이용한 경우들을 비교한 것으로 전체 map 내에서 임의로 선택한 1000개의 검색 point를 대상으로 검색을 수행하는데 걸리는 시간(초)을 측정한 것이다. 또한 경로 1의 시간은 동체가 그림 10에서 ○로 표현된 경로로 운동한다고 가정하는 경우에 검색에 소요되는 시간이고, 경로 2의 시간은 그림 10에서 ○로 표현된 경로로 운동한다고 가정하는 경우에 검색에 소요되는 시간으로 이는 각각 30회의 검색을 수행하는데 걸리는 시간이다.

그 결과, 수치시험에서 일반적인 T-search에 LOD기법과 NN기법을 적용한 새로운 검색 방법이 뛰어난 성능을 입증하였다. 새로운 방법들에서 검

색속도의 감소율은 17%에서 54% 이다. 만약 trial-and-error를 최소화 할 수 있다면, 검색속도를 추가적으로 감소시킬 수 있을 것이다.

표 1. 수치시험결과 (sec)

	T-search	LOD 기법	NN 기법
1000 points	0.93	0.77	0.82
경로 1	0.66	0.59	0.37
경로 2	0.54	0.43	0.25

#### 4. 결론

Delaunay triangulation 분할방법으로 형성된 map에서 LOD 기법을 도입한 새로운 검색방법을 시도하였으며, 수치시험 결과, 검색에 소요되는 시간이 일반적인 T-search 방법보다 빠르게 나타나 검색 시간을 단축시키는데 효과적인 방법임을 알 수 있었다. 특히, 시뮬레이션의 대상이 되는 동체의 경로를 미리 예측할 수 있는 경우, NN 기법을 도입하면 수치시험에서 확인한 바와 같이 보다 효과적으로 검색시간을 단축시킬 수 있었다. 만약 trial-and-error를 최소화 할 수 있다면, 검색속도를 추가적으로 감소시킬 수 있을 것이다.

#### 참고문헌

- [1] Hanselman Duane, Littlefield Bruce, "Mastering MATLAB5", Prentice-Hall, Inc., 1998.
- [2] Edwin Boender, "Reliable Delaunay-Based Mesh Generation and Mesh Improvement", Communications in Numerical Methods in Engineering, Wiley, Vol 10, pp.773-783, 1994.
- [3] Baker, Timothy J., "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation", Engineering with Computers,

- Springer-Verlag, Num 5, pp.161-175, 1989.
- [4] Kotropoulos C., Pitas I. and Maglara A.,  
"Voronoi Tessellation and Delauney  
Triangulation Using Euclidean Disk Growing in  
Z<sup>2</sup>", Int. Conf. on Acoust. Speech and Signal  
Processing, Mineapolis, 1993.
- [5] Besl P.J. and McKay N.D., "A Method for  
Registration of 3-D Shapes", IEEE Trans.  
PAMI, vol 14, no 2, pp.239-256 February 1992.
- [6] Huiskamp Wim, Jens Hans, and Janssen  
Hank, "An HLA Based Flight Simulation  
Architecture", Proceedings of AIAA Modeling  
and Simulation Conference, AIAA 2000-4401,  
Aug. 14-17, 2000.
- [7] Cohen-Or Daniel and Levononi Yishay,  
"Temporal continuity of levels of detail in  
Delaunay triangulated terrain", Proceedings of  
IEEE Visualization'96, 1996, pp.37-42.
- [8] Hoppe Hugues, "View-dependent refinement  
of progressive meshes", In SIGGRAPH 97 Proc.,  
August 1997, pp.189-198.
- [9] Lindstrom Peter, Koller David, Ribarsky  
William, Hodges Larry F., Faust Nick and  
Turner Gregory A., "Real-time, continuous level  
of detail rendering of height fields", Proceedings  
of SIGGRAPH'96, 1996, pp.109-118.
- [10] 조용준, 허준, 최인규, "Neural Connection 을  
이용한 데이터마이닝 신경망분석", 고려정보산업,  
1999.