

항공시뮬레이터에서 예측 기반의 동적 LOD 적용방안

김동진 · 임주호 · 김기일*

Prediction Based Dynamic Level of Detail in Flight Simulator

DongJin Kim · Juho Lim · Ki-Il Kim*

Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 52828, Korea

요 약

비행 시뮬레이터에서는 조종사에게 실제와 흡사한 가상현실을 제공하기 위해서는 원활하면서 현실감 있는 이미 지 처리를 위한 일정한 렌더링 속도가 요구된다. 하지만 고해상도 지형 데이터를 사용하는 경우, 개체의 증가로 인해 렌더링 속도가 감소하거나 속도의 변동이 큰 경우가 발생하는 경우가 발생하므로 기존의 연구들은 이를 해결하기 위하여 Level of Detail (LOD)를 적용한다. 하지만, LOD기법의 경우 정점 수가 늘어난 것을 감지한 이후 레벨의 변경을 적용하므로 화면의 변화가 감지되는 문제점이 발생한다. 이를 해결하기 위하여 본 논문에서는 예측 알고리즘을 통해 다음 시점에서 늘어날 렌더링 데이터에 대한 과부하를 예측하고 미리 LOD의 레벨을 조절하는 방법을 제안한다. 또한, 제안된 방법을 검증하기 위해서 예측 데이터와 실제 데이터와 비교하였고 그 결과 LOD 레벨을 조절하였을 때 정점수가 줄어들고 FPS 수치가 변화하는 모습을 보여주었다.

ABSTRACT

Fast rendering speed is one of key functions to provide realistic scenes in flight simulator. However, since flight simulator mostly operates with high volume terrain data, rendering speed is reduced and changed very rapidly when it handles file containing too much vertexs. So, previous schemes make use of Level of Details (LOD) scheme to prevent this problem. But, since LOD is applied after the large number of vertexs are detected, transition between scenes is not completely smooth. To solve this problem, in this paper, we propose a new dynamic LOD scheme which controls LOD level in advance through prediction of vertex overload. To verify the proposed scheme, we implement the proposed scheme in our flight simulation through OpenSceneGraph(OSG) and identify the reduced number of vertexs and enhanced Frame Per Second (FPS) by comparing real data with predicted one.

키워드 : LOD, 예측, 렌더링, FPS

Key word : Level of detail, Prediction, Rendering, Frame per second

Received 14 March 2016, Revised 18 March 2016, Accepted 01 April 2016

* Corresponding Author Ki-Il Kim (E-mail:kikim@gnu.ac.kr, Tel:+82-55-772-1373)

Department of Informatics, Engineering Research Institute, Gyeongsang National University, Jinju 52828, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2016.20.7.1363>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

최근 항공기를 통한 임무 및 이동이 늘어남에 따라 항공기 운행을 책임지는 조종사 양성의 중요성이 커지고 있다. 항공기 사고는 대부분 조종사의 과실에서 발생하게 되므로 안전한 비행을 위한 조종사의 훈련이 중요하지만 훈련 비용 및 안전성에 있어서 어려움을 겪고 있다. 이를 해결하기 위한 방법으로 비행 시뮬레이터가 사용되고 있으며 이를 통하여 현실 세계와 유사한 훈련 환경을 제공하여 조종사가 실제 비행 훈련과 같은 몰입감을 줄 수 있는 시각화에 대한 연구개발이 진행되고 있다. 현재 개발되고 있는 비행 시뮬레이션 소프트웨어는 크게 상용 시뮬레이션 소프트웨어와 오픈소스 시뮬레이션 소프트웨어가 있다. 대표적인 상용 시뮬레이션 소프트웨어로는 Microsoft사에서 개발한 Flight Simulator X[1]와 라미나르 리서치의 X-Plane[2]이 있으며, 오픈소스 시뮬레이션으로는 Flight Gear[3]가 있다. Microsoft Flight Simulator X는 게임을 위한 용도로 개발된 소프트웨어이다. 1980년에 최초로 Flight Simulator라는 이름으로 발표된 이후 2006년에 Flight Simulator X까지 개발되었다. 그 뒤 미국의 록히드 마틴사에서 Flight Simulator X를 개량하여 Prepar3D를 만들었다. 항법 및 동역학이 포함되어 실제 항공 시뮬레이션으로도 사용되고 있다.

위와 같은 항공 시뮬레이션 소프트웨어는 사용자에게 실제와 유사한 가상현실을 제공해야 하므로 원활하면서도 현실감 있는 환경 구현이 요구된다. 이러한 기능구현의 핵심은 렌더링 영상이 끊김이나 튀는 현상이 없이 부드럽게 표현되어야 한다. 하지만 항공 시뮬레이션과 같은 가상현실을 표현하는 렌더링 영상은 현실감이 높이기 위해 고해상도 데이터가 요구되고 있지만 해상도가 높아짐에 따라 처리해야 할 데이터는 점점 증가되기 때문에 렌더링 속도 저하 또는 끊김 현상이 발생하는 문제가 생긴다. 이러한 문제로 인해 사용자에게 현실적인 훈련 환경을 제공하는데 어려움이 있다.

이를 해결하기 위한 기존의 연구들은 Level of Detail (LOD)를 사용함으로써 처리하는 버텍스의 수를 조절한다. 하지만, LOD를 적용하는 경우 어떻게 언제 레벨을 선택하는지가 가장 중요한 요소이다. 만약 대용량의 정점들을 확인한 후 LOD를 적용하게 되면 화면의 급격한 변화가 발생하게 된다.

이러한 문제점을 해결하기 위해 본 논문에서는 예측 알고리즘을 통해 동적 LOD를 적용하는 방안을 제안하였다. 제안하는 방안은 정점수를 지수가중이동평균을 이용하여 예측하고 다음 시점에서 늘어날 렌더링 데이터에 대한 과부하를 예측함으로써 LOD 레벨을 조절한다. 이를 통하여 시뮬레이션 중 발생하는 과부하에 의한 렌더링 속도의 저하 및 변동 폭을 줄일 수 있다. 이러한 기법에 대한 성능 평가를 OpenSenceGraph(OSG)를 이용하여 직접 개발한 시뮬레이터에 적용하여 그 성능을 검증하였다.

본 논문은 다음과 같이 구성된다. 연구의 목적 및 배경을 설명한 서론에 이어 2장에서는 관련연구를 소개한다. 3장에서는 제안된 메커니즘을 설명하고 성능 평가 결과는 4장에서 설명한다. 마지막으로 5장에서는 결론 및 향후연구를 제시한다.

II. 관련연구

본절에서는 항공 시뮬레이션 소프트웨어와 렌더링 속도 향상을 위한 LOD기법을 활용한 연구를 소개한다.

먼저 대표적인 항공 시뮬레이션 소프트웨어로는 X-Plane과 Flight Gear가 있다. 라미나르 리서치에서 개발한 X-Plane은 실제 항공기와 아주 비슷한 수준의 움직임을 보이는 엔진을 사용한다. 그래픽 수준이 높으며 조작 가능한 항공기 모델과 군용으로 쓰이는 기체들을 제공하고 있다. Flight Gear는 오픈소스로 개발된 항공 시뮬레이션 소프트웨어로 비행 조종 훈련과 상용 분야 등에서 널리 사용되고 있으며 오픈 소스로 제공되기 때문에 개발자들이 개발하고자 하는 기능을 직접 구현 및 수정이 가능한 장점이 있다.

이러한 시뮬레이터에서 적용되고 있는 렌더링 속도 향상 방안 중에 하나인 LOD는 3D 그래픽스 프로그래밍으로 렌더링 대상이 되는 객체와 카메라 사이의 거리에 따라 정밀도를 다르게 하여 렌더링 속도를 향상시키는 기법이다[4]. LOD의 종류는 크게 정적 LOD와 동적 LOD로 나뉜다. 동적 LOD와 관련된 대표적 연구로는 대용량 지형 데이터의 렌더링 속도 향상을 위해 고정 쿼드트리를 이용한 동적 LOD를 구현한 메커니즘이 있다[5]. 해당 논문에서는 쿼드트리를 이용하여 기존의 동적 LOD 구현 알고리즘을 개선하고 쿼드트리의 문제

점인 T-Type fracture 현상을 제거하였다. 또한, 연구[6]에서는 LOD Level의 선택을 위해 오브젝트의 정점들을 측정하여 통계를 통한 동적 LOD를 위한 프레임 워크를 제안하였다. 건물을 예로 들면 LOD A단계에서는 건물의 블록을 표현하며, B단계에서 지붕과 외벽 등을 표현하고 C단계에서 굴뚝과 창문 등을 표현하는 방식이다. 이러한 방식으로 10단계의 동적 LOD를 제공하며 CityGML을 사용하여 모델을 통합하였다.

위의 관련 연구들은 대부분 LOD 레벨 판별 속도를 향상시키거나 정점 수를 측정하여 많은 단계의 LOD 레벨을 둬으로써 LOD의 성능 향상에 초점이 맞추어져 있다. 하지만 이러한 연구들은 불특정 구간에 발생하는 과부하에 대한 동작으로 렌더링 속도가 하락되고 난 후에 LOD레벨을 조절 하는 이유로 렌더링 속도의 변동 폭이 크고 과부하에 대한 대처가 늦은 단점이 있었다. 따라서 본 논문에서는 과부하에 대한 예측을 통해 렌더링 속도의 저하와 변동 폭을 줄이는 방안을 제안한다.

III. 제안 메커니즘

본 연구에서 제안하는 메커니즘은 항공기 시뮬레이션 중 특정 구간에서 발생하는 정점수의 증가에 따른 과부하를 막기 위해 정점의 증감 추세와 평균을 이용하여 과부하 지점을 예측하고 예측한 정보와 현재의 정보를 종합하여 LOD 레벨을 조절함으로써 렌더링 속도를 향상시키는 방법을 제안하고자 한다.

항공시뮬레이터에 사용하는 하나의 지형데이터에도 정점수가 차이나는 지점이 존재한다. 이런 지점에서 현재의 데이터와 최근의 추세를 비교하여 예측 여부 판단 후 필요에 따라 예측을 하며 렌더링 속도 향상을 위해 LOD 레벨을 조절하고 렌더링 데이터를 줄임으로써 렌더링 속도를 향상시키는 방법이 본 논문의 핵심이다.

보다 자세히 제안하는 방법은 그림 1의 5단계 과정과 그림 2의 4단계 과정으로 구성된다. 그림 1의 과정에서 예측 판단 후 만약 예측 데이터가 요구되는 경우 그림 2의 과정을 수행함으로써 레벨 조절을 위한 값을 계산하게 된다. 반면 예측 데이터가 필요 없을 시 그림 1의 과정만을 수행하며 자세한 설명은 다음과 같다.

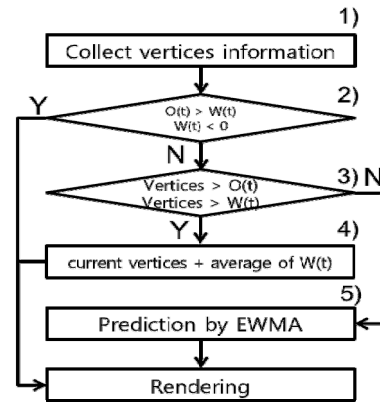


Fig. 1 Procedure for prediction

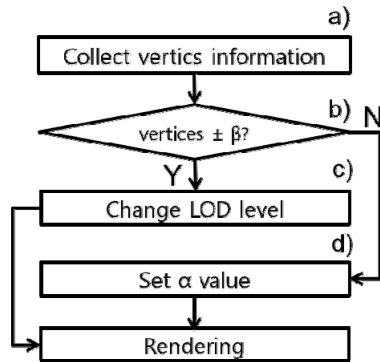


Fig. 2 Compute prediction data

3.1. 예측 여부 판단 과정

예측 기법으로는 이동평균(MA : Moving Average)을 사용하였다. 이동평균이란 추세의 변동을 알 수 있도록 구간을 옮겨 가면서 구하는 평균을 말한다. 본 논문에서는 가중이동평균과 지수이동평균의 개념을 혼용한 지수가중이동평균(EWMA : Exponentially Weighted Moving Average)을 이용하였다.

우선 예측 여부를 판단하기 위한 데이터로 현재, 과거, 최근의 정점 정보를 수집한다. 현재의 데이터는 현재 렌더링 되어 표현된 정점들을 의미하며 과거의 데이터 O(t)는 지금 시점부터 과거 시점까지의 데이터 평균을 의미한다. 최근의 데이터 W(t)는 현재의 시점부터 멀지않은 과거 시점까지의 가중치가 적용된 데이터로써 최근의 추세를 알기위해 사용된다.

단계 2)와 단계 3) 두 단계는 예측 여부를 판단하기 위한 과정에 해당한다. 첫 번째 판단 과정으로 단계 1)

에서 수집된 정점 데이터를 사용하여 해당 데이터들이 과거 평균 데이터 $O(t)$ 와 현재 데이터보다 적을 경우 과부하가 발생하지 않을 확률이 높다고 판단하며 다음 두 가지 방법으로 비교한다.

평균과 과거의 평균 $O(t)$ 와 최근 추세의 데이터인 $W(t)$ 를 비교하여 $O(t)$ 가 $W(t)$ 보다 크다면 최근의 추세가 평균에 미치지 못한다는 의미로 예측 여부 판단이 필요하지 않은 근거 중 하나로 사용한다. 그리고 최근의 추세를 의미하는 $W(t)$ 가 0보다 작다는 것은 그래프의 기울기가 음수임을 의미하며 이 역시 예측 여부 판단이 필요하지 않은 근거가 된다. 해당 두 가지의 조건을 모두 만족한다면 예측이 필요 없다고 판단, 추가 연산 없이 렌더링 과정을 진행하며 그림 2의 예측 데이터 적용 과정을 진행하지 않고 그림 1의 예측 여부 판단 과정을 다시 진행하게 된다.

예측 여부 판단 두 번째 과정으로 수집한 데이터 중 현재의 데이터가 예측 범위 밖일 경우에 예측이 아닌 다른 방법으로 과부하에 대한 처리를 위한 과정이다. 우선 EWMA 수식에 따르면 현재 시점의 예측할 데이터 $E(t)$ 는 $O(t)$ 와 $W(t)$ 의 범위를 벗어날 수 없다. EWMA는 과거의 평균과 최근의 추세를 지수 α 로 조절하여 두 데이터를 더해주는 방법인데 현재의 데이터가 두 데이터의 범위를 벗어나버리면 예측 데이터 값의 오차가 많이 발생한다.

따라서 두 데이터의 범위를 벗어나는 경우는 과정 4)를 수행한다. 최근의 기울기 추세를 현재 데이터에 더해줌으로써 다음 프레임의 과부하에 대한 예측을 대신해 적용한다. 이 과정도 예측을 필요로 하지 않기 때문에 그림 2의 과정을 진행하지 않는다.

단계 2)와 단계 3)의 과정에 모두 해당하지 않을 경우 예측이 필요하다고 판단하여 EWMA의 수식대로 예측 데이터를 연산하게 된다. 현재 시점의 예측 데이터를 다음 프레임으로 넘겨준 뒤 렌더링을 하는 것으로 Fig. 1의 과정을 끝내게 되며, 다음 프레임에서는 그림 1의 과정을 다시 수행하지 않고 넘겨받은 예측 데이터를 사용하여 그림 2의 과정을 진행하게 된다.

3.2. 예측 데이터 적용 과정

해당 과정은 그림 1의 단계 1)과 동일한 데이터를 수집하는 과정을 적용하며 예측이 맞았는지 틀렸는지를 판단하기 위한 과정이다. β 는 예측의 오차 범위 값으

로 사용되며 현재의 정점 데이터와 앞 프레임에서 받아들인 예측 데이터 값을 비교하여 그 차이가 오차 범위면 예측이 맞았다고 판단하여 단계 c)의 과정으로 이동하며 오차범위 밖이라면 지수 α 의 값을 조절하여 과거의 평균이나 최근의 추세 중 한곳에 가중치를 부여한다. 그림 2의 b)과정은 예측이 맞았다고 판단하여 LOD 레벨을 조절하는 과정이다. LOD 레벨 간격을 감소시키며 결과적으로 정점의 수를 감소시키는 과정이다. 이 때 LOD 레벨을 조절하는 정도는 앞 프레임에서 예측한 데이터인 $E(t)$ 와 과거의 평균인 $O(t)$ 의 차이만큼 조절하게 된다.

앞 프레임에서 예측한 정점 데이터와 현재 데이터와의 차이가 오차범위를 벗어날 경우 예측 데이터는 사용하지 않고 현재 프레임의 정점 수와 과거 평균인 $O(t)$ 차이를 비교하여 지수 α 를 조절한다. 이 때 현재의 정점 수가 $O(t)$ 보다 더 많다면 지수 α 를 크게 하여 과거의 평균보다 최근의 추세를 보다 잘 반영할 수 있게 한다. 반대로 현재의 정점 수가 $O(t)$ 보다 적다면 정점 수의 증가에 따른 부하가 없다고 판단하여 LOD 레벨을 조절하지 않는다.

IV. 성능 평가

제안된 방법으로 렌더링 속도의 향상을 위한 예측과 그 결과에 대한 검증을 위하여 OpenSceneGraph를 이용하여 자체 개발한 항공시뮬레이터[7]에 기능을 추가하였다. 시뮬레이터는 CPU Inter core I5-3570 3.40Ghz, RAM 4GB DDR, GPU NVIDIA Geforce GTX 570의 PC환경에서 테스트가 이루어 졌으며 OSG에서 사용한 2개의 지형 데이터 정보는 표 1과 같다.

Table. 1 Experimental data

Terrian data	Yecheon Airpot around 100Km and Republic of Korea
Data size	3.1GB and 13.7GB
LOD Level	Three level LOD
Fight information	Flight at 300Knot speed for 10 minites

예측 알고리즘에 대한 검증을 위해서 예측 데이터와 실제 데이터와 비교를 하였다. 그리고 지형의 정점의

수가 다를 때 비교를 위해 서로 크기가 다른 지형 데이터를 사용하였다.

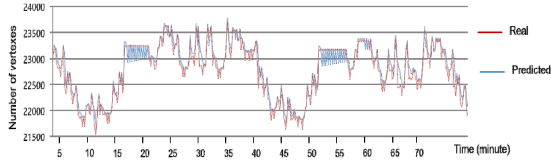


Fig. 3 Comparison of predicted data with actual data (Yecheon airport around 100km)

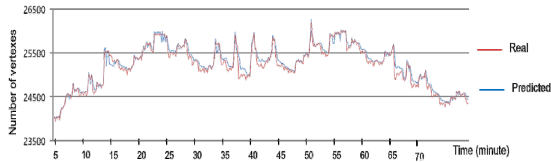


Fig. 4 Comparison of predicted data with actual data (Republic of Korea)

그림 3과 그림 4의 그래프를 보면 예측 데이터와 실제 데이터가 겹치는 구간이 예측이 맞은 구간이고 비교적 차이가 큰 부분이 예측의 오차 범위가 발생한 부분이다. 그림 3을 보면 예측 데이터와 실제 데이터의 차이가 큰 부분이 일정구간 반복되는 곳이 있다. 이곳은 예측 여부 판단의 단계 3)에 해당하는 구간으로 예측 범위를 벗어나는 데이터가 증가함에 따라 기울기가 급격하게 늘어나다가 갑자기 일정하게 유지가 됨으로써 예측에 오차가 비교적 크게 나타난 지점을 의미한다.

그림 3은 예천 공항 주변 100Km의 지형 데이터를 실험한 결과로 변동 폭이 그림 4의 대한민국 전 지역보다 더 크음을 알 수 있다. 그 이유는 높은 고도의 테스트 환경에서 예천 공항의 경우 지형 데이터가 남한 전 지역 지형 데이터에 비해 지형의 상대적으로 작기 때문에 시야 범위(View Frustum)에 지형 데이터를 모두 채우지 못하기 때문이다.

그림 5와 그림 6은 예측한 결과가 오차 범위 내일 때 예측 데이터를 사용하여 LOD 레벨을 조절하였을 때 줄어든 정점 수이며, 그림 7과 그림 8은 정점이 줄어들어 변화하는 초당 프레임 수(FPS : Frame Per Second)이다. 또한, 그림 5와 그림 6을 비교해보면 예측 데이터들 간의 비교에서와 같이 변동 폭의 차이가 있음을 알 수 있다.

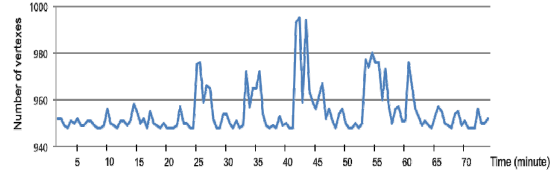


Fig. 5 Number of reduced vertices (Yecheon airport around 100km)

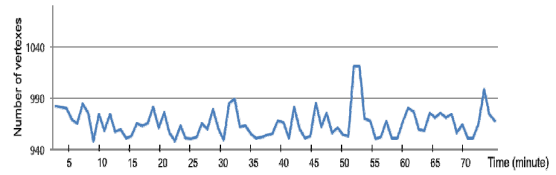


Fig. 6 Number of reduced vertices (Republic of Korea)

이는 정점의 증가와 감소가 예천 공항 주변이 많은 편이고 대한민국 전 지역이 비교적 덜하다는 결과로 볼 수 있다. 정점의 감소 수치가 일정 수 이하로 내려가지 않는 그래프의 특성을 가지는 것은 정점 수가 감소할 경우 LOD 레벨을 조절을 하지 않기 때문이며 일정 수준 이상의 증가폭을 예측 할 경우만 LOD 레벨을 조절하기 때문이다.

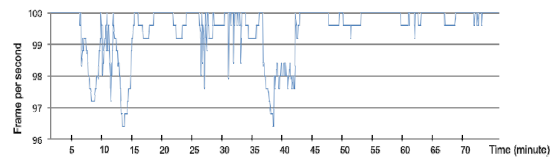


Fig. 7 Measured FPS (Yecheon airport around 100km)

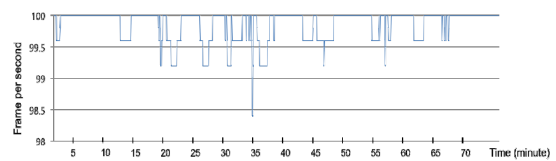


Fig. 8 Measured FPS (Republic of Korea)

마지막 결과인 FPS의 변화도 역시 앞서 보았던 두 가지 결과와 같이 그림 7과 그림 8에서 변동 폭이 차이는 것을 알 수 있다. 대개의 경우 정점의 증가와 감소는 FPS와는 반대의 패턴 즉 정점수가 늘어남에 따라 연산이 늘어나고 1초당 렌더링 하는 프레임 수인 FPS가

하락되는 결과를 가져온다. 예천 공항 주변의 경우 최고 3.5 프레임, 대한민국 전 지역의 경우 1.5 프레임의 변동 폭을 보여주고 있으며 변동 폭이 큰 구간의 발생 빈도는 잘못된 예측이 발생하는 빈도와 일치함을 알 수 있다. 그 외의 예측이 맞았을 경우 FPS 변동 폭이 좁게 나타남을 알 수 있다.

V. 결론

본 논문에서는 항공 시뮬레이터에서 렌더링 속도 향상 및 급격한 변화를 방지하기 위해 EWMA 예측 알고리즘을 통해 사전에 정점 수의 증가와 같은 부하를 예측하고 LOD 레벨을 조절하는 방법을 제안하였다. 이를 통하여 부하가 발생하는 지역에서 정점의 수를 감소시킴으로써 일정 수준의 렌더링 속도 향상과 렌더링 속도의 변화 폭을 줄일 수 있었다. 그러나 급격한 LOD 단계 조절이 발생할 경우 인접한 노드와의 LOD 레벨차이로 인한 균열 현상이 발생하는 문제점이 있었다. 향후 해당 균열 현상과 일정하지 않는 지형 이외의 오브젝트 추가에 따른 과부하 현상에 대처할 수 있는 방안에 대하여 연구할 예정이다.

REFERENCES

- [1] Microsoft. Flight simulator [Internet]. Available: <https://www.microsoft.com/Products/Games/FSInsider/product/Pages/default.aspx>
- [2] Laminar Research. X-plane [Internet]. Available: <http://www.x-plane.com/desktop/home/>
- [3] FlightGear Project. FlightGear Flight Simulator [Internet]. Available: <http://www.flightgear.org/>
- [4] D. Luebke, B. Watson, J. Cohen, M. Reddy and A. Varshney, *Level of detail for 3D graphics*, San Francisco, Elsevier Science Inc., 2002.
- [5] W. Ying, W. Yanjie, "Implementation of a fast simulation algorithm for terrain based on dynamic LOD," *International Conference on Electronic Measurement & Instruments (ICEMI)*, pp.189-193, Aug. 2011.
- [6] F. Biljecki, H. Ledoux, J. Stoter, J. Zhao, "Formalisation of the level of detail in 3D city modelling," *International Journal of Computer Environment and Urban Systems*, pp. 1-15, Nov. 2014.
- [7] C. Lee, S. Kang, S. Kang, K. Kim and K. Kim, "Development of key functions for flight simulator," *International Journal of Control and Automation*, vol.9, no.1, pp.347-358, Jan. 2016.



김동진(DongJin Kim)

2014.02: 경상대학교 학사
2016.02: 경상대학교 공학석사
2016.03 ~ 현재: 아피아 엔지니어링
※관심분야: 항공기 시뮬레이터, 항공기 소프트웨어



임주호(JuHo Lim)

2013.02: 경상대학교 학사
2015.03 ~ 현재: 경상대학교 석사과정
※관심분야: 항공기 시뮬레이터, 항공기 소프트웨어



김기일(Ki-II Kim)

2002.02: 충남대학교 이학석사
2005.02: 충남대학교 이학박사
2006.03 ~ 현재: 경상대학교 정보과학과 부교수
※관심분야: 센서네트워크, 이동네트워크, 항공기소프트웨어