

논문 2014-09-08

항공시뮬레이터에서 기상 효과를 위한 적응적 컬링기법 (Adaptive Culling Mechanism for Weather Phenomena Effect in Flight Simulator)

차 영 준, 김 종 범, 김 기 일*

(YoungJun Cha, JongBum Kim, Ki-Il Kim)

Abstract : Temporal disappearance of weather phenomena effect is frequently observed in flight simulator when large volume of terrain data are processed. This problem was solved by employing culling scheme at static ratio in the existing scheme. However, since this approach causes the irregular rendering speed according to volume of data, it is necessary to develop a new culling scheme to maintain steady rendering speed by adjusting the culling ratio dynamically. In this paper, we propose a new culling scheme to make use of distance of the visibility to determine culling ratio depending on volume of terrain data. The experimental results show that rendering speed is preserved by the proposed scheme without affecting the visibility at rendering the scene and weather phenomena effect together.

Keywords : Culling, Flight simulator

I. 서론

비행 시뮬레이션 소프트웨어는 탑승자에게 실제와 같은 가상 비행 상황을 제공함으로써 항공기 조종 훈련 및 탑승 체험에 사용된다. 이러한 비행 시뮬레이션 소프트웨어에서 기상 효과는 파일럿이 실제 비행 시 직면할 수 있는 다양한 기상의 변화를 사전에 가상으로 경험할 수 있게 도와줌으로써 사실적인 비행 훈련에 필수적이다. 이러한 요구를 만족시키기 위하여 다양한 그래픽 처리 플랫폼에서는 기상효과 구현을 위한 모듈을 기본적으로 포함하고 있다. 예를 들어, OSG(Open Scene Graph)[1-3]를 이용하여 기상효과를 구현하고자 하는 경우에는

*Corresponding Author (kikim@gnu.ac.kr)

Received: 30 Oct. 2013, Revised: 29 Nov. 2013.

Accepted: 4 Dec. 2013.

Y.J Cha, J.B. Kim, K.I. Kim: Gyeongsang National University, Engineering Research Institute

※ 본 논문은 교육부와 한국연구재단의 지역혁신인력양성사업(No. 2011-02-기-01-010)과 미래창조과학부 및 정보통신산업진흥원의 고용계약형 SW석사과정 지원사업의 연구결과임

Customized module[4] 또는 Predefined module[5]을 이용하는 방법이 있다.

이러한 기상 효과의 구현은 가상 시뮬레이션 소프트웨어의 현실감을 향상시켜 주는 반면 추가적인 데이터 처리로 인한 렌더링 속도 저하를 야기하며 렌더링 데이터 양의 증가에 따라 실제 구현된 기상 효과가 사라지는 문제점이 발생한다.

앞에서 언급된 문제를 해결하기 위해서는 대용량 데이터와 상관없이 렌더링 속도를 보장하기 위한 방안이 적용되어야 하며 대표적인 방식으로는 LOD(Level of Detail)와 컬링이 있다. 전자인 LOD는 맵 생성 과정에서 거리에 따라 지형 데이터를 따로 생성해야 하는 방식으로 시뮬레이션 실행 중 동적 조정이 불가능하다. 반면, 후자의 방법은 시뮬레이션 실행 중 카메라의 시야 범위 비율을 조절함으로써 동적인 렌더링 속도를 향상시킬 수 있기 때문에 이를 이용해 기상 효과 추가로 인한 렌더링 속도 저하 방지가 가능하다. 이러한 이유로 인하여 LOD보다는 컬링을 이용하여 렌더링 속도를 보장하는 것이 다양한 기상효과를 구현하여야 하는 항공시뮬레이터에 적합하다. 하지만, 현재 제안 항공시뮬레이터를 위하여 적합한 시각 절두체 컬링 기법의 경우 컬링 비율이 고정되어 있어 다양한 상황에

다른 일정한 속도 보장이 어려운 문제점이 있다.

이러한 문제점을 해결하기 위하여 본 논문에서는 컬링기법을 적응적으로 적용하는 방안을 제안하였다. 제안된 방안은 실세계와 같이 비나 눈이 내리는 상황에 따라 가시거리를 동적으로 변경한다. 이를 위하여 설정된 가시거리에 따라 컬링의 near-far 평면의 비율을 조정하여 far평면이 가시거리 내의 객체들을 컬링하지 않는 최대 비율과 각 비율에 따른 렌더링 속도를 측정한다. 측정된 결과를 이용하여 기상 효과 추가 시 눈, 비의 밀도에 따른 저하된 렌더링 속도를 보충시킬 수 있도록 컬링 비율을 적용하여 일정한 렌더링 속도를 보장한다. 이를 통하여 기상효과 추가 시 급격한 속도 저하를 방지할 수 있으며 다양한 시나리오를 통하여 이를 검증하였다.

본 논문은 다음과 같이 구성된다. 본 연구배경을 1장에서 설명하였고 2장에서는 기상 효과 구현 및 컬링에 관련된 기존 연구를 설명한다. 3장에서는 동적 컬링 적용 방안을 설명하고 4장에서는 구현된 컬링 메커니즘을 검증하였다. 마지막 5장에서는 결론 및 향후 연구 내용을 제시한다.

II. 관련연구

시뮬레이터상의 기상효과에 관련된 연구는 크게 실제 구현에 관한 연구와 기상효과 추가에 따른 효과를 처리하기 위한 방안으로 크게 나누어진다.

첫 번째로 기상효과 구현을 위한 대표적인 연구로는 OpenGL을 이용한 방법[6, 7]과 Direct3D를 이용한 방법[8]이 있다. 참고문헌 [6, 8]은 눈이나 비의 파티클을 텍스처 매핑으로 처리하고 기상 효과의 움직임은 파티클 시스템을 이용하는 방식을 사용한다. 한편, 참고문헌 [7]에서는 OpenGL의 Display list와 Bulletin board 기술을 사용하는 방식을 제안하였으나 지형과 함께 기상 효과 구현 시 소프트웨어의 프레임 레이트가 매우 낮게 측정되는 문제점이 발생하게 된다.

두 번째 기상효과 추가에 따른 렌더링 속도 보장을 위한 연구로는 LOD와 컬링을 적용하는 방안이 제안되고 있다. LOD는 카메라와 객체의 거리에 따라 가까운 곳은 고해상도로, 먼 곳은 저해상도로 표현하는 기법이고 컬링은 카메라의 시야 범위를 지정하여 범위 밖의 객체들을 잘라내어 렌더링되는 데이터를 줄이는 기법을 말한다. 전자의 경우, 맵 생성 과정에서 거리에 따라 지형 데이터를 따로

생성해야 하므로, 시뮬레이션 실행 중 동적 조정이 불가능하다. 반면, 후자의 방법은 시뮬레이션 실행 중 카메라의 시야 범위 비율을 조절하여 동적인 렌더링 속도 향상이 가능하므로 이를 이용해 기상 효과 추가로 인한 렌더링 속도 저하를 해결하는데 보다 적합하다.

컬링의 대표적인 예로는 후면 컬링 (Backface Culling), 시각 절두체 컬링 (View Frustum Culling), 차폐 컬링 (Occlusion Culling) 이 있다. 이러한 기법들 중 렌더링 속도 향상을 위한 연구는 크게 차폐 컬링[9]과 시각 절두체 컬링을 이용한다. 먼저 차폐 컬링을 이용한 연구로는 교차점 감지를 통한 효율적 차폐 컬링 알고리즘[10], 복잡한 기계 장치의 3D 구현 시 차폐 컬링 전략[11], 숲 속에서 나무들의 가림 현상을 이용한 차폐 컬링 적용 방안 [12] 등이 제안되었다. 하지만 차폐 컬링을 이용한 방법은 위의 연구들처럼 장면 속의 객체들 간의 가림 현상이 심한 경우에만 렌더링 속도 향상 효과가 나타나므로 카메라의 시선이 위에서 아래로 내려다 보는 비행 시뮬레이션에서는 주로 시각 절두체 컬링 기법을 이용한다.

절두체 컬링 기법을 통한 대표적 렌더링 속도 향상방안으로는 2D 변환을 이용한 방식이 제안되었다. [13]. 또한 복잡한 도시 장면의 고속 렌더링을 위한 기여도 컬링 기법[14], 시각 절두체 선별을 통한 고해상도 타일 가시화 성능 향상[15]과 같이 시각 절두체 컬링을 통한 렌더링 속도 향상 연구가 제안되었다.

III. 항공시뮬레이터를 위한 동적컬링 방안

앞서 살펴본 바와 같이 시각 절두체 컬링을 통한 렌더링 속도 향상에 관한 다수의 연구가 많이 이루어지고 있지만 대부분의 연구는 시뮬레이션의 전반적인 속도 향상에만 중점을 두고 있어 아직까지 시뮬레이션 작동 중 기상효과등과 같은 처리 데이터의 추가와 같이 동적인 상황에 따른 렌더링 속도 보장을 위한 연구는 제안되지 않았다. 따라서, 컬링 비율을 일정하게 유지하는 방식을 적용하고 있다. 하지만, 항공시뮬레이터의 경우 다수의 기상 효과와 같은 동적인 객체가 자주 추가되기 때문에 이러한 상황에서도 일정한 렌더링 속도 보장을 위한 방안이 요구된다. 이를 위하여 본 절에서는 기상 변화에 따른 컬링 적용 방안을 설명한다. 제안된 방법은 그림 1과 같이 크게 네 부분의 과정으로 나누

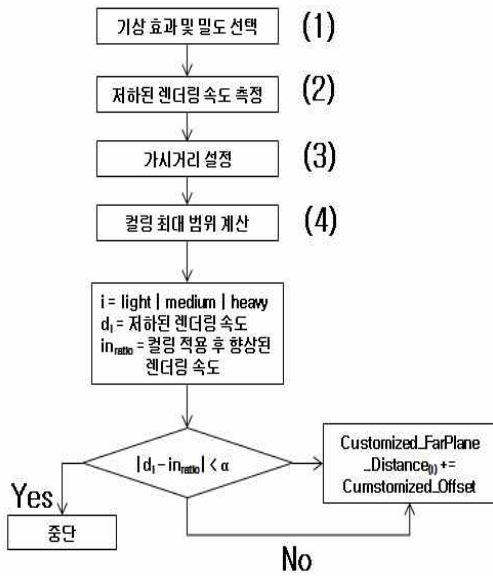


그림 1. 기상변화에 따른 동적 컬링 적용 방안 구현 과정

Fig. 1 Procedure for adaptive culling scheme

어진다. 각 과정에 대한 자세한 설명은 다음과 같다.

1) 일단 기상 효과 추가 시 저하되는 렌더링 속도를 측정한다. 현재 개발된 시뮬레이터 상에서 기상 효과(눈, 비)의 경우 밀도에 따라 light, medium, heavy로 구분하기 때문에 각 단계에 따른 렌더링 속도를 측정한다.

2) 기상 효과 구현 시 연무로 인하여 좁혀지는 시야 범위와 같이 가시거리는 동적으로 변하므로 변경된 가시거리를 설정하기 위해서는 실제계의 관측 자료를 이용한다. 하지만 실제계에서 가시거리는 기상에 따라 그 편차가 심하므로 기준이 되는 가시거리 값을 개발자가 현실감 및 렌더링 속도를 고려하여 임의로 설정하고 기상 효과의 각 단계에 따라 그 값을 일정한 비율로 축소하여 적용한다.

3) 1단계에서 측정된 렌더링 속도를 보정하기 위해서 3단계에서는 컬링 범위에 대한 설정이 필요하다. 이 컬링 범위는 2단계에서 설정된 가시거리보다 먼 거리에 있는 객체들을 제거할 수 있도록 설정되어야 한다. 이를 위하여 컬링범위의 시작점을 가시거리와 동일하게 설정한다. 이 시작점의 거리는

다음 식 (1)과 같이 구해진다. (식 1)에서 Default_NearPlane_Distance는 카메라 시야에서 가장 가까운 뷰 평면까지의 거리를 나타내며 Customized_FarPlane_Distance는 2단계에서 설정된 가시거리로 정의된다.

$$Near-Far\ Ratio = \frac{Default_NearPlane_Distance}{Customized_FarPlane_Distance} \quad (1)$$

4) 3단계에서 구해진 비율의 경우 최대 컬링 범위를 의미하기 때문에 이 값이 1단계에서 측정된 렌더링 감소량보다 클 경우 프레임 속도를 일정하게 유지하기 위한 방안이 필요하다. 이를 위하여 (식 1)의 Customized_FarPlane_Distance를 증가시킴으로써 새로운 컬링 비율을 계산한다. 이후 새로운 컬링 비율에 따른 렌더링 속도 향상을 측정하고 이 값을 1단계에서 측정된 감소 속도와 비교한다. 이러한 작업은 증가와 감소 비율의 차가 적절한 범위 내로 계산된 경우 중단된다.

IV. 성능 평가

제안된 방법의 일정한 렌더링 속도 보장에 대한 검증은 위하여 본 논문에서는 윈도우 플랫폼에서 OSG를 이용하여 기능을 구현한다. 개발 하드웨어는 Intel Core i5-3570 3.40Ghz 프로세서와 4GB DDR3 RAM, NVIDIA GeForce GTX 550 Ti 그래픽 카드로 구성하였다. 기상효과의 구현은 OSG의 PrecipitationEffect 클래스를 사용하였다. 이 클래스를 통하여 사용자는 미리 정의되어 있는 snow(), rain() 함수를 통하여 쉽게 눈과 비 기상효과를 구현할 수 있다. 또한, 기상효과의 밀도, 색상 설정이 가능하며 setParticleSystem() 함수를 통하여 파티클이 아래 방향으로 내리는 속도를 설정 가능하다.

일단 기능의 검증에 위한 다수의 정점을 포함하는 환경에서 눈 / 비 효과를 구현한 결과를 비교한다. 이를 위해 화면에 출력되는 지형 데이터의 양이 다른 상황에서 실험을 하였다.

그림 2는 화면에 출력되는 지형 데이터의 정점의 수가 76,238인 지점으로 기상효과가 정상적으로 렌더링 되고 있다. 하지만 그림 3과 같이 화면에 출력되는 지형 데이터의 정점이 82,144로 증가하게 되면 기상효과와 입자 크기가 점진적으로 작아지며 하단 영역에는 렌더링이 되지 않고 사라지는 모습

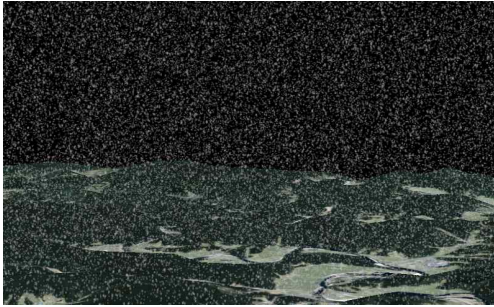


그림 2. 저용량 지형 데이터에 따른
Predefined방법 기상 효과

Fig. 2 Predefined method in high-resolution terrain data

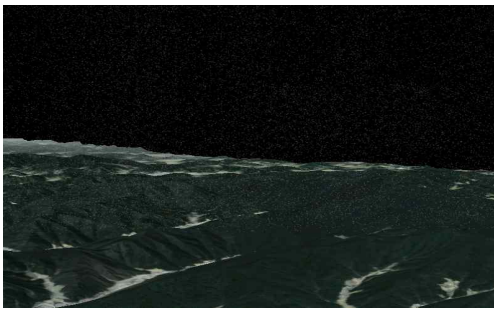


그림 3. 대용량 지형 데이터에 따른
Predefined방법 기상 효과

Fig. 3 Predefined method in low-resolution terrain data



그림 4. 컬링 기법을 적용한 기상효과

Fig. 4 Weather effect through culling scheme

을 보인다. 이는 지형 데이터의 정점의 수가 증가함에 따라 렌더링 데이터 양이 증가하여 기상효과가 사라지는 문제점을 보여준다. 이러한 문제를 해결하기 위해 본 논문에서 제안하는 컬링 기법을 적용

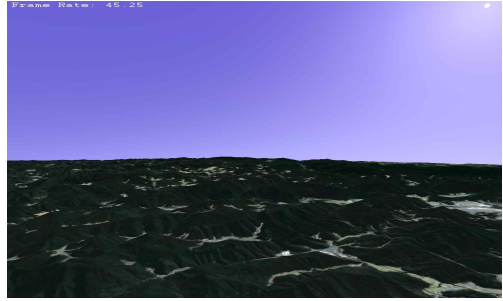


그림 5. 기상이 맑은 날 렌더링 장면

Fig. 5 Scene for rendering shiny day

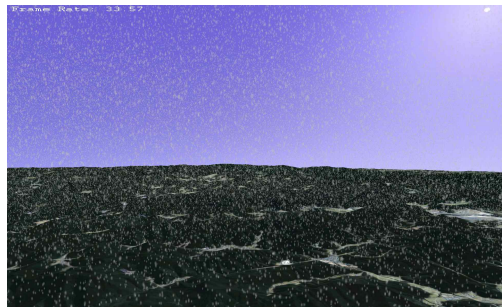


그림 6. 기상 효과가 추가된 렌더링 장면

Fig. 6 Scene for rendering weather condition

한 모습을 그림 4에서 보여준다. 그림 3과 동일한 위치이지만 컬링 기법을 적용하여 정점의 수를 줄여 화면에 지형 데이터의 양이 많이 출력되는 환경에서도 기상효과가 정상적으로 렌더링 된다.

이러한 컬링기법의 적용에 따른 현실감에 대한 비교는 그림 5, 그림 6과 같이 같은 지형에서 기상 효과의 유무에 따른 비교분석을 수행하였다. 먼지가시거리의 동적인 변화에도 렌더링 장면에서 부자연스러움이 없음을 보여주기 위해 동일한 좌표 값과 FOV(Field Of View)값을 설정하였다. 그림 5와 그림 6의 지형을 비교해 보면 후자의 경우 지형의 끝단을 컬링 하였지만 산의 형세나 지평선 끝단의 높이에 있어 큰 변화를 느낄 수 없음을 보여준다. 게다가 지형에 끝단 처리를 위한 fog효과와 haze처리를 통하여 시뮬레이션 사용자가 느끼는 가시거리의 변화는 전혀 없다.

기존의 정적 기법과 제안된 방법을 검증하기 위하여 렌더링 속도 저하를 유발하도록 정점의 개수를 변화시키는 상황에서 기상효과 추가 시 제안된 방법을 적용한 경우 렌더링 속도를 측정하였다.

표 1은 위와 같이 자연스러운 가시거리 설정을

표 1. 정점 양에 따른 기상 효과 추가 시 렌더링 속도 비교

Table 1. Comparison of rendering speed
단위: FPS

정점	기상 효과 밀도	고정 비율 컬링	제안 메커니즘
100,000 (1)	Default	60	60
	light	49	53
	medium	45	50
	heavy	43	50
120,000 (2)	Default	47	47
	light	40	45
	medium	38	47
	heavy	36	46
140,000 (3)	Default	44	44
	light	39	44
	medium	36	44
	heavy	35	44

바탕으로 고정 비율 컬링을 적용한 경우와 본 논문에서 제안한 메커니즘을 적용한 경우로 나누어 렌더링 속도를 측정한 결과이다. 측정은 다양한 시뮬레이션 상황에서도 일정한 렌더링 속도의 보장을 보여주기 위해 정점 수가 각각 (1) 100,000, (2) 120,000, (3) 140,000인 지점에서 테스트를 하였다.

표 1을 보면 (3)번째 경우에서 고정 비율 컬링을 적용한 경우 렌더링 속도가 44~35프레임으로 9프레임의 차이를 보이고 있다. 반면 제안 메커니즘을 적용한 경우 모든 단계에서 44프레임으로 일정한 렌더링 속도를 유지함을 보여준다. 또한 (2)의 경우에도 1~2프레임 정도 편차는 있지만 렌더링 속도를 일정하게 보장한다. 하지만 (1)의 경우 고정 비율 컬링에 비해 전반적인 렌더링 속도는 향상되었지만 일정한 렌더링 속도를 보장하지는 못하였다.

이러한 현상은 기상효과 추가로 인해 증가되는 렌더링 데이터양과 컬링으로 인해 감소되는 렌더링 데이터양의 차이로 인해 발생한다. 현재 설정된 컬링 비율은 정점의 수가 140,000인 지점에서 테스트를 통해 그 값을 조정하였다. 따라서 정점의 수가 140,000인 지점에서 기상 효과 추가로 인한 데이터 증가량과 컬링되는 정점의 양이 동일해지는 지점을 컬링 적용의 시작점으로 설정하였기 때문에 (1)처럼 전체 정점 수가 감소된 경우 그 만큼 컬링되는 정점의 수도 감소하므로 렌더링 속도 향상 효과 또한

줄어든다. 이러한 문제점을 해결하기 위해 정점이 적은 100,000인 지점을 기준으로 컬링 비율을 측정하여 적용해보았지만 이러한 경우 140,000인 지점에서 너무 많은 정점을 컬링 함으로써 기본 시뮬레이션 상황(기상이 맑은 날)의 렌더링 속도보다 높은 렌더링 속도를 보이는 오버프레임 현상이 발생하였다.

V. 결 론

본 논문에서는 항공시뮬레이터 개발 시 요구되는 기상효과를 구현 시 렌더링 데이터 양의 증가에 따라 기상 효과가 사라지는 문제점을 해결하기 위한 방안을 제안하였다. 제안된 방안은 추가되는 기상 효과로 인한 성능 저하를 방지하고 일정한 렌더링 속도를 보장하기 위해 가시거리에 따라 컬링 비율을 동적으로 적용함으로써 기상효과 구현으로 인해 추가적으로 증가되는 렌더링 데이터 양에 관계없이 일정한 렌더링 속도를 보장함을 실험을 통하여 증명하였다.

본 연구와 관련하여 프레임 락을 이용하여 오버프레임 현상을 막는 방안과 정점의 양에 따라 동적으로 컬링 비율을 적용하는 방안이 추가적으로 연구될 것이다.

References

- [1] R. Wang, X. Qian, OpenSceneGraph 3 Beginner's Guide, Packt Publishing Ltd., 2012.
- [2] R. Wang, X. Qian, OpenSceneGraph 3 Cookbook, Packt Publishing Ltd., 2010.
- [3] OpenSceneGraph, <http://www.openscenegraph.org/>
- [4] Y. Zhang, L. Zou, J. Liu, "Simulation of snow effects in visual simulation of virtual campus based on OSG," Proceedings of the International Conference on Multimedia Technology, pp.3658-3662, 2011.
- [5] L. Li, W. Wan, X. Li, Z. Wang, "Weather phenomenon simulations in 3D virtual scenes based on OSG particle system," Proceedings of IET International Communication Conference on Wireless Mobile and Computing, pp.254-257, 2011.
- [6] X. Huang, N. Li, D. Zhao, R. Xu, "Design and Application of General Data Structure for

- Particle System,” Proceedings of International Conference on Computer Science and Electronics Engineering, Vol. 1, pp.489-492, 2012.
- [7] N. Fan, N. Zhang, “Real-time Simulation of Rain and Snow in Virtual Environment,” Proceedings of International Conference on Industrial Control and Electronics Engineering, pp.29-32, 2012.
- [8] Y. Tan, X. Zhang, C. Wang, Q. Zhao, “Real-Time Snowing Simulation Based on Particle Systems,” Proceedings of International Workshop on Education Technology and Computer Science, Vol. 3, pp.7-11, 2009.
- [9] T. Akernine-Moller, E Haines, “Real-time rendering 2nd edition,” 2002.
- [10] J. Yi, “An efficient occlusion culling algorithm of line segment intersection based on large-scale scene,” Proceeding of Information Science and Digital Content Technology, pp. 128-130, 2012.
- [11] I. Mansa, A. Amundarain, A. Garcia-Alonso, L. Matey, “Occlusion culling for stereoscopic display of complex mechanical sets,” Proceedings of International Conference on Information Visualization, 2007.
- [12] Q. Xiong, X.Y. Huang, “Speed Tree-Based Forest Simulation System,” Proceedings of IEEE International Conference on In Electrical and Control Engineering, pp.3033-3036, 2010.
- [13] C. Wang, H. Xu, H. Zhang, D. Han, “A fast 2D frustum culling approach,” Proceeding of Computer Engineering and Technology, pp.414-417, 2010.
- [14] B.J. Lee, J.S. Park, “A Contribution Culling Method for Fast Rendering of complex Urban Scenes,” Journal of Korea Game Society, Vol. 7, No. 1, pp.43-52, 2007(in Korean).
- [15] H.S. Kim, K.W. Kim, J.Y. CHO, “Enhanced High Resolution Tiled Display through View Frustum Culling,” Proceedings of Conference On The Korean Society for Aeronautical and Space Sciences, pp.533-536, 2005(in korean).

저 자 소 개

차영준



2013년 경상대학교 정보과학과 학사.

현재, 경상대학교 정보과학과 석사과정.

관심분야: 항공 시뮬레이션 소프트웨어.

Email: atheaven@nate.com

김종범



2012년 경상대학교 정보과학과 학사.

현재, 경상대학교 정보과학과 석사과정.

관심분야: 항공 시뮬레이션 소프트웨어.

Email: karma15513@nate.com

김기일



2002년 충남대학교 컴퓨터과학과 석사.

2005년 충남대학교 컴퓨터과학과 박사.

현재, 경상대학교 정보과학과 부교수.

관심분야: 네트워크 & 항공SW, 프로토콜 설계.

Email: kikum@gnu.ac.kr