

논문 2016-11-06

프레임률을 고려한 항공기 시뮬레이터의 영상 동기화 방안

(Display Synchronization Scheme for Flight Simulator Considering Frame Per Second)

이 순 영, 문 대 환, 이 충 재, 김 기 일*

(SunYoung Lee, Dae-Han Mun, ChungJae Lee, Ki-Il Kim)

Abstract: According to general architecture of flight simulator made up of several independent rendering display systems, display synchronization problem between them naturally happens. In addition, since the flight simulator is usually implemented in the same networks where network delay is not big concern, it is necessary to consider different factors of existing synchronization technique. Among them, in this paper, we propose a new display synchronization scheme for flight simulator where each system has different rendering capacity. In the proposed scheme, each system is synchronized by considering maximum and minimum frames per second (FPS) while considering level of detail and latency. Also, experimental results are given to demonstrate the suitability of the proposed scheme for display synchronization scheme.

Keywords : Flight simulator, Synchronization, Level of detail

1. 서 론

항공 시뮬레이터는 조종사에게 비행 훈련뿐만 아니라 위기 상황에서의 대처 및 무장 훈련 등에 활용된다. 이러한 항공기 시뮬레이터는 훈련 효과를 높이기 위해 실제 세계와 동일한 환경을 제공해야 하므로 위성영상과 고도 정보를 화면에 표시한다. 이러한 영상 데이터를 화면에 표시하는 경우 훈련 효과를 높이기 위해 시야각 (Field of View, FOV) 을 높임으로써 조종사에게 사실적인 환경을 제공한다. 하지만 시야각의 증가는 그래픽 하드웨어와 CPU의 처리량을 증가시키므로 시야각에 따른 분할 처리 방식이 사용된다.

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

Received: 3 Mar. 2015, Revised: 8 Apr. 2015,
Accepted: 28 May 2015.

S.Y. Lee, D.H. Mun, C.J. Lee, K.I. Kim:
Gyeongsang National University, Engineering
Research Institute

※ 본 논문은 BK21 플러스 무인항공기SW플랫폼
연구사업팀과 미래창조과학부의 2015년 고용계약
형 SW석사과정의 지원받아 수행되었음

이렇게 시야각을 분할하여 처리하는 영상 시스템의 경우 독립적인 시스템에서 시야각에 따라 화면을 분할하여 표시해야하기 때문에 각 영상 시스템간의 정밀한 동기화가 필수적이다. 이를 해결하기 위한 기존의 동기화 기법에는 네트워크 지연을 보정함으로써 서로간의 데이터를 동기화 한다 [1-3]. 하지만 항공기 시뮬레이터의 경우 네트워크 지연은 0.001ms 이하이므로 동기화 문제를 해결 시 네트워크 지연은 큰 문제가 되지 않는다. 반면, 각 영상 시스템의 구성 시 렌더링 속도가 다를 경우 이를 보정하기 위한 방안이 요구된다.

이를 위해 본 논문에서는 각 영상 시스템의 렌더링 속도에 따라 최대 및 최소화면 갱신률 (Frame Per Second, FPS)을 활용한 영상 동기화 방안을 제안한다. 첫 번째 방안의 경우, 상대적으로 낮은 FPS를 가진 시스템의 성능을 향상시켜야 하므로 적은 버텍스 (vertices)를 처리할 수 있도록 Level of Detail (LOD)를 적용한다. 반면, 두 번째 방안의 경우, 상대적으로 높은 FPS를 감소시켜야 하므로 3차원 렌더링 시 오브젝트를 지형을 구성하는 버텍스를 줄이는 대신 렌더링 과정에서 지연을 추가하는 방안이다. 두 방안은 이러한 제안하는 두

가지 방안의 구현은 오픈 소스 그래픽 라이브러리인 OSO (Open Scene Graph)를 사용하였으며 서로 다른 렌더링 성능을 가진 세 개의 시스템을 사용하여 성능 평가하였다.

본 논문의 구성은 다음과 같다. 연구의 목적을 설명한 서론에 이어 2장에서는 관련 연구를 소개한다. 3장은 제안 메커니즘을 제시하며 4장은 실험 결과를 분석한다. 마지막으로 5장에서는 결론이 제시된다.

II. 관련 연구

본 절에서는 네트워크 동기화 기법인 Network Time Protocol (NTP)과 IEEE1588과 렌더링 속도에 따른 동기화 방안을 위한 LOD에 대해 설명한다. 첫 번째로 네트워크 동기화 기법 중 NTP는 네트워크 지연을 보정하여 각 시스템의 시간을 동기화 하는 기법을 의미하며 그림 1에서 제시한다.

NTP는 네트워크로 연결되어 있는 호스트 간의 클럭 시간을 동기화하는 프로토콜이다 [4]. NTP는 미국 델라웨어 대학의 데이빗 밀스에 의해 처음 개발되었으며 동작원리는 그림 1과 같이 A 시스템은 B 시스템으로 정확한 시간을 요청하기 위해 클라이언트의 시간을 전송한다. 이 후 B 시스템은 오프셋 시간을 계산하여 함께 전송하며 A 시스템은 전송된 결과값에 따라 시간을 동기화 한다. 하지만 NTP 동기화 기법은 전송 지연을 고려하지 않으므로 이를 해결하기 위해 제안된 기법은 IEEE1588이다.

IEEE1588 기법은 마스터와 슬레이브 사이에 시간 동기화 프로토콜인 IEEE1588 프로토콜을 사용하여 시간 동기화 패킷과 네트워크 지연 측정 패킷을 사용하여 2초마다 시간을 동기화 시킨다. 서로 동기화 된 마스터와 각 슬레이브 사이의 네트워크 지연을 2초마다 측정하여 각 네트워크 지연 중 가장 짧은 지연을 선택하게 된다. IEEE1588 프로토콜을 사용한 동기화 기법은 그림 2에서 제시한다 [5]. 그림 2와 같이 선택된 지연은 호스트에서 디스플레이 패킷을 슬레이브로 전송할 때 최대 지연을 포함한 디스플레이 타임스탬프를 포함하여 전송한다. 수신된 데이터는 각 슬레이브에서 최대 네트워크 지연만큼의 지연 시간을 디스플레이 시간에 적용하여 각각의 슬레이브는 동시에 같은 영상 정보를 화면에 표시 할 수 있다. 하지만 이와 같은 시간 동기화 기법들은 렌더링 속도 차이에 따른 동기화 문제를 고려하지 않는다. 또한 본 논문에서는 상대적으로 낮은 FPS의 성능을 보이는 시스템의 FPS를

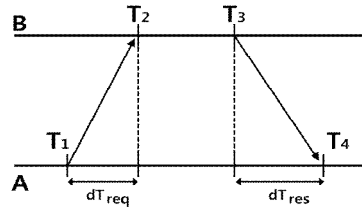


그림 1. NTP 작동 원리
Fig. 1 NTP Principle of Operation

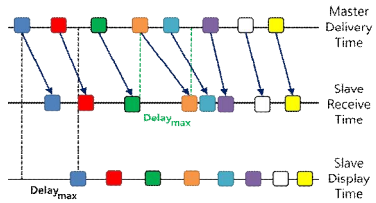


그림 2. IEEE1588 동기화 기법
Fig. 2 NTP Principle of Operation

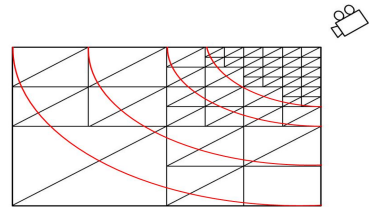


그림 3. 동적 LOD
Fig. 3 Dynamic LOD

항상시켜야하기 때문에 렌더링 연산에 포함된 객체의 수를 감소 시켜야한다. 이를 위해 본 논문에서는 LOD기법을 적용하였다. LOD 기법은 렌더링 하고자 하는 객체와 카메라 사이의 거리에 따라 그 정밀도를 달리하여 렌더링 속도를 향상 시키는 기법으로 정적 및 동적 LOD로 구분한다. 본 논문에서 제안하는 방안은 지형을 구성하는 버텍스 수를 조절하여 렌더링 데이터 처리량을 감소하기 때문에 실시간 렌더링 시 객체의 LOD 단계에 따라 데이터를 변경하는 동적 LOD 방식을 사용한다 [6, 7]. 동적 LOD는 그림 3에서 제시한다.

그림 3과 같이 동적 LOD는 객체에 대한 하나의 모델만을 가지고 객체와 카메라와의 거리에 따라 실시간으로 정점 수를 조절하여 정밀도를 조절하는 방식을 말한다. 해당 기법의 경우, 정적 LOD에 비해 LOD의 단계 변화에 따른 지형의 변화가 자연스

럽고 낭비되는 메모리가 감소하여 지형 데이터와 같은 대용량 객체 데이터에 사용한다. 이러한 기법을 사용하여 본 논문에서는 렌더링 성능에 따른 동기화 기법을 제안한다.

III. 제안 메커니즘

본 절에서는 본 논문에서의 제안하는 메커니즘인 최소 FPS를 기준으로 동기화 하는 방안과 LOD를 사용하여 최대 FPS를 기준으로 동기화 하는 방안을 설명한다.

1. 제안하는 시스템 구조

본 논문에서는 영상 시스템의 동기화 알고리즘에 대한 구현 및 성능 평가를 위해 실제 비행 시뮬레이터와 동일한 환경을 사용한다. 시스템 구조는 그림 4에서 제시한다.

그림 4와 같이 전체 시스템은 항공기의 움직임 을 묘사하는 역학 데이터를 포함한 호스트 시스템과 호스트로부터 전송된 데이터에 따라 화면에 데이터를 렌더링하여 출력하는 영상 시스템으로 나눈다. 영상 시스템은 전체 180°의 시야각을 제공하기 위해 각각 60°의 시야각을 가지는 3개의 디스플레이 시스템으로 나눈다. 또한 각 영상 시스템은 서로 다른 데이터 처리량을 가진 하드웨어로 구성되어 있으며 영상을 렌더링하는 FPS 또한 서로 다르게 구성되어 있다. 이렇게 구성된 시스템은 다음 절의 메커니즘에 의해 동작하며 서로 다른 처리량을 가진 시스템에서도 동기화가 가능하게 한다.

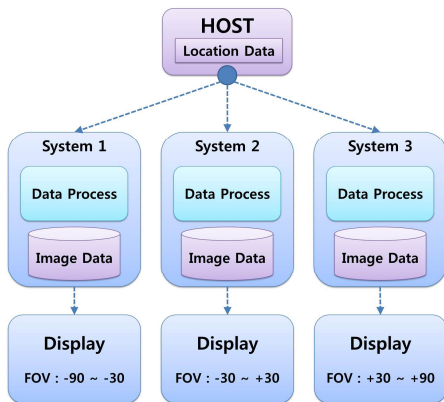


그림 4. 시스템 구조

Fig. 4 System Structure

2. 최소 FPS 기준 동기화 기법

최소 FPS를 기준으로 동기화 하는 기법은 이전 절의 서로 다른 처리량을 가진 영상 시스템 1,2,3으로 구성된 시스템에서 적용된다. 최소 FPS 기준 동기화 기법은 그림 5에서 제시한다.

그림 5와 같이 각각의 시스템은 40, 50, 60 FPS의 처리 속도를 가진 영상 시스템으로 구성된다. 이러한 시스템에서 FPS가 높은 시스템은 낮은 시스템에 비해 렌더링 시간이 상대적으로 더 길다. 따라서 동일한 화면을 같은 시간에 렌더링 하여도 동일한 시간에 각 분할 영상 시스템에서 같은 화면을 시현하지 못하게 되어 화면들은 동기화가 이루어지지 않는다. 이러한 비 동기화된 화면을 동기화하기 위해서는 렌더링 속도가 상대적으로 가장 낮은 최소 FPS를 가진 시스템 1을 기준으로 다른 두 시스템의 처리 시간을 동일하게 만들어 준다. 그림 6에서 호스트와 영상 시스템간의 동기화를 위한 과정을 제시한다.

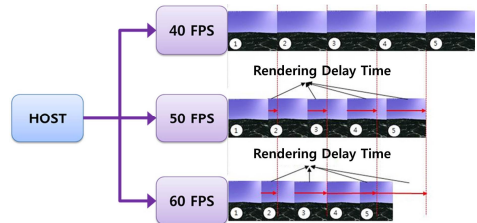


그림 5. 최소 FPS 기준 동기화 기법

Fig. 5 Synchronization Scheme based on Minimum FPS

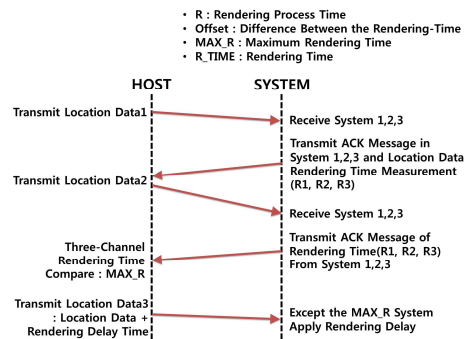


그림 6. 최소 FPS 기준 동기화 과정

Fig. 6 Synchronization Process based on Minimum FPS

우선 호스트는 연결된 각각의 영상 시스템으로부터 현재 위치 데이터를 전송한다. 이 후 영상 시스템은 위치 데이터에 따라 렌더링 과정을 거치고 이에 따른 소요 시간을 측정한다. 이렇게 측정된 소요 시간은 호스트에게 송신하며 호스트는 연결된 모든 영상 시스템의 소요 시간이 도착했을 경우 다음 위치 데이터를 전송한다. 호스트는 모든 영상 시스템이 송신한 두 번째 ACK 수신 시 각각의 영상 시스템에 대한 FPS를 비교한다. 이 후 호스트는 비교한 결과 값 중 최대 렌더링 소요 시간을 결정하고 이 결과값을 각 영상 시스템으로 전송한다. 이렇게 전송된 값에 따라 각각의 영상 시스템은 현재 시스템의 FPS를 감소하고 모든 영상 시스템은 동일한 처리 속도로 동작하게 되며 각 시스템의 영상은 동기화가 이루어진다.

3. 최대 FPS 기준 동기화 기법

최소 FPS 기준 동기화 기법은 상대적으로 높은 FPS를 가진 시스템이 렌더링 지연 시간의 추가로 인하여 전체 시스템의 효율성이 낮아지게 되므로 본 논문에서는 전체 시스템의 FPS 향상을 위해 최대 FPS를 기준으로 동기화 하는 기법을 제안한다. 최대 FPS를 기준으로 동기화 하는 기법은 그림 7에서 제시한다.

그림 7과 같이 최대 FPS 기준 동기화 기법은 이전 절에서 설명한 기법과 동일한 환경을 가진다. 하지만 이전 메커니즘과는 반대로 FPS가 가장 높은 시스템 3을 기준으로 시스템 1과 2의 렌더링 소요 시간을 감소하여 가장 높은 FPS에 따라 동작한다. 이때 상대적으로 짧은 렌더링 소요 시간을 가진 시스템에서는 렌더링 소요시간을 증가시키기 위해 앞선 2.3절에서 언급한 지형 데이터에 LOD 기법을 적용하고 원하는 렌더링 소요시간을 가질 때까지 데이터를 감소시킨다. 최대 FPS 기준 동기화 과정은 그림 8에서 제시한다.

최대 FPS 기준 동기화 과정은 3.2절에서 언급한 최소 FPS 기준 동기화 기법과 동일하다. 하지만 두 기법의 차이는 호스트가 각 영상 시스템의 렌더링 소요 시간을 수신한 후 최대 소요시간이 아닌 최소 소요 시간을 구한다. 이렇게 연산된 최대 FPS는 각 영상 시스템으로 전송되며 전송된 결과값에 따라 렌더링 소요 시간을 감소하기 위해 LOD 단계 조절 기법이 적용된다. LOD 단계 조절 기법은 현재 지형 데이터에 적용된 LOD의 단계별 거리를 감소하는 방식으로 지형을 이루는 버텍스의 수를 급격히 감소하여 렌더링에 소요되는 시간을 감소하는 방식이다.

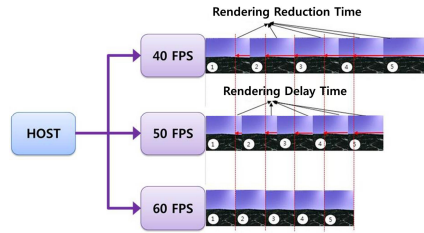


그림 7. 최대 FPS 기준 동기화 기법
Fig. 7 Synchronization Scheme based on Maximum FPS

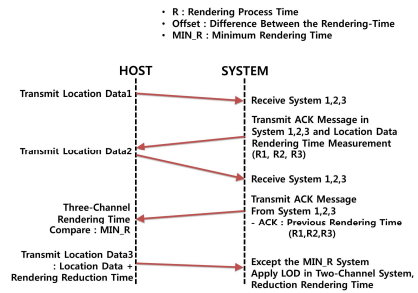


그림 8. 최소 FPS 기준 동기화 과정
Fig. 8 Synchronization Process based on Maximum FPS

각 영상 시스템은 전송된 FPS가 현재 FPS보다 높을 경우 전송된 FPS까지 도달하기 위해 지형 데이터에 LOD 단계 조절 기법을 적용하여 FPS를 증가시키고 모든 시스템은 동일한 FPS에 따라 동작하게 된다. 이렇게 각각의 영상 시스템은 동일한 FPS를 가진 시스템으로 변경되며 화면 동기화를 달성하게 된다.

IV. 실험 및 결과

본 논문에서는 제안하는 영상 동기화 기법을 실험하기 위하여 OSG 기반 비행 시뮬레이션 소프트웨어를 사용하였다. 또한 실험환경은 표 1와 같다.

표 1. 실험 환경
Table 1. Test Environment

	Host	IG1	IG2	IG3
CPU	i7-930 (2.80Ghz)	i7-930 (2.80Ghz)	i7-930 (2.80Ghz)	i7-930 (2.80Ghz)
RAM	4GB	4GB	4GB	4GB
VGA	GeForce GTX 650	GeForce GTX 250	AMD R9 270X	GeForce GTX 550

표 2. 최소 FPS 기준 동기화된 시간에 따른 호스트와 각 시스템의 항공기 자세 정보와 동기화 기법 사용 전의 자세 정보

Table 2. Aircraft State Data of Host and Display System according to Synchronization Scheme based on Minimum FPS and Aircraft State Data of without Synchronization Scheme

(a) 동기화 기법 사용 전

(a) Without Synchronization Scheme

출력 시각		12:52:000	12:52:025	12:52:050	12:52:075	12:52:100
HOST	Lat	35.3399634	35.3399636	35.3399637	35.3399638	X
	Long	128.5819729	128.5819730	128.5819730	128.5819731	
	Alt	2010ft	2011ft	2011ft	2010ft	
SYSTEM 1	Lat	X	35.3399634	35.3399634	35.3399636	35.3399636
	Long		128.5819729	128.5819729	128.5819730	128.5819730
	Alt		2010ft	2010ft	2011ft	2011ft
SYSTEM 2	Lat		35.3399634	35.3399634	35.3399637	35.3399637
	Long		128.5819729	128.5819730	128.5819730	128.5819730
	Alt		2010ft	2010ft	2011ft	2011ft
SYSTEM 3	Lat		35.3399634	35.3399636	35.3399637	35.3399638
	Long		128.5819729	128.5819730	128.5819730	128.5819731
	Alt		2010ft	2011ft	2011ft	2010ft

(b) 최소 FPS 기준 동기화

(b) Synchronization based on Minimum FPS Scheme

출력 시각		12:52:000	12:52:025	12:52:050	12:52:075	12:52:100
HOST	Lat	35.3399634	35.3399636	35.3399637	35.3399638	X
	Long	128.5819729	128.5819730	128.5819730	128.5819731	
	Alt	2010ft	2011ft	2011ft	2010ft	
SYSTEM 1	Lat	X	35.3399634	35.3399636	35.3399637	35.3399638
	Long		128.5819729	128.5819730	128.5819730	128.5819731
	Alt		2010ft	2011ft	2011ft	2010ft
SYSTEM 2	Lat		35.3399634	35.3399636	35.3399637	35.3399638
	Long		128.5819729	128.5819730	128.5819730	128.5819731
	Alt		2010ft	2011ft	2011ft	2010ft
SYSTEM 3	Lat		35.3399634	35.3399636	35.3399637	35.3399638
	Long		128.5819729	128.5819730	128.5819730	128.5819731
	Alt		2010ft	2011ft	2011ft	2010ft

본 논문에서 제안하는 FPS를 기반으로 한 동기화 방안의 경우 상대적 비교를 위한 구체적인 방안이 아직까지 제시되지 않았으므로 이를 통한 성능 분석은 불가능하다. 따라서, 본 실험에서는 제안된 방법을 적용하였을 경우의 동기화율이 향상정도를 측정하였다.

표 1과 같이 구성된 환경에서 동기화 검증 확인을 위하여 IEEE1558 시간 동기화 프로토콜을 이용하여 각 서브 시스템에 시간을 동기화 한다. 이 후

영상 시스템은 호스트로부터 전송된 위치 데이터에 따라 사전에 설정된 시각각을 렌더링하여 화면에 표시한다. 본 논문에서 제안한 동기화 기법을 검증하기 위해 각 시스템에서 같은 시간에 시현되는 위치 정보와 벡터의 수를 비교하였다. 표 2는 최소 FPS 기준 동기화 기법의 실험 결과이며 표 3은 최대 FPS 기준 동기화 기법의 실험 결과를 보여준다. 표 2에서와 같이 호스트에서 전송된 데이터를 기준으로 각 영상 시스템은 동일한 시간에 동일한 데이

표 3. 최대 FPS 기준 동기화된 시간에 따른 호스트와 각 시스템 사이의 항공기 자세 정보와 동기화 기법 사용 전의 자세 정보

Table 3. Aircraft State Data of Host and Display System according to Synchronization Scheme based on Maximum FPS and Aircraft State Data of without Synchronization Scheme

(a) 동기화 기법 사용 전
(a) Without Synchronization Scheme

출력 시각		1:13:000	1:13:016	1:13:032	1:13:049	1:13:065
HOST	Lat	24.3399632	24.3399640	24.3399642	24.3399641	X
	Long	112.5819729	112.5819730	112.5819732	112.5819733	
	Alt	1910ft	1910ft	1911ft	1911ft	
SYSTEM 1	Lat	X	24.3399632	24.3399640	24.3399640	24.3399642
	Long		112.5819729	112.5819730	112.5819730	112.5819732
	Alt		1910ft	1910ft	1910ft	1911ft
	Vertex		171548	171435	171754	171145
SYSTEM 2	Lat	X	24.3399632	24.3399632	24.3399642	24.3399642
	경도		112.5819729	112.5819729	112.5819732	112.5819732
	Alt		1910ft	1910ft	1911ft	1911ft
	Vertex		170578	170345	170587	170489
SYSTEM 3	Lat	X	24.3399632	24.3399640	24.3399642	24.3399641
	Long		112.5819729	112.5819730	112.5819732	112.5819733
	Alt		1910ft	1910ft	1911ft	1911ft
	Vertex		170776	170864	170899	170867

(b) 최대 FPS 기준 동기화

(b) Synchronization based on Maximum FPS Scheme

출력 시각		1:13:000	1:13:016	1:13:032	1:13:049	1:13:065
HOST	Lat	24.3399632	24.3399640	24.3399642	24.3399641	X
	Long	112.5819729	112.5819730	112.5819732	112.5819733	
	Alt	1910ft	1910ft	1911ft	1911ft	
SYSTEM 1	Lat	X	24.3399632	24.3399640	24.3399642	24.3399641
	Long		112.5819729	112.5819730	112.5819732	112.5819733
	Alt		1910ft	1910ft	1911ft	1911ft
	Vertex		120203	120239	121256	125656
SYSTEM 2	Lat	X	24.3399632	24.3399640	24.3399642	24.3399641
	Long		112.5819729	112.5819730	112.5819732	112.5819733
	Alt		1910ft	1910ft	1911ft	1911ft
	Vertex		150123	150154	150645	150112
SYSTEM 3	Lat	X	24.3399632	24.3399640	24.3399642	24.3399641
	Long		112.5819729	112.5819730	112.5819732	112.5819733
	Alt		1910ft	1910ft	1911ft	1911ft
	Vertex		170776	170864	170899	170867

터 사용하여 렌더링 되는 것을 확인 할 수 있다. 동기화 기법 사용 전과 비교하면 최소 FPS 기준 동기화 기법은 시간별로 각 시스템은 동일한 위치

데이터를 렌더링 하는 반면 동기화 기법 사용 전은 세 개의 시스템이 서로 동기화가 이루어지지 않음을 확인 할 수 있다. 이는 각 시스템의 서로 다른

렌더링 소요에 따라 발생하는 결과로써 본 논문에서 제안하는 기법은 이러한 문제를 해결함을 보인다. 표 3은 최대 FPS 기준 동기화 기법의 실험 결과이며 LOD를 적용하여 상대적으로 낮은 FPS 성능을 보이는 영상 시스템 1과 2의 시스템에 LOD가 적용되므로 영상 시스템 3과 비교하면 상대적으로 낮은 삼각망을 렌더링하고 있음을 확인하였다. 이와 같은 실험 결과 제안하는 두 메커니즘 결과 모두 동일한 시간에 동일한 위치 정보를 렌더링 함으로써 호스트에 연결된 모든 영상 시스템의 동기화를 확인하였다.

하지만, 두 가지 메커니즘 중 최소 FPS 기준 동기화 방안은 렌더링 과정에서 소요되는 시간이 상대적으로 빠른 시스템의 경우 높은 FPS를 유지할 수 있음에도 불구하고 인위적인 지연시간을 추가하기 때문에 시스템 효율성을 감소시켜 CPU 실속 현상이나 GPU 기아현상을 발생 시킨다. 또한 각 시스템간의 렌더링 소요 시간 격차가 증가할수록 낭비되는 자원은 비례하기 때문에 이를 보완하는 방법이 요구되며 이는 최대 FPS 기준 동기화 방안으로 해결이 가능하다. 하지만 이러한 방안은 지형에 LOD를 적용하기 때문에 지형의 급격한 변화 시 크랙 현상이 발생하고 이를 해결하기 위한 대처가 추가적으로 요구된다.

V. 결 론

본 논문은 시야각에 따라 서로 다른 렌더링 속도를 가진 독립된 영상 시스템에서 화면 동기화 방안을 제안한다. 이를 위해 각 영상 시스템에서 최소 및 최대 렌더링 성능 기반 동기화 기법을 적용하여 각 메커니즘의 성능을 측정하였고 실험 결과 두 메커니즘의 동기화가 이루어짐을 확인하였다.

하지만 현재 메커니즘은 호스트로부터 고정적인 데이터 속도 기반으로 동기화를 이루어졌으며 향후 유동적인 데이터 속도 기반에서의 동기화에 대한 연구를 진행 할 예정이다.

References

- [1] F. Cristian, "Probabilistic clock synchronization," *Distributed Computing*, Vol. 3, No. 3, pp. 146-158, 1989.
- [2] J.L. Lin, K.S. Hwang, Y.S. Hsiao, "A synchronous display of partitioned images

broadcasting system via VPN transmission," *IEEE Systems Journal*, Vol. 8, No. 4, pp. 1031-1039, 2014.

- [3] J.W. Song, Y.M. Kwon, "Network synchronization for collaborative work in distributed environment," *Journal of Korea society of IT services*, Vol. 10, Issue 3, pp. 203-212, 2011.
- [4] Y. Lin, K. Guo, S. Paul, "Sync-MS : Synchronized Message Service for Real-Time Multi-Player distributed Games," *International Conference on Network Protocols*, pp. 155-162, 2002.
- [5] C. Yun, J. Hwang, D. Paik., "Effective message synchronization methods for multiplayer online games with map," *Journal of Computers in Human Behavior*, Vol. 24, pp. 2477-2485, 2008.
- [6] W. Ying, W. Yanjie, "Implementation of a fast simulation algorithm for terrain based on Dynamic LOD," *International Conference on Electronic Measurement & Instruments*, pp. 189-193, 2011.
- [7] H. He, Y. Xing, T. Wang, "Real-time rendering system of large-scale terrain in flight simulation: design and implementation," *International Conference on Computer Graphics & Imaging and Visualization*, pp. 180-185, 2009.

SunYoung Lee (이 승 영)



He received the B.S. degree in Computer Science from Gyeongsang National University. He is currently working toward M.S. degree in Department of Informatics at Gyeongsang National University. His research interests include Wireless Sensor Network.

Email: astesia@icloud.com

Dae-Han Mun (문 대 환)

He received the B.S. degree in Electrical Engineering from Gyeongsang National University and M.S. degree in Department of Converged Software from Gyeongsang National University. He is currently Research Associate in APPIA Engineering Ltd. His research interests include Avionics Software.
Email: Belldandy00@hanmail.net

Chungjae Lee (이 충 재)

He received the B.S. degree in avionics and simulation from Hanseo University and M.S. degree in aerospace engineering from Gyeongsang National University. He is currently working toward Ph.D. degree at Gyeongsang National University. His research interests include Flight Simulation Software.
Email: cjlee@gnu.ac.kr

Ki-Il Kim (김 기 일)

He received M.S. and Ph.D. degrees in computer science from the Chung-Nam National University, Daejeon, Korea. He is currently with the Department of Informatics at Gyeongsang National University. His research interests include ad hoc/sensor networks and avionics software.
Email: kikum@gnu.ac.kr