현실환경과 가상 환경을 연동하는 맥락 인식 기반 U-VR 시뮬레이터

Context-aware U-VR Simulator for Linking Real and Virtual Environments

오유수, Yoosoo Oh*, 강창구, Changgu Kang**, 우운택, Woontack Woo***

요약 본 논문에서는 유비쿼터스 가상 현실 개념에 기반하여 맥락 정보를 이용하여 현실과 가상 환경을 이음매 없이 연결하는 U-VR 시뮬레이터를 제안한다. U-VR 시뮬레이터는 스마트 홈 환경을 시뮬레이션 할 수 있고, 센서, 액추에이터, 서비스와 같은 새로운 가상 개체들을 기존의 실제 개체에 추가할 수 있게 한다. 제안된 시뮬레이터는 기존의 장치와 시뮬레이션 할 장치를 동시에 활용함으로써 빠른 개발과 비용 절감의 효과가 있다. 또한, 제안된 시뮬레이터는 어플리케이션 개발자들이 유비쿼터스 가상 현실 응용을 쉽고 빠르게 개발하고, 적절한 도메인으로 확장하게 한다. 추후 연구로는 제안된 시뮬레이터가 현실 환경에 가상의 개체들을 지능적으로 증강하도록 확장할 것이다.

Abstract In this paper we propose the U-VR Simulator which seamlessly connects entities in real and virtual environments by exploiting context, based on the concept of ubiquitous virtual reality. Using the U-VR Simulator, a smart home environment can be simulated and new virtual entities such as sensors, actuators, and services can be added to existing real entities. The proposed approach is fast and cost-effective because the simulator effectively utilizes both existing and simulated devices. In addition, our approach allows application developers to rapidly develop U-VR applications and extend them to relevant domains. As a future work, we will expand our simulator to intelligently augment virtual entities into real environments.

핵심어: 시뮬레이션, 유비쿼터스 가상 현실, 맥락 인식, 현실과 가상 환경

본 연구는 문화체육관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소육성사업의 연구결과로 수행되었음

*주저자 : 광주과학기술원 정보통신공학과 박사과정 e-mail: yoh@gist.ac.kr

**공동저자 : 광주과학기술원 정보통신공학과 석사과정 e-mail: ckang@gist.ac.kr

***교신저자 : 광주과학기술원 정보통신공학과 교수; e-mail: wwoo@gist.ac.kr

1. 서론

유비쿼터스 컴퓨팅 환경의 시뮬레이션은 센서, 서비스와 환경이 포함된 다양한 시나리오에 적용된다 [1]. 유비쿼터스 컴퓨팅 환경 중의 하나인 스마트 홈 환경에서 센서, 서비스, 액추에이터와 같은 개체들은 상호 연동되어 있다. 현실 환경에서 이와 같은 개체들을 동시에 테스트하는 것은 대체적으로 많은 시간과 노력이 소모된다. 예를 들어, 기존의 개체들을 교체하거나 새로운 개체들을 추가하는 것은 개체들의 설치, 실험과 복구로 인하여 재정적 비용을 소모한다. 또한, 다양한 종류의 센서, 액추에이터와 많은 수의 맥락 정보들이 산재해있는 현실 환경에서 그런 개체들을 테스트하는 것은 비용 소모가 크다 [2]. 더욱이, 주어진 프레임워크나 기반 구조를 테스트할 때마다 모든 사용 가능한 개체들을 배치하는 것은 어려운 일이다. 따라서, 실제 스마트 홈 환경에서의 새로운 개체의 추가나 확장은 이미 고정된 구조물과 높은 재정적 비용 때문에 쉬운 일이 아니다 [3].

가상 환경에서의 개체들의 시뮬레이션은 상대적으로 비용 절감 효과가 있지만, 실시간 디버깅, 반응, 시스템 행동 등과 같은 시스템에서의 실제 동작을 지원하기가 어렵다 [1,2]. 따라서, 현실과 가상 환경에서의 테스트가 고려되어야 한다. 유비쿼터스 가상 현실 (U-VR: Ubiquitous Virtual Reality) 환경 [4] 은 현실과 가상 환경에서의 컴퓨팅을 가능하게 한다. 이와 같은 U-VR 환경에서는 개체들의 시뮬레이션이 특정 어플리케이션의 개발을 증진시킨다. U-VR 환경과 관련된 연구로써, 본 논문에서는 스마트 홈에서의 시뮬레이터들을 조사하였다. 최근 스마트 홈 시뮬레이터 관련된 연구로는 UbiREAL [2], eHomeSimulator [3], TATUS [5], UBIWISE [6], CASS [7], CAST [8], C@SA [9] 등이 있다. 그러나 관련 연구들은 새롭게 설계된 개체들에 대해서만 고려하였고, 기존에 이미 구축되어 있는 실제 개체들에 대한 효율적인 활용 사례가 없었다. 즉, 새롭게 추가되는 가상 개체들과 기존의 실제 개체들의 혼합 환경에서의 개체들 간의 시뮬레이션에 대한 지원이 없었다.

이와 같은 문제점을 해결하기 위하여 본 논문에서는 유비쿼터스 가상 현실 개념에 기반하여 맥락 정보를 이용하여 현실과 가상 환경을 이음매 없이 연결하는 U-VR 시뮬레이터를 제안한다. U-VR 시뮬레이터는 스마트 홈 환경을 시뮬레이션 할 수 있고, 센서, 액추에이터, 서비스와 같은 새로운 가상 개체들을 기존의 실제 개체에 추가할 수 있게 한다. 제안된 시뮬레이터는 기존의 장치와 시뮬레이션 할 장치를 동시에 활용함으로써 빠른 개발과 비용 절감의 효과가 있다. 또한, 제안된 시뮬레이터는 현실 환경과 가상 환경의 통합 디버깅 테스트를 구현하여 소스 코드 라인 감소 및 빠른 프로토타이핑과 같은 장점을 지닌다. 마지막으로 제안된 시뮬레이터는 어플리케이션 개발자들이 유비쿼터스 가상 현실 응용을 쉽고 빠르게 개발하고, 적절한 도메인으로 확장하게 한다.

2. U-VR 시뮬레이터

2.1 맥락 인식 프레임워크

이질적인 환경에서 제안된 시뮬레이터가 동작하도록 제안된 시뮬레이터는 맥락 인식 프레임워크인 UCAM (Unified Context-aware Application Model) [10]을 활용하였다. UCAM 은 물리적인 센서와의 연결을 위한 논리적인 센서 역할의 UCAM Sensor 와 맥락 정보를 관리하기 위한 UCAM Service 로 구성된다. UCAM 은 사람, 현실 오브젝트, 그리고 가상 오브젝트 간의 사회적 관계를 관리한다. 현실과 가상 개체들 간의 이음매 없는 연동을 위해서 본 논문에서는 수정된 버전의 UCAM 을 개발하여 제안된 U-VR 시뮬레이터에 적용하였다.

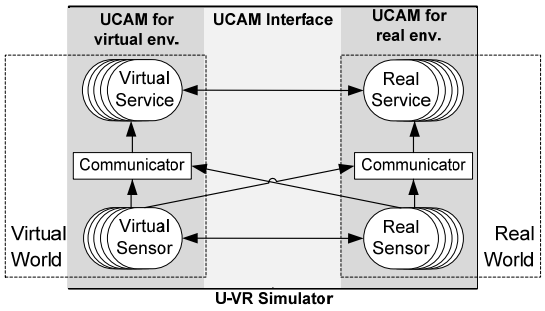


그림 1. U-VR 시뮬레이터에서의 현실 환경과 가상 환경 간의 맥락 흐름도

U-VR 시뮬레이터는 현실-현실, 현실-가상, 그리고 가상-가상 환경을 연동시킨다. 이와 같은 기능을 지원하기 위하여, 수정된 UCAM 은 맥락 정보 교환을 통해서 이질적인 환경 간의 동적 연결을 제공한다. 그림 1 은 U-VR 시뮬레이터에서의 현실 환경과 가상 환경 간의 맥락 흐름도를 나타낸다. 그림 1 에서 실선은 맥락 흐름을 나타낸다. 그림 1 의 모든 개체들 (e.g., real/virtual services, real/virtual sensors)은 동적으로 상호 통신한다.

더욱이, 현실과 가상 환경 간의 직접적인 통신으로 인하여 제안된 시뮬레이터는 다양한 개체들의 동작을 검증하고, 현실 환경과 구현될 다른 환경을 동시에 시뮬레이션 할 수 있다. 기존 관련 연구들과는 달리, 제안된 시뮬레이터는 이미 구축된 하드웨어나 소프트웨어 프레임워크를 이용하여 새로운 개체를 시뮬레이션 한다. 그래서 제안된 시뮬레이터는 새로운 개체와 기존의 개체들을 연동하여 조화롭게 하고, 실시간 시뮬레이션으로 물리적 개체의 개발 시간을 단축시킨다.

2.2 3D 스마트 홈 시뮬레이터

U-VR 시뮬레이터는 스마트 홈 환경을 위한 3 차원 그래픽 인터페이스를 가지며, 이 인터페이스가 동적으로 현실과 가상 스마트 홈 환경에서 작동하게 한다. U-VR 시뮬레이터의 모든 개체들은 UCAM 에 기반하여 상호

작동하며, 실감적으로 맥락 정보를 생성한다. 또한, 시뮬레이터의 가상 스마트 홈에서의 개체들은 실제 스마트 홈 환경에서의 다른 개체들과 연결되어 있어서, U-VR 시뮬레이터는 개발자들이 쉽게 새로운 개체들을 추가하거나 배치하게 한다.

U-VR 시뮬레이터는 맥락 인식을 위한 UCAM 인터페이스와 3 차원 가상 환경을 관리하기 위한 Space Manager 를 포함한다. 그림 2 는 U-VR 시뮬레이터의 구성요소를 나타낸다. Network Module 은 통신 채널로써 UCAM 과 연결되는 인터페이스 역할을 한다. Repository 는 실제 센서 데이터와 시뮬레이터 행동을 기록하고, 이때의 실제 센서 데이터는 자동적으로 수집되고 맥락 정보로 변환된다. 수집된 데이터를 활용하여 가상 센서는 실제 센서의 특성과 행동을 학습한다. 기록된 시뮬레이터 행동은 실제 스마트 홈을 테스트할 때 물리적인 개체들로 인하여 많은 비용이 소모되고 복잡한 과정이 요구되기 때문에 개발자가 실제 장치가 없는 스마트 홈을 테스트할 때 이용된다. 그러므로 기록된 시스템 행동을 통하여 우리는 정확하고 효율적으로 스마트 홈을 시뮬레이션 할 수 있다.

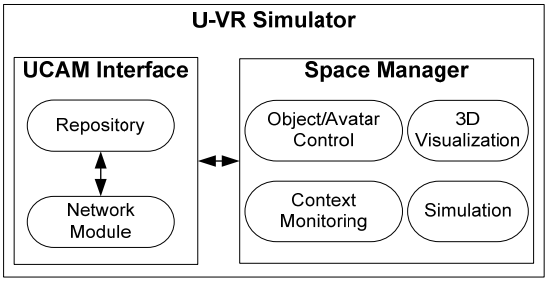
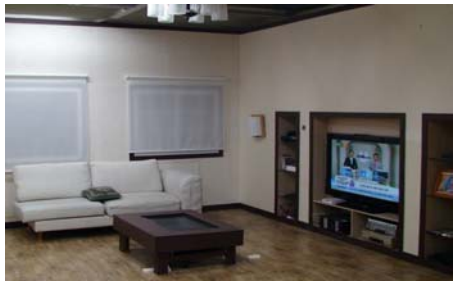


그림 2. U-VR 시뮬레이터 구성 요소

Space Manager 는 control, visualization, monitoring 과 simulation 의 4 가지 기능을 가진다. Object/Avatar Control 모듈은 가상 스마트 홈에서의 모든 개체들을 제어하고, 각 개체 (아바타, 오브젝트, 센서, 서비스)를 제어하기 위한 3 차원 사용자 인터페이스를 관리한다. 3D Visualization 모듈은 가상 개체들의 시각적 효과를 디스플레이하며, 시각적 애니메이션을 통해서 어떻게 각 개체가 맥락 정보에 기반하여 행동하는지를 나타낸다. Context Monitoring 모듈은 실시간으로 맥락 정보의 흐름을 감지하여 시스템 행동을 추적한다. 마지막으로 Simulation 모듈은 환경을 구성하고, 센서, 서비스와 아바타의 행동을 등록한다. 이 모듈은 실제 스마트 홈에서의 새로운 개체들의 추가나 기존 개체들의 수정을 시뮬레이션한다. 이와 같은 기능을 이용하여, U-VR 시뮬레이터는 물리적인 장치 없이 스마트 홈을 시뮬레이션하고 가상으로 테스트할 수 있게 한다.

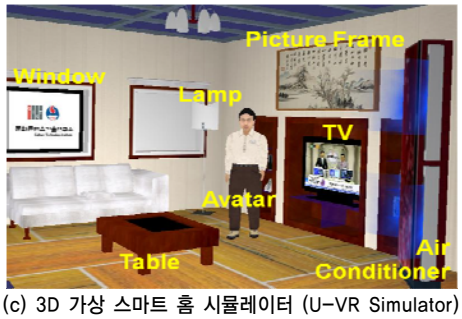
3. 구현

제안된 시뮬레이터의 효과를 검증하기 위해서 우리는 그림 3 과 같이 U-VR 시뮬레이터에 기반한 스마트 홈을 구축하였다. 먼저, 본 논문에서는 기 구축된 실제 스마트 홈 [그림 3(a)]을 모델링하여, Macromedia Flash 9 에 기반하여 2 차원 시뮬레이터 [그림 3(b)]를 구현하였고, 시각적 효과와 개체 배치의 자유도를 증가시키기 위하여 제안된 3 차원 시뮬레이터 [그림 3(c)]를 구현하였다. 제안된 3 차원 시뮬레이터는 OpenSceneGraph, cal3d, osgcal 라이브러리를 이용하여 Visual Studio 2005(C++)에 기반하여 개발되었고, 시뮬레이터의 가상 환경은 3D Studio MAX 9 를 이용하여 구축되었다. 본 논문에서는 구축된 스마트 홈 [그림 3(a)] 에 다양한 스마트 센서들과 맥락 인식 서비스들을 구현하였고, 이것이 똑같이 반영된 시뮬레이터 상의 가상 스마트 홈 [그림 3(c)]을 구축하였다. 구현된 서비스들로는 실제 서비스들(TV, 전등, 윈도우, 테이블)과 가상 서비스들(TV, 전등, 윈도우, 테이블, 에어컨, 사진액자, 아바타)가 있다. 또한, 구현된 센서로는 실제 센서들(프로파일, activity, 조도, 온도, 생체신호)과 가상 센서들(프로파일, 조도, 온도, 위치)이 있다.



(a) 실제 스마트 홈

(b) 2D 가상 스마트 홈 시뮬레이터



(c) 3D 가상 스마트 홈 시뮬레이터 (U-VR Simulator)

그림 3. 구현된 현실 및 가상 환경들

현실과 가상 환경 사이의 통신을 위한 메시지 데이터 포맷은 획득된 맥락 정보로 표현된다. 예를 들어, 가상 시뮬레이터로부터 실제 스마트 홈 환경으로의 메시지 포맷은 "*VirtualSensor:0/Name:Simulator/Illuminance:5/Temperature:21/*" 와 같이 표현된다. 이 포맷에서 "VirtualSensor:0"는 개체의 종류와 아이디를 나타내고, 나머지는 획득된 개체들의 종류와 값을 나타낸다. 이렇게 획득된 개체들의 종류와 아이디 및 값은 맥락 정보를 해석하여 표현된다. 실제 센서들은 UCAM Sensor 에 기반하여 물리적 센싱 장치로 구현되었고, 실제 서비스들은 UCAM Service 에 기반하여 물리적인 서비스 장치 위에 구현되었다. 시뮬레이션(가상) 센서들은 실제 센서의 물리적 특성에 따라서 구현되었고, 센서 프로파일과 UCAM Sensor 에 기반하여 3 차원 시뮬레이터의 이벤트 방식의 오브젝트로 프로그래밍되었다. 시뮬레이션(가상) 서비스들 역시 시뮬레이션 센서와 마찬가지로 구현되었다.

구현된 U-VR 시뮬레이터는 그림 4 에서 보는 것처럼 가상 센서와 서비스, 실제 센서와 서비스, 그리고 시뮬레이터 코어로 구성되는 구조를 가진다. 가상 서비스는 그림 2 의 Space Manager 의 모든 기능과 UCAM Service 로 이루어지며, 가상 센서는 사용자의 입력에 따라서 반응하는 Manual Manipulation 과 실제 센서의 동작이 그대로 반영되어 스스로 반응하는 Automatic Manipulation 의 기능과 UCAM Sensor 로 이루어진다. 실제 서비스와 센서는 앞서 언급한 것과 같이 물리적인 장치 위에서 동작한다. 시뮬레이터 코어는 UCAM 기반의 네트워크를 통하여 현실 환경과 가상 시뮬레이션 환경이 연동되도록 하는 중요한 역할을 한다. 시뮬레이터 코어는 U-VR 시뮬레이터의 환경을 구성 및 제어하고, 개체들의 동작 히스토리를 저장하고 반영한다.

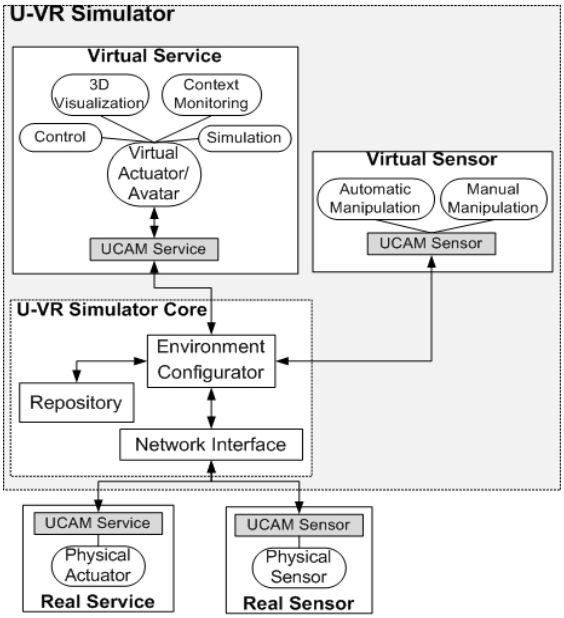


그림 4. 구현된 U-VR Simulator 구조

그림 5 는 U-VR 시뮬레이터에서 구현된 개체들의 사례를 나타낸다. U-VR 시뮬레이터에는 가상 아바타와 실제 사용자가 연동되어 있어서 실제 사용자의 위치에 따라서 아바타가 동시에 움직이며, 서로 연동된 가상과 실제 전등 서비스가 가상과 실제 센서나 가상과 실제 액추에이터(스위치)에 반응하여 상호 동작한다. 가상과 실제 TV 나 윈도우 디스플레이 장치들도 마찬가지로 가상과 실제 센서들과 사용자의 움직임에 따라서 실시간으로 동작된다. 즉, 가상과 실제 개체들은 모두 U-VR 시뮬레이터에서 맥락 정보를 통해서 상호 연동되어 있다.

(a) 가상 아바타와 실제 사용자

(b) 가상과 실제 전등 장치

(c) 가상과 실제 TV 장치

(d) 가상과 실제 윈도우 디스플레이

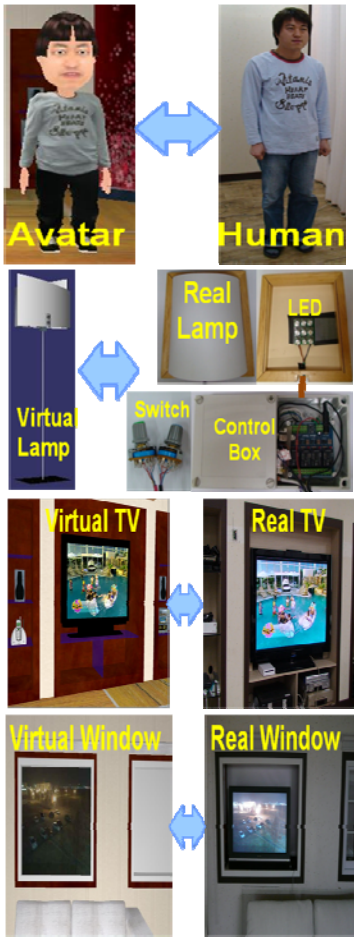


그림 5. U-VR 시뮬레이터에서 구현된 개체들

4. 실험

본 장에서는 제안된 U-VR 시뮬레이터가 현실과 가상 환경에 있는 개체들을 이음매 없이 연동하는 것을 검증하기 위한 실험을 하였다. 실험은 현실과 가상 환경의 개체들 간의 입력에 대한 반응 시간을 측정함으로써 상호 연동 정도를 비교하였다. 실험 환경은 하드웨어 장치로는 각 개체에 해당하는 laptop(Windows Vista, Intel Core2 Duo 1.66 GHz, 2 GB RAM)과 Particle 센서 (조도센서) [15]가 이용되었고, 구현된 U-VR 시뮬레이터를

이용하였다. 각 UCAM Service 에서의 맥락 정보 처리 간격은 50ms 로 하고, 센서와 서비스에서의 맥락 정보는 실시간으로 생성하여 전송하였다. 실험에 사용된 개체로는 실제 전등 서비스와 가상 전등 서비스, 실제 조도 센서와 가상 조도 센서를 활용하였다. 그리고 현실과 가상 환경의 개체들 간의 연동으로는 가상 조도 센서-가상 전등 서비스(A), 가상 조도 센서-실제 전등 서비스(B), 실제 조도 센서-가상 전등 서비스(C), 실제 조도 센서-실제 전등 서비스(D)의 4 가지 경우가 이용되었다. 네트워크 상황에 영향이 없는 정확한 측정을 위하여 본 실험에서는 각 경우에 대해서 1000 개의 조도 센서 입력에 대한 1000 개의 전등 서비스 반응에 대하여 각각의 시간(센싱 시간과 서비스 반응 시간)을 측정하고 2 개 시간 합의 평균을 구하였다. 그림 6 은 반응 시간에 대하여 측정된 결과이다.

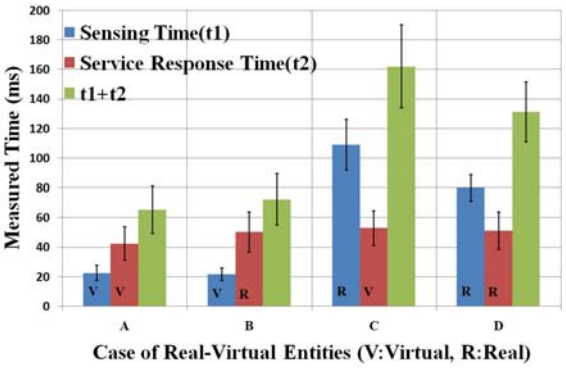


그림 6. 현실과 가상 환경의 개체들 간 측정된 반응 시간

실험 결과, 가상 조도 센서-가상 전등 서비스(A)와 가상 조도 센서-실제 전등 서비스(B)의 경우는 상대적으로 총 시간이 작게 측정된 반면, 실제 조도 센서-가상 전등 서비스(C)와 실제 조도 센서-실제 전등 서비스(D)의 경우는 총 시간이 비교적 높게 측정되었다. 각 경우의 t1 과 t2 의 시간을 관찰하면, 실제 조도 센서가 가상 조도 센서보다 센싱 시간이 길기 때문에 총 시간을 높게 하는 요인임을 알 수 있다. 그럼에도 불구하고, 전체적인 연동 시간은 약 65~160ms 로 빠른 시간 안에 현실과 가상 개체들 간의 연동이 이루어짐을 알 수 있다. 또한, 실제 센서에서의 센싱 시간은 물리적인 센서의 영향이기 때문에 제안된 U-VR 시뮬레이터의 성능에는 지장을 주지 않으므로, 제안된 시뮬레이터가 측정된 4 가지 경우에서도 비교적 균일하게 동작하는 성능을 보임을 알 수 있다.

4. 결론

본 논문에서는 스마트 홈 환경에서 새로운 가상의 개체들을 기존의 실제 개체들에 추가하여 동시에 시뮬레이션할 수 있는 U-VR 시뮬레이터를 제안하였다. U-VR 시뮬레이터는 UCAM 을 이용하여 이음매 없이 현실과 가상 환경에 있는 개체들을 연동시킨다. 제안된 시뮬레이터는 개발자가 쉽게 맥락 정보의 흐름을 확인할 수 있게 하고, 문제가 발생했을 때 빠르게 각 개체를 디버깅할 수 있게 한다. 또한, 제안된 시뮬레이터는 기존의 갖춰진 실제 장치(개체)를 이용하고 그 개체들을 쉽게 배치할 수 있어서 비용 절감의 효과가 있다. 추후 연구는 제안된 시뮬레이터가 현실 환경에 가상의 개체들을 지능적으로 증강하도록 확장하는 것이다.

참고문헌

[1] V. Reynolds, V. Cahill, and A. Senart, "Requirements for a ubiquitous computing simulation and emulation environment", ACM InterSense, 2006, vol. 138.

[2] H. Nishikawa, S. Yamamoto, M. Tamai, K. Nishigaki, T. Kitani, N. Shibata, K. Yasumoto, and M. Ito, "UbiREAL: Realistic Smartspace Simulator for Systematic Testing", UbiComp, LNCS4206, 2006, pp. 459-476.

[3] I. Armac and D. Retkowitz, "Simulation of Smart Environments", IEEE ICPS, 2007, pp. 322-331.

[4] Y. Lee, S. Oh, C. Shin, and W. Woo, "Recent Trends in Ubiquitous Virtual Reality", IEEE ISUVR, 2008, pp. 33-36.

[5] E. O'Neill, M. Klepal, D. Lewis, T. O'Donnell, D. O'Sullivan, D. Pesch, "A testbed for evaluating human interaction with ubiquitous computing environments", Tridentcom, 2005, pp. 60-69.

[6] J.J. Barton and V. Vijayarahgavan, "Ubiwise: A Ubiquitous Wireless Infrastructure Simulation Environment", tech. report HPL2002-303, HP Labs, 2002.

[7] J. Park, M. Moon, S. Hwang, and K. Yeon, "CASS: A Context-Aware Simulation System for Smart Home", IEEE SERA, 2007, pp. 461-467.

[8] I. Kim, H. Park, Y. Lee, S. Lee, H. Lee, and B. Noh, "Design and Implementation of Context-Awareness Simulation Toolkit for Context learning", IEEE SUTC, 2006, pp. 96-103.

[9] B. De Carolis, G. Cozzolongo, S. Pizzutilo, and V.L. Plantamura, "Agent-Based Home Simulation and Control", ISMIS, LNCS3488, 2005, pp. 404-412.

[10] Y. Oh and W. Woo, "How to build a Context-aware Architecture for Ubiquitous VR", ISUVR, CEUR-WS, 2007, pp. 032-033.

[11] OpenSceneGraph2.2, http://www.openscenegraph.org

[12] cal3d library, http://home.gna.org/cal3d

[13] osgcal library, http://osgcal.sourceforge.net

[14] Autodesk 3D StudioMax9, http://usa.autodesk.com/adsk

[15] Particle, TECO, http://particle.teco.edu/.