3축 가속도 센서를 이용한 증강현실 기반의 동적 상태변환 알고리즘

Augmented Reality based Dynamic State Transition Algorithm using the 3-Axis Accelerometer Sensor

장유나, 박성준
호서대학교 게임공학과

Yu-Na Jang(joyrisk@gmail.com), Sung-Jun Park(sjpark@hoseo.edu)

요약

스마트폰의 도입으로 인하여 증강현실이 널리 알려짐에 따라 대중들의 관심은 이에 집중되고 있으며 휴대성으로 인하여 모바일기기에서의 증강현실 연구가 하나의 흐름으로 자리 잡고 있다. 기존의 증강 현실 관련 응용 기술들이 많이 연구되고 있지만 실제 게임에서 사용되고 있는 인공 지능과 결합된 연구는 이루어지고 있지 않다. 본 논문에서는 스마트 폰의 기능중 하나인 3축 가속도 센서를 이용하여 증강 현실 환경에서 3D 에이전트의 상태를 동적으로 변환하는 인공 지능 알고리즘을 제안한다. 인공지능이 적용된 에이전트의 상태를 제어하기 위한 전통적인 방식은 사용자가 직접 입력해 주거나 이를 인식하는데 마커를 사용하여 해결하였다. 정확한 상태 변환을 위하여 가속도 센서의 값에 임계값을 주었으며 임계값은 실험을 통하여 결정되었다. 본 논문에서는 증강 현실 구현을 위해 마커리스 추적 기술을 사용하였고 3축 가속도 센서를 이용하여 동적으로 에이전트의 상태를 변환하도록 하였다.

■ 중심어 : | 모바일 | 증강현실 | 인공지능 |

Abstract

With the introduction of smart phones, the augmented reality became popular and is increasingly drawing attention. The augmented reality in the mobile devices is becoming an individual area to study. Many applications of the augmented reality have been studied, but there are just a few studies on its combination with artificial intelligence in games. In this study, an artificial intelligence algorithm was proposed, which dynamically converts the state of the 3D agent in the augmented reality environment using the 3-Axis acceleration sensor in the smart phone. To control the state of the agent to which the artificial intelligence is applied, users used to directly enter the data or use markers to detect them. The critical values, which were determined via test, were given to the acceleration sensor to ensure accurate state conversion. In this paper, markerless tracking technology was used to implement the augmented reality, and the state of the agent was dynamically converted using the 3-Axis acceleration sensor.

■ keyword : | Mobile | Augmented Reality | Artificial Intelligence |

* 본 연구는 2010년 호서대학교 재원으로 학술연구비 지원을 받아 수행된 연구임 (2009-0627)

접수번호 : #100812-001

접수일자 : 2010년 08월 12일

심사완료일 : 2010년 10월 01일

교신저자 : 박성준, email : sjpark@hoseo.edu

I. 서론

오늘날의 모바일기기들은 더 이상 통신을 위한 수단만이 아니라 여러 부가적인 기능들이 포함되면서 휴대용 기기로서의 많은 이점들을 보여주고 있고 이와 관련된 많은 응용 연구가 진행되고 있다. 특히 스마트폰의 GPS와 고성능 카메라의 기능은 모바일기기에서 실시간 증강현실을 가능하게 하는 주요 요소이다. 대표적인 응용 어플리케이션으로서는 위치정보를 사용자에게 전달해주는 Layer 등이 있다. 기존의 대다수의 증강현실 어플리케이션들은 카메라로 얻은 실시간 이미지 정보와 GPS를 사용하여 얻은 위치정보를 보여주는 Layer 어플리케이션과 유사한 방식을 사용하고 있다.

현재 증강 현실의 에이전트들이 3차원 객체를 사용하면서 좀 더 현실감 있는 환경을 제공해주고 있다. 특히 증강 현실 환경에서 3차원 에이전트의 중요성이 커지고 있는 추세이며 좀 더 현실감 있게 움직임을 표현하기 위한 그래픽 응용 기술들이 많이 연구되고 있다. 3차원 에이전트의 지형 충돌, 그림자 생성등과 같은 것이 좋은 보기라고 할 수 있다. 그러나 3차원 에이전트의 현실감 있는 시각화 기술이 발전하고 있음에도 불구하고 정작 게임에 적용된 사례는 아직까지 이루어지고 있지 않다. 특히 게임에서 중요한 기능 중의 하나인 인공지능은 증강 현실 기반에서 현장감을 높이는 중요한 기능이지만 몇몇 로봇공학 분야에서 적용된 이외에는 아직까지 특별하게 연구되고 있지 않다.

본 논문에서는 증강현실과 게임 인공지능을 활용한 응용 기술을 제안한다. 증강 현실 구현에 있어서 본 논문에서 제안한 방법은 특징 점 추출 기반의 3차원 월드 생성 방법을 사용하였다. 생성된 월드는 인공지능이 적용된 에이전트에서 사용하고 있으며, 스마트폰의 3축 가속도 센서를 활용하여 3차원 에이전트의 인공지능 상태를 동적으로 변환할 수 있도록 하였다. 3차원 월드의 생성은 스마트 폰에서 실시간으로 하기에는 많은 비용이 소요되기 때문에 실제 구현된 시스템의 구조를 준비단계와 실행단계로 나눈 뒤 준비 단계에서 처리할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 증강현실

관련 응용기술을 소개하고, 3장에서는 본 논문에서 제안한 알고리즘을 제시한다. 4장에서는 제안된 알고리즘에 대한 실험을 소개하고 마지막으로 결론 및 향후 과제를 설명한다.

II. 관련 연구

최근 모바일 기반의 증강현실 어플리케이션들은 대부분이 스마트폰에서 구현되었으며 이는 카메라로 얻은 이미지 위에 원하는 위치정보를 그려주는 방식을 사용하고 있다. 이러한 방식을 레이어 방식이라고 부른다. [그림 1]은 레이어 방식을 사용한 대표적인 증강현실 어플리케이션이다.

그림 1. 증강현실 어플 "Layer"

레이어 방식은 사용자가 쉽게 구현할 수 있어서 많은 사용자가 증강현실 어플리케이션을 개발할 때 주로 사용하는 방식이다. 그러나 레이어 방식은 증강 현실의 주요한 3가지 요소인 현실과 가상의 정합, 상호 작용, 3D 구현을 완벽하게 지원하기가 힘든 방식이다. 레이어 방식을 사용하지 않고, 보다 정교한 증강 현실을 구현하기 위해서는 마커 추적 기술[1]이나 마커리스 추적 기술[2]을 사용한다. 마커 추적 기술은 불필요한 정보인 마커가 사용자에게 노출되는 단점이 있지만 쉽고 정확성 있게 현실과 가상을 정합시켜 준다. 그리고 마커리스 추적 기술은 마커 추적 기술의 단점을 해결하기 위하여 진행된 연구의 결과물로서, 카메라로 얻은 영상들의 특징 점을 추출하고 추출된 특징 점들을 사용하여

현실에 가상을 정합하기 위한 연산을 수행한다.

마커리스 추적 기술은 SURF(Speeded Up Robust Features) 알고리즘[3]이나 SIFT(Scale Invariant Feature Transform) 알고리즘[4] 등을 사용하여 특징점을 추출할 수 있다. 대부분 특징점 추출 시 영상의 크기 변화와 회전에 불변하는 특성을 가지는 SIFT 알고리즘을 사용한다. 보다 정교하게 특징점을 추출할 수 있다는 장점이 있지만 연산량이 많다는 단점이 있다. SURF 알고리즘은 SIFT의 단점을 개선한 알고리즘으로서 정확성은 SIFT 보다 떨어지지만 빠르게 증강 현실을 표현할 수 있어서 모바일 환경에서 사용하기에 적합한 알고리즘이다. 마커 추적 기술과 마커리스 추적 기술은 모바일 기기에서 제약된 사항이 많다. 이는 추적 기술의 연산량이 모바일기기의 성능을 오버하기 때문이다. 이를 해결하기 위하여 제안된 연구들은 서버-클라이언트의 구조를 가지고 있다[5]. 이러한 구조는 모바일기기를 단순히 영상의 획득과 증강된 결과물을 보이기 위한 도구로 사용하며 증강을 위하여 모바일기기에서 전달된 영상을 서버에서 연산하는 방식을 사용한다.

기존의 증강현실과 관련된 응용 기술 중 게임에서 사용되는 인공 지능과 결합된 연구들은 이루어지지 않고 있다. 또한 게임에서 사용되고 있는 인공지능에 동적인 요소가 있는 연구들은[6][7] 주로 에이전트의 추적, 경로 탐색, 게임 환경등에 대하여 연구되어졌으며 동적 상태 변환에 관한 연구는 이루어지지 않았다.

본 논문에서는 마커리스 추적 기술 중 하나인 SURF 알고리즘을 사용하여 증강 현실 구현을 처리했으며 실시간 입력되는 3축 가속도 센서를 통해 3차원 에이전트의 상태를 동적으로 변환하는 알고리즘을 개발하였다.

III. 증강현실 기반의 인공지능 시스템

1. 시스템 구조

본 논문에서 제안하는 시스템은 크게 전처리 단계와 실행 단계로 구분된다. 전처리단계에서는 3차원 에이전트가 사용할 3차원 가상 월드를 구성하는 역할을 수행한다. 실행 단계에서는 전처리 단계에서 만들어진 가상

월드 기반에서 실시간으로 입력되는 영상 처리를 통해 증강 현실을 구현하고 3축 가속도 센서의 입력에 따라 동적으로 에이전트의 상태가 변환하는 인공지능을 수행한다. [그림 2]는 이러한 과정을 도식화한 것이다.

그림 2. 시스템 처리

2. 준비 단계에서의 특징점 추출 및 월드 생성

특징점의 추출은 준비단계와 실행단계에서 모두 수행한다. 전처리 단계에서 특징점을 추출하는 목적은 가상의 월드를 생성하기 위해서이고, 실행 단계에서는 마커리스 추적 기술을 위해서이다. 특징점의 추출에는 SURF 알고리즘이 사용되었다. 대부분 특징점 추출에는 영상의 회전, 스케일 변화에 강건한 SIFT 알고리즘이 사용되지만 연산량이 많기 때문에 속도 개선 버전인 SURF를 사용하였다. [그림 3]은 두 가지의 특징점 추출 알고리즘을 사용하여 특징점을 추출한 결과 화면이다.

SURF를 사용한 방법(왼쪽) SIFT를 사용한 방법(오른쪽)보다 특징점이 덜 추출되는 것을 알 수 있다.

그림 3. 특징점 추출 결과화면
(a) SURF (b) SIFT

3D 월드를 생성하기 위하여 우선 카메라 영상으로부터 추출한 특징점들을 모아서 가상공간을 생성하였다.

생성된 가상공간은 특징점들의 정보를 포함하는 평면이며, 자세를 추정하기 위하여 POSIT(Pose from Orthography and Scaling with Iteration) 알고리즘을 사용하였다. POSIT알고리즘으로 2차원 평면상의 정보와 이에 대응되는 3차원상의 정보를 가지고 객체의 회전과 이동정보를 구할 수 있으며, 본 논문의 시스템에서는 3차원상의 정보를 임의로 지정하여 사용하고 있다.

가상공간의 자세를 추정한 후에는 RANSAC 알고리즘[8]을 사용하여 3D 월드의 기준이 될 좌표계를 생성하며, 좌표계를 기준으로 에이전트가 사용된 3D 월드를 생성한다. [그림 4]는 전처리 단계를 거쳐서 3D 월드가 생성되는 과정을 나타낸 그림이다.

그림 4. 전처리 단계
(a)입력된 영상 (b)특징점 추출 (c)좌표계 생성 (d)생성된 월드에 에이전트가 증강된 모습

3. 실행 단계에서의 증강현실 구현

실행 단계에서는 전처리 단계에서 생성된 3D 월드를 사용하여 증강현실을 구현하고 있으며, 마커리스 추적 기술을 사용하여 3D 월드와 증강된 에이전트의 기하변환을 수행하고 있다. 기하변환을 위해서 실시간으로 입력되는 영상들의 특징점을 추출한다. 새롭게 추출된 특징점을 이전 영상의 특징점과 매칭을 하여 추적에 적합한 특징점들을 선별한다. 그리고 선별된 특징점으로 옵티컬 플로우 방식의 추적 기술[9-11]을

사용하여 변환행렬을 생성해준다. 생성된 변환행렬을 3D 월드와 증강된 물체에 적용하여 기하변환을 시켜준다. [그림 5]는 앞서 설명한 기하변환을 그림으로 보여주고 있다.

그림 5. 월드의 기하변환
(a) 입력된 영상 (b) 특징점 추출
(c), (d) 변환행렬이 적용된 모습

4. 상태 변환 알고리즘

본 논문에서 제안하는 알고리즘은 애플사의 아이폰을 기반으로 구현하였고 자체 내장 기능인 3축 가속도 센서(3-Axis Accelerometer Sensor)를 사용하였다. 기존의 아이폰 가속도센서는 자동회전, 게임 상의 객체를 제어하기 위한 입력장치로서 사용되었지만 본 논문에서는 에이전트의 상태를 변환하는데 사용되고 있다.

가속도 센서는 x, y, z의 세 방향에 대한 가속도의 크기를 측정할 수 있기 때문에 3차원 공간에서의 움직임과 중력을 감지 할 수 있다. 가속도 센서의 반환 값이 1.0이면 특정 방향으로 1G의 힘이 작용하고 있다는 것이며 1G는 자연 상태의 중력가속도를 나타낸다. 예를 들어 가속도 센서의 값이 $[x:0.0, y:-1.0, z:0.0]$ 으로 측정되었다면 홈 버튼이 하단에 위치하며 아이폰이 세워져 있는 상태인 것을 알 수 있다. 3축 가속도 센서를 사용하여 변환해야 할 상태들은 5가지로 [그림 6]과 같이 대기, 이동, 점프, 전투, 회복으로 나뉜다.

그림 6. 상태 전이 다이어그램

에이전트의 상태를 동적으로 변환하기 위해서는 3차원 가상 환경에서 아이폰의 현재 상태를 알아야 한다. 아이폰의 상태를 알기 위해서는 가속도 센서의 값을 측정하고, 측정된 가속도 센서의 값을 각 상태의 임계값과 비교해야 한다. 임계값은 아이폰의 각 상태에 따라 측정된 가속도 센서의 값들을 식(1)과 식(2)를 사용하여 최대값과 최소값을 구한 후에 다른 임계값들과 겹쳐지지 않는 값으로 정한다.

$$Threshold_{\alpha} = MAX\{A_x, A_y, A_z\} \quad (1)$$

$$Threshold_{\beta} = MIN\{A_x, A_y, A_z\} \quad (2)$$

각 상태에 따라 측정된 가속도 센서의 값은 [그림 7]에 나타내었으며 [그림 7]의 가속도계 값과 식(1), 식(2)를 사용하여 각 상태에 따라 측정된 임계값을 [표 1]에 정의하였다.

그림 7. 상태별 3축 가속도계 값

표 1. 상태별 임계값

상태 \ 축	X	Y	Z
대기	0.0 ~ 0.02	-0.95 ~ -0.96	-0.06 ~ -0.09
이동	0.02 ~ 0.05	-0.79 ~ -0.95	-0.09 ~ -0.6
점프	$ X > 1.5$	$ Y > 1.5$	0.05 ~ -1.0
전투	0.0 ~ -0.49	-0.83 ~ -0.96	-0.09 ~ -0.18
회복	0.05 ~ 0.43	-0.83 ~ -0.96	-0.1 ~ -0.17

대기상태는 아이폰이 세워져 있으며 홈 버튼이 하단에 위치할 경우를 나타낸다. 이때 움직임이 없어서 0으로 출력되어야 할 가속도 센서의 절대값은 실제로는 0보다 크지만 1보다는 작은 미세한 값들을 갖는다. 이동상태는 대기 상태에서 스마트폰을 뒤로 기울인 상태이다. 이동시 z축의 값이 급격하게 변하는 것을 확인할 수 있다. 점프상태는 스마트폰을 위 아래로 흔들었을 때의 상태를 나타내며 스마트폰을 세게 흔들었을 경우에 스마트폰을 흔든 방향에 따라서 1.0을 넘는 큰 값이 들어오는 것을 그림에서 확인할 수 있다. 전투 상태는 대기 상태에서 스마트폰을 왼쪽으로 기울인 상태를 말하며 이 때 x축의 값이 음의 값인 것을 확인할 수 있다. 회복 상태는 대기 상태에서 스마트폰을 오른쪽으로 기울인 상태를 말한다. 전투상태와는 반대이며 x축의 값이 양의 값인 것을 확인할 수 있다.

[표 1]의 임계값을 사용하여 아이폰의 현재 상태를 판정한 후에는 [그림 6]의 상태 전이 다이어그램에 따라서 에이전트의 상태를 변경해주면 된다. 이러한 상태 변환 알고리즘을 간략하게 나타낸 것이 [그림 8]이다.

```

가속도계 값의 획득
//아이폰의 상태 값 획득
IF(대기 임계값 = 가속도계 값) 상태 값 = 대기;
IF(이동 임계값 = 가속도계 값) 상태 값 = 이동;
IF(점프 임계값 = 가속도계 값) 상태 값 = 점프;
IF(전투 임계값 = 가속도계 값) 상태 값 = 전투;
IF(회복 임계값 = 가속도계 값) 상태 값 = 회복;

IF(상태 변환 가능) 현재 상태 = 아이폰의 상태 값
ELSE 현재 상태 유지
  
```

그림 8. 상태변환 의사코드

기존의 상태 변환 알고리즘들은 사용자가 직접 입력하여 에이전트의 상태를 제어하는 반면에 본 논문에서

제안된 알고리즘은 가속도계의 값을 사용하여 에이전트의 상태를 동적으로 변환하고 있다. 또한 기존의 알고리즘들이 일정한 프레임마다 실행되는 반면에 제안된 알고리즘은 0.1초마다 발생하는 가속도계의 값을 획득하여 에이전트의 상태를 변경한다. 그리고 제안된 알고리즘은 가속도 값이 발생되기 위한 약간의 아이폰 조작과 방향 설정 등을 위한 터치입력등 사용자의 간단한 동작을 요구하기 때문에 한 손으로 쉽게 조작을 할 수 있다. 이는 사용자의 상태와 위치정보들을 사용하는 체감형 콘텐츠에서 다른 조작법보다 편리하게 사용될 것으로 기대하고 있다.

IV. 실험 및 분석

본 논문에서는 신뢰성있는 에이전트의 상태 변환을 위하여 상태변환 알고리즘을 실험하였으며 정확도를 측정하였다. 실험에 사용된 모바일 기기는 iPhone 3GS이며, 어플리케이션은 iMac(Mac OS X Snow Leopard)에서 xCode를 사용하여 개발되었다. 실험을 위하여 상태변환 조건을 [그림 9]에 20가지의 경우로 정의하였으며 정의된 조건들은 W(대기), M(이동), J(점프), B(전투), R(회복) 등 5가지의 상태를 조합하여 정의하였다.

상태변환 조건			
1	W - M - J - B - R - W	11	W - J - R - M - B - M
2	W - M - B - J - R - M	12	W - M - B - J - R - W
3	W - J - M - B - R - W	13	W - M - R - B - J - M
4	W - M - R - B - J - W	14	W - J - B - M - R - M
5	W - M - J - R - B - R	15	W - M - J - B - R - W
6	W - M - B - R - M - J	16	W - M - W - J - R - B
7	W - J - R - B - M - B	17	W - J - B - R - M - J
8	W - J - M - R - B - R	18	W - M - R - W - J - B
9	W - M - W - J - B - R	19	W - M - R - W - B - J
10	W - M - B - J - W - R	20	W - M - B - R - W - J

그림 9. 상태변환 조건

[그림 10]은 [그림 9]의 조건에 따라 실험한 결과를 나타내고 있다. Theoretical Value는 상태변환 조건에 따라서 각 상태들이 호출되어야 할 횟수를 나타내며

Experimental Value는 상태변환 조건을 기반으로 실험을 하여 각 상태들이 실제로 호출된 횟수를 나타낸다.

그림 10. 상태변환 호출횟수

그림 11. 상태변환 호출횟수의 합과 평균

[그림 11]은 [그림 10]에서 Theoretical Value와 Experimental Value의 호출횟수의 합과 평균을 나타낸 것이다. 그림을 보면 Experimental Value가 전체적으로 Theoretical Value보다 호출된 횟수가 적다는 것을 확인할 수 있다. 이는 각 상태들의 호출이 제대로 이루어지지 않았기 때문이다. 실험 결과를 분석하여 이러한 결과가 나타난 원인을 알 수 있었으며, 원인은 다른 상

태에서 대기상태로의 전환이 제대로 이루어지지 않았다는 것이다. 특히, 실험의 결과 값 중에서 조건 9의 경우를 보면 M에서 W로의 변환이 제대로 수행되지 않았기 때문에 W 다음에 변환되어져야할 상태들이 호출되지 않았다.

본 논문에서 제안된 알고리즘은 실험과 실험 결과의 분석을 통하여 대기상태의 임계값 범위가 적절하지 않았기 때문에 상태 변환이 제대로 수행되지 못하였으며 Theoretical Value값과 Experimental Value값이 서로 틀린 결과를 보인 것을 확인하였다.

V. 결론 및 향후연구

본 논문에서는 마커리스추적기술을 사용하여 증강현실을 구현하였으며 아이폰의 3축 가속도 센서를 사용하여 에이전트의 상태를 동적으로 변환시켰다. 동적인 에이전트의 인공지능 상태는 실시간으로 입력되는 스마트폰의 3축 가속도 센서의 값을 측정하여 스마트폰의 움직임의 상태에 따라 실험 결과로서 얻은 임계값과의 매칭을 통해 이루어진다. 특히 실시간으로 입력되는 3축 가속도 센서의 값은 정지 상태에서도 미세한 값들로 측정되기 때문에 실험을 통해 얻은 임계 범위에 따라 임계값을 설정 하여 대기, 이동, 점프, 전투, 회복 등 5가지의 인공지능 상태를 증강현실 기반에서 실험하였다. 실험 결과 대기 상태 이외의 상태에서 대기 상태로의 전환이 정확히 이루어지지 않는다는 것을 알 수 있었다. 향후과제로서는 대기상태의 임계값을 보정하여 정확한 상태 변환이 이루어지도록 할 것이며 본 논문에서 제안된 알고리즘에서 인공지능을 위하여 유한 상태 머신 알고리즘을 사용하고 있지만 향후에는 게임에서 사용되는 인공지능 기술을 증강현실 기반에서 다양하게 적용 할 것이다. 또한 3축 가속도 센서를 사용하여 측정된 사용자의 걷기, 뛰기 등의 상태와 위치정보시스템을 사용하여 얻어낸 이동속도를 연구에 반영하여 좀 더 동적인 환경에서 인공 지능을 이용한 응용 기술을 연구 개발 할 것이다.

참 고 문 헌

- [1] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," IWAR99, pp.85-95, 1999.
- [2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," ISMAR, pp.225-234, 2007.
- [3] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," European Conference on Computer Vision, Vol.3951, pp.404-417, 2006.
- [4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," IJCV, Vol.60, No.2, pp.91-110, 2004.
- [5] 홍동표, 우운택, "모바일 증강 현실 시스템에 대한 연구 동향", 정보과학회지 제26권, 제1호, pp.5-14, 2008.
- [6] 권오익, 황보택근, "3D게임에서 이동 장애물을 고려한 동적 경로 탐색 기법", 한국게임학회 논문지, 제6권, 제3호, pp.3-12, 2006.
- [7] 유한하, 조경은, 엄기현, "실시간 동적인 환경에서 다중 에이전트의 협동 기법", 한국게임학회 논문지, 제6권, 제3호, pp.13-22, 2006.
- [8] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," Comm of the ACM, Vol.24, pp.381-395, 1981.
- [9] T. Lee and T. Höllerer, "Hybrid Feature Tracking and User Interaction for Markerless Augmented Reality," Proc. IEEE Conf. Virtual Reality (VR '08), pp.145-152, 2008.
- [10] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.

- [11] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," International Joint Conference on Artificial Intelligence, pp.674-679, 1981.

저 자 소 개

장 유 나(Yu-Na Jang)

준회원

- 2008년 2월 : 호서대학교 게임공학(공학사)
- 2009년 3월 ~ 현재 : 호서대학교 컴퓨터공학과(공학석사)

<관심분야> : 증강현실, 게임, 인공지능

박 성 준(Sung-Jun Park)

정회원

- 1997년 2월 : 호서대학교 컴퓨터공학과(공학사)
- 1999년 2월 : 건국대학교 컴퓨터공학과(공학석사)
- 2005년 2월 : 건국대학교 컴퓨터공학과(공학박사)

- 2006년 3월 ~ 현재 : 호서대학교 게임공학과 조교수
- 현재 : 한국콘텐츠학회 정회원
- 현재 : 한국게임학회 정회원
- 현재 : 한국컴퓨터정보학회 정회원

<관심분야> : 게임, 가상현실, HCI, Computer Vision