

증강현실 게임에서의 동작 기반 상호작용 설계 및 구현*

박종승[○], 전영준*

인천대학교 컴퓨터공학과^{○*}

jong@incheon.ac.kr, 0961144@incheon.ac.kr

Design and Implementation of Motion-based Interaction in AR Game

Jong-Seung Park[○], Young-Jun Jeon*

Dept. of Computer Science and Engineering, University of Incheon

요 약

본 논문에서는 증강현실 게임에서의 동작 기반 상호작용의 설계 및 구현 기법을 기술한다. 증강현실에서는 상호작용의 도구로 고전적인 입력 장치보다 사용자의 동작에 의존한다. 동작 기반 증강현실 게임은 신체적 움직임에 의한 상호작용을 사용하는 실감 게임의 유망한 분야이나 현재의 동작 인식 기술의 불안정성으로 인해서 증강현실 응용에서 동작 인식 장치를 주 입력 장치로 사용하는 데에는 한계가 있다. 이러한 한계에 대처하기 위해서 각 증강현실 응용들은 여러 보완수단들을 동원하고 있고 이로 인해서 증강현실 게임의 구조가 불규칙적이며 비효율적으로 설계되는 경향이 있다. 본 논문에서는 동작 기반 증강현실 게임의 효율적인 개발 방법론을 제시한다. 실제 탁구채를 사용하는 동작 인터페이스를 가지는 증강현실 탁구게임의 프로토타입을 제시된 방법론에 따라서 설계하고 이를 구현하였다. 구현된 증강현실 게임에서의 상호작용에 있어서 탁구채를 고속으로 인식하고 지연 없이 상호작용 됨을 확인하였다.

ABSTRACT

This article proposes a design and implementation methodology of a gesture-based interface for augmented reality games. The topic of gesture-based augmented reality games is a promising area in the immersive future games using human body motions. However, due to the instability of the current motion recognition technologies, most previous development processes have introduced many ad hoc methods to handle the shortcomings and, hence, the game architectures have become highly irregular and inefficient. This article proposes an efficient development methodology for gesture-based augmented reality games through prototyping a table tennis game with a gesture interface. We also verify the applicability of the prototyping mechanism by implementing and demonstrating the augmented reality table tennis game. In the experiments, the implemented prototype has stably tracked real rackets to allow fast movements and interactions without delay.

Keyword : Motion-based Interface, Feature Tracking, Visual Interface

접수일자 : 2009년 07월 17일

심사완료 : 2009년 08월 07일

* 본 연구는 지식경제부 지방기술혁신사업(RTI05-03-01)의 지원으로 수행되었음

1. 서 론

증강현실(AR: Augmented Reality) 게임은 사용자의 신체적인 움직임에 따라 감각적으로 상호작용할 수 있는 유망한 게임 기술이다. 최근 AR 게임의 잠재력을 보여주는 여러 프로토타입들이 제안되고 있다. AR 게임에서는 키보드나 조이스틱과 같은 고전적 입력 기기가 아닌 신체적 움직임에 의한 입력 방법을 사용하여 사용자의 실 환경과 가상의 게임 환경을 융합하기 위한 다양한 시도를 하고 있다. 사용자의 실 환경에서의 동작에 따른 게임 인터페이스의 테스트베드가 제시되고 있다[1].

기존의 게임들을 AR화한 AR 게임을 구현하는 것은 다양한 방법으로 가능하다. 그 예로 테이블 상에서의 자동차 경주 게임이나 볼링 게임 등이 있다[2]. Sony사의 PlayStation3용 카드 게임인 "Eye of Judgement"는 2008년 출시된 AR 게임으로 특정 카드를 인식해서 테이블 보드 상에서 3차원적으로 가상 캐릭터들을 보여준다[3]. 또한 제한된 실내 공간에서 사용자의 움직임을 허용하는 AR 게임도 있다[4]. 이 방법에서는 천장에 많은 마커들을 부착하고 이들을 검출하여 사용자의 자세를 인식하고 사용자의 손 제스처를 인식하여 상호작용한다. 보다 넓은 실외 공간에서의 AR 게임의 예로, 한 고전 게임인 "Pacman"의 AR 버전인 "Human Pacman"이라는 실외용 체험 게임도 제안되었다[5]. 이 게임은 GPS와 관성센서 등을 사용하여 사용자의 자세를 인식한다. 사람들이 실제로 실외 환경에서 사회적인 상호작용할 수 있는 RPG의 형태를 갖춘 게임이다. 또한 "Second Life"와 같은 많은 사용자가 온라인으로 동시 접속하는 MMO(massively multiplayer online) 게임에 대해서도 AR 기술이 적용될 수 있음이 제시되었다[6].

동작 기반 인터페이스는 상호작용의 강력한 수단으로 AR 게임에서 대표적으로 사용될 수 있는 기법이다. 그러나 사람의 손동작의 자유도는 매우 크기 때문에 손동작을 정확히 인식하는 문제는 매

우 어렵다[7]. 손동작을 안정적이고 정확히 인식할 수 있어야 동작 기반 인터페이스를 AR 게임에 활용할 수 있다. 동작 인식은 전적으로 시각적인 인식에 의존할 수도 있지만 그 대안 또는 보완책으로 움직임 센서를 도입할 수 있다. 이 경우에는 센서 장치의 착용으로 인한 사용자의 불편함을 감수해야 한다. 또한 상호작용은 센서가 지원하는 기능에 한정되어서 자유로운 상호작용의 구현에 어려움이 있다.

스포츠 장르의 컴퓨터 게임에서 AR 기술을 사용하는 것은 큰 장점들을 가진다. 가장 큰 장점은 사용자의 신체적 측면과 정신적 측면을 일치시켜서 게임의 몰입감을 높일 수 있는 것이다. 본 논문에서는 사용자의 실 환경에서 플레이되는 AR 스포츠 게임을 디자인하는 방법에 대해서 다룬다. 카메라 이외의 부가적인 입력 센서는 사용하지 않는다고 가정한다.

패들(paddle) 게임 장르는 패들을 사용하여 공을 제어하는 인기 있는 게임 부류이다. 이 장르는 탁구 게임, 테니스 게임, 벽돌깨기 게임 등의 많은 스포츠 게임들을 포함한다. 이들 게임에서는 패들과 공의 물리적인 상호작용이 중요한 역할을 한다. 공은 패들뿐만 아니라 벽, 탁구대, 네트 등과 같은 물체와도 충돌될 수 있다. 어떤 패들 게임은 실제 공과 실제 라켓을 사용하며[8,9], 어떤 패들 게임은 실제 라켓을 사용하지만 가상의 공을 사용하기도 한다[10]. 패들 게임 장르에서 탁구 게임은 가장 상호작용적인 게임으로 알려져 있다[11]. 본 논문에서 제안하는 AR 게임 프로토타이핑 디자인 방법의 예로 탁구 게임의 경우에 대한 디자인 및 구현을 진행하였다.

패들의 움직임의 인식을 위해서 센서를 포함하는 패들 트래커 장치를 사용할 수도 있지만 이는 장치의 부착으로 인한 이물감, 센싱 정보의 한계, 움직임 반경의 제약, 경제성 등의 측면에서 적절하지 않다. 본 논문에서는 카메라를 사용한 패들의 움직임 인식에 대해서 다룬다. 패들의 자세나 특정한 움직임은 입력 장치에서의 입력 명령어에 대응

되도록 바뀌어 AR 게임 인터페이스에 입력된다. 사용자의 움직임은 임의적일 수 있으므로 어떠한 움직임도 해석되고 처리될 수 있도록 디자인해야 한다. 또한 동작 인식은 의도된 제스처와 우연한 신체 움직임을 구별할 수 있어야 한다. 우연한 움직임으로부터는 상호작용이 유발되지 않도록 해야 하며 의도된 움직임으로부터만 연관된 입력 명령어를 인터페이스 시스템으로 전달하도록 해야 한다. 동작 인식 알고리즘은 안정적이어야 하며 또한 다양한 사용자에 대해서 올바르게 동작해야 한다. 예상치 못한 주변 환경이나 움직임에 대해서도 올바르게 동작해야 한다.

2장에서는 AR 탁구 게임의 경우에 있어서의 동작 기반 프로토타이핑 기법에 대해서 설명하고, 3장에서는 AR 탁구 게임에서의 휴먼 플레이어 동작의 인식 기법에 대해서 다룬다. 4장에서는 실제 구현 기법과 실험 결과에 대해서 설명하고 5장에서 결론을 맺는다.

2. AR 게임 프로토타이핑

2.1 동작 인식의 단서들

AR 응용에서 동작 인식에 적용될 수 있는 단서들이 있다. 첫째, 응용에 따른 맥락(context)이 시각적 해석의 다양성을 제한시킬 수 있다. 이는 사람의 동작 해석에 대한 자유도를 충분히 감소시킬 수 있으므로 문제의 해결이 가능하다. 또 하나의 유의한 단서로 AR 응용은 즉각적인 피드백을 사용자에게 제공하므로 사용자는 주어지는 피드백을 활 단서로 사용자의 동작을 자신의 의도대로 맞추어갈 수 있는 점용자에게 러한 속성들은 시각적인 인식을 사용하는 상호작용을 AR 응용에 활 단서 문제를 쉽게 한다. 또한 대부분의 의 자신의 의에서의 플레이어의 동작의 종류가 제한되어 있고 또한 고급 지식들이 거의 필요하지 않는다는 점도 활용 문제를 쉽게 한다. 동작에 필요한 최소한의 정보만 인식하면 되기 때문이다. 이러한 단서들은 자유도

가 높은 실 환경에서도 게임의 상호작용에 필요한 정보를 획득할 수 있도록 한다.

AR 응용의 개발에 있어서 가장 중요한 문제는 대상 물체 추적의 정확성이다[12]. 물체의 정확한 추적을 위해서는 캡처되는 영상에서 물체의 외형이 선명하게 구분되도록 함이 중요하다. 물체의 외형에 대한 가정을 하지 않는 경우에는 관심 물체의 빠른 탐지가 어렵고 인식 절차에서 애매성이 존재한다. 따라서 범용 컴퓨팅 환경에서 실시간 추적이 가능한 구현을 위해서는 관심 물체의 외형에 대한 충분한 제한이 필요하다.

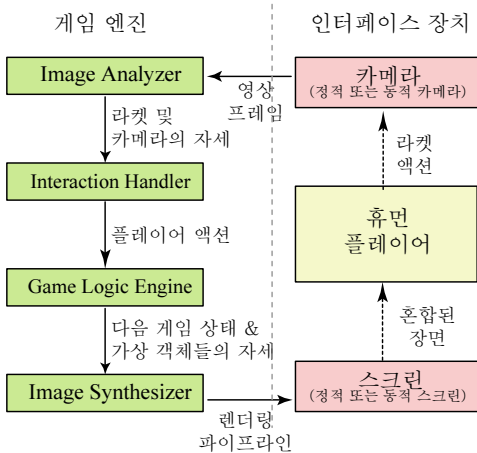
실제 라켓을 사용하는 AR 탁구 게임의 경우에는 플레이어의 동작 인식을 위해서 라켓을 인식해야 한다. 라켓 위치의 탐색에 있어서 탁구 라켓이 일반적으로 붉은색의 원색이라는 점이 큰 단서가 된다. 특정 색만을 필터링하여 탐색 후보 위치를 빠르게 찾고 또한 그 범위를 상당히 제한시킬 수 있다. 라켓 인식의 다른 방법으로 흑백 마커를 패들에 부착하는 방법이 있다. 평면 흑백 마커를 사용하여 패들 자세의 인식에 있어서 정확성을 유지하면서 고속 처리가 가능한 방법이다.

2.2 AR 탁구 게임의 프로토타이핑

본 절에서는 상에서 열거한 단서들과 제약들을 적용하여 AR 탁구 게임에서 강건한 제스처 인터페이스의 프로토타이핑 기법을 제시한다. 탁구 게임은 한 테이블 상에서 라켓을 사용하여 공을 움직이는 게임이다. AR 탁구 게임에서는 실제 공 대신 가상의 공을 사용한다. 본 논문에서는 실제 테이블도 가상의 테이블로 대체한다. 이는 테이블이 없는 임의의 공간에서도 탁구 게임이 가능하도록 한다. 라켓의 경우에도 가상의 라켓을 사용할 수 있지만 본 연구에서는 실제 라켓을 사용하여 탁구 게임의 실감을 유지하도록 하였다.

AR 탁구 게임의 전체적인 블록 다이어그램이 [그림 1]에 있다. 플레이어의 라켓 액션은 카메라로 캡처되고 캡처된 영상 프레임은 영상 분석기(image analyzer)로 보내진다. 영상 분석기는 라켓

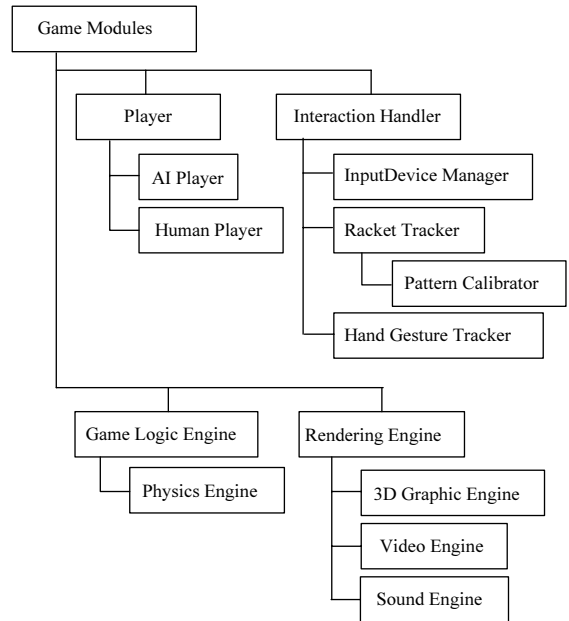
의 속성값들을 검출하고 이를 추적하여 라켓의 위치와 방향을 추정한다. 상호작용 처리기(interaction handler)는 인식된 라켓 및 카메라의 자세로부터 플레이어의 동작을 결정하고 정한다.따른 액션을 발생시킨다. 게임 로직 엔진(game logic engine)은 플레이어의 액션을 받아들여서 새로운 게임 상태를 결정하고 또한 각 게임 객체들의 모든 물리 현상을 계산하여 가상 객체들의 새로운 상태들을 결정한다. 가상 객체들의 새로운 속성값들은 영상 합성기(image synthesizer)로 전송되어서 가상 객체들과 실제 영상의 합성이 이루어지도록 한다. 영상 합성기는 모든 가상 객체들을 렌더링하고 합성된 장면을 스크린으로 출력한다.



[그림 1] AR 탁구 게임의 전체적인 다이어그램

제안된 게임 시스템은 [그림 2]와 같이 기능적으로 분리된 여러 단위 모듈들을 가지고 있다. Player 모듈은 게임 규칙에 따라서 각 플레이어의 상태를 제어한다. InteractionHandler 모듈은 플레이어와의 모든 상호작용들을 관리한다. 이 모듈은 플레이어의 라켓 액션을 검출하는 RacketTracker 하부 모듈을 포함한다. GameLogicEngine은 사실적인 물리 시뮬레이션 기능을 제공한다. 게임 세계에서의 물리 시뮬레이션은 플레이어에 대한 주관적인 표현이 목적이므로 실세계에서의 물리 현상과

정확히 일치하지 않아도 된다. 게임 세계에서의 물리 현상들은 뉴턴 물리학과 강체 역학의 간단한 이론들로 충분히 모델링될 수 있다. Rendering Engine은 가상 객체들을 합성하고 비디오 프레임으로 출력한다.



[그림 2] AR 탁구 게임의 시스템 모듈들

2.3 게임의 규칙과 상태의 정의

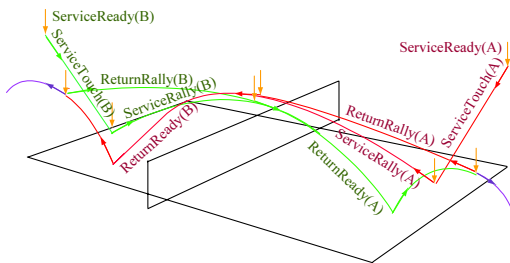
게임 시스템의 상태는 게임의 규칙으로부터 결정된다. 탁구 게임의 규칙으로부터 게임의 상태를 정의하는 방법을 알아보자. 탁구 게임의 규칙은 단순하다. 탁구 게임은 공을 서비스하는 단계와 리턴하는 단계로 구분된다. 게임은 먼저 서비스 단계로부터 시작된다. 서비스 단계에서는 플레이어가 라켓으로 공을 쳐서 그 공이 테이블의 네트 안쪽 절반 영역에 단 한번만 부딪혀야 하고 이후에 테이블의 상대방 쪽 절반 영역에 한번 이상 부딪혀야 한다.

서비스 단계 이부터는 각 플레이어는 리턴 단계를 수행한다. 리턴 단계에서는 플레이어가 공을 쳐서 그 공이 테이블의 상대방 쪽 절반 영역에 한번

이상 부딪히도록 해야 한다. 한 플레이어가 실수를 행하면 스코어 점수가 상대 플레이어에게 주어지고 게임은 서비스 단계부터 다시 시작한다.

이러한 게임의 규칙으로부터 게임 상태를 정의해야 한다. 게임의 흥미와 빠른 진행을 위해서 탁구의 공식 규칙을 그대로 적용하기 보다는 단순화된 규칙을 사용하였다. 예를 들어, 점수를 획득한 쪽이 항상 다음번의 서비스를 하도록 하였고 서비스 시의 요건에는 까다로운 제한을 두지 않았다.

단순화된 게임 규칙을 사용하면 게임 상태는 공의 현재의 상태에서부터 결정될 수 있다. 각 플레이어에 대해서 5개씩의 상태를 정의하여 총 10개의 상태를 정의하였다. [그림 3]은 각 플레이어에 정의되는 5개의 상태를 보여준다. ServiceReady는 서비스를 하기 전의 대기 상태이고, ServiceTouch는 서비스를 한 후에 공이 테이블 면에 닿을 때까지의 상태이고, ServiceRally는 공이 테이블에 부딪친 후에 이동하는 상태이다. 서비스된 공이 네트를 지나면 서비스한 플레이어는 ReturnReady 상태로 되어 되돌아올 공의 리시빙을 위해 대기하게 된다. 되돌아온 공을 친 경우에는 상태가 ReturnRally로 바뀌게 되고 그 공이 네트를 넘어가면 다시 ReturnReady 상태로 된다. 상대 플레이어에 대해서 동일한 상태 전이를 수행한다. 이러한 상태들은 게임 중에 발생할 수 있는 모든 경우들을 고려하고 있다. 상태의 전이는 이벤트에 따라서 이루어진다. 이벤트에는 플레이어가 공을 치는 것, 공이 테이블 면에 닿는 것, 공이 네트를 넘어가는 것, 공이 지면에 닿는 것 등이 있다.



[그림 3] 공의 위치에 따라 결정되는 게임의 상태

이러한 단순화된 탁구 게임을 구현하는 것은 간단하다. 게임의 시나리오에 필요한 기하 모델의 수가 많지 않으며 게임의 상태 및 전략의 수도 제한적이기 때문이다. 상태를 더 많이 세분화하면 더욱 정교한 게임 규칙의 적용이 가능하다.

3. 휴먼 플레이어 액션의 인식

AR 게임에서의 플레이어의 상호작용은 실제 세계에서의 플레이어의 움직임에 기반을 둔다. 따라서 AR 게임에서 게임 세계에서의 플레이어의 반응을 인지하기 위해서는 플레이어의 동작을 인식하는 기술이 필요하다. AR 탁구 게임의 구현 과정에서 가장 어려운 문제는 플레이어의 라켓의 움직임을 빠르고 안정적으로 인식하는 모듈의 구현이다.

액션의 인식을 위한 손 추적 모듈 및 패들 모션 추적 모듈들을 구현하고 이를 AR 탁구 게임에 적용하였다. AR 게임에의 적용을 위해서 처리 속도와 안정성에 가장 큰 비중을 두었다.

상호작용을 담당하는 모듈의 구현은 세 가지로 나누어서 접근하였다. 첫 번째는 맨손으로의 체스 처 기반 상호작용이고, 두 번째는 실제 라켓을 사용하는 상호작용이고, 세 번째는 마커가 부착된 라켓을 사용하는 상호작용이다. 각 상호작용 카테고리에 대해서 해당하는 모듈을 구현하여 전체 게임 프로토타입 시스템에 적용하였다.

플레이어는 모니터로 관찰되는 게임 장면을 보면서 실제 라켓으로 가상의 공을 친다. 가상의 공과 실제 라켓은 모두 모니터로 관찰되므로 플레이어는 자신의 위치에 상대적인 실 공간에서의 가상의 공의 위치를 각각할 수 있다. 더욱 일치화된 동작을 위해서 [그림 4]에서와 같이 HMD를 착용할 수도 있다.



[그림 4] 휴먼 플레이어의 동작

3.1 맨손 행동의 인식

입의 미지의 공간에서의 손 움직임의 강건한 인식은 매우 어려운 문제이다. 시간적 및 공간적으로 변하는 조명 환경, 인식을 어렵게 하는 애매한 배경, 불규칙적인 물체의 움직임 등의 다양한 변화 요인들이 인식을 어렵게 한다.

맨손을 사용한 상호작용의 경우에 대해서 피부색 영역의 획득을 위한 컬러 필터링 기법을 사용하였다[13]. 필터링된 피부색 영역들에 대해서 손 위치의 후보 위치들을 구한다. 각 후보 위치들 중에서 가장 유력한 후보의 위치를 손의 위치로 결정한다. 영역의 모양 및 손의 움직임 궤적에 대한 정보를 사용한 검증 절차를 통해서 손의 영역인지에 대한 신뢰성을 높이도록 하였다.

다른 중요한 문제는 사용자의 의도된 움직임인지 우연한 신체적인 움직임인지를 구분해야 하는 것이다. 의도된 움직임의 경우에는 움직임에 일관성과 연속성이 있어야 한다. 이를 위해서 손의 움직임 궤적을 분석하고 그 궤적의 곡선과 공의 궤적의 연관성을 분석하여 의도된 손의 움직임만을 인식할 수 있도록 하였다.

손동작의 인식에 있어서 인식의 정확성을 높이는 것은 매우 어렵다. 손 움직임의 변화가 불규칙하며 주변 환경도 복잡할 수 있기 때문이다. 또한 피부색 필터링 기법을 사용하므로 손 이외의 신체 노출 부위로 인한 오인식 및 배경이 피부색을 포함한 경우에 있어서의 오인식의 가능성이 있다. 이는 피부색을 사용한 손동작의 인식은 제한된 환경에서만 안정적으로 동작할 수 있다는 것을 의미한다.

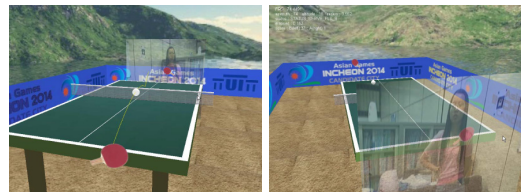
3.2 실제 라켓의 인식

플레이어의 손을 인식하는 방법보다 현실적인 접근 방법은 실제 라켓을 인식하는 방법이다. 사람의 손에 비해서 라켓은 강체이므로 모양의 변화가 심하지 않다. 또한 라켓의 표면은 특정 원색 컬러이므로 영상에서 영역 탐색이 쉽다.

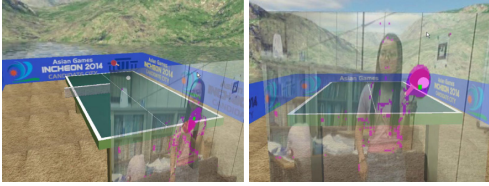
영상의 분석 단계에서 라켓의 붉은색 영역을 필터링하여 라켓의 후보 위치를 결정한다. 탁구 라켓은 붉은색 고무로 덮혀져 있다. 입력 영상을 HSV(hue-saturation-value) 컬러 공간으로 변환한 후에 hue 요소들을 체크하여 라켓 칼라 픽셀들을 검출한다. hue 요소는 조명 조건에 거의 영향을 받지 않으므로 hue 요소를 사용한 컬러 필터링은 조명에 강건한 특성을 가진다.

라켓의 위치를 찾기 위해서 라켓 색을 가진 픽셀들만을 고려하여 blob들을 찾는다. blob의 위치는 히스토그램 분석을 통해서 수행된다. 영상의 가로 방향과 세로 방향으로 라켓 색 픽셀 개수를 세어 히스토그램을 만든다. 두 방향의 히스토그램의 피크 위치들을 찾고 그 위치의 교점에 대해서 라켓 모양의 blob이 존재하는지를 검사한다. 각 히스토그램의 최대 피크 위치만 고려할 수도 있고 또는 여러 피크 위치들을 모두 고려하여 여러 blob 위치들 중에서 최선의 영역을 선택하는 방법도 가능하다. 라켓 색이 실제 라켓 주위에 배경 성분으로 분포하는 경우에도 안정적인 라켓 검출이 가능하다.

게임 플레이어는 3차원 공간에서의 가상 탁구공을 붉은색 탁구 라켓을 사용하여 조작하게 된다. 단 가상 공의 제어는 가상 공과 실제 라켓이 모두 영상의 시야에 들어온 경우에만 가능하다.



[그림 5] 가상 코트에서 AR 탁구 게임 플레이 모습



[그림 6] 라켓 표면 컬러를 필터링하여 실시간 라켓 위치를 추적하는 기법의 적용

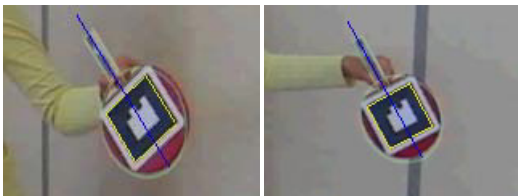
실제 비디오 스트림은 [그림 5]와 같이 수평으로 뒤집어서 플레이어 쪽에 배치하여 보여주도록 해서 자신을 관찰할 수 있도록 하였다. 플레이어는 모니터를 보면서 가상의 공을 칠 수 있다. [그림 6]은 라켓의 색을 필터링하여 라켓을 추적하는 과정을 보여준다. 그림에서 라켓 색으로 판정된 위치를 자주색으로 표시하였고 추정된 라켓 위치의 중심을 녹색으로 표시하였다.

3.3 마커를 부착한 실제 라켓의 인식

컬러 필터링을 사용한 라켓 추적은 빠르고 또한 조명에 강건하게 라켓의 위치를 영상 좌표 기준으로 제공한다. 그러나 영상 내에서의 라켓의 2차원 정보만을 제공하므로 게임 공간에서의 3차원적인 라켓의 위치와 자세를 계산할 수 없다. 라켓의 3차원적인 위치의 계산을 위해서는 보다 복잡한 접근 방법이 필요하다.



[그림 7] 라켓 표면에 부착된 마커를 검출하여 실시간 라켓 위치를 추정하는 기법의 적용



[그림 8] 추정된 라켓의 위치와 방향

가장 먼저 고려해볼 수 있는 방법은 미리 알려진 마커를 라켓에 부착하는 방식이다. 많은 AR 응용에서는 흑백 패턴의 평면 마커를 사용한다. 흑백 패턴 마커는 영상에서 빠르게 검출할 수 있으며 또한 마커로부터 카메라 자세를 바로 계산할 수 있는 장점을 가진다. 흑백 마커를 라켓의 표면에 부착하고 게임 플레이 시에 영상에서 마커를 검출하여 라켓의 3차원적인 위치와 방향을 계산하였다.

[그림 7]의 왼쪽 그림은 마커를 부착한 라켓의 모습이고 오른쪽 그림은 검출된 마커로부터 계산된 라켓의 자세를 표시한 것이다. 인식된 라켓의 표면이 노란색으로 표시되어 있고 마커의 모서리점과 방향도 선으로 표시하였다. [그림 8]에 인식된 라켓의 자세를 확대하여 표시하였다.

4. 게임 요소들의 구현

4.1 물리 시뮬레이션

실제 탁구 게임에서 탁구공은 테이블 위를 최대 175km/h의 속도로 매우 빠르게 날아간다. 이것은 0.1초 이내에 테이블을 완전히 통과한다는 의미이다. 범용 비디오카메라의 일반적인 프레임 캡처 속도가 초당 30 프레임인 점을 고려하면 플레이어가 비디오 프레임을 보면서 공을 제어하는 것은 거의 불가능하다. 따라서 공이 진행되는 속도를 충분히 낮추어서 플레이어가 공의 움직임에 대처할 수 있도록 해야 한다. 탁구공의 움직임은 탁구 게임에 있어서 매우 중요하게 작용한다. 탁구공의 역학을 담당하는 간단한 물리 엔진을 구현하였다. AR 탁구 게임에서 게임 로직 엔진은 가상의 공과 다른 가상 객체들과의 모든 충돌을 계산한다. 가상 테이블, 가상 네트, 실제 바닥과 일치된 가상 바닥과의 충돌은 게임의 상태를 변화시키는 중요한 요인이다.

탁구공의 운동 역학은 주로 공을 치는 동작인 스트로크, 공이 날아가는 동작, 공이 튀어 오르는 바운싱 동작에 적용된다. 시간에 의존되는 공의 위

치 함수 $\mathbf{x}(t)$ 는 뉴턴의 물리 법칙에 따라 다음과 같이 계산된다.

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0(t - t_0) + \frac{1}{2}\mathbf{a}(t - t_0)^2$$

여기서 가속도 \mathbf{a} 는 상수로 가정한다. \mathbf{x}_0 와 \mathbf{v}_0 는 시간 t_0 에서의 초기 위치와 초기 속도이다. 가속도와 초기 값들이 주어지면 위의 위치 함수는 4차 Runge-Kutta 적분 방법으로 구할 수 있다.

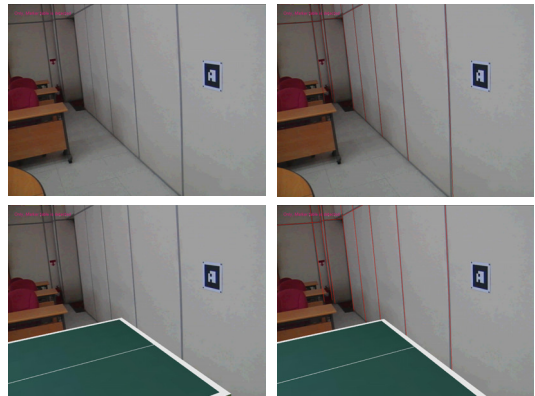
실세계에서의 탁구공의 역학은 더욱 복잡하다. 공은 중력의 영향으로 아래로 떨어지는 특성을 가진다. 공이 빠른 속도로 주행하는 경우에는 테이블을 통과하는 시간 동안에 약 2cm 정도밖에 아래로 떨어지지 않지만 공의 진행 속도에 따라서 공이 떨어지는 정도는 크게 달라진다. 따라서 공의 속도에 따라서 스트로크의 속도와 방향에 매우 주의해야 한다. 공을 최고 속도로 칠 수 있는 순간은 공이 상대편 쪽 테이블의 목표지점으로 직선 주행할 수 있을 만큼 공이 높이 떠 있는 경우이다. 그 외에 경우에는 적절히 조절된 힘과 속도로 공을 쳐야 네트에 부딪히지 않고 목표하는 지점에 다다를 수 있다. 이런 점이 탁구 게임을 흥미롭게 만드는 주요 요인이다.

탁구에서의 공의 움직임에는 스핀(spin) 특성이 있다. 숙련된 선수는 초당 150회전 이상의 스핀을 구사할 수 있고 라켓으로 공을 약 50m/s의 속도로 보낼 수 있다. 공은 공기 저항으로 점차 느려지지만 그래도 여전히 빠른 속도에 해당한다. 스핀은 공이 네트보다 아래에 있는 경우라도 플레이어가 공을 세계 쳐서 원하는 지점에 다다를 수 있도록 한다. 백스핀(backspin)은 공을 뜨게 만들고 탑스핀(topspin)은 공을 가라앉게 만든다. 탑스핀을 심하게 가한다면 위쪽 방향으로 세계 쳐서도 상대방 테이블 위치에 다다르게 할 수 있다. 또한 사이드스핀(sidespin)으로 공이 좌우로 휘도록 할 수도 있다. 플레이어는 수많은 복합적인 스핀을 구사할 수 있으며 이러한 특성이 탁구의 재미를 더한다.

스핀에 변화를 주어 상대방 플레이어를 속이도록 하는 기술이 탁구의 승패를 결정한다. 따라서 탁구의 재미를 높이기 위해서는 플레이어가 자유로운 스핀을 구사할 수 있도록 스핀을 위한 플레이어의 모션을 인식하도록 함이 필요하다.

4.2 가상 테이블의 합성

마커를 사용하는 경우에는 마커에 상대적인 카메라의 3차원적인 위치와 방향을 추정할 수 있다. 가상 탁구 테이블을 합성하기 위해서 고정 마커를 추가적으로 사용하였다. [그림 9]와 같이 마커를 벽과 같은 고정된 위치에 부착하고 그 위치에 상대적으로 정해진 위치에 가상 테이블을 합성하도록 하였다.

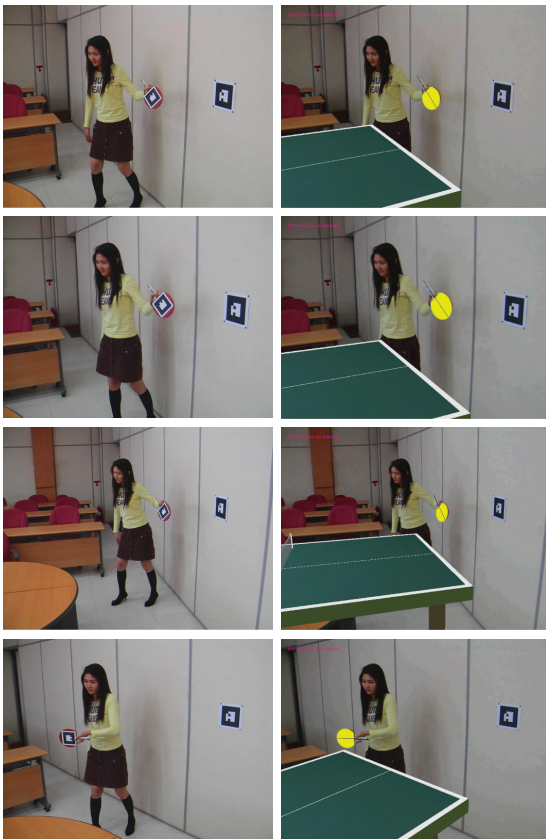


[그림 9] 벽의 마커에 상대적으로 가상의 테이블을 배치하는 기법의 적용

고정 위치에 부착된 마커를 검출한 다음 카메라의 상대적인 위치와 방향을 추정한다. 추정된 카메라 변환 행렬을 사용하여 가상 테이블을 3차원 공간의 고정 위치에 삽입하고 렌더링하였다. [그림 9]의 하단 그림은 가상 테이블이 합성된 모습을 보여준다. 오른쪽 그림의 붉은색 선은 마커 주변의 미리 알려진 방의 구조를 마커를 기준으로 일치한 결과를 보여준다. 미리 알려진 방의 구조가 실제 영상에서 잘 일치됨을 확인할 수 있다.

고정된 벽에 부착된 마커와 라켓에 부착된 마커는 각각 별도의 패턴을 가지므로 두 마커들이 겹쳐진 후에 서로 명확히 구분할 수 있다. 라켓에 부착된 마커는 공을 칠 때의 각도나 스핀을 조절하거나 플레이어의 전부 또는 일부를 가상 객체로 대체하기 위한 용도로 이용된다. [그림 10]은 라켓이 움직이는 상황에서의 입력 장면들과 합성된 출력 장면들을 보여주고 있다.

플레이어가 움직이는 동안에 고정 마커가 가려지거나 시야에서 벗어나는 경우가 발생할 수 있다. 이러한 문제의 해결을 위해서 변환 행렬 추정에 대해 dead reckoning 기법을 사용하거나 상이한 패턴의 고정 마커들을 다중으로 부착하는 방법을 사용해야 한다.



[그림 10] 움직이는 카메라 환경에서의 게임 플레이. 입력 장면(왼쪽)과 합성된 출력 장면(오른쪽)

4.3 처리 시간 분석

구현된 AR 탁구 게임 프로토타입을 2.6GHz Pentium 4 PC 환경에서 실험하였다. 캡처되는 비디오 프레임 크기는 320x240으로 하였다. 세 가지 액션 인식 타입에 대한 각 모듈의 처리 속도는 [표 1]과 같다. 타입 1은 맨손 행동의 인식 방식을, 타입 2는 실제 라켓의 인식 방식을, 타입 3은 마커를 부착한 실제 라켓의 인식 방식을 의미한다. 전체적인 프레임 처리 속도는 타입 1과 타입 2의 경우에 약 17 FPS이었고 타입 3의 경우에 약 12 FPS이었다.

[표 1] 게임 시스템의 처리 속도의 비교

처리 시간 \ 타입	타입 1	타입 2	타입 3
프레임 캡처(ms)	33	33	33
동작 인식(ms)	22	22	47
물리 엔진(ms)	1	1	1
렌더링 엔진(ms)	2	2	2
전체 시간(ms)	58	58	83
전체 FPS	17	17	12

전체적인 물리 엔진 및 렌더링 엔진의 처리 속도는 300 FPS 이상의 빠른 처리속도를 보인다. 그러나 카메라 장치의 한계로 인해서 프레임 캡처율은 최대 30 FPS를 넘지 않았다. 따라서 영상 분석 모듈의 속도도 30 FPS에 제한되어 있고 실제 패들의 움직임이 가상 패들의 움직임으로 반영되는 속도도 프레임 처리 속도에 제한되어 있다.

타입 1이나 타입 2에 있어서 피부색 픽셀 및 붉은색 픽셀의 필터링과 히스토그램 누적 절차는 손 및 라켓의 후보 위치들을 결정하기 위한 과정이다. 이 과정에서 여러 후보들에 대해서 blob의 모양을 분석하여 더욱 신뢰성 있는 손 및 라켓의 위치를 결정하는 방법이 가능하다. 그러나 정확도를 높이기 위해 계산량이 많은 부가적인 절차들을 도입하면 처리 시간이 길어지고 따라서 사용자에 대한 즉시적인 피드백이 어려워진다. 따라서 실시간성을

보장하는 한도에서 최대의 정확성을 추구함이 바람직하다. 정확성이 조금 떨어지는 것은 전체적인 측면에서 게임 진행에 심각한 영향을 미치지 않는다. 왜냐하면 플레이어의 반응이 빠르게 시스템으로 전달되고 또한 빠른 피드백 시간으로 플레이어가 다음 동작을 자연스럽게 진행할 수 있다면 정확성의 차이는 플레이어의 감각에 큰 영향을 주지 않는다.

5. 결 론

본 논문에서는 AR 스포츠 게임의 구현에 필요한 세부 기법들을 제시하였다. 특히 빠른 처리 시간과 계산량에 따른 적절한 개발 방법론을 제시하였다. 동작 기반 인터페이스를 사용하는 AR 탁구 게임을 프로토타입으로 구현하였다. 동작 기반 인터페이스에서의 상호작용은 고전적인 입력 장치들을 전혀 사용하지 않고 사람의 손 또는 라켓 동작에만 의존하도록 하였다.

AR 탁구 게임의 게임 시나리오의 구현은 간단하게 이루어질 수 있다. 게임의 규칙들의 개수가 작고 필요한 기하 모델들이나 상태 및 전략들이 제한적이기 때문이다. 가장 어려운 문제는 라켓의 움직임은 빠르고 안정적으로 인식하는 문제이다.

패들 모션을 빠르게 추적하는 모듈을 구현하여 AR 탁구 게임에 적용하여 보았다. 플레이어의 패들을 추적하는 모듈의 구현은 각기 다른 세 가지 방식으로 수행하였으며 각 방식에 있어서의 응용성을 비교하였다. 플레이어의 실제 움직임이 즉시 게임 진행 상황에 반영되어야 하므로 구현은 실제 라켓의 실시간 추적을 요구조건으로 하였다. 빠른 반응 시간으로 인해서 플레이어는 즉시적으로 시각적인 피드백을 확인할 수 있고 그에 따라서 진행 동작을 조절할 수 있었다.

참고문헌

- [1] Nilsen, T., Looser, J., "Tankwar tabletop war gaming in augmented reality," Proceedings of 2nd International Workshop on Pervasive Gaming Applications, 2005.
- [2] Dong, Q., Sun, Z., Namee, B.M., "Physics-based table-top mixed reality games", Conference of the International Simulation & Gaming Association, 2008.
- [3] <http://www.eyefofjudgment.com/>
- [4] 김기영, 이민경, 박영민, 이종원, 우운택, "ARPushPush: 실내 환경 증강 현실 게임," KHCI, pp. 354-359, 2005.
- [5] Cheok, A.D., Fong, S.W., Goh, K.H., Yang, X., Liu, W., Farzbiz, F., Li, Y., "Human Pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area," Mobile HCI 2003, pp. 209 - 224, 2003.
- [6] Lang, T., MacIntyre, B., Zugaza, I.J., "Massively multiplayer online worlds as a platform for augmented reality experiences," Virtual Reality, pp. 67-70, 2008.
- [7] Pavlovic, V., Sharma, R., Huang, T.S., "Visual interpretation of hand gestures for human-computer interaction: A review," IEEE T-PAMI 19(7), pp. 677-695, 1997.
- [8] Govil, A., You, S., Neumann, U., "A video-based augmented reality golf simulator," ACM Multimedia, pp. 489-490, 2000.
- [9] Ishii, H., Wisneski, C., Orbanes, J., Chun, B., Paradiso, J., "Pingpongplus: design of an athletic-tangible interface for computer supported cooperative play," Conference on Human Factors in Computing Systems, pp. 394-401, 1999.
- [10] Woodward, C., Honkamaa, P., Jappinen, J., Pyokkimies, E., "Camball - augmented virtual table tennis with real rackets," ACE Singapore, pp. 275-276, 2004.
- [11] Aitkin, A.L., "Playing at reality: exploring the potential of the digital game as a medium for science communication," PhD thesis, Institution The Australian National University, 2005.
- [12] Pentenrieder, K., Meier, P., "The need for accuracy statements in industrial augmented reality

applications," International Symposium on Mixed and Augmented Reality, 2006.

- [13] Park, J.S., Yoon, J.H., Kim, C., "Towards an efficient implementation of a video-based gesture interface," Lecture Notes in Computer Science 4304, pp. 463-472, 2006.



박종승(Park, Jong Seung)

1992년 경북대학교 전자계산학과 (이학사)
1994년 포항공과대학교 컴퓨터공학과 (공학석사)
1999년 포항공과대학교 컴퓨터공학과 (공학박사)
2004년~현재 인천대학교 컴퓨터공학과 조교수

관심분야 : 게임공학, 3D 비전, 증강현실



전영준(Jeon, Young Jun)

2001년 인천대학교 컴퓨터공학과 (공학사)
2005년 인천대학교 컴퓨터공학과 (공학석사)
2005년~현재 인천대학교 컴퓨터공학과 박사과정

관심분야 : 게임공학, USN, 소프트웨어공학
