■ 2005년 정보과학 논문경진대회 수상작

Flying Cake: 모바일 단말기를 이용한 실감형 게임

(Flying Cake: An Augmented Game on Mobile Device)

박 안 진 † 정 기 철 ††
(Anjin Park) (Keechul Jung)

요 약 언제, 어디서, 누구나 대용량 네트워크를 사용할 수 있는 유비쿼터스(ubiquitous) 시대가 다가오면서, 카메라가 장착되어 있고 무선 통신이 가능한 PDA, 웨어러블(wearable) 컴퓨터와 같은 휴대용 장치가 가까운 미래에는 일상의 한 부분이 될 것이다. 이런 상황을 반영하듯, 휴대용 장치를 이용한 실감형 게임(augmented game)에 관한 다양한 연구가 진행되어 왔다. 기존의 실감형 게임들은 전통적으로 'backpack' 시스템이나 패턴마커(pattern marker)를 이용하였다. 'backpack' 시스템은 비싸고, 거추장스러우며, 사용하기 불편한 단점이 있으며, 패턴마커를 사용하면 미리 정한 장소에서만 게임을 해야 하는 단점을 가지고 있다. 본 논문에서 소개하는 게임 *Flying Cake*는 거추장스러운 장비 대신, 가볍고 휴대 가능한 PDA를 이용하며, 실제 세계에서 가상의 물체를 접목(overlay)하기 위해, 수동적인 패턴마커 대신 얼굴 영역을 이용한다. *Flying Cake*는 PDA만을 이용하여 실제 세계를 돌아다니며 카메라에 의해 입력된 영상에 접목된 가상의 캐릭터를 공격하는 일인용과 무선 랜을 통해 전송되는 상대방의 영상에 접목된 가상의 캐릭터를 공격하는 이인용을 제공하는 실감형 슈팅 게임이다. 얼굴 추출 기술을 이용하여 입력 영상의 얼굴 영역에 가상의 캐릭터를 접목하며, 사용자는 가상의 캐릭터를 공격하며 게임을 즐긴다. *Flying Cake*는 얼굴 추출 기술을 이용하여 PDA카메라를 통해 입력된 실제 세계와 가상의 물체 사이의 상호작용을 제공하는 새로운 패러다임(paradigm)을 제공함으로써 사용자에게 새로운 즐거움을 제공할 것이다.

키워드 : 실감형 게임, 모바일 비전(PDA), 얼굴 추출, 3D 슈팅 게임, CAMShift

Abstract In the ubiquitous computing age which uses a high quantity network, mobile devices such as wearable and hand-held ones with a small camera and a wireless communication module will be widely used in near future. Thus, a lot of researches about an augmented game on mobile devices have been attempted recently. The existing augmented games used a traditional 'backpack' system and a pattern marker. The 'backpack' system is expensive, cumbersome and inconvenient to use, and because of the pattern marker, it is only possible to play the game in the previously installed palace. In this paper, we propose an augmented game called *Flying Cake* using a face region to create the virtual object(character) without the pattern marker, which manually indicates an overlapped location of the virtual object in the real world, on a small and mobile PDA instead of the cumbersome hardware. *Flying Cake* is an augmented shooting game. This game supplies us with two types: 1) a single player which attacks a virtual character on images captured by a camera in an outdoor physical area, 2) dual players which attack the virtual character on images which are received through a wireless LAN. We overlap the virtual character on the face region using a face detection technique, and users play *Flying Cake* though attacking the virtual character. *Flying Cake* supplies new pleasure to players with a new game paradigm through an interaction between the user in the physical world captured by the PDA camera and the virtual character in a virtual world using the face detection.

Key words : Augmented Game, Mobile Vision(PDA), Face Detection, 3D Shooting Game, CAMShift

· 이 논문은 2006년 게임연구센터(Game Research Center) 지정 및 연구수행 지원에 의하여 연구되었음

† 학생회원 : 숭실대학교 미디어학부
anjin@ssu.ac.kr
†† 종신회원 : 숭실대학교 미디어학부 교수
kcjung@ssu.ac.kr

논문접수 : 2005년 9월 15일
심사완료 : 2006년 11월 21일

1. 서 론

1950년대 초반 최초의 컴퓨터 게임이 등장한 후, 상품성과 상업성을 인정받으면서 다양한 장르의 게임, 다양한 플랫폼이 개발되어 왔고, 지금도 새로운 장르, 새

로운 플랫폼에 대한 다양한 연구가 진행되고 있다. 표 1은 게임의 역사를 간단하게 정리한 내용이다[1,2].

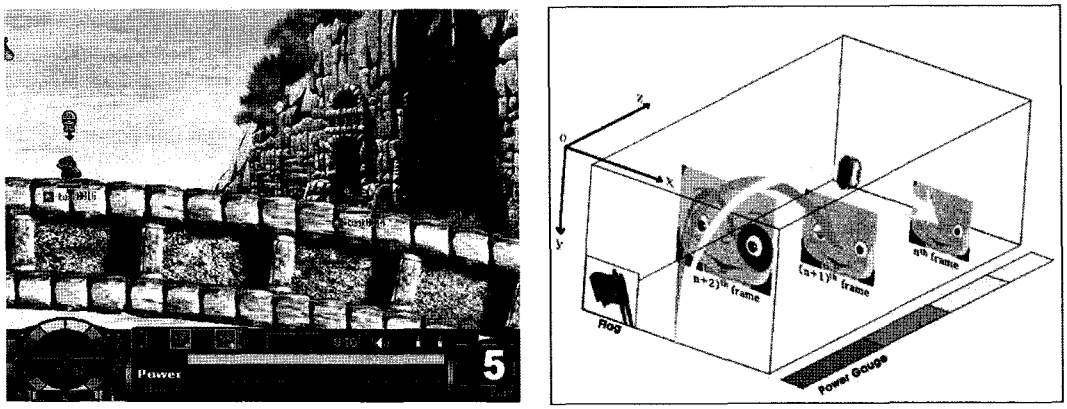
최근의 게임시장은 휴대용 게임기가 점점 확대되고 있으며, 2005년에는 시장 규모가 약 40억불 내외로 추정되고 있다. 현재 휴대용 게임기 시장의 90% 이상을 장악하고 있는 Nintendo는 지금의 자리를 확고히 하기 위해, 기존의 인터페이스에서 벗어나 스크린을 직접 스타일러스 펜으로 조작할 수 있고 스크린이 두 개인 휴대용 시스템 *Nintendo DS*를 출시하였고, Sony는 Nintendo의 독주를 막기 위해 기존의 콘솔 게임 PlayStation을 이용한 화려한 그래픽의 PSP(PlayStation Portable)을 선보였다.

휴대용 게임기의 성장과 더불어 모바일 게임 시장 역시 크게 성장하고 있다. 국내 모바일 게임 시장은 1999년 인터넷 머그수준의 게임이 처음 출시되었고, 각 이동통신사별로 다운로드가 가능한 그래픽 게임이 소개되었다. 2001년부터 게임이 대량 출시되기 시작했으며, 2003년 컬러폰의 보급으로 RPG게임등이 대작화되어 가고 '타이쿤'이라는 인기 장르의 발굴로 게임매니아들도 생겨났다. 2003년 노키아는 게임기의 기능을 합친 모바일폰을 처음으로 출시하였으며, 이때부터 3D 게임폰에 대한 개발이 본격으로 시작되었다. 우리나라는 2004년 팬택&큐리텔에서 게임폰을 처음 출시한 이후 각 통신사 별로 3D 게임폰과 3D 게임 개발이 진행중이며, 일본은 이미 3D 게임폰 보급에 맞춰 '드래곤퀘스트', '파이널판타지', '삼국지' 등 PS2와 PC게임으로 인기를 얻은 히트작을 대거 모바일 게임으로 출시했으며, 이로 인해 3D게임폰의 보급률이 급격히 증가하고 있다[3,4].

이런 시대를 반영하듯 모바일 폰, 휴대용 게임기와 같은 휴대용 장치를 이용하여 게임을 제작, 기획하는 많은 일들이 진행되고 있다. 특히 최근에는 카메라가 장착된 휴대용 장치를 이용하여 실제 세계를 배경으로 게임을 하는 실감형 게임(augmented game)에 대한 다양한 연구들이 진행되고 있다[5-11]. Szalavari[5] 등은 실제 환경에서 두 명 이상이 가상의 마작 패를 이용하여 게임을 진행하는 협동적인 실감형 게임 *마작(Mah-Jongg)*을 소개하였다. Ohshima[6] 등은 두 사람이 실제 환경인 하나의 테이블과 가상의 퍽(puck)을 공유하는 실감형 게임 *air-hockey*를 소개하였다. Starner[7] 등은 네트워크를 이용하여 여러 사람이 같이 즐길 수 있는 실감형 게임 환경을 제안하였으며, 제스처(gesture)를 통해 사용자들 사이에서 공격과 수비를 할 수 있는 *쿵푸* 게임에 적용하였다. Thomas[8] 등은 일인용 슈팅 게임

표 1 게임의 역사

년도	개발자	개발명	Description
1952	A. S. Douglas	*Noughts and Crosses*	The first documented computer game
1958	Brookhaven National Lab.	*Tennis for two*	The first interactive computer game
1961	MIT's Steven Russel	*Spacewar*	The first shooting game
1971	Nolan Bushnell	*Computer Space*	The first coin arcade-style game
1972	Ralph Baer(ATARI)	*pong*	The first home game console
1977	ATARI	*Atari 2600*	The first multi-game home console
1980	Namco	*Pac-man*	The first animated main character with its own name
1982	Microsoft	*Flight Simulation*	Microsoft joins the game industry
1984	Nintendo	*Famicom*	The first 8-bit system
1986	Nintendo	*Nintendo Entertainment System(NES)*	The console system with 8-bit color graphics and elaborate sound effects
1989	Sega	*Sega Genesis*	The first 16-bit console game
1989	Nintendo	*Game Boy*	The hand-held game system
1994	Sony	*PlayStation*	
1995	Sega	*Saturn System*	The first 32-bit CD-based console
1996	Nintendo	*Nintendo 64*	The first true 64-bit game system
2000	Sony	*PlayStation2*	The first console with 128-bit graphics and using DVD technology
2001	Nintendo	*Game Boy Advance*	The first system to allow gamers to connect wireless phones to it for Internet Access
2001	Microsoft	*XBox*	The first console system to contain hard disk space and support HDTV
2004	Nintendo	*Nintendo DS*	The first touch screen-based portable game
2005	Sony	*PlayStation Portable*	



(a) 포트리스[14] (b) *Flying Cake*

그림 1 게임 환경

인 *Quake*를 옥외에서 플레이할 수 있는 실감형 게임으로 확장한 최초의 게임 *ARQuake*를 제안하였으며, 사용자가 실제 세계에서 이동할 수 있도록 하며, 동시에 컴퓨터가 만들어내는 그래픽 유령과 물체들을 경험하고 제거할 수 있도록 하였다. Cheok 등은 실제 환경인 도시 전체를 게임 공간으로 확장한 *Game-City*를 소개하였으며, 웨어러블 기기와 증강 현실을 함께 유비쿼터스 환경에 적용하여 개발하였다[9]. 이렇게 개발된 *Game-City* 환경에 사용자가 돌아다니며 가상의 쿠키를 모우는 실감형 게임 *팩맨(pacman)*을 적용하였다[10].

위에서 언급한 게임들은 실제 세계에 가상의 물체를 접목하기 위해 HMD, 카메라, 추가적인 지원 장비(노트북 등)로 구성된 전통적인 'backpack' 시스템을 이용하였다. 그러나 이 시스템은 비싸고, 거추장스러우며, 사용하기 매우 불편한 단점을 가지고 있다. 게다가 HMD는 가로 시각 범위가 34도[1] 밖에 되지 않기 때문에, 사용자는 시각적으로 매우 불편함을 느낀다[6]. 이런 불편함을 해결하기 위해 거추장스러운 시스템 대신 PDA를 이용하여 수행한 실감형 게임이 소개되었다[11]. 그러나 이 게임은 실제 세계에서 가상의 물체를 접목(overlay)할 위치를 지정하기 위해 패턴마커(pattern marker)를 이용하였으며, 패턴마커가 미리 놓여 있는 장소에서만 게임이 가능한 단점을 가지고 있다.

본 논문에서는 *Flying Cake*라는 실감형 게임을 소개한다. 이 게임은 가상의 케이크를 던져서 적의 얼굴 영역에 접목되어있는 가상의 캐릭터를 공격하는 게임으로써, 기존의 거추장스러운 'backpack' 시스템 대신 가볍고, 휴대 가능하며, 네트워크와 쉽게 연동할 수 있는 PDA를 이용하며, 가상의 캐릭터를 접목하기 위해, 항상 휴대하고 있어야 게임이 가능한 수동적인 패턴마커 대신, 언제 어디서나 게임을 즐길 수 있는 얼굴 영역을 이용한다. 우리는 얼굴 영역을 추출하기 위해 스킨칼라(skin-color) 모델[12]을 이용하며, PDA에서 효과적인 수행 시간을 보장하기 위해서 CAMShift 알고리즘[13]을 이용한다.

본 논문의 구성은 다음과 같다. 제 2장에서 *Flying Cake*의 전체적인 개요에 대해 설명하고, 제 3장에서 입력 영상에 가상의 캐릭터를 접목할 위치를 계산하기 위한 얼굴 영역 추출 방법을 기술하며, 제 4장에서는 3차원 공간에서 가상의 케이크를 던지고, 공격의 성공여부를 확인하기 위해 필요한 3차원 정보에 대해 기술한다. 마지막으로 제 5장에서 실험 및 결과에 대해 언급하고, 제 6장에서 결론 및 향후 연구 방향을 기술한다.

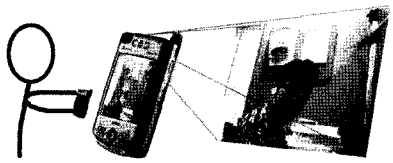
2. *Flying Cake*

*Flying Cake*는 포트리스[14]와 유사한 게임으로 실제 사람을 타깃(target)으로 한 3차원 슈팅게임이다. 포트리스는 컴퓨터 화면의 2차원 공간에서 게임을 수행하며, 탱크가 포를 쏘아 적을 제거하는 게임으로, 포는 포물선을 그리며 날아가고 사용자 의도에 의해 수동으로 조정되는 파워와 컴퓨터에 의해 생성되는 바람에 영향을 받는다(그림 1(a)). *Flying Cake*는 PDA 카메라를 통해 입력된 실제 환경의 3차원 공간에서 수행하며, 포 대신 가상의 케이크를 가상의 캐릭터가 접목되어있는 적의 얼굴을 향해 던지는 게임으로, 포트리스와 같이 가상의 케이크 역시 포물선을 그리며 날아가고, 파워(power gauge)와 바람(flag)에 영향을 받는다(그림 1(b)).

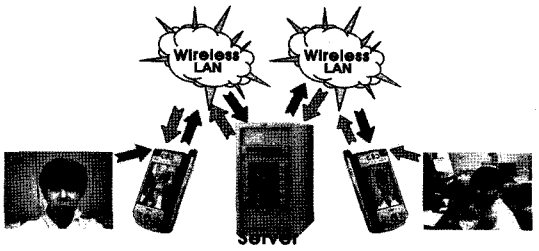
2.1 게임 설계(design)

*Flying Cake*는 싱글(single) 모드와 듀얼(dual) 모드로 진행되며, 그림 2는 각각의 시스템 구성을 보여준다. 그림 2(a)는 싱글 모드로, 하드웨어(hardware)는 카메라

1) 사람의 가로 시각 범위는 약 100도 정도이다[6].



(a) 싱글 모드



(b) 듀얼 모드

그림 2 *Flying Cake*의 시스템 구성

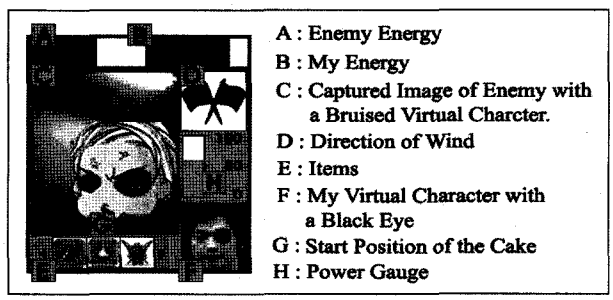
가 장착된 PDA만을 필요로 한다. 카메라는 상대방(적)을 향하며, PDA 화면에는 가상의 캐릭터가 접목된 상대방이 보인다. 그림 2(b)는 듀얼 모드이며, 하드웨어는 카메라가 장착된 PDA, 무선 통신 모듈, 서버로 구성되어 있다. 카메라는 사용자 자신에게 향하며, PDA 화면에는 무선 랜을 통해 수신된 상대방의 카메라를 통해 입력된 영상이 보인다.

그림 3(a)는 게임 중인 영상을 나타내며, 그림 3(b)는 게임 중인 PDA 화면을 보여준다. 심벌(symbol) C는 얼굴 영역에 가상의 캐릭터가 접목되어 있는 상대방을 보여주는 부분이다. 심벌 D는 바람의 방향(그림 4)이며, 케이크가 날아갈 x축 방향에 영향을 준다. 심벌 F는 내 가상 캐릭터이며, 내 캐릭터를 점점 붉은색으로 변화시킴으로써 적의 케이크가 가까워짐을 알려준다. 심벌 G는 공격을 수행할 때 케이크의 출발 위치이고, 심벌 H는 케이크를 던지기 위한 파워게이지(power gauge)이며, 케이크가 날아갈 깊이 정보에 영향을 준다. 그림 5는 가상 캐릭터의 예를 보여준다. 적의 공격으로 에너지가 줄어들면, 그림 5처럼 가상 캐릭터는 점점 더 많은 상처가 생기게 되며, 그림 3에서 적의 캐릭터가 내 가상 캐릭터보다 더 많은 상처가 있는 건, 적과 나의 에너지 차이 때문이다.

2.2 전체 흐름도

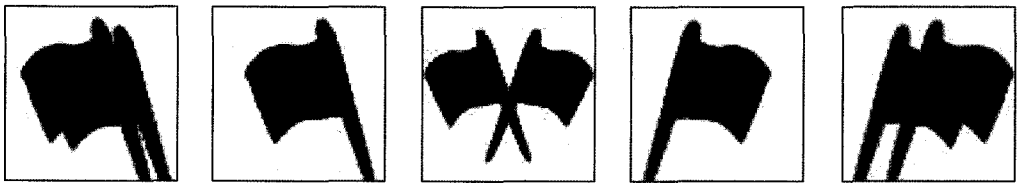
실감 현실(augmented reality) 시스템은 다음의 3가지 특성을 가지고 있어야 한다[13]. 첫째 실제 세계에 가상의 물체가 접목되어야 한다. 둘째 실시간 상호작용이 있어야 한다. 셋째 가상의 물체는 3차원 상에 정합(registration)되어야 한다. *Flying Cake*는 얼굴 영역을 추출한 결과를 바탕으로 PDA 화면에 보이는 실제 세계에 가상의 캐릭터를 접목하며, 날아오는 케이크를 피하기 위해 얼굴을 움직임으로써 실시간 상호작용을 수행하고, 빠른 수행 시간을 위해 가상 캐릭터의 좌우방향

(a) 게임 중인 영상



(b) 게임이 진행중인 PDA화면

그림 3 게임이 진행중인 영상



(a) (b) (c) (d) (e)

그림 4 바람의 방향: (a) 왼쪽으로 강하게 부는 바람, (b) 왼쪽으로 약하게 부는 바람, (c) 무풍(無風), (d) 오른쪽으로 약하게 부는 방향, (e) 오른쪽으로 강하게 부는 방향

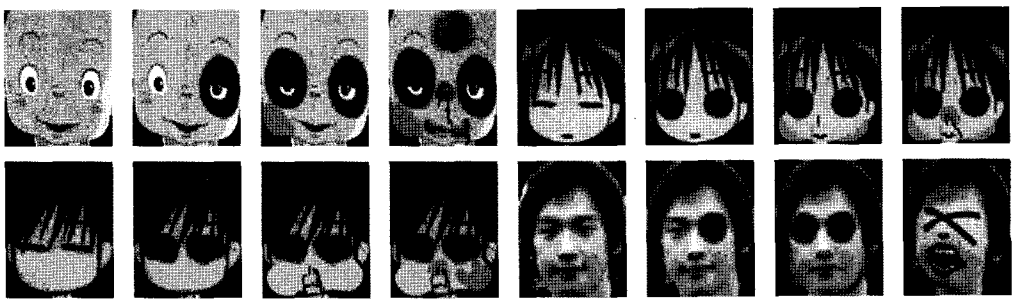


그림 5 가상 캐릭터의 예

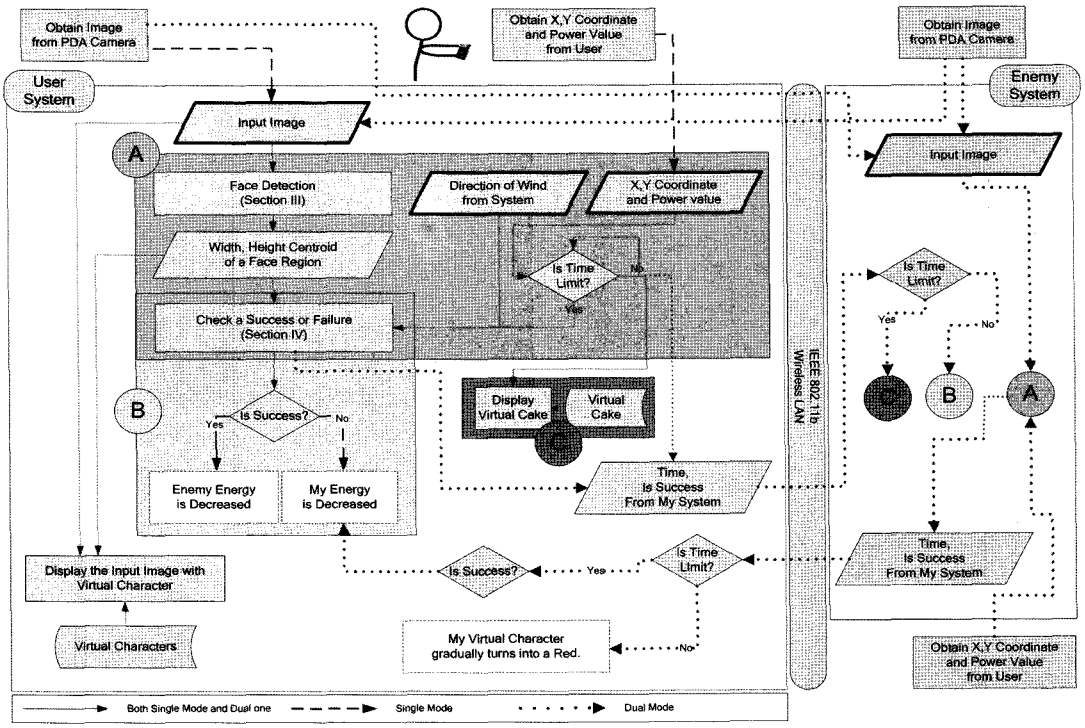


그림 6 전체 흐름도

(yaw), 상하방향(pitch), 회전(roll)을 고려하지 않는다는 가정 하에 3차원상의 정합을 2차원적으로 접목하여 게임을 수행한다.

그림 6은 *Flying Cake*의 전체 흐름도를 보여준다. *Flying Cake*는 3가지 입력 값을 받는다. 첫째 PDA 카메라에서 실제 환경을 영상으로 입력 받고, 둘째 사용자가 공격을 수행할 때 사용자로부터 공격할 위치의 x, y 좌표와 깊이 정보에 영향을 줄 파워(power) 값을 입력 받으며[2], 셋째 게임에서 자동으로 바람의 방향을 입력 받는다. 입력 영상에서 얼굴 추출(face detection)의 결과를 이용하여 가상 캐릭터를 접목하며, 사용자에게 적의 공격을 알려주기 위해 케이크가 날아오는 시간(time) 만큼 내 캐릭터의 색을 변화시키고, 공격의 성공여부는 입력영상의 3차원 정보와 사용자에 의해 입력 받은 값 의해 구할 수 있다(3D information). 싱글 모드는 사용자의 PDA 카메라에서 획득한 영상을 이용하여 게임을 수행하며, 싱글 모드에서 공격이 실패할 때 에너지가 줄어드는 것은, 적의 공격이라는 개념이 없기 때문이다. 듀얼 모드는 적의 PDA 카메라에서 획득한 영상을 이용하여 게임을 수행하며, 듀얼 모드에서 적의 시스템으로부터 적의 공격 성공여부(IsSuccess)를 전달받는 것은, 사용자의 시스템에서 입력받은 영상이 적의 시스템으로

2) x, y 좌표는 공격을 위해 PDA 화면을 클릭할 때 획득되며, 클릭한 시간만큼 파워 값은 증가한다.

전달되고 공격 성공여부 결정 역시 적의 시스템에서 수행하기 때문이다.

3. 얼굴 추출(Face Detection)

*Flying Cake*는 PDA 카메라에 의해 입력된 영상의 얼굴 영역에 가상의 캐릭터를 접목한다. PDA에서 사용하는 대부분의 CPU는 실수연산 구성요소가 없는 정수형 CPU를 사용하며, 실수연산이 많은 비전(vision) 시스템이나 영상처리를 수행하는데 오랜 수행시간이 소요된다[16]. 우리는 연산자원이 부족한 PDA에서 빠르게 얼굴영역을 추출하기 위해, 기존 스킨칼라 모델[12]과 스킨칼라 모델의 전역 탐색에 의한 수행 시간을 줄이기 위해 CAMShift 알고리즘[13]을 함께 이용한다.

3.1 스킨칼라 모델

우리는 얼굴 영역을 추출하기 위해 픽셀기반의 스킨칼라 모델을 이용하며, 모델링을 위한 칼라공간(color space) 선택 단계와 스킨칼라 분포를 모델링하는 단계로 이루어져있다. 우리는 칼라공간으로 PDA에서 빠른 수행을 위해 색상 성분의 개수가 적고, 변환식이 간단한 그리고 밝기값[3]에 영향을 덜 받는 정규화된 RGB 모델(noramalized RGB)을 이용하며, 다음 식과 같이 기술한다.

$$r=\frac{R}{R+G+B},\quad g=\frac{G}{R+G+B},\quad b=\frac{B}{R+G+B} \tag{1}$$

여기서 r과 g값만을 이용하며, 스킨칼라 분포를 모델링하기 위해 다음 식과 같은 베이지안 분류기(Bayesian classifier)을 이용한다[17,18].

$$P(skin\,|\,\mathbf{c})=\frac{P(\mathbf{c}\,|\,skin)P(skin)}{P(\mathbf{c})} \tag{2}$$

베이지안 분류기를 이용해서 스킨칼라 분포를 모델링할 때는 두 가지 경우가 있다.[4] 첫째 칼라공간에서 얼굴영역일 확률과 얼굴영역이 아닐 확률이 같은 경우 ($P(skin)=P(\sim skin)$)이며(maximum likelihood estimation[19]), 이때는 $P(\mathrm{c}|skin)$만을 이용해서 모델링하다. 둘째 다른 경우(maximum a posteriori estimation [19]), 다음 식 (3,4)와 같이 비율을 이용하며 1보다 크면 얼굴 픽셀, 작으면 비-얼굴 픽셀로 구분한다.

$$\frac{P(skin\,|\,\mathbf{c})}{P(\sim skin\,|\,\mathbf{c})}=\frac{P(\mathbf{c}\,|\,skin)P(skin)}{P(\mathbf{c}\,|\sim skin)P(\sim skin)} \tag{3}$$

$$\frac{P(\mathbf{c}\,|\,skin)}{P(\mathbf{c}\,|\sim skin)}>\frac{1-P(skin)}{P(skin)} \tag{4}$$

우리는 칼라공간에서 얼굴영역일 확률과 아닐 확률이 같다는 가정하에, $P(\mathrm{c}|skin)$만을 이용해서 스킨칼라 분포를 모델링하며, 다음과 같이 가우시안 확률 밀도 함수(Gaussian probability density function)를 이용한다[17].

$$\boldsymbol{\mu}=(\mu_r,\mu_g),\quad \mu_r=\frac{1}{N}\sum_{i=1}^{N}r_i,\mu_g=\frac{1}{N}\sum_{i=1}^{N}g_i \tag{5}$$

$$\boldsymbol{\Sigma}=\begin{bmatrix}\sigma_{rr} & \sigma_{rg}\\ \sigma_{gr} & \sigma_{gg}\end{bmatrix} \tag{6}$$

$$p(\mathbf{c}\,|\,skin)=\frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}}e^{-\frac{1}{2}(\mathbf{c}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{c}-\boldsymbol{\mu})} \tag{7}$$

N은 총 픽셀의 수, $\mathbf{c}$는 입력 벡터(r,g)이며, $\boldsymbol{\mu},\boldsymbol{\Sigma}$는 스킨칼라 분포의 매개변수(parameter)로써, 평균 벡터와 공분산 행렬(covariance matrix)을 나타낸다. 그림 7은 스킨칼라 분포를 모델링하기 위해 다양한 환경에서 획득한 샘플들이며, 그림 7(a)는 실내 환경에서 조명이 어두운, 그림 7(b)는 조명이 밝은 곳에서, 그림 7(c)는 실외 환경에서 획득한 샘플의 예이다. 표 2는 샘플들을 기반으로 계산한 매개변수이며, 그림 8은 모델링 된 스킨칼라 분포의 그래프를 보여준다.

3.2 CAMShift 알고리즘

스킨칼라 모델을 이용해서 얼굴 영역을 추출하면 입력 영상 전체를 수행해야 하기 때문에 오랜 수행시간이 소요되는 단점이 있다. 우리는 이런 단점을 해결하기 위해 CAMShift 알고리즘을 이용한다(그림 9). 그림 9에서 x, y는 x, y축, 0, (t-1), t는 실행순서(iteration), $L\varepsilon_x$, $L\varepsilon_y$, $S\varepsilon_x$, $S\varepsilon_y$는 임계값을 나타낸다.

CAMShift 알고리즘은 시작 단계에서 검색창의 초기 중심 좌표($mean_{x0}$, $mean_{y0}$)와 크기($width_0$, $height_0$)를 결정하고, 연속된 일련의 단계에서 2차원 모멘트(moment)를 계산하여 얼굴 영역의 크기와 위치를 구하며, 검색창의 크기를 얼굴 영역의 크기에 비례해서 수정한다. 이차원 p+q차 모멘트는 다음과 같이 기술할 수 있으며, $FPI(x,y)$는 x, y좌표의 픽셀이 얼굴 픽셀이면 1, 비-얼굴 픽셀이면 0을 나타낸다.

$$M_{pq}=\sum_x\sum_y x^p y^q FPI(x,y) \tag{8}$$

얼굴 영역의 중심 좌표(sample mean location)는

$$mean_{x_t}=\frac{M_{10}}{M_{00}},\quad mean_{yt}=\frac{M_{01}}{M_{00}} \tag{9}$$

로 설정할 수 있고, 얼굴 영역의 폭(width)과 높이(height)는 다음과 같이 표현할 수 있다.

$$width_t=\sqrt{2(a+c)+2\sqrt{b^2+(a-c)^2}},$$

$$height_t=\sqrt{2(a+c)-2\sqrt{b^2+(a-c)^2}} \tag{10}$$

3) 피부색은 밝기값에 매우 민감하다[12].

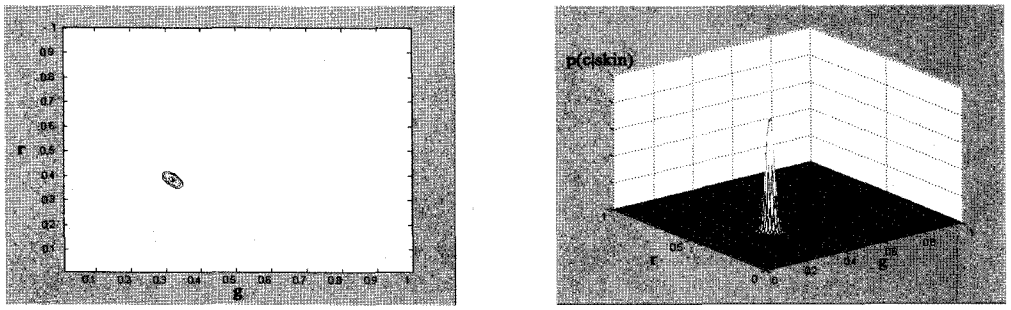
4) $P(\mathrm{c})$는 입력 벡터(r,g)가 나타날 확률이며, 베이지안 분류기의 특성에 의해 고려하지 않는다.

(a) (b) (c)

그림 7 스킨칼라 분포를 모델링하기 위해 사용한 샘플의 예

표 2 다양한 환경에서 획득한 샘플들을 기반으로 계산한 매개변수

	μ_r	μ_g	σ_{rr}	σ_{rg}	σ_{gr}	σ_{gg}
값	0.383383	0.321816	0.000710	-0.000184	-0.000184	0.000263



(a) (b)

그림 8 스킨칼라 모델의 가우시안 분포도: (a) 2차원 그래프, (b) 3차원 그래프

■ Set up the initial location ($mean_{x0}$, $mean_{y0}$) and size ($width_0$, $height_0$) of search window W
■ Do
- Generate the face probability image(FPI) within W using skin-color model.
- Based on the mean shift vector, derive the new location ($mean_{xt}$, $mean_{yt}$) and size ($width_t$, $height_t$) of the face region
- Modify W according to the derived values.
- Increment the iteration number t.

While(($\|mean_{xt}-mean_{x(t-1)}\|>L\varepsilon_x$ or $\|mean_{yt}-mean_{y(t-1)}\|>L\varepsilon_y$) or ($\|width_t-width_{(t-1)}\|>S\varepsilon_x$ or $\|height_t-height_{(t-1)}\|>S\varepsilon_y$))

그림 9 얼굴 추출을 위한 CAMShift 알고리즘

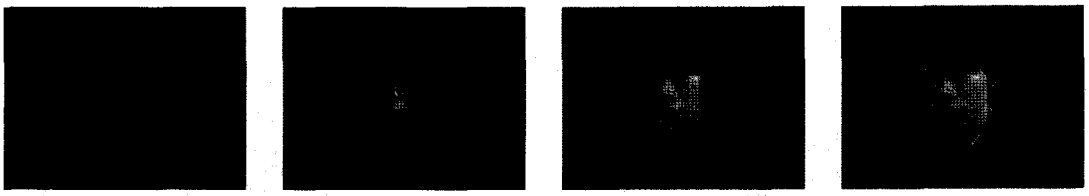
이때 $a=\frac{M_{20}}{M_{00}}-\left(\frac{M_{10}}{M_{00}}\right)^2$, $b=2\left(\frac{M_{11}}{M_{00}}-\frac{M_{10}M_{01}}{M_{00}^2}\right)$, $c=\frac{M_{02}}{M_{00}}-\left(\frac{M_{01}}{M_{00}}\right)^2$

이다.

얼굴 영역의 폭과 높이를 계산한 후에, 얼굴영역의 중심좌표와 크기에 비례하여 검색창의 크기를 변경하며, 미리 정한 임계값보다 검색창의 크기가 변하지 않을 때까지 반복 수행한다.

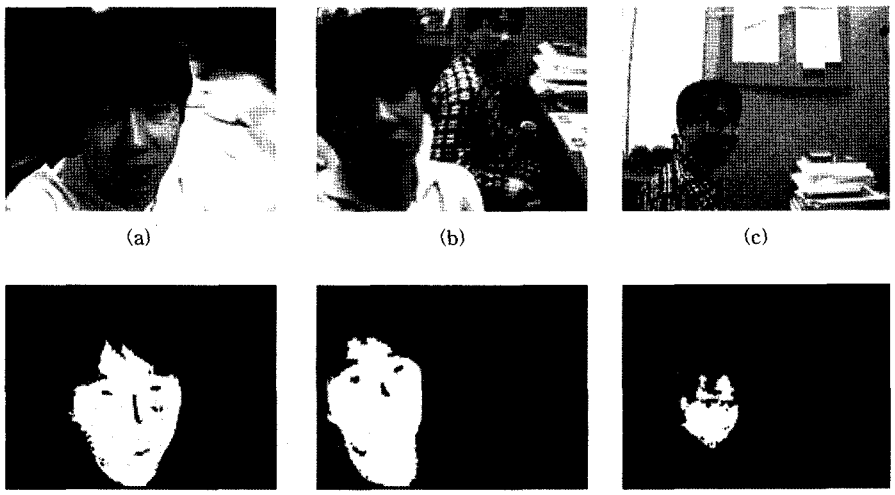
그림 10은 검색창의 변화에 따라 추출된 얼굴 영역이며, 그림 11은 CAMShift 알고리즘을 이용하여 추출된 얼굴 영역 결과 영상이다. 그림 11(a-c)는 입력 영상이며, 그림 11(d-f)는 얼굴 영역은 흰색, 비-얼굴 영역은 검은색으로 표시된 결과 영상이다. 입력 영상 11(b)는 두 개의 얼굴 영역을 가지지만, 결과 영상 11(e)를 보면 하나의 얼굴 영역만을 추출한다. 이것은 미리 정한 임계값보다 검색창의 변화가 작기 때문에[5] CAMShift 알고리즘이 더 이상 수행되지 않아 생긴 결과이며, 이전 프레임에서 결정된 얼굴 영역의 위치(그림 11(d))가 다음 영역의 초기 값으로 들어가기 때문에 왼쪽 얼굴 영역이 추출된다. 결과적으로, CAMShift 알고리즘은 입력 영상 전역 탐색에 관한 문제를 해결하며, 영상에서 얼굴 영역이 많이 차지하지 않을 때 계산량을 줄일 수 있다.

5) 반복 수행에서 왼쪽 얼굴영역이 먼저 추출되고, 변화된 검색창에서 왼쪽 얼굴과 오른쪽 얼굴의 간격 때문에 오른쪽 얼굴영역이 추출되지 않으며, 이로 인해 얼굴영역의 중심좌표와 크기가 변하지 않는다.



(a) (b) (c) (d)

그림 10 검색창의 크기 변화에 따라 추출된 얼굴 영역: (a)부터 (d) 순서



(a) (b) (c)

(d) (e) (f)

그림 11 얼굴 추출 결과: (a-c) 입력 영상, (d-f) 추출 결과 영상

4. 3차원 정보

4.1 가상 캐릭터 접목

실감 현실 시스템은 실제 환경에 가상의 물체를 3차원적으로 정합해야 한다. *Flying Cake*는 입력 영상의 얼굴 영역에 가상 캐릭터를 접목하며, 3차원적으로 정합하기 위해서는 얼굴 영역의 3차원 정보를 알아야 한다. 그러나 얼굴 영역의 3차원 정보를 알기란 매우 복잡하고 어려우며 추가적인 수행시간이 소요된다. PDA에서 게임의 빠른 수행을 위해 3차원 정합에서 가상 캐릭터의 좌우방향, 상하방향, 회전을 고려하지 않는다는 가정하에 접목된 가상 캐릭터는 PDA 화면에서 2차원으로 보이기 때문에(그림 12), 실제 환경에서 가상 캐릭터를 접목할 때 얼굴 추출의 결과인 폭, 높이, 중심값만을 이용하여 2차원적으로 접목한다.

공격에 사용하는 케이크 역시 2차원적으로 접목한다. 케이크는 포물선을 그리며 날아가며, 그림 13은 가상의 케이크를 접목하기 위한 순서도(flow chart)이다. 케이크의 시작 위치는 입력 영상의 하단 중앙이며(X_Coord =IMG_HEIGHT, Y_Coord=IMG_WIDTH/2), x 좌표는

(a) (b) (c)

그림 12 가상 캐릭터가 접목된 입력 영상

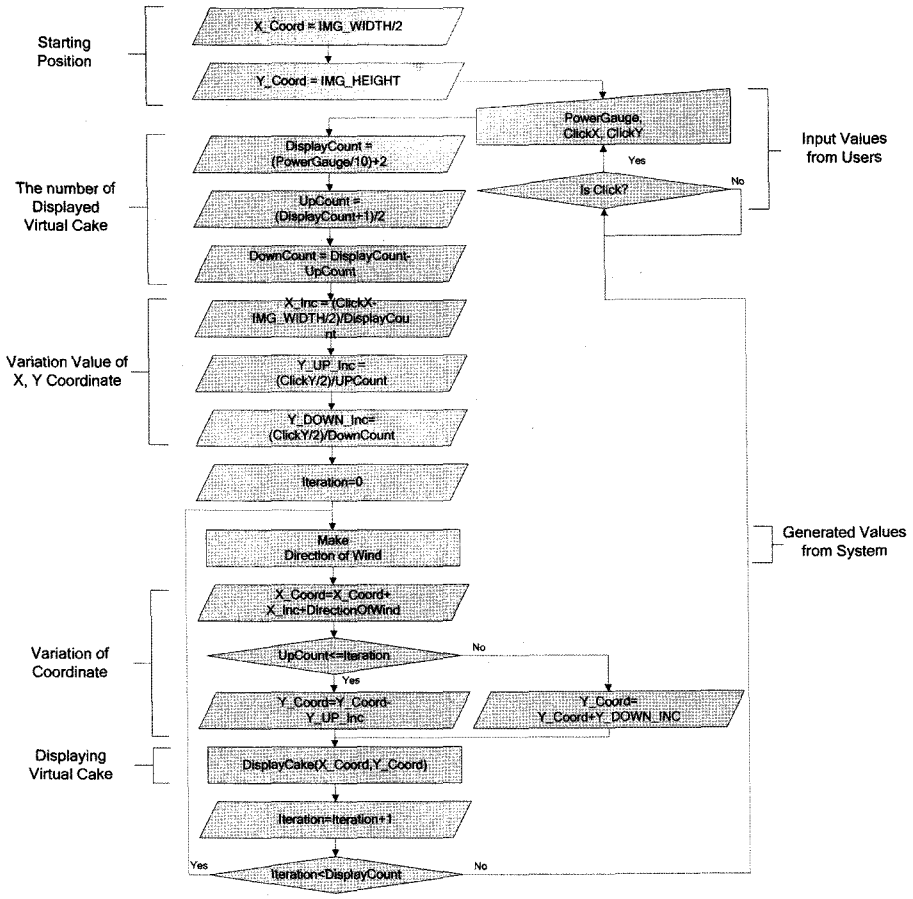
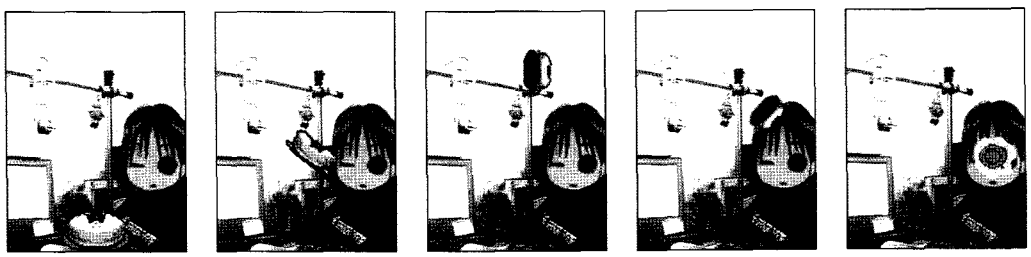


그림 13 가상의 케이크를 접목하기 위한 순서도



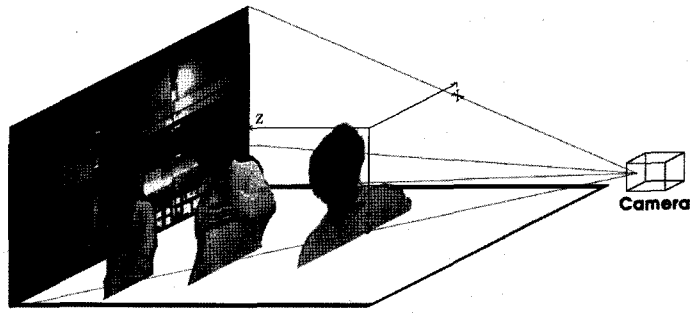
(a) (b) (c) (d) (e)

그림 14 연속 영상에서 케이크의 움직임: (a)부터 (e)까지

케이크가 공격 시 날아가는 프레임 수(DisplayCount)에 영향을 받는다(X_Inc=(ClickX-IMG_WIDTH)/2)/DisplayCount). y 좌표는 날아가는 프레임 수를 반으로 나누어((DisplayCount+1)/2) 올라갈 때와 내려갈 때로 구분하며, 올라갈 때는 y 좌표를 감소시키고(Y_Coord=Y_Coord-Y_UP_Inc), 내려올 때는 좌표를 증가시킨다(Y_Coord=Y_Coord+ Y_DOWN_Inc). 그림 14는 연속 영상에서 가상 케이크의 움직임을 보여주는 영상이다.[6] 그림 14(a)는 케이크의 시작 위치이며, 그림 14(a-c)는 올라갈 때, 그림 14(d-e)는 내려올 때이다. 그림 14(c)는 케이크가 날아가는 최고 높은 위치이며, 사용자에 의해 입력된 y 좌표와 입력 영상의 상위(y 좌표가 0인) 좌표의 중심값으로 계산한다.

6) 사용자에 의해 입력된 x, y 좌표는 얼굴의 중심이고, 파워게이지는 20에서 40사이이며, 시스템에 의해 입력된 바람의 정보는 무풍이다.

(a)



(b)

그림 15 2차원 입력 영상을 3차원 좌표계로 변환한 영상: (a) 2차원 입력 영상, (b) 3차원 좌표계 영상

4.2 공격성공 여부

*Flying Cake*에서 공격을 수행할 때 케이크는 3차원 공간에 포물선을 그리며 날아가며, 사용자에 의해 입력된 x, y 좌표와 파워 값에 의해 날아갈 위치가 결정된다. 3차원 공간에서 날아오는 케이크의 공격성공 여부를 확인하기 위해서는 PDA에서 입력된 2차원 영상의 3차원 정보가 필요하다. 그림 15는 2차원 입력 영상을 3차원 좌표계로 변환한 영상을 보여준다. 3차원 정보의 x, y 좌표는 CAMShift 알고리즘에서 얻어진 얼굴 영역의 중심값을 이용하였으며, 우리는 추가적으로 깊이 정보(z 좌표)가 필요하다. 그림 15에서 우리는 깊이 정보가 얼굴 영역의 크기와 비례함을 알 수 있다. 이 가정을 기반으로 우리는 얼굴 영역의 크기를 이용하여 대략적인 깊이 정보를 구한다. 우리는 PDA상에서 복잡한 계산을 피하고 빠른 수행을 위해 얼굴 영역의 크기에 비례해서 깊이 정보를 5단계로 구분(그림 16)하고, 얼굴 영역의 크기와 파워게이지를 이용해서 공격의 성공과 실패를 구분한다. 그림 16은 파워게이지에 의해 날아가는 케이크의 범위를 보여주며, 얼굴 영역의 크기와 파워게이지의 관계(relationship)[7]를 보여준다.

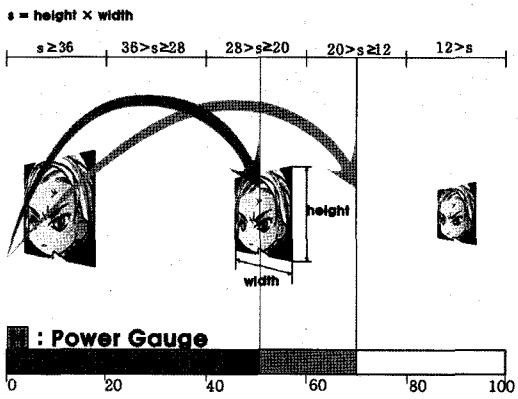
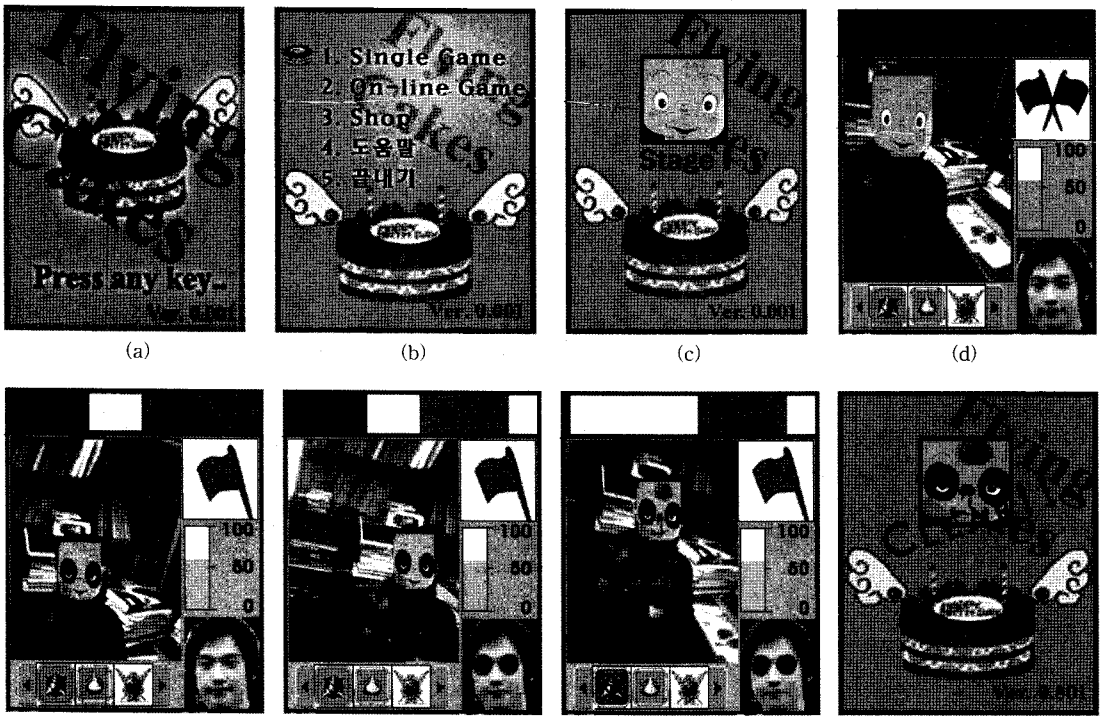


그림 16 발사대의 파워게이지에 의해 케이크가 날아가는 깊이 범위

5. 실험 및 결과

그림 17은 PDA화면에 보여지는 *Flying Cake*이다. 그림 17(a)는 시작 화면, 그림 17(b)는 메뉴를 선택하는 화면, 그림 17(c)는 싱글 모드일 때 Stage 1을 위한 시작 페이지이며, 그림 17(d-g)는 게임이 진행중인 화면이다. 그림 17(f)에서 내 에너지가 줄어든 것은 일정시간 동안 공격을 하지 않기 때문이며, 그림 17(g)는 아이템을 사용하여 공격한 화면으로 폭탄 아이템이 비활성화 되어있음을 보여주며, 에너지도 많이 줄어들었음을 확인할 수 있다. 그림 17(h)는 적의 에너지가 모두 소모되어 Stage 1이 종료되었음을 보여준다. 현재는 총 4개

7) 예를 들어 얼굴 영역의 크기가 20에서 28사이이고, 파워게이지가 40에서 60사이이면 공격은 성공한다.



(a) (b) (c) (d)

(e) (f) (g) (h)

그림 17 PDA화면에 보여지는 *Flying Cake*

의 Stage로 구성되어 있으며, 앞으로 더욱 많은 캐릭터와 아이템을 이용하여 게임을 더욱 재미있는 게임을 제공할 예정이다.

우리는 실험 및 결과를 두 가지로 분류하여 기술한다. 첫째, 연산 자원이 부족한 PDA에서 수행되는 게임 측면에서의 실험 및 결과로써, 사용한 시스템, 게임이 수행되는 속도 등을 기술한다. 둘째, 사용자 측면에서의 실험 및 결과로써, 사용자가 직접 경험을 하고 우리가 제시한 질문의 답을 분석한다.

5.1 *Flying Cake*에 대한 실험 및 결과

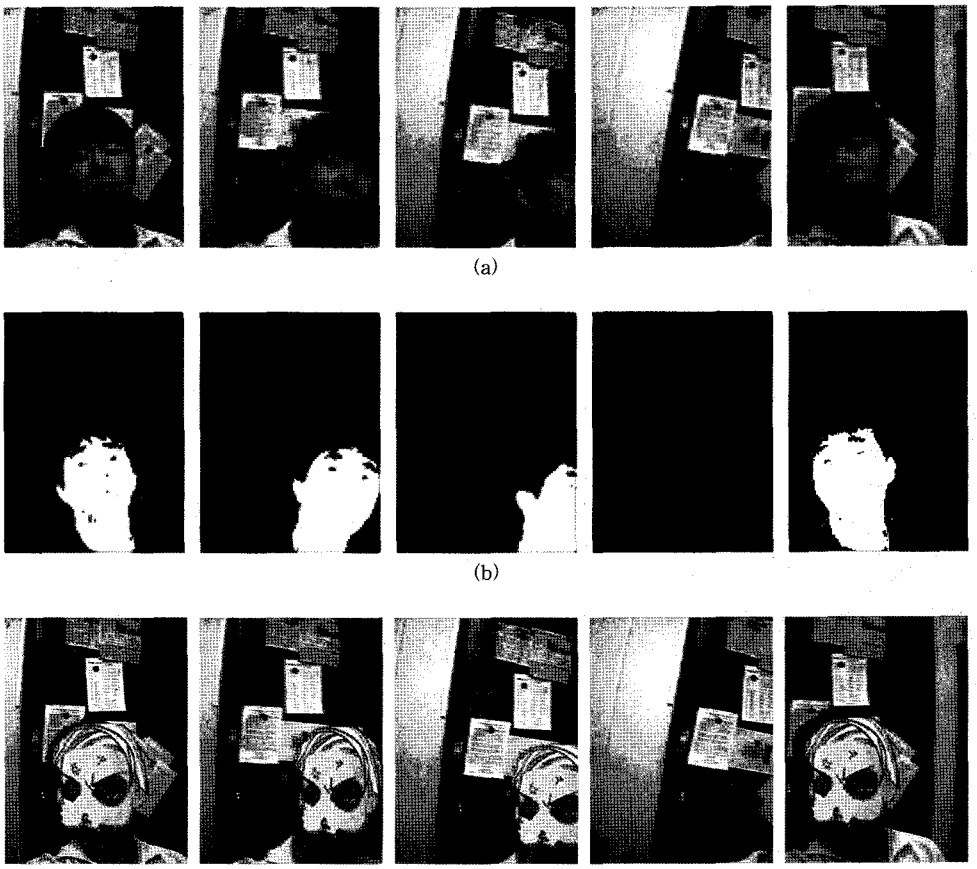
본 실험에서 사용한 PDA는 Pocket PC 2003기반의 카메라가 장착된 POZ x301을 사용하였다. POZ x301 모델은 400MHz 인텔 프로세서를 사용하며, 64MB SDRAM/120MB Flash ROM을 사용한다. 카메라는 32만 화소(pixel)이다.

그림 18은 연속된 입력 영상에 대해 PDA 화면에 보이는 상대방(적)의 영상이다. 그림 18(a)는 입력 영상, 그림 18(b)는 얼굴 영역을 흰색, 비-얼굴 영역을 검은색으로 표시한 얼굴 추출 결과 영상, 그림 18(c)는 PDA 화면에 보이는 영상이다. 표 3은 PDA에서 얼굴 영역을 추출하는 평균 수행시간을 보인다.[8] 표 3에서 보는 바와 같이 스킨칼라 모델과 CAMShift 알고리즘(1×1 간격)[9]을 같이 사용하면, CAMShift 알고리즘을 사용하지 않는 방법(스킨칼라 모델)보다는 약 4배정도 빠르게 수행되지만, 이 수행시간은 PC 게임에 익숙한 사용자에게 매우 답답한 느낌을 주게 된다.

우리는 빠른 수행시간을 위해 검색창 내에서 3×3 간격으로 얼굴 영역을 추출하며, 그림 19는 3×3 간격으로 수행하는 CAMShift 알고리즘을 보여준다. 한 픽셀에 대해 스킨칼라 모델을 이용하여 얼굴 영역인지 판별하고 3칸 이동해서 다음 픽셀을 판별하며(그림 19(a)), 만약 얼굴 영역 픽셀이면 3×3 모폴로지 연산(morphological operator)을 이용하여 주위 8개 픽셀을 모두 얼굴 영역으로 채운다(그림 19(b)). 그림 19(a)에서 1, 2, 3, 4는 스킨칼라 모델을 수행하는 픽셀이며, 그림 19(b)는 그림 19(a)의 1과 4픽셀이 얼굴 영역으로 판별되어 주위 3×3 영역이 모두 얼굴 영역(흰색)으로 표시된 영상이다. 그림 20은 실제 입력 영상에 대해 3×3 간격으로 얼굴 영역을 추출한 결과이다.

8) 연속되는 여러 영상에서 측정된 결과의 평균이며, 그림 21은 수행시간을 측정하기 위한 여러 연속영상 중 한 연속영상의 예와 수행시간을 보여준다.

9) 검색창의 모든 픽셀에 대해 얼굴 영역을 추출하는 경우이다.

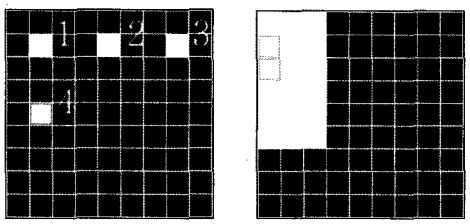


(a) (b) (c)

그림 18 연속된 입력 영상에 대해 PDA 화면에 보이는 결과 영상: (a) 입력 영상, (b) 얼굴영역을 흰색으로 표시한 얼굴 추출 결과 영상, (c) PDA 화면에 보이는 영상

표 3 PDA에서 얼굴이 추출되는 평균 수행시간

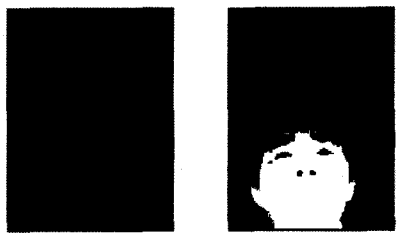
	스킨칼라 모델	CAMShift 알고리즘 (1×1 간격)	CAMShift 알고리즘 (3×3 간격)
시간(ms)	5000	900	150



(a) (b)

그림 19 3×3 간격으로 수행하는 CAMShift 알고리즘: (a) 얼굴 영역을 추출하는 픽셀, (b) 추출 결과에 의해 반응하는 픽셀

CAMShift 알고리즘에서 검색창의 중심 좌표와 크기



(a) (b)

그림 20 실제 영상에 대해 3×3 간격으로 얼굴 영역을 추출한 영상: (a) 3×3 간격, (b) 주위 3×3영역을 모두 얼굴 영역을 표시한 영상

를 구하기 위해서는 2차원 모멘트를 이용하며, 검색창 내에서 3×3 간격으로 얼굴 영역을 추출할 때 2차원 모멘트는 다음과 같이 바꾸어서 계산한다.

$$M_{pq} = \sum_{j=-1}^{1}\sum_{k=-1}^{1}(x+j)^p(y+k)^q FPI(x+j,y+k) \quad (11)$$

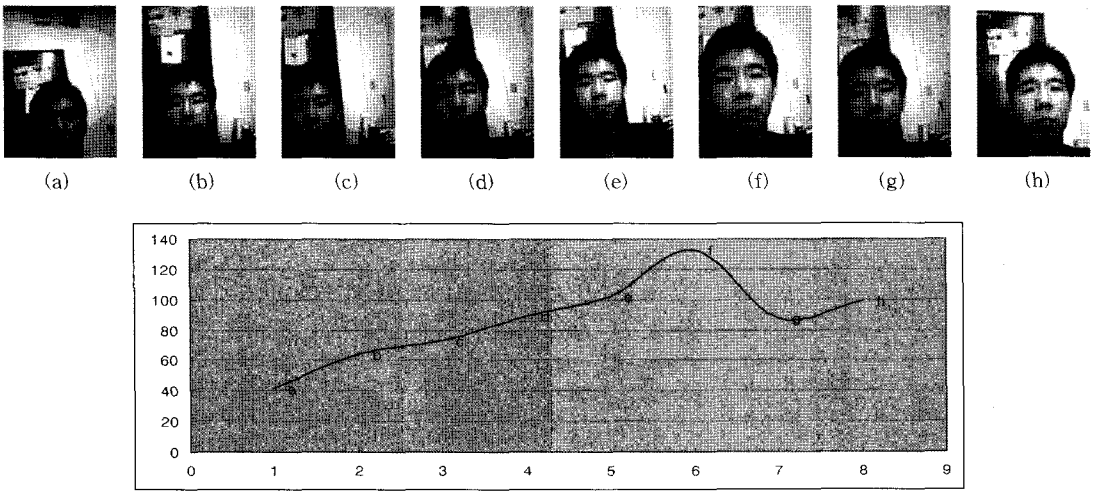


그림 21 연속 영상에서 얼굴 크기변화에 따른 CAMShift 알고리즘의 수행시간: (a-i) 입력영상, (j) 수행시간

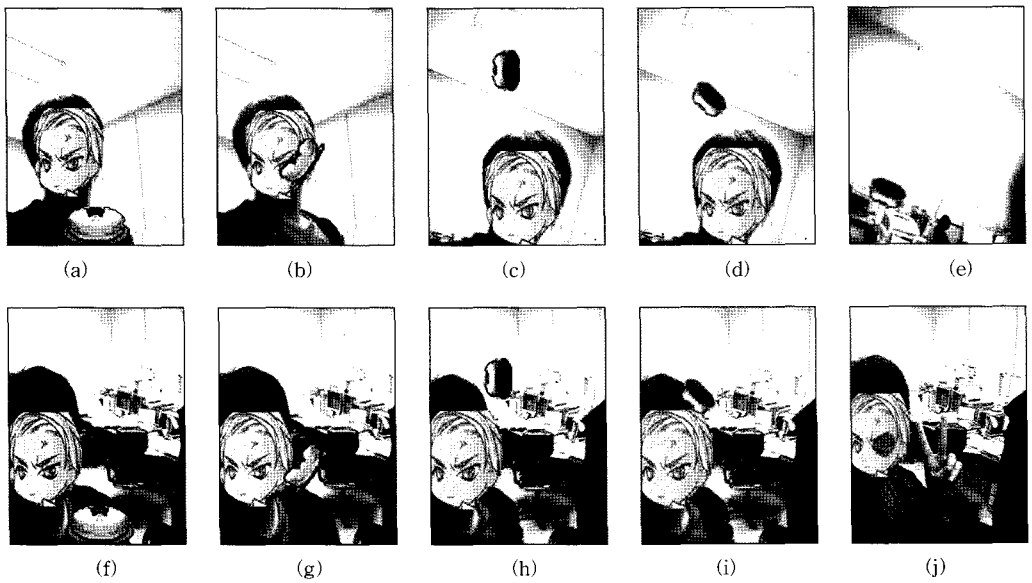


그림 22 연속 영상에서 가상 케이크의 움직임: (a-e) 공격 실패, (f-j) 공격 성공

픽셀 단위로 얼굴 추출률을 평가(표 4)하였으며, 아래의 두 식(정확도(precision)(식 (12)), 검출률(recall)(식 (13))을 이용하였다. CAMShift 알고리즘을 이용하면 검색창 내에서만 얼굴영역을 추출하기 때문에 검출률이 떨어지는 결과를 보이지만, 초기 검색창내에 얼굴색깔과 유사한 다른 물체가 아닌 실제 얼굴영역이 추출되면, 반복수행에서 높은 정확도를 보인다.[10]

표 4 얼굴 추출에 대한 정확도와 검출률 비교

	스킨칼라 모델	CAMShift 알고리즘
precision(%)	74.7	87.3
recall(%)	85.2	67.5

$$\text{precision}(\%) = \frac{\#\text{ of correctly detected face pixels}}{\#\text{ of detected face pixels}} \times 100 \quad (12)$$

$$\text{recall}(\%) = \frac{\#\text{ of correctly detected face pixels}}{\#\text{ of face pixels}} \times 100 \quad (13)$$

10) 정확도와 검출률을 평가할 때, 초기에는 실제 얼굴영역이 추출되었다고 가정한다.

표 5 사용자 평가를 위한 질문들

i.	*Flying Cake*를 포트리스와 비교했을 때의 오락성은? 1(포트리스가 더 오락성이 있다)~7(*Flying Cake*가 더 오락성이 있다) 사이 값으로 표시 바랍니다.
ii.	PDA에서 실제 세계를 돌아다니며 하는 게임이 편합니까?
iii.	가상의 캐릭터를 가상의 케이크로 공격하는 방법에 대한 당신의 생각은?
iv.	키보드를 이용한 포트리스와 비교했을 때, 실제 세계를 배경으로 한 *Flying Cake*의 흥미도? 1(낮다)~7(높다) 사이 값으로 표시 바랍니다.
v.	사람 얼굴에 증강된 가상의 캐릭터에 대해서 어떻게 생각합니까?
vi.	*Flying Cake*를 플레이하는 게 재미있습니까?
vii.	다른 컴퓨터 게임과 비교했을 때 어떻습니까?
viii.	서바이벌 경기장에서 서바이벌 형식으로 *Flying Cake*를 제공한다면 이용하겠습니까?
ix.	*Flying Cake*를 더욱 향상시킬 수 있는 방법과 단점에 대해 기술바랍니다.

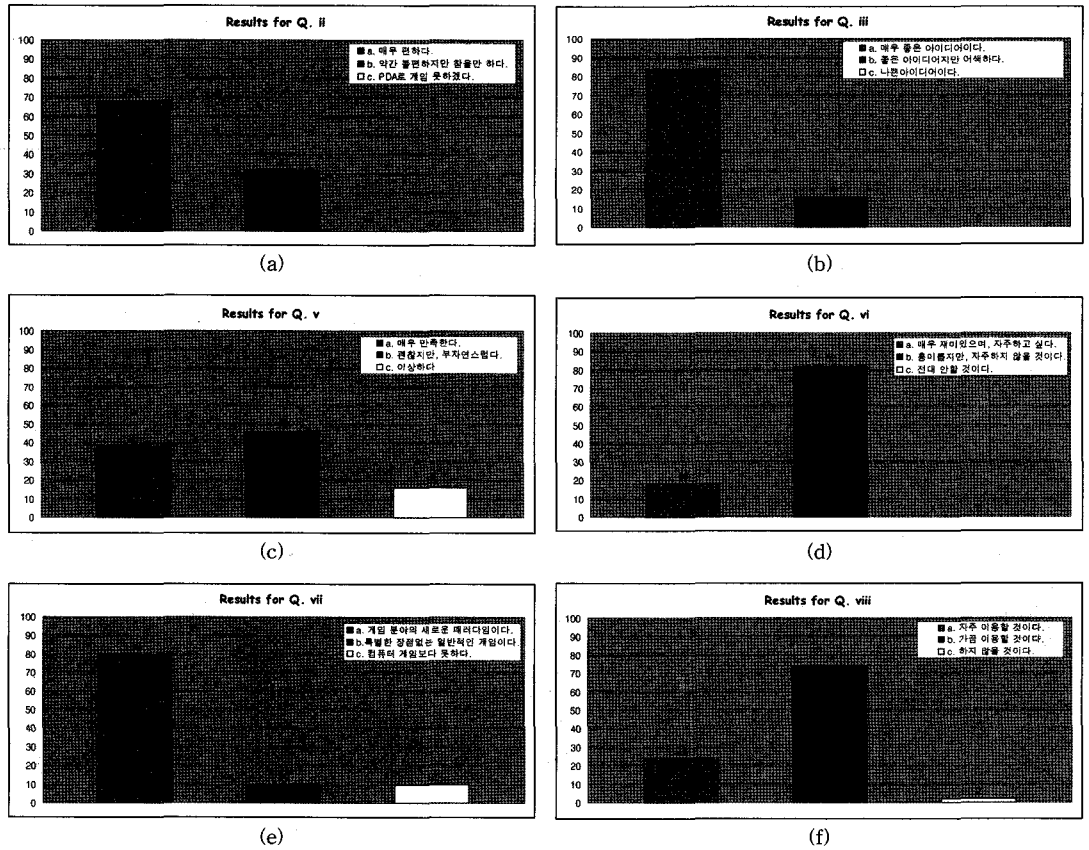


그림 23 질문 i, iv를 제외한 사용자 평가에 대한 결과

그림 22는 연속 영상에서 가상 케이크의 움직임을 보여주며, 그림 21(a-e)는 공격 실패, 그림 22(f-j)는 공격 성공을 나타낸다. 그림 22(j)는 가상 케이크의 공격에 의해 멍이 든 가상 캐릭터를 보여준다.

5.2 사용자에 의한 게임 분석(User Study)

우리는 사용자로부터의 평가를 얻기 위해 설문조사를 실시하였다(표 5, Appendix 1).[11] 사용자 평가를 위해 50명이 참석하였으며, 이들은 19세에서 35세 사이이고, 남자 35명, 여자 15명으로 구성되어 있다. 실험을 위한 사전 단계로 일반 컴퓨터에서 약 10분 정도 포트리스를 하고, 그리고 게임을 이해시켜주기 위해 약 3분 정도 동영상을 보며 게임을 설명하였으며, 약 10분 정도 참석자가 직접 게임을 진행하였다. 표 4는 사용자 평가를 위한 질문들이며, 그림 23은 i, iv를 제외한 사용자 평가에 대한 결과이다.

설문조사의 목적은 *Flying Cake*의 흥미도(표 4((i,iv, vi))와 아이디어(표4(ii,iii,v,vii,viii))를 알아보기 위함이

11) 우리는 Cheok[10] 등이 사용한 사용자 평가를 바탕으로 설문조사 및 평가를 수행하였다.

다. 먼저 흥미도 측면에서 포트리스와 비교했을 때의 오락성(질문 i)에 대한 답변으로 평균 3 정도(분산 1.5) 표시하였고, 기존의 포트리스보다는 오락성에서 떨어지는 결과를 보였으며, 오락적인 측면에서 게임을 발전했으면 좋겠다는 답변이 있었다. 하지만 실제 세계를 배경으로 한 *Flying Cake*의 흥미도에 대해서는 평균 4.5(분산 1)로 다소 높은 점수를 주었지만 아직 키보드에 익숙한 사용자에게 스타일러스 펜으로 게임을 즐기는 것에 대해 약간의 거부감을 표시하였다. 이런 단점으로 게임에 대한 아이디어는 흥미롭지만 자주하지 않을 것이라고 대다수(82%)의 사용자가 답변하였다(질문 vi). 아이디어 측면에서 PDA를 들고 실제 환경을 돌아다니면서 하는 게임에 대해 68%는 만족하였으나, 32%는 작은 화면에 대한 불편, 상대방(적)이 뛰어서 움직일 때 불편함을 지적하였다. 가상 케이크를 이용하여 실제 사람의 얼굴 영역에 접목된 가상 캐릭터를 공격하는 점에 대해 다수의 사용자(84%)가 좋은 아이디어라고 체크해 주었으며, 몇몇 이용자(16%)는 한 종류뿐인 가상 케이크의 식상함을 지적하였다. 그러나 가상 캐릭터에 대해서는 38%만이 만족함을 표시했고, 2차원적으로 접목된 가상 캐릭터의 부자연스러움과 얼굴영역이 아닌 얼굴 색깔과 유사한 다른 곳에서 캐릭터가 접목되는 부 정확성을 지적하였다(질문 v). 더욱 게임성 부여를 위해 계획중인 서바이벌 형식의 *Flying Cake*에 대한 질문의 답으로 24%만이 자주 이용할 것이라고 답하였으며, 빠르게 움직이는 적에 대한 공격이 어렵다라는 단점을 지적해주었다. 그러나 대부분의 사용자들이 게임의 새로운 패러다임으로 위의 단점들만 보완한다면 더욱 재미있는 게임으로 발전해갈 것 같으며, 이런 패러다임을 더 많은 아이템에 적용할 수 있을 것이라고 답변해주었다.

6. 결 론

기존의 실감형 게임은 비싸고, 거추장스러우며, 사용하기 불편한 'backpack' 시스템과 미리 지정되어 있는 장소에서만 가상의 오브젝트 생성이 가능한 패턴마커를 이용하였다. 우리는 실감형 게임 *Flying Cake*를 제안하였으며, 거추장스러운 장비 대신 PDA를 이용하였고, 실제 세계에서 가상의 물체를 접목할 위치를 미리 지정하는 패턴마커 대신 얼굴 영역을 이용하였다. *Flying Cake*는 적은 연산 자원을 가진 PDA상에서 실시간으로 수행되며, 가상의 오브젝트와 실제 세계 사용자 사이의 상호작용을 제공하는 새로운 게임 패러다임을 제공함으로써 사용자에게 새로운 즐거움을 제공한다.

현재 게임의 싱글모드에서 적 개개인에 대한 에너지를 알 수 없다. 그래서 적의 에너지를 알기 위해 PDA의 연산 자원에 적합한 얼굴 인식, GPS를 이용한 싱글과 듀얼 모드의 결합과 같은 연구가 진행 중이며, 더욱 실감적인 게임을 위해 3차원 가상 캐릭터의 접목에서 얼굴 영역의 좌우방향, 상하방향, 회전에 대한 연구도 진행 중이다. 그리고, 사용자들이 지적한 단점들을 보완하여 더욱 게임성있는 게임으로 개발이 진행 중이다.

APPENDIX I

*Flying Cake*에 대한 설문지

i. *Flying Cake*를 포트리스와 비교했을 때의 오락성은? 1(포트리스가 더 오락성이 있다)~7(*Flying Cake*가 더 오락성이 있다) 사이 값으로 표시 바랍니다.

ii. PDA에서 실제 세계를 돌아다니며 하는 게임이 편합니까?
   a. 매우 편하다.
   b. 약간 불편하지만 참을만하다.
   c. PDA로 게임 못하겠다.

iii. 가상의 캐릭터를 가상의 케이크로 공격하는 방법에 대한 당신의 생각은?
   a. 매우 좋은 아이디어이다.
   b. 좋은 아이디어지만 어색하다.
   c. 나쁜 아이디어이다.

iv. 키보드를 이용한 포트리스와 비교했을 때, 실제 세계를 배경으로 한 *Flying Cake*의 흥미도는? 1(낮다)~7(높다) 사이 값으로 표시 바랍니다.

v. 사람 얼굴에 증강된 가상의 캐릭터에 대해 어떻게 생각합니까?
   a. 좋은 아이디어이다.
   b. 괜찮지만, 부자연스럽다.
   c. 이상하다.

vi. *Flying Cake*를 플레이하는 게 재미있었습니까?
   a. 매우 재미있으며, 자주하고 싶다.
   b. 흥미롭지만, 자주하지 않을 것이다.
   c. 절대로 안할 것이다.

vii. 다른 컴퓨터 게임과 비교했을 때 어떻습니까?
   a. 게임 분야의 새로운 패러다임이다.
   b. 특별한 장점 없는 일반적인 게임이다.
   c. 지금의 컴퓨터 게임보다 못하다.

viii. 서바이벌 경기장에서 서바이벌 형식으로 *Flying Cake*를 제공한다면 이용하겠습니까?

a. 자주 이용할 것이다.
b. 가끔 이용할 것이다.
c. 하지 않을 것이다.

ix. 게임을 더욱 향상시킬 수 있는 방법과 단점에 대해 기술바랍니다.

참 고 문 헌

[1] http://www.pbs.org/kcts/videogamerevolution/history/timeline_flash.html/
[2] http://www.gamemeca.com/
[3] http://www.inew24.com/
[4] http://www.thegames.co.kr/
[5] Z. Szalavari, E. Eckstein and M. Gervautz, "Collaborative Gaming in Augmented Reality," Proceedings of ACM Symposium on Virtual Reality Software and Technology, pp. 195-204, Nov. 1998.
[6] T. Ohshima, K. Satoh, H. Yamamoto and H. Tamura, "AR2Hockey: A Case Study of Collaborative Augmented Reality," Proceedings of IEEE Virtual Reality Annual International Symposiums, pp. 268-275, Mar. 1998.
[7] T. Starner, B. Leibe, B. Singletary and J. Pair, "MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game," Proceedings of International Conference on Intelligent User Interface, pp. 256-260, Jan, 2000.
[8] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi and W. Piekarski, "First Person Indoor/Outdoor Augmented Reality Application: ARQuake," Personal and Ubiquitous Computing, Vol. 6, Issue 1, pp. 75-86, Feb. 2002.
[9] A. Cheok, F. Wan, X. Yang, W. Weihua, L. Huang, M. Billinghurst and H. Kato, "Game-City: A Ubiquitous Large Area Multi-Interface Mixed Reality Game Space for Wearable Computers," Proceedings of the International Symposiums on Wearable Computers, pp. 156-157, Oct. 2002.
[10] A. Cheok, H. K. Goh, W. Liu, F. Farbiz, S. Fong, S. Teo, Y. Li and X. Yang, "Human Pacman: a Mobile, Wide-area Entertainment System based on Physical, Social and Ubiquitous Computing," Personal and Ubiquitous Computing, Vol. 8, Issue 2, pp. 71-81, May 2004.
[11] D. Waqner, T. Pintaric and F. Ledermann, "Towards Massively Multi-User Augmented Reality on Handheld Devices," Proceedings of the International Conference on Pervasive Computing, to be published, May 2005.
[12] J. Yang and A. Waibel, "A Real-Time Face Tracker," Proceedings of IEEE Workshop Applications of Computer Vision, pp. 2-4, Dec. 1996.
[13] G. Bradski and V. Pisaresvsky, "Intel's Computer Vision Library: Applications in Calibration, Stereo, Segmentation, Tracking, Gesture, Face and Object Recognition," Proceedings of IEEE Conference of Computer Vision and Pattern Recognition, Vol. 2, pp. 796-797, 2000.
[14] http://fortress2.x2game.com/
[15] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MaxIntyre, "Recent Advances in Augmented Reality," Computer and Graphics, Vol. 21, Issue 6, pp. 34-47, Nov. 2001.
[16] J. Zhang, X. Chen, J. Yang and A. Waibel, "A PDA-based Sign Translator," Proceedings of the IEEE International Conference on Pattern Recognition, pp. 216-219, Oct. 2002.
[17] V. Vezhnevets, V. Sazonov and A. Andreeva, "A Survey on Pixel-based Skin Color Detection Techniques," Proceedings of International Conference on Computer Graphics and Vision, pp. 85-92, Sep. 2003.
[18] S. L. Phung, A. Bouzerdoum and D. Chai, "Skin Segmentation using Color Pixel Classification: Analysis and Comparison," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 1, pp. 148-154, Jan. 2005.
[19] B. D. Zarit, B. J. Super, F. K. H. Quek, "Comparison of Five Color Models in Skin Pixel Classification," Proceedings of International Workshop on Recognition, Analysis and Tracking of Face and Gestures in Real-Time System, pp. 58-63, 1999.

박 안 진

2004년 인제대학교 정보컴퓨터공학과 학사. 2006년 숭실대학교 IT대학 미디어학부 미디어공학(공학석사). 2006년~현재 숭실대학교 IT대학 미디어학부 미디어공학(박사과정). 관심분야는 패턴 인식, 컴퓨터비전, 실감형게임, Mobile Vision

정 기 철

1996년 경북대학교 컴퓨터공학과 공학석사. 1999년 방문 연구원, Intelligent User Interfaces Group, DFKI(The German Research Center for Artificial Intelligence GmbH), Germany. 1999년 방문 연구원, Machine Understanding Division, Electro Technical Laboratory, Japan. 2000년 경북대학교 컴퓨터공학과 공학박사. 2000년~2002년 PRIP Lab., Michigan State University, U.S. 박사후연구원. 2003년~현재 숭실대학교 정보과학대학 미디어학부 교수. 관심분야는 HCI, Interactive Contents, 영상처리/컴퓨터비전, 패턴인식, 증강현실, 인공지능