

비디오 게임 인터페이스를 위한 인식 기반 제스처 분할

정기철[†], 한은정^{**}, 강 현^{***}

요 약

키보드나 조이스틱 대신 카메라를 통해 입력되는 사용자의 제스처를 이용하는 시각 기반 비디오 게임 인터페이스를 사용할 때 자연스러운 동작을 허용하기 위해서는, 연속 제스처를 인식할 수 있고 사용자의 의미없는 동작이 허용되어야 한다. 본 논문에서는 비디오 게임 인터페이스를 위한 인식과 분할을 결합한 제스처 인식 방법을 제안하며, 이는 주어진 연속 영상에서 의미있는 동작을 인식함과 동시에 의미없는 동작을 구별하는 방법이다. 제안된 방법을 사용자의 상체 제스처를 게임의 명령어로 사용하는 1인칭 액션 게임인 *Quake II* 게임에 적용한 결과, 연속 제스처에 대해 평균 93.36%의 분할 결과로써 비디오 게임 인터페이스에서 유용한 성능을 낼 수 있음을 보였다.

Recognition-Based Gesture Spotting for Video Game Interface

Keechul Jung[†], Eunjung Han^{**}, Hyun Kang^{***}

ABSTRACT

In vision-based interfaces for video games, gestures are used as commands of the games instead of pressing down a keyboard or a mouse. In these interfaces, unintentional movements and continuous gestures have to be permitted to give a user more natural interface. For this problem, this paper proposes a novel gesture spotting method that combines spotting with recognition. It recognizes the meaningful movements concurrently while separating unintentional movements from a given image sequence. We applied our method to the recognition of the upper-body gestures for interfacing between a video game (*Quake II*) and its user. Experimental results show that the proposed method is on average 93.36% in spotting gestures from continuous gestures, confirming its potential for a gesture-based interface for computer games.

Key words: Automatic Gesture Recognition(제스처 인식), Gesture Spotting(제스처 분할), Video Games(비디오 게임), Recognition-Based(인식 기반)

1. 서 론

다양한 형태의 비디오 게임이 많은 사람들에게 인기를 끌고 있다. 비디오 게임을 위한 시각 기반 인터페이스(vision-based interface)는 조이스틱이나 키

보드를 누르는 대신 사용자의 제스처를 통해서 게임에 명령을 내림으로써, 사용자를 더욱 현실감있게 게임에 몰입할 수 있게하는 새로운 입력 방법으로써 많이 연구되고 있다. 시각 기반 인터페이스는 사용자의 제스처가 담긴 동영상상을 입력받아서 사용자의 행

* 교신저자(Corresponding Author) : 정기철, 주소 : 서울시 동작구 상도동(156-743), 전화 : 02)828-7260, FAX : 02)822-3622, E-mail : kcjung@ssu.ac.kr

접수일 : 2004년 11월 25일, 완료일 : 2005년 5월 6일

[†] 정희원, 숭실대학교 정보과학대학 미디어학부 교수

^{**} 숭실대학교 정보과학대학 미디어학부 콘텐츠 공학 박사과정

(E-mail : hanej@ssu.ac.kr)

^{***} 한국전자통신연구원 디지털콘텐츠단 연구원

(E-mail : hkang@etri.re.kr)

* 본 연구는 숭실대학교 교내연구비 지원으로 이루어 졌음.

동의 의도를 이해함으로써 컴퓨터와의 인터페이스를 가능하게하는 방법[1-5]으로써, Freeman 등의 *Decathlete* 시스템은 시각 기반 인터페이스의 특징을 보여주는 좋은 예이다[6].

비디오 게임을 위한 시각 기반의 인터페이스는 (1)반응속도, (2)신뢰도, (3)비용, 이상 세 가지 측면을 모두 고려하여야한다: (1)사용자는 명령을 위한 동작을 취한 후 컴퓨터가 반응하기까지의 지연시간을 허용하지 않는다. (2)사용되는 알고리즘이 사용자나 사용 환경에 무관하게 높은 신뢰도를 가져야한다. (3)기존의 게임 인터페이스에 쓰인 장비들은 저가의 장비로써, 이러한 장비들을 대체하기 위해서는 가격 경쟁력을 갖추어야한다. 위에서 기술한 문제는 게임을 위한 시각 기반 인터페이스의 기본적인 요구사항이라고 할 수 있다. 이외에도 시각 기반 인터페이스는 직관적이고 자연스러운 동작(intuitive and natural gesticulations)을 지원해야 한다. 사용자가 쉽게 게임에 몰입하기 위해서는 직관적으로 인지할 수 있는 동작 명령어(gestural commands)를 통해 자연스러운 동작을 지원해야하며, 사용자의 연속되는 제스처들의 분리인식, 즉 제스처의 분할을 수행할 수 있어야한다.

게임을 위한 시각 기반 인터페이스에서, 연속적인 제스처(continuous gestures)를 인식할 때 다른 제스처와의 구별과 함께 제스처가 아닌 쓸모없는 동작(unintentional movements)을 가려내는 과정(제스처 분할:gesture spotting)이 필요하다. 이는 사용자의 행동을 관찰한 연속 영상으로부터 사용자의 의도가 담긴 제스처의 시작과 끝을 찾아내는 작업을 말하며, 영상 내의 제스처의 시·공간적인 변형 등에 의해서 매우 어려운 일로 알려져 있다.

이현규와 김진형[7]은 임계치 HMM 모델을 이용한 제스처 분할 방법을 제안하였다. 입력시퀀스에 대해서 임계치 모델의 유사도와 제스처 인식 모델의 유사도를 비교함으로써 입력시퀀스의 제스처 여부를 가려낸다. 임계치 모델은 제스처 각 모델의 상태를 모음으로써 현재 시퀀스의 제스처 정도(gesture-ness)를 계산하게 되며, 임계치가 자동적으로 계산되는 장점이 있다. Nishimura와 Okaf[8]는 Continuous Dynamic Programming 기법을 사용하여 제스처를 분할한다. 이전의 연구들처럼 입력된 시퀀스에 대한 인식값을 바로 계산하지 않고, 현재 입력된 프레임이

제스처 내부 시퀀스 중 어디와 가장 잘 어울리는지 계산하여, 입력된 프레임 시퀀스와 제스처 내부 시퀀스가 잘 어울릴 때 제스처를 인식하는 방법이다. Sharma 등[9]은 3차원 가상 분자 시뮬레이션 시스템 'MDScope'을 개발하면서, 제스처가 항상 음성 데이터와 병존한다는 가정하에 음성데이터를 기준으로 제스처를 분할함으로써 제스처 분할 문제가 음성 분할 문제로 전환된다.

본 논문에서는 비디오 게임 인터페이스를 위한 인식 기반(recognition-based)의 제스처 분할 방법을 제안한다. 제스처의 인식과 분할의 밀접성으로 인해서 가-제스처(tentative gesture)를 이용한 인식 기반의 분할 방법으로써, 연속 시퀀스에서 제스처가 될만한 부분(가-제스처)을 생성하고, 이들 가-제스처들에 대한 인식 결과를 기준으로 최종적으로 제스처를 분할한다. 이러한 제스처 분할 방법을 이용한 제스처 인식기를 구현하여 Quake II 게임 인터페이스에 적용함으로써 유용성을 검증한다.

논문의 구성은 다음과 같다. 2장에서는 제안된 방법에 대해 설명하며, 2.1절에서는 제스처의 표현 방법, 2.2절에서는 제스처 분할방법, 2.3절에서는 제스처 인식 엔진, 2.4절에서는 제안된 방법을 탑재한 게임 인터페이스에 대해 기술한다. 3장에서는 제안된 방법을 실험한 결과를 도시하며 결론을 4장에서 맺는다.

2. 인식 기반의 제스처 분할

입력시퀀스에 포함된 의미없는 동작들은 연속 제스처 인식을 어렵게 하기 때문에 이를 분리하는 제스처 분할 과정이 필요하다. 제스처 분할을 어렵게 하는 요인들 중 하나인 제스처 혼동(gesture confusion)은 그림 1에서와 같이 'Duck'과 'Walk' 제스처[1]에서 유발될 수 있으며, 이로인해 제스처 인식 결정의 순간이 항상 지금 즉시가 아닐 수 있음을 알 수 있다.

그림 2는 본 논문에서 사용하는 제스처 명령어들이다. 사용자에게 직관적인 제스처 명령어 셋을 제공하기 위해서 쉽게 이해할 수 있는 동작들로 구성하였으며, 총을 쏘는 동작만을 마우스를 사용하였는데 이

1) 본 논문에서 정의한 게임 Quake II를 위한 제스처 명령어 중의 일부이다.

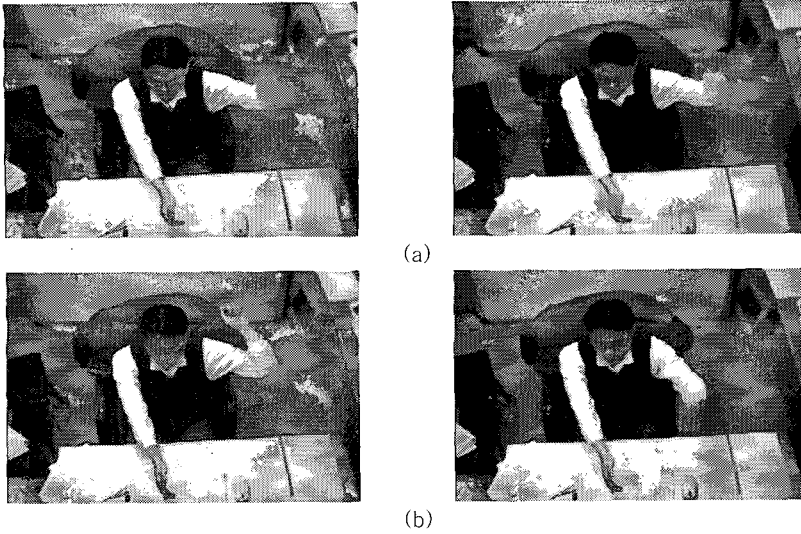


그림 1. 제스처 혼동: (a) 'Duck'. (b) 'Walk'.

는 동작 명령어보다 쉽게 사용할 수 있기 때문이다.

자의 포즈로 결정된다.

2.1 제스처 표현

하나의 입력 영상에 나타난 사용자 상체의 특정 부분의 위치 값을 사용하여 포즈를 정의한다. 사용자 상체 포즈는 실시간 게임 인터페이스 제공을 위해 머리와 양손의 위치 정보만을 사용하며, 색깔 정보를 이용한 블립(blob) 기반의 방법을 사용함으로써 실시간에 안정적인 결과를 기대할 수 있다. (1)은 t 번째 프레임에 담긴 포즈에 관한 특징벡터를 보여준다.

$$x_t = (h_x, h_y, l_x, l_y, r_x, r_y) \tag{1}$$

여기서 h, l, r 은 각각 머리, 왼손, 오른손의 영상 내 위치(x, y)를 의미하며, t 는 프레임 번호를 의미한다.

영상 내의 대표적인 상체 위치 값들을 추정하기 위해서, 제스처 영상 데이터를 클러스터링(clustering)한다. 클러스터링의 사용으로 수작업 라벨링의 어려움을 없애주면서도 만족할만한 분류결과를 기대할 수 있다.

$$p_i = \arg \min_{1 \leq i \leq K} \{D(c_i, x_t)\}, K : \text{the number of classes} \tag{2}$$

여기서 $D(\cdot, \cdot)$ 는 유클리디안 거리를 뜻하며 사용자의 현재 포즈는 i 번째 클러스터 c_i 와 현재 프레임의 사용자 포즈의 특징정보 x_t 간의 거리에서 최소가 되는 클러스터의 인덱스를 t 번째 프레임에 나타난 사용

2.2 제스처 분할

본 장에서는 입력된 연속 영상에서 의미있는 제스처의 위치를 찾아내는 방법, 즉 제스처 분할 방법을 기술한다. 서두에서 기술한 바와 같이 제스처의 시작과 끝 프레임을 찾아내는 일은 제스처 인식과 분리하여 생각하기 어렵기때문에, 다음과 같이 인식을 병행한 제스처 분할 방법을 사용한다. 먼저 영상 시퀀스 내에서 시공간(spatio-temporal)적인 정보를 이용하여 가-제스처(tentative gestures)들을 생성한다. 이러한 가-제스처들 중 가장 높은 인식값을 가지는 부분을 최종 제스처로 선택하며, 이와 동시에 제스처 분할이 수행되게 된다. 제안된 방법은 가-제스처들을 생성하는 과정과 제스처를 결정하는 과정, 두 부분으로 나뉜다.

2.2.1 가-제스처 생성

가-제스처는 입력된 영상 시퀀스의 부분 중, 제스처 가능성이 있는 부분을 의미한다. 영상 시퀀스에서 가-제스처를 찾는 일은 제스처의 시작과 끝이 될 후보(candidate) 프레임을 찾는 일로써, 시작과 끝이 될 만한 프레임을 후보 컷(candidate cut)이라고 부르며, 다음과 같은 제스처의 특징이나 외부 정보를 이용하여 정해진다.

| 제스처 명령어 | 그림 설명 | 제스처 명령어 | 그림 설명 |
|---|--|---|-------|
| System Setting (S) Unfold your arms within 3 seconds, then fold them within 3 seconds to set the interface. | | Body Right Move (BR) Keep your hands together, and move your upper body to the right so that the virtual character steps right. | |
| Walk (W) Shake your left hand in a forward direction for the virtual character to walk forward. | | Body Left Move (BL) Keep your hands together and move your upper body to the left so that the virtual character steps left. | |
| Run (R) Shake your hands for the virtual character to run forward. | | Duck (D) Raise your left hand so that the character ducks at his position. | |
| Gun Left Move (GL) Keep your hands together and move your hands to the left so that the virtual character turns his direction to the left. | | Move Stop (MS) Put your hands on the desk so that the virtual character holds a movement. | |
| Gun Right Move (GR) Keep both hands together and move your hands to the right so that the virtual character turns his direction to the right. | | Stop System (Q) Cross your arms to stop the interface. | |
| Shot/Weapon Change | Wireless mouse left/right click so that the virtual character fire his weapon. | | |

그림 2. Quake II 게임을 위한 제스처 명령어

Def. 1: 제스처를 시작하기 전과 마친 후에는 일정시간 동작을 멈추는 특징을 이용하여 속도를 기준으로 제스처를 분할.

Def. 2: 제스처의 시작과 끝에서 많이 나타나는 특정 포즈를 기준으로 제스처 분할.

Def. 3: 심한 굴곡점을 가진 프레임 기준으로 제스처 분할.

그림 3은 속도, 특정 포즈, 심한 굴곡점의 정의를 가지고 입력 영상 시퀀스로부터 가-제스처를 생성한 결과이다. 이런 후보점을 기준으로 가-제스처를 생

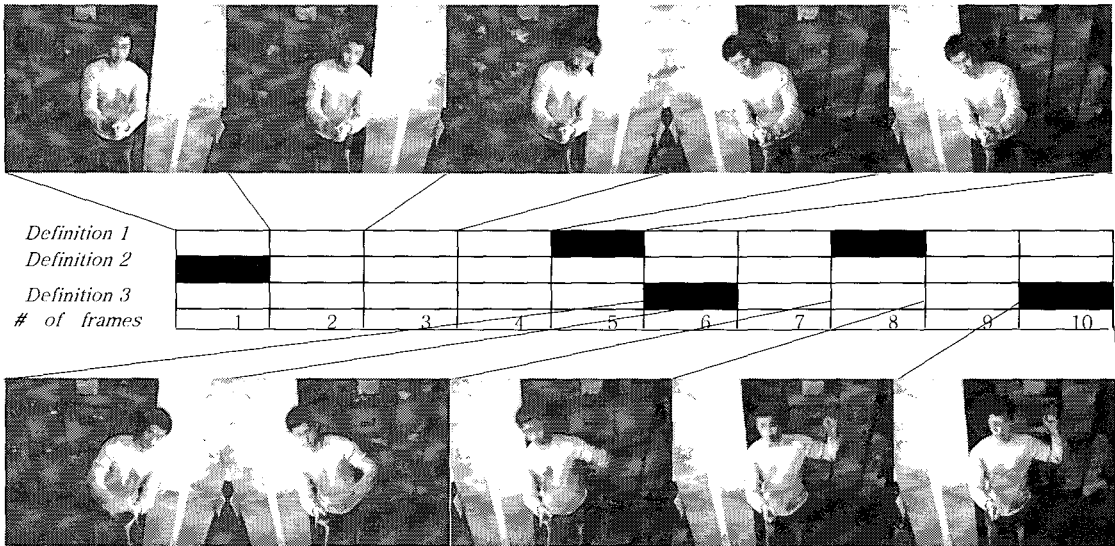


그림 3. 후보컷의 예: Def.1은 급격한 속도변화, Def.2는 특정포즈 제스처, Def.3은 심한 굴곡점(검은 영역이 후보컷)

성하게 되는데, 그림 3에서 찾은 후보컷으로는 다음과 같은 가-제스처 11개가 생성된다: 1-5, 1-6, 1-7, 1-8, 1-10, 5-6, 5-8, 5-10, 6-8, 6-10, 8-10. 이 숫자들은 프레임의 번호이며, 1-5는 1번 프레임부터 5번 프레임까지를 말한다.

2.2.2 제스처 분할 과정

가-제스처들의 인식값은 탐색 범위를 결정하는 슬라이딩 윈도우(sliding window)에 저장 된다. 윈도우의 크기는 제스처의 길이에 의존한다. 슬라이딩 윈도우의 크기가 클수록 긴 제스처를 분할할 가능성이 커지며, 짧은 제스처를 놓칠 확률도 커진다. 본 논문에서 윈도우의 크기는 실험적으로 구해진 값을 사용하였다. 슬라이딩 윈도우 내에 있는 가-제스처 중에서 가장 높은 인식값을 가진 가-제스처가 최종 제스처로 결정되며, 슬라이딩 윈도우 내에 어떤 제스처도 입력되지 않았을 경우에는 가장 높은 인식값을 내는 가-제스처의 삭제 여부를 결정하기 위해서 인식값을 실험에 의해서 결정된 임계치와 비교한다. 임계치는 피실험자 5인의 데이터로부터 결정되며, 동작을 정확히 알고 있는 5인의 동작에 대한 평균 인식값과 분산을 기준으로 95% 신뢰구간을 설정하고 이 구간의 최저 인식값을 임계치로 정한다. 인식 결정 후에 슬라이딩 윈도우는 인식된 제스처의 다음으로 이동한다.

그림 4는 두개의 연속 제스처의 인식과 슬라이딩 윈도우의 이동을 보인다. 첫 슬라이딩 윈도우에서 0-4 가-제스처가 'Walk' 제스처로 결정되면 슬라이딩 윈도우가 (4,5)로 이동하게 된다. 특히 첫 슬라이딩 윈도우에서 0-2 가-제스처 또한 높은 인식값을 가지고 있는데 이는 'Duck' 제스처를 의미한다. 이 두 제스처는 혼동이 일어나는 제스처들인데, 슬라이딩 윈도우에 의해서 0-2대신 0-4가 선택되며 이러한 제스처 혼동 문제를 해결하게 된다. 두번째 슬라이딩 윈도우에서는 'Body Right Move'가 인식이 되었다. 그 사이에 사용자의 의도하지않은 동작들(손을 잠시 떴었다 붙이는 동작)이 삽입되었는데, 이런 동작들은 제안된 방법에 의해 배제되며 인식된 'Body Right Move' 제스처에 어떤 영향도 미치지 않는다. 3번째 슬라이딩 윈도우에서는 사용자가 아무런 행동을 하지 않았을 때 가장 높은 인식값을 가진 가-제스처가 임계치와 비교후 삭제되었다.

2.3 제스처 인식

주어진 가-제스처에 대해 제스처 인식 프로세스가 제스처 인식을 수행한다. 보통 제스처 인식에서는 은닉 마르코프 모델(HMM)[7], Dynamic Time Warping (DTW)[11,12]과 Finite State Machines (FSM)[13]과 같은 방법을 많이 사용한다. 제안된 연구에서는 DTW 방법에 의한 인식을 하였다. DTW 방법에 의

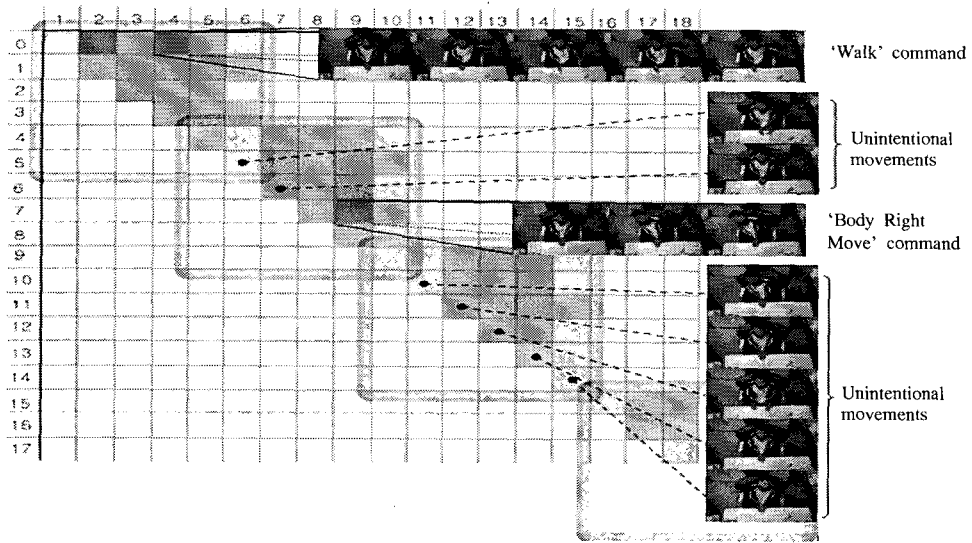


그림 4. 슬라이딩윈도우의 동작 예

한 제스처 인식에서 재귀법칙은 다음 식과 같다. 여기서 $dist_{i,j}$ 는 i -번째 심볼과 j -번째 심볼의 최소 유사도를 나타내며, $cost(i, j)$ 는 함수로서 심볼간의 거리를 나타낸다. 본 논문에서는 대표 포즈(pose template)간의 위치값의 유클리디안 거리를 사용하였다.

$$dist_{i,j} = cost(i, j) + \min\{dist_{i-1,j}, dist_{i-1,j-1}, dist_{i,j-1}\} \quad (3)$$

2.4 제스처 기반 인터페이스

제스처 기반 인터페이스는 동영상으로부터 제스처를 인식하는 일을 하며, 게임이나 다른 응용 프로그램들과 데이터 교환을 한다. 제스처 기반 인터페이스는 크게 두개의 모듈로 구성되어있다(그림 5). 인터페이스 시스템은 포즈 추정 모듈(Pose Estimation Module: PEM)과 제스처 인식 모듈(Gesture Recognition Module: GRM)로 구성되어있다. PEM은 각 프레임마다 사용자의 머리, 양손을 추출하고 이에 대한 대표적인 포즈를 결정하게 되며, GRM은 포즈 심볼 열에서 제안된 제스처 분할방법을 통해 인식과 분할 수행한다.

프레임에 나타난 사용자의 상체 포즈는 영상 내 피부색(skin color) 영역을 이용하여 사용자의 머리와 양손의 위치로 결정한다. 피부색 픽셀을 결정하기 위해서 샘플 영상에서 추출한 Hue 상의 히스토그램

을 이용하여 피부색 확률값 영상을 구한다. 피부색 영역을 구하기 위해서 다음과 같은 3 단계 영상 처리 기법이 도입된다[1]: 피부색 픽셀을 분류하는 부분, 모폴로지 연산을 통해 이진 영상내의 잡음을 제거하는 부분, 연결 요소 분석(connected component analysis)을 통해서 신체 영역을 찾아내는 부분. 피부색 픽셀로 분류된 이진영상으로부터 모폴로지 연산으로 잡음을 제거한다. 피부색 영역은 사용자의 머리와 양손 영역에서 군집하여 나타나게 되므로 연결 성분 분석에 의해서 영역을 구한다. 이 영역들을 머리, 왼손, 오른손으로 결정하기 위한 간단한 휴리스틱 룰을 도입한다; (1)모든 신체부분은 영역 크기가 제한된 크기보다 커야 한다. (2)가장 큰 영역이 머리이다. (3) 왼손은 영상 내 오른쪽에 위치하고 오른손은 왼쪽에 위치한다. (4)영역이 단 한 개 있을 때는 머리, 양손이 합쳐져 보이는 것이다. (5)영역이 단 두개 있을 때에는 상위의 영역이 머리, 하위는 양손의 위치이다.

이러한 과정에 의해서 프레임에 나타난 머리, 양손의 위치가 결정되며 대표적인 포즈와의 거리로부터 포즈가 결정된다. 대표적인 포즈는 정의된 제스처 동영상 데이터로부터 k-means 방법으로 클러스터링된 k개의 클러스터가 각각의 포즈가 된다. 이런 방법은 상당한 양의 동영상 데이터의 라벨링에 의한 분류방법보다 수월하며 만족할만한 성능을 보인다.

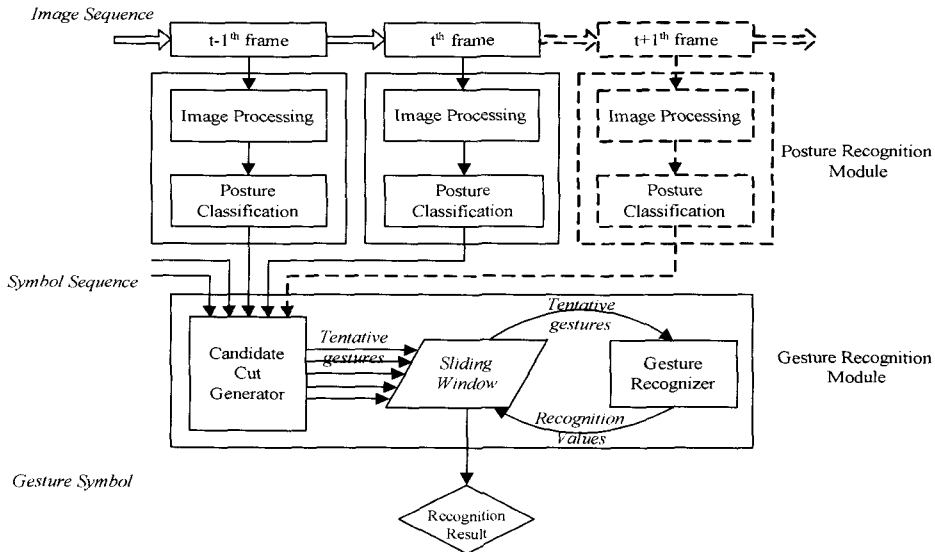


그림 5. 전체적인 시스템 구조

3. 실험 및 결과

제안된 인식 기반의 제스처 분할 방법을 평가하기 위해서 Id Software[®]에서 개발한 1인칭 슈팅 게임인 Quake II를 위한 제스처 기반의 인터페이스 시스템을 개발하였다(<http://www.idsoftware.com/games/quake/quake2/>). 본 논문에서 제안한 제스처 인식 시스템을 게임에 접목하기 위해서, Id Software[®]에서 제공하는 Dynamic Linked Library(DLL)을 사용하였다(<http://www.quake2.com/dll/>). 그림 2는 10개의 제스처와 1개의 마우스 동작으로 구성된 Quake II를 위한 명령어를 보인것이다. 이러한 제스처는 사용자가 게임에 쉽게 집중할 수 있도록 구성되었으며, 일반적이며 직관적인 동작들로 구성되어있으며, 제스처의 인식결과가 Quake II DLL을 통해서 Quake II의 입력 명령어로 사용되게 된다. 제스처 기반 인터페이스의 학습과 실험을 위해서 45명의 제스처 데이터를 사용하였으며, 5명의 제스처 데이터가 대표적인 포즈 추출을 위해서 사용되었다. 제스처 기반 인터페이스는 Microsoft Visual C++ 6.0과 OpenCV를 이용하여 만들었다. 이들은 1.3GHz의 표준 PC에서 구현되었으며, 영상은 부가적인 영상 캡처 장비없이 USB 1.1 방식의 웹카메라만으로 구현되었다.

Quake II 게임을 위한 게임 인터페이스 시스템을 위해 그림 6과 같은 환경을 구현하였다. 카메라는 사

용자의 상단에 위치하여 사용자의 상체 전체의 움직임을 촬영할 수 있으며, 카메라 자체의 움직임은 없도록 정하였다.

표 1은 독립 제스처 인식에 대한 결과를 보이고 있다. '독립'의 의미는 제스처의 시작과 끝을 인식하지 않고, 미리 지정한 데이터로 제스처 인식기 자체 성능을 보인것이다. 'Duck'과 같은 제스처가 인식을 못했는데 이는 포즈 추정 모듈의 부정확성 때문에 다른 포즈와의 혼동에 기인한다.

연속 제스처 인식 실험의 오류는 제스처의 삽입(insertion), 삭제(deletion), 대체(substitution) 에러가 있다. 삽입 에러는 제스처 분할이 행해진 제스처 외에 다른 제스처가 있다고 결정하는 것을 말한다. 삭제 에러는 행해진 제스처의 결과 자체를 삭제한 것을 말하며, 대체 에러는 행해진 제스처의 인식 대신에 다른 제스처라고 결정하는 것이다. 제스처 분할을 평가하기 위해 이러한 에러 요소를 고려한 [7]의 에러 평가 기준을 도입한다.

$$Detection\ ratio = \frac{correctly\ recognized\ gestures}{the\ number\ of\ input\ gestures}$$

$$reliability = \frac{correctly\ recognized\ gestures}{the\ number\ of\ input\ gestures + the\ number\ of\ insertion\ errors}$$

검출률(Detection ratio)은 삽입 에러의 영향을 고려하지 않으며, 신뢰도(reliability)에서는 이를 고려

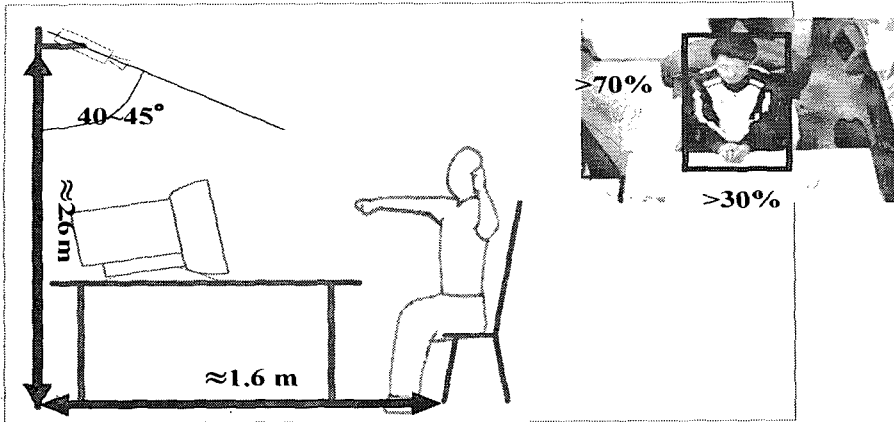


그림 6. 레이아웃 B의 구조: 사용자는 폭의 30% 이상, 높이의 70% 이상을 차지해야한다

표 1. 미리 분할된 제스처의 인식률

| Gestures | S | Q | GL | GR | BL | BR | W | D | MS | R |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Recognition rate (%) | 95.60 | 98.80 | 97.40 | 99.80 | 98.20 | 97.60 | 95.20 | 90.60 | 95.20 | 97.40 |

하여 평가를 내릴 수 있다. 표 2는 연속 제스처 인식에 대한 제안된 제스처 분할 결과이며, 독립 제스처 인식결과에 비해 낮은 결과를 보이나 그다지 크지는 않다.

그림 7을 보면 두개의 제스처가 연속적으로 행해질 때 삽입되는 extra-movements(그림에서 화살표)가 앞 뒤의 제스처 인식에 영향을 미침을 알 수 있다. 이러한 extra-movements는 가-제스처 생성에서 분리되는 경우에는 제스처 인식되는 경우가 있었으나,

서로 다른 신체 부분에서 일어나는 extra-movement의 경우(그림 7)는 제스처 인식에 실패하였다.

슬라이딩 윈도우의 크기를 결정하기 위해서 제스처 분할의 성능과 속도면의 검증을 하였다. 그림 8과 같이 슬라이딩 윈도우의 크기가 커질수록 검출률과 신뢰도가 증가하다가 6이상에서 감소하였다. 속도는 윈도우가 커질수록 계산해야되는 가-제스처의 개수도 많기므로 지수함수적으로 증가하였다. 우리는 이 실험을 토대로 6이라는 슬라이딩 윈도우 크기를 결정하였다.

표 2. 슬라이딩 윈도우의 크기에 따른 분할 결과

| | # of Gestures | Correct | Error types | | | Detection ratio | Reliability |
|----|---------------|---------|-------------|----------|--------------|-----------------|-------------|
| | | | Insertion | Deletion | Substitution | | |
| S | 185 | 175 | 3 | 4 | 6 | 94.59 | 93.09 |
| Q | 195 | 186 | 3 | 4 | 5 | 95.38 | 93.94 |
| GL | 200 | 193 | 5 | 4 | 3 | 96.50 | 94.15 |
| GR | 200 | 195 | 5 | 2 | 3 | 97.50 | 95.12 |
| BL | 200 | 194 | 3 | 3 | 3 | 97.00 | 95.57 |
| BR | 198 | 191 | 2 | 2 | 5 | 96.46 | 95.50 |
| W | 166 | 157 | 3 | 4 | 5 | 94.58 | 92.90 |
| D | 163 | 149 | 6 | 7 | 7 | 91.41 | 88.17 |
| MS | 160 | 150 | 2 | 5 | 5 | 93.75 | 92.59 |
| R | 195 | 182 | 4 | 5 | 8 | 93.33 | 91.46 |
| 합계 | 1862 | 1772 | 36 | 40 | 50 | 95.17 | 93.36 |

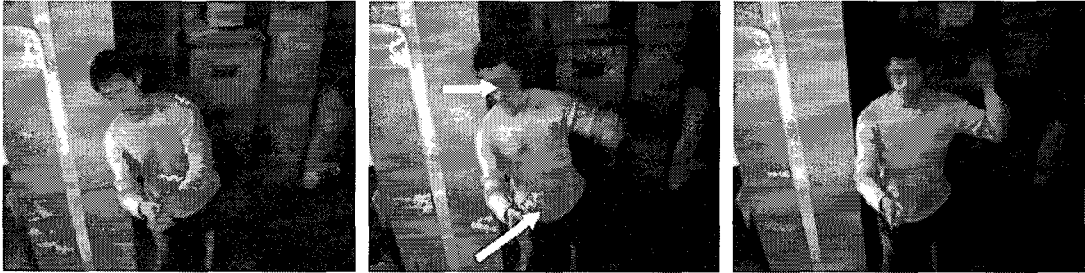


그림 7. 'Body Right Move'와 'Duck' 제스처 사이의 혼동

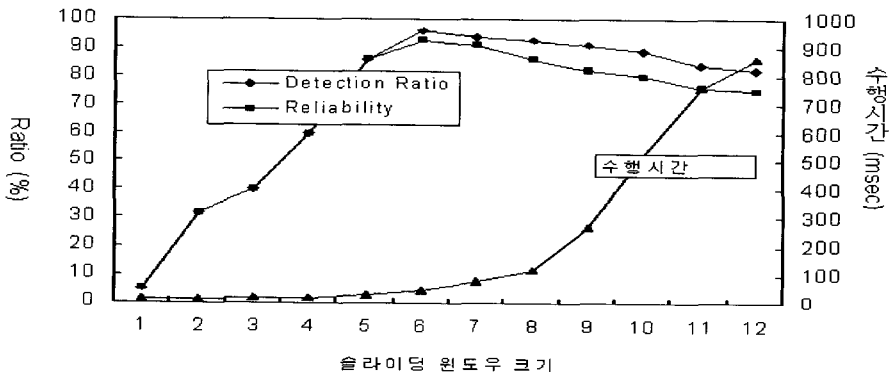


그림 8. 슬라이딩윈도우의 크기에 따른 분할 결과 및 속도 비교

4. 결 론

본 논문은 인식 기반의 제스처 분할 방법을 제안하고, 이를 Quake II 게임을 위한 제스처 기반 인터페이스에 적용함으로써 유용성을 검증하였다. 제안된 방법은 구현된 게임 시스템에서 93.36%의 신뢰도를 보였으며 이러한 성능은 Quake II 게임을 즐기기에 충분하였다. 효과적인 분할 능력으로 사용자가 정의된 제스처 외에 다른 동작하는 것을 허용하며, 실시간 게임인터페이스에서도 적용 가능한 실시간 수행 성능을 보였다.

연속 제스처 인식에서 발생하는 제스처 혼동 문제는 슬라이딩 윈도우 기법을 도입하여 해결하였다. 현재 우리는 좀더 다양한 환경 하에서의 신뢰도있는 게임 인터페이스를 구현하기 위해서 노력하고 있다. 다양한 조명조건에서의 신체 부분을 찾고 추적하는 모듈을 개발하고 있다. 또한 이런 게임인터페이스 외에도 증강현실(Augmented Reality)분야의 어플리케이션에서 사용가능한 가벼운 제스처 기반 인터페이스를 구현 중이다.

참 고 문 헌

- [1] Kang, H., Lee, C.W., Jung, K., and Kim, H. J., "A Clustering Approach to the Vision-based Interface for Interactive Computer Games," *Lecture Notes in Computer Science* 2690, pp. 246-253, 2003.
- [2] Kendon, A., "Current Issues in the Study of Gesture. The Biological Foundation of Gestures: Motor and Semiotic Aspects," Lawrence Erlbaum Associate, pp. 23-47, 1986.
- [3] Min, B.W., Yoon, H.S., Soh, J., Ohashi, T., and Ejima, T., "Gesture-based Editing System for Graphic Primitives and Alphanumeric Characters," *Engineering Applications of Artificial Intelligence*, Vol. 12, No. 4, pp. 429-441, 1999.
- [4] Pavlovic, V. I., Sharma, R., and Huang, T. S., "Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review," *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, Vol. 19, No. 7, pp. 677-695, 1997.

- [5] Wu, Y. and Huang, T. S., "Vision-based Gesture Recognition: a Review," *Lecture Notes in Artificial Intelligence*, Vol. 1739, Gesture Workshop, pp. 103-115, 1999.
- [6] Freeman, William T., Anderson, David B., Beardsley, Paul A., Dodge, Chris N., Roth, Michael, Weissman, Craig D., and Yerazunis, William S., "Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphic and Application*, Vol. 18, pp. 42-53, 1998.
- [7] Lee, H.K. and Kim, J.H., "An HMM-based Threshold Model Approach for Gesture Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 10, pp. 961-973, 1999.
- [8] Nishimura, T. and Oka, R., "Towards the Integration of Spontaneous Speech and Gesture based on Spotting Method," *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 433-437, 1996.
- [9] Sharma, R., Zeller, M., Pavlovic, V. I., Huang, T. S., Lo, Z., Chu, S., Zhao, Y., Philips, J. C., and Schulten, K., "Speech/Gesture Interface to a Visual-Computing Environment," *IEEE Computer Graphics and Applications*, pp. 29-37, May/April 2000.
- [10] Jung, K. and Kim, H.J., "On-line Recognition of Cursive Korean Characters using Graph Representation," *Pattern Recognition*, Vol. 33, No. 3, pp. 399-412, 2000.
- [11] Black, M. and Jepson, A., "Recognition Temporal Trajectories using the Condensation Algorithm," *International Conference on Automatic Face and Gesture Recognition*, Japan, pp. 16-21, 1998.
- [12] Gavrila, D. M., "Vision-based 3-D Tracking of Humans in Action," Ph.D. Thesis, Depart-

ment of Computer Science, University of Maryland, College Park, 1996.

- [13] Davis, J. and Shah, M., "Visual Gesture Recognition," *Vision Image and Signal Processing*, Vol. 141, No. 2, pp. 101-106, 1994.



정 기 철

1994년 경북대학교 컴퓨터공학과(공학사)
 1996년 경북대학교 대학원 컴퓨터공학과(공학석사)
 1999년 Intelligent User Interfaces group at DFKI (The German Research Center for Artificial Intelligence, GmbH), Germany, 방문연구원
 2000년 Machine Understanding Division, ElectroTechnical Laboratory in Japan, 방문연구원
 2000년 경북대학교 대학원 컴퓨터공학과(공학박사)
 2000년~2001년 성균관대학교 박사후연구원
 2001년~2002년 PRIP Lab, Michigan State University Postdoc
 2003년~현재 숭실대학교 정보과학대학 미디어학부 교수
 관심분야: HCI, 콘텐츠공학, 인터랙티브 게임, 영상처리/컴퓨터비전, 증강현실, 인공지능



한 은 정

2003년 신라대학교 국어국문학과(인문학사)
 2005년 동서대학교 소프트웨어대학원 소프트웨어학과(공학석사)
 2005년 현재 숭실대학교 정보과학대학 미디어학부 콘텐츠 공학(박사 과정)
 관심분야: e-learning, 모바일 콘텐츠, HCI, CBIR



강 현

1999년 경북대학교 대학원 컴퓨터공학과(공학석사)
 2004년 경북대학교 대학원 컴퓨터공학과(공학박사)
 2000년~2002년(주)네비웍스 인공지능연구소 책임연구원
 2004년~현재 한국전자통신연구원 디지털콘텐츠단 연구원
 관심분야: 컴퓨터 비전, 증강 현실, HCI, 게임