컴퓨터 비전 기술 기반 체감형 복싱 게임

박지영*, 김기찬**, 박정우***, 이준호****

Interactive Boxing Game Based on Computer Vision

Park, Ji-Young*· Kim, Gi-Chan**· Park, Jung-Woo***· Yi, June-Ho****

ABSTRACT

본 연구는 컴퓨터 비전기술을 사용하여 사용자의 제스처를 인식하고 이를 통해 일반 PC 용으로 제작된 게임을 보다 현실감 있게 제어할 수 있도록 하는 기술을 제안한다. 키보드를 사용하여 제어하는 일반 PC 용 복싱 게임에 본 연구에서 제안하는 효율적인 시스템 구성과 정확하고 빠른 제스처 인식 기법을 적용하여 그 유용성을 검증하였다. 복싱 게임에서 요구되는 플레이어의 좌우이동과 펀치 동작을 정확하고 빠르게 인식하기 위하여 각각의 제스처를 구분하는 비전 시스템을 개별적으로 구성한다. 각 시스템에서 인식한 동작 데이터의 전송을 위해서는 수평 구조의 클라이언트-서버 모델을 적용하였으며, 제안된 기법을 적용하여 실험한 결과 기존에 쉽게 접할 수 있는 일반 PC 용 게임의 인터페이스를 대화형 인터페이스로 대체할 수 있음을 보였다. 이와 같이 단순히 키보드의 버튼을 누르거나 센서가 장착된 장비를 사용하지 않고 사람의 몸동작을 사용하여 게임을 제어하는 경우 게임의 현실감 증대는 물론 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간 친화적인 인터페이스가 가능하다.

---

*성균관대학교 정보통신공학부 박사과정, **성균관대학교 정보통신공학부 석사과정
***성균관대학교 정보통신공학부 학부과정, ****성균관대학교 정보통신공학부 부교수

1. 서 론

현재 PC 용으로 개발된 모든 게임들은 키보드, 마우스, 조이스틱과 같이 매우 제한적인 하드웨어를 이용하여 동작하도록 구성되어있다. 이와 같은 장비의 사용은 공간상의 제약은 물론 게임의 현실성을 떨어뜨리는 주된 요인이라고 할 수 있다. 최근 발표되고 있는 체감형 아케이드 게임기들은 장비에 센서를 부착하여 체감성을 증대시키는 효과를 보이나 부가적인 하드웨어를 수반한다. 단순한 키보드조작이나 센서가 장착된 장비를 사용하지 않고 사람의 몸동작 만으로 게임을 제어한다면 게임의 현실감 증대는 물론 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간 친화적인 인터페이스가 가능하다. 본 연구는 컴퓨터 비전기술을 사용하여 사용자의 제스처를 인식하고 이를 통해 일반 PC 용으로 제작된 게임을 보다 현실감 있게 제어할 수 있도록 하는 것을 목적으로 한다.

최근 컴퓨터 비전기술을 기반으로 하는 제스처 인식 알고리즘을 적용하여 다양한 대화형 시스템을 구축하기 위한 연구들이 활발히 진행되고 있다. 신체 특정부위의 동작을 추적하여 분석하는 연구로서 손 제스처 또는 얼굴의 움직임을 인식하는 알고리즘들이 개발되어 수화인식 시스템[1] 또는 컴퓨터 제어를 위한 휴먼 마우스 시스템[2]에 적용되고 있다. 게임을 동작하기 위한 제스처들은 이들과는 달리 큰 동작과 다양한 움직임이 요구된다. 뿐만 아니라 게임 제어를 위해서는 무엇보다 빠르고 정확하게 제스처를 인식하는 기술이 필요하다. 이러한 요구들 때문에 게임 제어를 위한 제스처 인식을 위해서는 인식 동작에 적합한 게임을 개발하고 속도 개선을 위한 장비를 추가하는 연구[3]가 보고 된바 있다. 본 연구에서는 기존의 키보드 제어방식으로 제작된 PC 용 복싱 게임에 효율적으로 구성된 시스템과 제스처 인식 알고리즘을 적용하여 대화형 인터페이스를 사용할 수 있음을 보였다. 제스처 인식을 위해 사용자의 인지정도를 뛰어넘는 정도의 정밀한 계산은 필요하지 않다. 따라서 계산속도가 빠른 칼라영상분할 및 이진영상처리 알고리즘을 사용하였으며 이를 통해 컴퓨터 비전 기반의 게임플레이가 가능함을 실제 복싱게임의 구현을 통해 입증하였다.

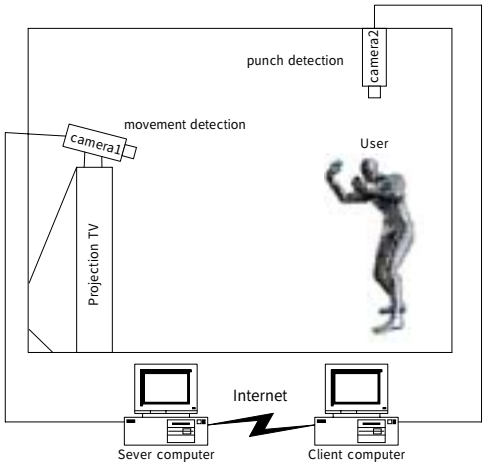
복싱게임을 위해서는 일반적으로 사용자가 컴퓨터 플레이어의 펀치를 피하기 위한 동작과 사용자가 컴퓨터 플레이어에게 펀치를 가하는 동작이 필요하다. 본 연구에서는

단순히 펀치를 피하거나 공격하는 상태만 존재하는 것이 아니라 사용자와 컴퓨터 플레이어가 서로 좌우로 이동하며 피하고 사용자가 컴퓨터 플레이어에게 직접 다가가 빠른 펀치 동작을 통해 공격하는 것이 가능하도록 하였다. 보다 정확하고 빠르게 좌우이동과 펀치 제스처를 인식하기 위하여 각각의 제스처를 구분하는 비전 시스템을 개별적으로 구성하였다. 각각의 비전 시스템의 데이터전송을 위한 네트워크는 인터넷 망을 매개로 한다. 이와 같이 각각의 비전 시스템들을 네트워크로 구성하는 경우 여러 개의 독립적인 비전시스템을 하나의 통합 네트워크 시스템으로 구성할 수 있으므로 다양한 기능을 더욱 효율적으로 수행할 수 있다[7]. 이렇게 인터넷을 매개로 하여 컴퓨터 비전 네트워크 시스템을 구성할 경우 원거리 사용자들 간의 인터페이스가 가능해진다. 본 연구에서는 인터넷상에서 다양한 정보의 전송으로 인한 속도 저하를 방지하고 네트워크 사용의 효율성을 증대하기 위해, 한 시스템을 구성하는 두 컴퓨터간의 연결은 크로스오버 케이블을 사용하였다. 그러나 네트워크용의 게임 애플리케이션이 존재한다면 인터넷 망을 매개로 하여 다수의 사용자가 참여하여 게임을 즐길 수 있는 인터페이스로도 확장이 가능하다.

본 논문의 구성은 다음과 같다. 2 절에서는 제안된 시스템의 전체 구성을 설명한다. 3 절에서는 좌우 이동 동작인식을 위한 방법을, 4 절에서는 펀치 동작인식을 위한 방법에 관하여 기술한다. 5 절에는 좌우 이동 동작 인식과 펀치 동작 인식의 연동을 가능하게 하는 네트워크 비전시스템 구성을 설명하며 6 장에서는 제안된 기법들을 실제 게임에 적용한 실험 결과를 보여준다.

2. 전체 시스템 개요

복싱게임을 위한 시스템 구성은 그림 1 과 같으며 게임 진행 화면을 보여주기 위한 디스플레이 장치 그리고 좌우 이동, 펀치동작 인식을 위한 카메라와 PC 로 구성되어 있다. 두 대의 PC 는 각각 서버와 클라이언트의 형태로 네트워크에 연결되어 있어 좌우이동과 펀치 동작의 연동이 가능하다. 시스템을 구성하는 서버와 클라이언트 컴퓨터는 펜티엄 2GHz 모델이며 데이터 전송을 위한 서버와 클라이언트간의 연결은 인터넷환경을 이용하였다.

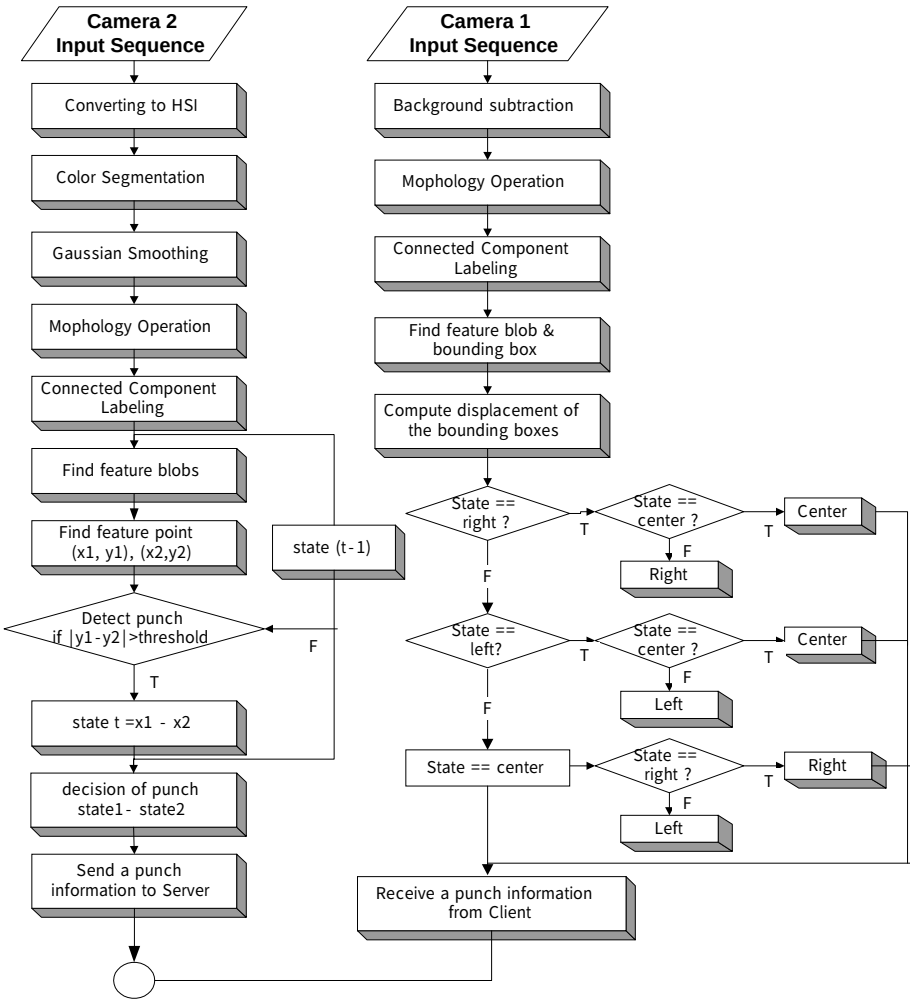


[그림 1] 전체 시스템 구성도

그림 2 의 전체 시스템은 크게 세 부분으로 나누어 볼 수 있다. 첫째, 그림 1 의 camera1 과 서버 컴퓨터를 사용하여 사용자의 좌우 이동 제스처를 인식하고 좌우 이동 키를 발생시키는 과정이 있다. 두 번째 과정은 camera2 와 클라이언트 컴퓨터를 사용하여 펀치 동작을 인식하고 펀치 키를 발생하는 과정이다. 세 번째로는 서버 컴퓨터에서 복싱 게임이 동작되는 동안 클라이언트 컴퓨터에서 발생하는 펀치동작을 서버로 전송하여 좌우 이동과 펀치가 동시에 발생할 수 있도록 네트워크를 구성하는 부분이 있다.

3. 좌우이동 동작 인식

컴퓨터 플레이어의 펀치를 피하는 동작과 사용자가 컴퓨터 플레이어에게 다가가 펀치를 가하는 동작 발생을 위해 사용자의 좌우이동 제스처를 인식한다. 사용자의 좌우이동 제스처는 상대적으로 빠른 움직임이 가능하도록 상체를 좌우로 기울이는 제스처로 가정한다(그림 3 참조). 본 연구는 일반적인 PC 환경에서 키보드의 키 조작을 통해 동작되는 게임에 제스처를 사용한 인터페이스를 적용하는 것을 목표로 하였으므로 사용자의 좌우이동 제스처를 기준으로 키를 누르는 것과 동일한 이벤트를 발생하여 게임을 제어한다.



[그림 2] 전체 시스템 흐름도

3.1 영상 분할

사용자의 움직임을 파악하기 위해서는 배경과 사용자를 구분하는 일이 선행되어야 한다. 복잡한 배경으로부터 사용자를 구분해 내는 일은 계산량이 많아 실시간 적용이 어려울 뿐만 아니라 정확성이 현저히 떨어지게 된다. 본 연구에서는 제스처 인식을 통해 실시간으로 게임을 동작시켜야 하므로 사용자의 움직임에 빠르고 정확하게 반응해야 하는 것이 필수적이다. 따라서 주위 배경을 단색으로 제한하고 입력영상으로부터 배경색을 제거하는 과정을 통해 빠르고 정확하게 사용자를 분리한 이진영상을 생성한다. 이러한 단순한 배경은 실제 아케이드형 게임 제작시 콘솔박스에

용이하게 적용할 수 있다. 그림 3 은 이러한 과정을 통해 배경과 사용자를 분리한 결과이다.

[그림 3] 영상 분할 결과

3.2 위치 추적을 위한 특징 blob 결정

3.2.1 모폴로지 연산

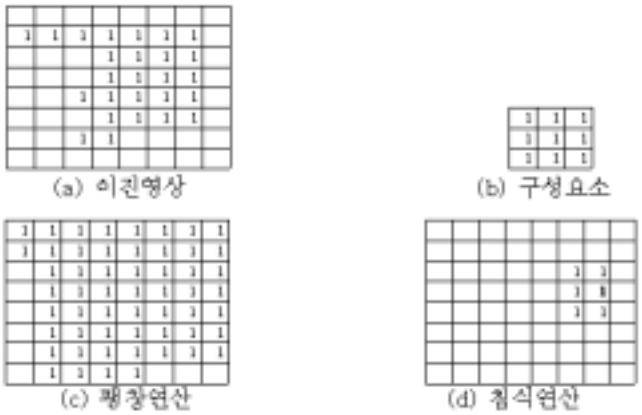
영상 분할 과정을 통해 얻은 이진 영상에는 작은 연결 영역들로 나누어진 사용자와 잡음들로 인해 많은 수의 블럽이 존재한다. 잡음을 제거하고 연산을 줄이기 위해 모폴로지 연산을 사용하여 블럽의 개수를 줄일 수 있다. 모폴로지 기법은 영상의 전처리 작업이나 초기 객체분류 또는 이러한 처리 뒤에 이어서 물체 원형의 구조를 명확히 하는데 주로 이용되며, 침식연산, 팽창연산, 제거연산, 채움 연산으로 나뉜다. 제거연산은 침식연산 다음에 팽창연산을 바로 사용하는 알고리즘으로서 침식연산으로 미세한 잡음을 제거한 후에 팽창연산으로 원영상의 크기를 복원한다. 채움연산은 그 반대의 기능을 하며, 이러한 제거연산과 채움연산을 조합하여 사용함으로써 효과적으로 잡음을 제거하고 원하는 영상을 얻을 수 있다.

식 (1)은 침식연산을 그리고 식 (2)는 팽창연산을 나타내며 그림 4 는 이진 영상 (a)에 (b)와 같은 구성요소를 적용하여 팽창(c)과 침식(d) 연산을 수행한 결과이다.

$$B \ominus S = \left\{ b \middle| b + s \in B \ \forall s \in S \right\} \tag{1}$$

$$B \oplus S = \bigcup_{b \in B} S_b \tag{2}$$

여기서 B 는 이진영상이며, b 는 B 의 한 픽셀, S 는 구성요소이며, s 는 S 의 한 픽셀을 나타낸다.



[그림 4] 모폴로지 연산

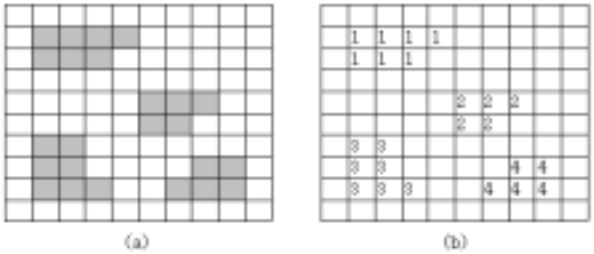
그림 5 는 3.1 절의 영상 분할 과정을 통해 얻은 영상에 제거연산과 채움연산을 적용하여 사용자를 나타내는 블럽을 병합하고 잡음을 제거한 결과를 보이고 있다.

[그림 5] 모폴로지 연산을 적용한 이진 분할 영상

3.2.2 연결요소 레이블링

영상분할을 위한 이진화 과정과 영역병합 및 잡음제거를 위한 모폴로지 연산을 수행한 영상에는 분할된 영역들이 존재한다. 서로 연결되어 있는 픽셀들의 집합을 하나의 블럽으로 간주하고 블럽들간의 구분을 가능하게 하는 연결요소 레이블링[4]

알고리즘을 사용하여 각 블럽을 구성하는 요소의 개수를 통해 블럽의 크기를 구할 수 있다. 그림 6 은 연결요소 레이블링의 간단한 예를 보이고 있다. 특징 블럽 검색의 연산을 줄이기 위해 블럽의 크기가 임계값보다 작은 경우는 무시한다.



[그림 6] 블럽들이 존재하는 이미지 (a)와 레이블링 된 이미지 (b)

3.2.3 특징블럽 결정

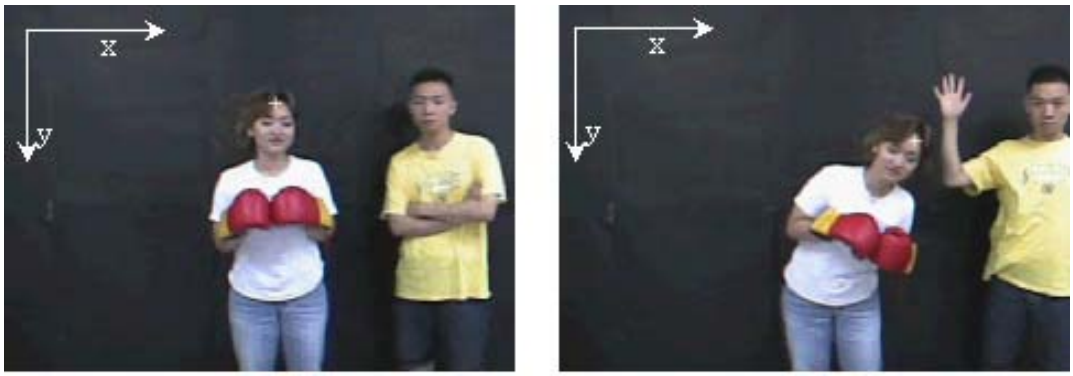
게임진행시 사용자는 카메라를 통한 입력영상을 볼 수 없으므로 입력영상 내부에 사용자 외에 다른 사람들이 존재할 가능성이 있다. 따라서 화면의 중앙으로부터 가장 가까운 곳에 위치한 블럽들 가운데에서 특징블럽을 결정하기 위하여 각 블럽의 중점을 구한다. $B(x,y)$가 영상의 x, y 위치에서 픽셀의 값을 나타낼 때 영상의 0 차, 1 차 모멘트는 아래와 같이 정의할 수 있다.

$$M_{00} = \sum_x \sum_y B(x,y), \quad M_{10} = \sum_x \sum_y xB(x,y), \quad M_{01} = \sum_x \sum_y yB(x,y) \tag{3}$$

각 블럽의 중점 x_c, y_c는 모멘트를 사용하여 다음과 같이 계산한다.

$$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}} \tag{4}$$

특징블럽은 중점의 좌표 x_c가 화면의 중앙에 근접해 있고 가장 작은 y_c값을 가지는 블럽으로 정의한다. 그림 7 은 입력영상에 사용자이외에 다른 사람이 존재하는 경우에도 특징 블럽 결정과정을 통해 사용자의 위치를 잘 찾아내고 있음을 보여준다. 십자로 표시한 위치는 특징블럽 내부에서 y_c좌표가 최소인 지점을 나타내고 있다.



[그림 7] 특징블럽 결정과정을 통한 사용자 위치 추적

3.3 움직임 벡터 누적을 통한 좌우 이동 동작 발생

사용자의 좌우이동 판단을 위해 특징블럽 내부에서 x와 y좌표가 최소인 지점을 특징점으로 정의하고 두 특징점의 이동을 벡터로 나타낼 수 있다. 이러한 특징점의 사용은 사용자가 배경과 동일한 색상의 옷을 입은 경우에도 이와 무관하게 움직임을 찾을 수 있다는 장점이 있다. 그림 8에서 나타나는 두 개의 십자선은 각각 이전프레임에서의 특징점 위치와 현재프레임에서의 특징점 위치를 나타내고 있다. 이러한 특징점의 위치변화를 기준으로 사용자의 좌우 이동을 판단할 수 있다.

(a)중앙→ 왼쪽 (b)왼쪽→ 중앙 (c)중앙→ 오른쪽 (d)오른쪽→ 중앙

[그림 8] 좌우 움직임에 따른 특징점의 이동

좌우이동 동작은 키보드의 사용방식과 동일하게 좌우로 몸을 기울인 상태를 유지하는 동안에는 연속적으로 키 이벤트를 발생하며 이동의 방향이 변경된 경우 즉시 키 이벤트 발생을 중단한다. 이때 상체를 기울인 상태에서 정지 동작시 발생할 수 있는 작은 움직임은 방향전환으로 인식하지 않는다. 이는 특징점의 이동을

나타내는 벡터의 크기를 각 방향에 따라 누적함으로써 가능하다. 그림 8 의 (a)와 같이 좌측으로 몸을 기울이는 경우 좌측 방향을 나타내는 벡터들을 누적하여 임계치 이상일 때 좌측 이동 키 이벤트를 발생한다. 벡터의 방향이 우측으로 바뀐 경우 역시 우측방향 벡터들을 누적하는데 이때 방향전환을 위한 임계치에 도달하지 않고 방향이 다시 좌측으로 변화하는 경우 우측방향 벡터의 누적값을 초기화 한다.그러므로 거의 멈춤 동작 상태에서의 상체진동인 경우 서로 다른 방향의 움직임들이 계속되어 정지상태 임을 나타낸다. 이와는 달리 우측으로의 방향전환을 위한 임계치에 도달한 경우에는 이전 상태 즉 좌측방향 벡터의 누적값을 초기화함으로써 또 다시 좌측방향으로의 이동이 발생하더라도 이동상태를 정확히 파악할 수 있다.

4. 펀치동작 인식

복싱 펀치동작의 특징은 움직임의 종류는 극히 제한적이면서 속도가 무척 빠르다는 점이다. 움직임을 검출하는 기존의 방식으로는 optical flow 방식, block matching 방식 등이 있으나, 이러한 방식들은 연산이 많아 실시간으로 처리하는 시스템에 적용하기에는 적합하지 않다[5][6]. 본 연구에서는 복싱게임의 펀치 동작 인식을 위해 최적화된 움직임 검출 및 특징점의 위치정보를 이용하였다. 펀치동작의 정확한 인식과 현실감 증진을 위해 사용자는 붉은색 글러브를 착용하며, 사용자의 머리위에 카메라를 설치하여 특별한 제약조건 없이 가장 효과적으로 펀치를 추출하고 인식한다.

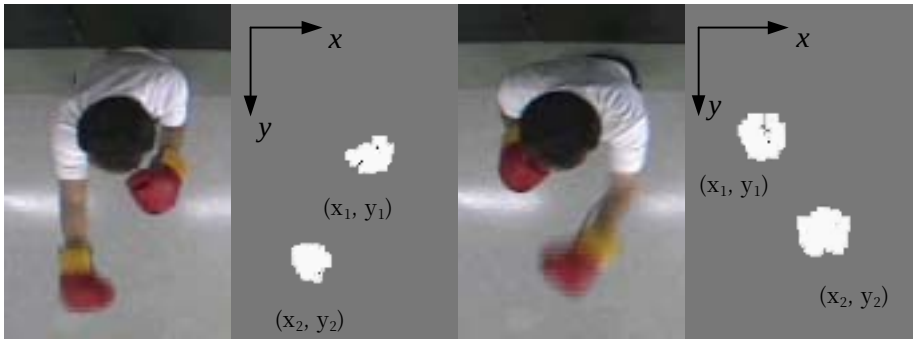
4.1 글러브 검출

본 연구에서는 빨간 글러브의 영역검출을 최대한 효과적으로 하기 위해 RGB 입력 영상을 매 프레임마다 Hue 영상으로 변환하였으며, 과정은 식 (5)와 같다.

$$H = \frac{a\cos(0.5\times((R-G)+(R-B)))}{\sqrt{((R-G)*(R-G)+(R-G)\times(G-B))}} \tag{5}$$

Hue 값은 칼라 성분의 값을 0°부터 360°까지 각각의 각도에 대응시켜 놓은 것으로, 색상이 하나의 특징 값으로 표시되는 장점이 있다. Hue 영상으로 변환된 영상은 적절한

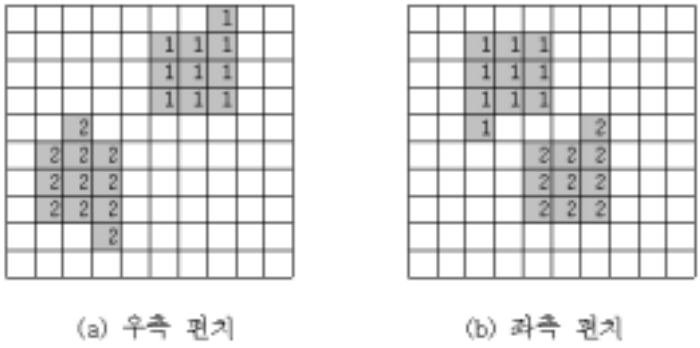
임계값을 적용하여 이진영상으로 바뀌지며, 가우시안 필터를 사용하여 작은 잡음성분들을 제거한다. 한 글러브 영역 내부에서 조각으로 검출되는 블럽들의 병합을 위해 영역확장기법을 적용하며, 모폴로지 연산을 통해 작은 블럽들을 제거한다. 그림 9 는 두개의 글러브 영역을 검출한 결과 영상이다. 잡음은 거의 완벽하게 제거되었으며 글러브 영역의 작은 블럽들도 하나의 블럽으로 병합되어 있음을 볼 수 있다.



(a) 우측 펀치 (b)좌측 펀치

[그림 9] 검출된 글러브 영역 검출 결과

3.2.2 절에서 언급한 레이블링 과정을 통해 남아있는 블럽들의 크기를 계산한다. 그 중 가장 큰 두개의 블럽을 선택하여 펀치 동작을 인식하며 검출된 두 개의 블럽은 그림 10 과 같이 레이블 된다.



[그림 10] 블럽 레이블링 결과

4.2 펀치동작 검출 및 인식

펀치동작은 왼쪽 펀치와 오른쪽 펀치, 펀치없음으로 정의한다. 글러브 영역 검출과정에서 결정된 두개의 블럽의 특징점을 이용하여 펀치여부와 좌, 우 펀치를 결정한다. 특징점은 영상의 Y 축을 기준으로 블럽의 최상위 좌표가 되며, 이는 두 주먹의 상대적인 위치를 비교하기에 좋은 기준이 된다. 특징점 위치정보를 이용한 펀치동작 인식 알고리즘은 다음과 같다.

1) 두 블럽의 특징점 y 좌표의 차이가 임계치보다 커지게 되면 펀치있음으로 정의한다.

$$If\ \ |y_1 - y_2| > T,\ then\ P = True$$
$$else\ P = False$$

y_1 과 y_2 는 각각 1 번과 2 번 블럽의 특징점 y 좌표이며, P 는 펀치동작의 여부를 나타내며 참인 경우 펀치있음의 상태이다.

2) P 가 참이면 좌, 우펀치 판단을 위해 특징점의 x 좌표들을 비교한다.

$$If\ \ x_1 - x_2 > 0,\ S = 1\ \text{(right punch)}$$
$$If\ \ x_1 - x_2 < 0,\ S = 2\ \text{(left punch)}$$

S 는 펀치의 좌, 우 상태를 나타내며 S 가 1 일 경우에는 오른쪽 펀치, 2 일 경우에는 왼쪽 펀치로 결정한다. 체감형 복싱게임에서는 사용자가 몸을 옆으로 숙이거나 좌우로 이동하면서 펀치동작을 하더라도 제대로 펀치를 인식해주어야만 현실감이 보장된다. 본 연구에서는 두 글러브 특징점의 상대적 위치정보를 고려하므로 그림 11 과 같이 사용자의 자세에 관계없이 펀치동작을 잘 찾아준다. 초당 30 프레임의 속도로 영상을 처리하는 경우, 주먹이 나가고 있는 상태나 주먹을 뻗어서 정지해있는 상태에서 두 특징점의 거리가 임계치를 넘어서면 계속해서 P 값이 참을 반환하게 되어 여러 번의 펀치가 나가는 오류가 발생할 수 있다. 이러한 문제점 해결을 위하여 이전 프레임의 펀치상태와 비교하여 현재 상태를 추정하는 과정이 필요하다.

[그림 11] 피하기동작 중의 펀치

3) S_t는 현재의 펀치상태, S_{t-1} 은 이전 프레임의 펀치상태를 나타낸다

$$if\ S_{t-1} = S_t, \quad P = \text{False}$$
$$if\ S_{t-1} \neq S_t, \quad S = S_t$$

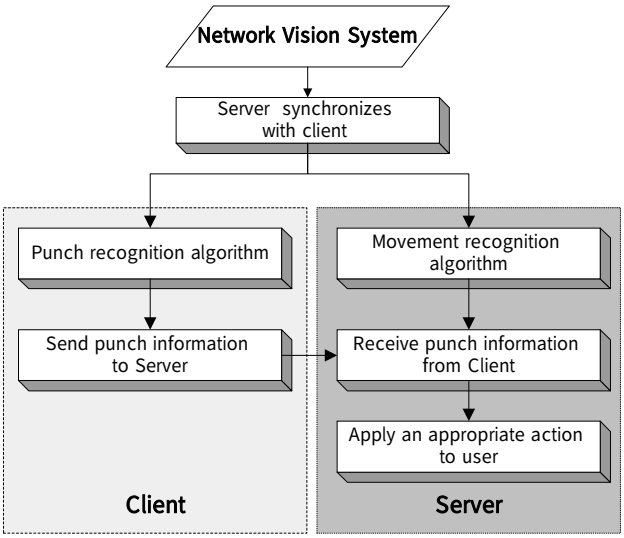
이전 프레임의 상태는 계속 갱신되어 현재 프레임의 상태와 비교되므로 펀치의 속도에 관계없이 정확한 펀치 검출이 가능하다.

5. 네트워크 비전 시스템 구성

복싱게임 인터페이스를 구성하는 두 컴퓨터 시스템이 구동될 때, 클라이언트 컴퓨터는 접속을 요구하는 패킷을 서버 컴퓨터에게 보내고, 서버에서는 접속을 허가하는 패킷을 클라이언트에게 보낸다. 이렇게 간단한 인증 과정을 거친 후, 두 컴퓨터 사이에는 사용자 펀치의 정보를 담은 패킷의 교환이 이루어지고, 이 연결된 네트워크는 전통적인 수직 클라이언트-서버 모델이 아닌 수평 구조를 갖는다.

인터넷을 매개로하는 네트워크에서 데이터 교환 속도는 인터넷 트래픽에 의해 좌우된다[7]. 따라서 실시간 구현과 시스템의 최적화를 목적으로 하는 본 연구에서는 임의의 인터넷 환경으로 인해 속도가 저하되는 현상을 방지하기 위하여, 두 컴퓨터는 크로스오버 케이블을 매개로하는 네트워크로 구성된다. 또한 상용의 네트워크 대전 플레이 게임에 본 연구에서 제안한 비전 시스템을 적용하면 인터넷을 매개로 하는 네트워크를 통해 다수의 원거리 플레이어들이 동시에 게임을 즐길 수 있다. 네트워크 비전 시스템은 클라이언트가 서버에 접속하여 인증을 획득한 후 형성된 네트워크

채널을 기반으로 사용자의 펀치 정보를 담은 패킷을 크로스오버 케이블을 매개로 하여 교환하도록 구성되어 있다. 전체적인 흐름은 다음과 같다.



[그림 12] 네트워크 비전 시스템의 구조

6. 실험결과

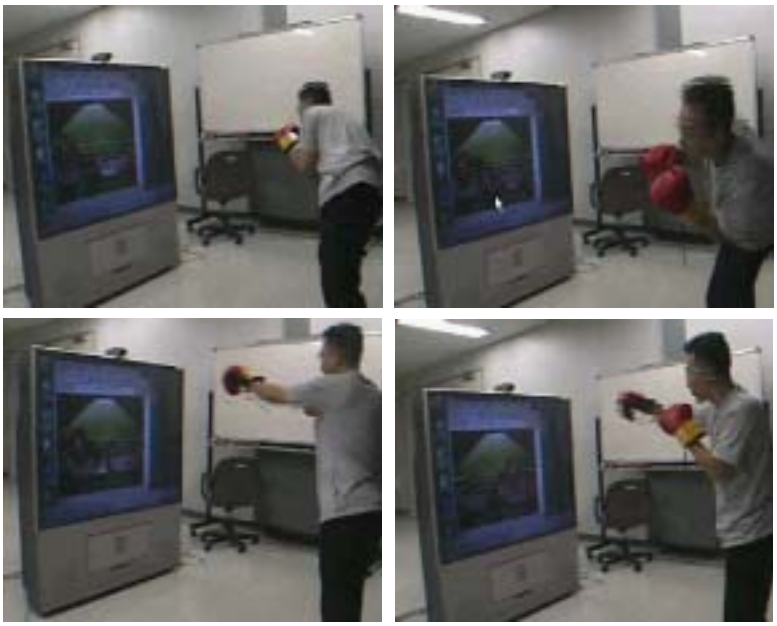
그림 13 은 펀치동작의 인식결과를 보여주고 있다. 빠른 속도의 펀치나 왼쪽, 오른쪽 펀치의 전환도 정확히 검출하고 있음을 알 수 있다.

(a) t 프레임 (b) t+1 프레임 (c) t+2 프레임 (d) t+3 프레임 (e) t+4 프레임 (f) t+5 프레임

[그림 13] 좌우펀치 연속동작

(d)의 경우 두 주먹의 위치가 펀치로 판단할 임계치를 넘었으나 이전 프레임에서 나왔던 펀치가 들어가는 중이므로 펀치로 인식하지 않았다.

본 논문에서 제안한 기법을 플래쉬로 제작된 PC 용 복싱게임 ‘O.J. Boxing’ 에 적용하여 실험하였으며 대화형 입력방식으로 기존의 키보드 사용을 대체하여 더욱 현실감 있는 게임진행이 가능함을 확인였다. 그림 15 은 좌우이동 동작과 펀치동작을 사용하여 실제로 복싱 게임을 동작하는 장면을 보여준다.



[그림 15] 복싱게임의 동작

7. 결 론

본 연구는 컴퓨터 비전기술을 사용하여 사용자의 제스처를 인식하고 이를 통해 일반 PC 용으로 제작된 게임을 보다 현실감 있게 제어할 수 있는 방법을 제안하였다. 그리고 기존의 키보드 제어방식으로 제작된 PC 용 복싱 게임에 본 연구를 통해 제안된 효율적인 시스템 구성방식과 제스처 인식 알고리즘을 적용하여 대화형 인터페이스를 사용할 수 있음을 보였다.

대화형 인터페이스를 통해 단순히 키보드의 버튼을 누르거나 센서가 장착된 장비를 사용하지 않고 사람의 몸동작만으로 게임을 제어하는 경우 게임의 현실감 증대는 물론 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간 친화적인 인터페이스가 가능함을 확인하였다. 향후 이와같은 기술을 충분히 활용할 수 있는 흥미롭고 다양한 게임 컨텐츠의 발굴이 기술개발과 병행되어야 할 것이다.

참 고 문 헌

[1] Ming-Hsuan Yang, Ahuja, N., and Tabb, M., "Extraction of 2D motion trajectories and its application to hand gesture recognition", IEEE Trans, on Pattern Analysis and Machine Intelligence, Vol. 24(8), pp. 1061-1074, Aug. 2002.

[2] Andrew Wu, Mubarak Shah and N. da Vitoria Lobo, "A Virtual 3D Blackboard: 3D Finger Tracking using a Single Camera" Proceedings of the Fourth International Conference on Atomatic Face and Gesture Reconition, pp. 536-543, 2000.

[3] Freeman, W.T., Anderson, D.B., Beardsley, P.A., Dodge, C.N., Roth, M., Weissman, C.D., Yerazunis, W.S., Kage, H., Kyuma, K., Miyake, Y., and Tanaka, K., "Computer Vision for Interactive Computer Graphics", IEEE Computer Graphics and Applications, Vol. 18, No. 3, pp. 42-53, May-June 1998.

[4] Milan Sonka, Vaclav Hlavac, and Roger Boyle, "Image Processing, Analysis, and Machine Vision" Second Edition, PWS Publishing.

[5] Pentland A. and Horowitz B. "Recovery of nonrigid motion and structure", IEEE Trans, on Pattern Analysis and Machine Intelligence, Vol. 13(7), pp.730-732, July 1991.

[6] M.Isard, A. blake, "CONDENSATION-conditional propagation for visual tracking", Int. J. Computer Vision, 1998.

[7] Gregory T. Kogut, Mohan M. Trivedi, “Maintaining the Identity of Multiple Vehicles as They Travel Through a Video Network”, IEEE Workshop Multi-Object Tracking, 2001. pp. 29-34,