
웹 카메라를 이용한 체감형 게임의 충돌감지를 위한 특징맵

이영재* · 이대호**

Feature Map for Collision Detection in Motion-Based Game using Web Camera

Young Jae Lee* · Dae Ho Lee**

요 약

체감형 게임 제작에서 필요한 충돌 감지를 위한 특징맵 방법을 제안한다. 특징맵은 충돌이 발생하는 물체들의 중첩 픽셀 비를 불변량으로 이용하는 방법으로 영상의 크기를 환경과 조건에 따라 조절해가면서 정할 수 있으므로 빠른 충돌감지가 가능하며 또한 충돌 부위도 감지 할 수 있는 장점이 있다. 이를 검증하기 위하여 web cam, 게임플레이어, 적캐릭터, 가상 오브젝트를 이용한 체감형 게임을 제작하고 특징맵을 사용하여 충돌 감지를 실험하여 그 타당성을 확인하였다. 이 방법은 게임에서 캐릭터 크기와 위치가 지속적으로 변화하는 경우 효과적인 충돌 감지를 위한 기본적인 알고리즘으로 응용될 수 있다.

ABSTRACT

We propose a feature map method to detect a collision for a motion-based game. The feature map can be made an optimally reduced motion data using subtraction image and virtual ball images according to image size and condition. And we calculate the overlapped ratio between moving image data and objects. This ratio is an invariant for detection even though image size is changed. And we compare this ration with collision detection constant, the feature map can detect fast collisions as well as the collided direction. To evaluate the method, we implemented a motion-base game that consists of a web cam, a player, an enemy, and some virtual balls, and we obtained some valid results for our method for the collision detection. The results demonstrated that the proposed approach is robust, and they can be used as a basic collide detection algorithm for a motion-based game where the size and the position of characters are continuously changing.

키워드

웹카메라, 체감형게임, 충돌, 특징맵

I. 서론

최근 문화산업의 핵심 화두가 되어있는 게임은 문화적 요소, 예술적 요소, 첨단 기술적 요소가 접목된 종합 문화 예술로써 국가경쟁력을 높일 수 있는 유망산업으

로 부각되고 있으며 그 수요와 잠재력이 무한한 산업으로 주목을 받고 있다[1-10]. 또한 멀티미디어 분야의 핵심 분야로 고부가가치를 창출할 수 있는 업종으로 인식되면서, 세계 각국에서는 새로운 장르 및 관련 기술개발에 치열한 경쟁을 벌이고 있다. 최근에는 사용자와 게임

* 전주대학교 문화산업대학 멀티미디어전공 교수

접수일자 : 2008. 2. 29

** 경희대학교 학부대학 교수

간의 양방향 상호작용에 의해 게임 사용자의 현실감과 몰입감 등을 극대화하여 사용자의 반응에 의한 변화를 피드백 받을 수 있도록 하는 게임이 인기를 끌고 있으며 서서히 기본적인 임의 기본조건으로 자리를 잡아가고 있고, 게임 화면 속 캐릭터의 모습을 바깥 모습을 통해 자유자재로 표현 하는 것처럼 눈과 손으로만 이용하는 게임과는 달리 온몸으로 느끼고 즐기는 체감형 게임이 세계 게임 시장의 새로운 트렌드로 떠오르고 있다. 인간의 경험은 직접적인 신체감각을 통해 다차원적으로 정보를 인식할 때 효율적이다. 따라서 직접적인 신체감각을 통해 이용하는 체감형 게임은 일반 게임과는 달리 사용자가 얻을 수 있는 만족감을 극대화 시켜준다[3]. 그러나 현재 PC용으로 개발 된 모든 게임들은 키보드, 마우스, 조이스틱과 같이 매우 제한적인 하드웨어를 이용하여 동작하도록 구성되어 있다. 이와 같은 장비의 사용은 공간상의 제약은 물론 게임의 현실성을 떨어뜨리는 주된 요인 이라고 할 수 있다. 최근 발표되고 있는 체감형 아케이드게임기들은 장비에 센서를 부착하여 체감성을 증대시키는 효과를 보이나 부가적인 하드웨어를 수반한다. 단순한 키보드 조작이나 센서가 장착된 장비를 사용하지 않고 사람의 몸동작만으로 게임을 제어한다면 게임의 현실감 증대는 물론 컴퓨터를 좀 더 사용자 중심으로 만드는 새로운 인간 친화적인 인터페이스가 가능하다[4].

본 연구는 web 카메라, 게이머, 적 캐릭터, 가상 볼 등으로 구성된 체감형 게임을 만들고 게임상의 가상물체와의 효과적인 충돌 감지를 제안한 알고리즘 사용하여 구현해볼 그 타당성을 검증한다. 본 논문의 구성은 I장 서론, II장 체감형 게임, III장 충돌, IV장 제안한 알고리즘, V장 실험 VI장 결론의 순으로 구성되어 있다.

II. 체감형 게임

2.1 체감형 게임의 정의[3]

체감형 게임은 직접적인 신체 감각의 접촉 및 신체의 움직임에 반영하여 실제적인 동작을 모방함으로써 실제감과 현실감을 느낄 수 있는 게임이다. 또한 체감형 인터페이스를 사용함으로써 더 큰 재미와 몰입감을 제공한다. 인간의 경험은 시각적 청각적 자극과 함께 물리적인 상호작용, 즉 직접적인 신체감각의 접촉을 통해 다차

원적으로 정보를 인식할 때 보다 효율적이다[5]. 특히 게임을 즐기므로써 얻을 수 있는 만족감 내지는 몰입감은 사용자에게 전달되는 다양한 신체감각에 대한 피드백이 복합적으로 이루어졌을 때 효과적이다[6]. 그러나 체감형 게임의 개발은 단순한 PC게임과는 달리 소프트웨어뿐만 아니라 하드웨어를 동시에 개발해야 한다는 어려움을 갖고 있다. 체감형 게임은 대형 오락실 등에서 제공되는 아케이드 체감형 게임, 게임 컨트롤러 체감기능을 사용하는 콘솔 체감형 게임, 개인용 컴퓨터를 기반의 PC 체감형 게임으로 나눌 수 있다[3]. 이중 PC 체감형 게임의 경우 콘솔게임이나 아케이드 게임에 비해 게임 그래픽이나 정보전달을 위한 피드백 지원 등 게임 리얼리티나 판지적인 게임 속성 구현에는 미흡하나 누구나 쉽게 접근할 수 있으며 PC 하드웨어를 이용한 저렴한 비용의 게임제작 등 장점이 있다.

2.2 Web-Cam을 사용한 체감형 게임 제작

소니의 아이토이(Eye-toy) 등 Web-Cam을 사용한 다양한 게임이 출시되고 있다. 그러나 저렴한 웹캠과 PC를 사용한 게임들은 많지 않으며 PC상에서 게이머의 움직임과 가상객체들간의 충돌, 특수효과, 등 다양한 이벤트에 대한 게임구현을 위한 기술연구가 활발하게 진행 중이다[1-8].

본 연구에서는 web cam을 사용하여 게이머와 가상의 볼(오브젝트), 적 캐릭터를 등장시켜 공을 터트려 득점하는 1인칭 체감형 게임을 만들고 중요 이벤트인 충돌을 제안한 특징맵 알고리즘을 사용하여 그 성능을 실험해 본다.

III. 충돌

충돌은 상대적으로 운동하는 물체 또는 입자가 근접 또는 접촉해서 상호작용을 미치는 현상이다. 따라서 이 같은 충돌현상에는 물체의 모양, 크기, 운동량, 물성 등 여러 가지 충돌에 영향을 미치는 물리학적인 파라미터가 존재하며 점과 점의 충돌로부터 2차원 충돌, 3차원 충돌 등 차원에 따라 구분할 수도 있다[7].

이 같은 충돌현상을 미리 예상하고 그에 대한 반응을 생각하기 위한 충돌감지 기능은 컴퓨터 그래픽, 애니메이션, 컴퓨터 게임 등 다양한 분야에 꼭 필요한 필수적인

기능이다. 특히 컴퓨터 게임 분야에서는 다양한 충돌 이벤트를 통해 가상과 현실과의 상호작용을 제공하고 더불어 게임을 재미있게 만드는 중요한 요소 중 하나이므로 이에 따른 적절한 충돌감지 알고리즘의 개발이 필요하다[7-10]. 컴퓨터 게임에서의 충돌은 실세계에서 일어나는 일들을 그대로 시뮬레이션하여 보여주는 다이내믹 시뮬레이션 시스템(Dynamic simulation system)과 같은 정밀성을 요하는 것은 아니지만 정확하고 빠른 감지가 가능한 특성을 만족해야 한다. 따라서 오브젝트의 전체형상을 하나의 단위로 감지하는 원, 사각형 사용방법이 주종을 이루었으나, 최근에는 프로세서의 성능향상에 기인하여 각종 다양한 방법의 개발과 더불어 애니메이션에서 사용하는 여러 기법 등도 적용해 사용하고 있다.

3.1 충돌 감지 알고리즘

3.1.1. 원을 사용한 방법

원을 사용하는 방법에서는 이들의 반지름을 사용한다. 즉 이들 두 원 반지름의 합을 상수로 정해 이 상수보다 작은 경우 충돌로 감지하는 방법이다.

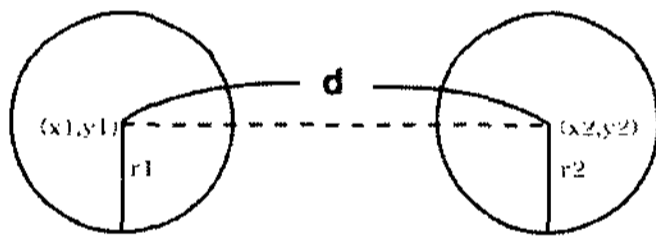


그림 1. 원을 사용한 충돌 감지
Fig. 1. Collision detection using circle

두 원 사이의 거리 d 는 식(1)과 같이 나타낼 수 있다.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

이 경우 $d \leq r_1 + r_2$ 이라면 두 원의 충돌이 발생한 경우이다. 원 둘레에 대한 좌표값은 $x_n^2 + y_n^2 = r_n^2$ 를 사용해 구할 수 있다.

3.1.2. 사각형을 사용한 방법

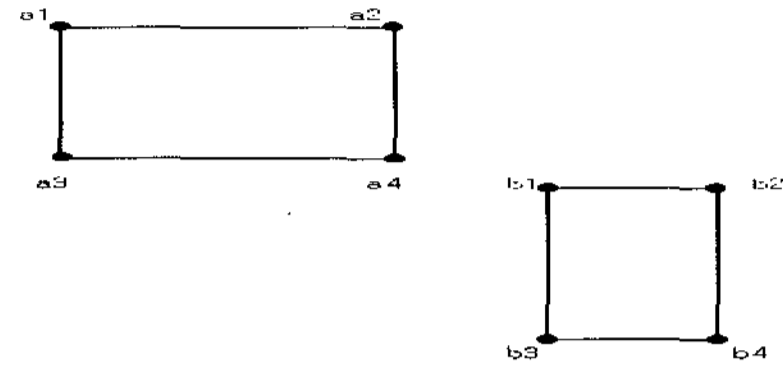


그림 2. 사각형을 사용한 충돌 감지
Fig. 2. Collision detection using rectangular

사각형은 4개의 특징점 좌표를 가지고 있으며 이들은 동일한 x 좌표, y 좌표를 가지고 있으므로 이들의 좌표값을 이용하여 충돌여부를 확인할 수 있다. 예를 들면 a4의 좌표를 (x4,y4)라 하고 b1의 좌표를 (x1',y1')라 하면 (x1' > x4 || y1' > y4) 경우와, a1의 좌표를 (x1,y1)라 하고 b4의 좌표를 (x4',y4')라 할 때 (x1 > x4' || y1 > y4') 경우 중 하나만 만족하면 충돌이 발생하지 않은 경우이다. 이 방법은 충돌체크가 간단한 장점은 있으나 오브젝트의 형상(예: 삼각형)에 따라 정밀한 감지가 어렵기 때문에 환경에 따라 제한적으로 사용할 수 있다.

3.1.3. 부분 분할 데이터를 이용한 방법

부분 분할 데이터(bounding data)[8]를 이용한 방법은 오브젝트의 트리나 특징점을 기준으로 원, 사각형을 이용해 오브젝트를 바운딩해 위치를 설정하고 이들 좌표를 데이터 베이스로 사용해 충돌을 감지하는 방법이다.

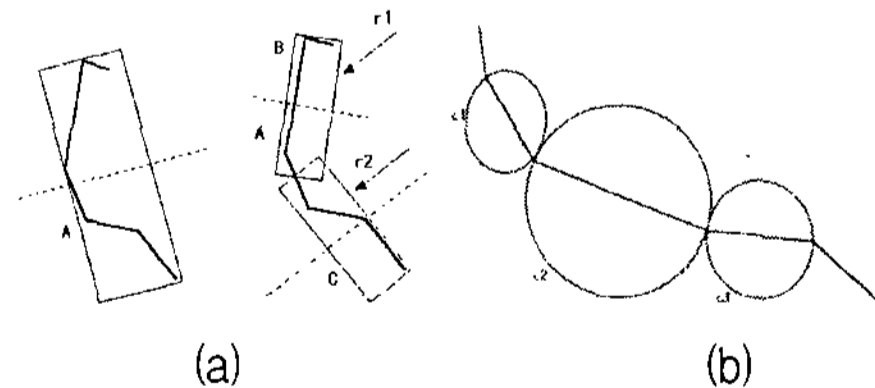


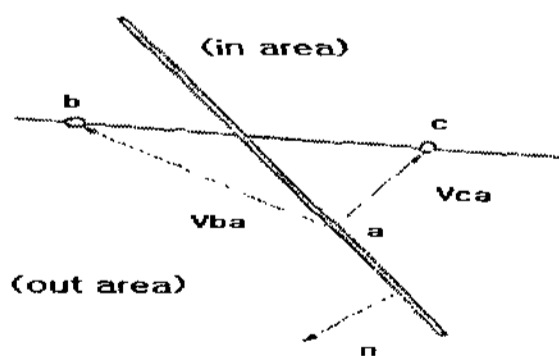
그림 3. 부분 분할 데이터를 이용한 충돌감지
Fig. 3. Collision detection using partial shape data

그림 3의 (a)(b)는 특징점 혹은 트리를 구성해 부분형상으로 나누고 이를 원이나 박스를 사용하여 오브젝트의 부분형상으로 나누어 데이터베이스로 활용하는 방법이다. 그림(a)에서는 c1,c2,c3 등 3개의 원을 사용하는 경우이며, 그림(b)의 경우 트리를 사용해 박스(r1,r2) 2개를 사용한 경우이다. 이 방법은 원과 박스를 오브젝트의

형상에 따라 세밀하게 나눔에 따라 정밀한 충돌 감지가 가능하나 충돌을 위한 비교 데이터 량이 많아지는 단점이 있다.

3.1.4 Cyrus-Beck clipping 알고리즘

Cyrus-Beck clipping 알고리즘[9-10]은 convex rigid body에 대한 충돌감지 알고리즘이다. 이 알고리즘에서는 충돌하는 경우를 다음 그림과 같이 두 가지 경우로 구분한다. 즉 한 물체의 꼭지점(vertex)이 다른 물체의 면(polygon)에 충돌하는 경우와 또 다른 경우로는 물체의 면(edge)과 다른 물체의 면 경계에(polygon boundary) 충돌하는 경우이다. 즉 한 물체의 면경계가 다른 물체의 면 경계와 충돌하는 경우이다. 그림 4에서 보는 것과 같이 굵은 선으로 표시된 무한직선에 대해 가는 선으로 표시된 무한 직선을 자른다고 생각해 보자. 즉, 굵은 선으로 표시된 무한직선의 양쪽을 그림 4와 같이 in, out으로 가정했을 때 가는 선의 in 쪽에 있는 부분만을 나타내하고자 한다. 이때 이 굵은 선의 외부 수직벡터(outward normal vector)를 그림과 같이 \vec{n} (vec n)으로 표시하고 굵은선 위의 임의의 한점을 a, 자르고자(clipping)하는 선 사이에 존재하는 임의의 점을 각각 b(굵은선 밖에 존재), c(굵은선 안에 존재)라 할 때 벡터 \vec{n} 과 $\vec{v}_{ba}(\vec{v}_{bo} - \vec{v}_{ao})$, $\vec{v}_{ca}(\vec{v}_{co} - \vec{v}_{ao})$ (0는 기준점) 사이에서 그림 6과 같은 관계가 성립한다. 즉, 굵은선 밖에 있는 임의의 점 b와 굵은선 상의 임의의 점 a 사이의 벡터 \vec{v} 와 \vec{n} 의 내적은 항상 0보다 크다.



$$n \cdot V_{ba} > 0 \quad n \cdot V_{ca} < 0$$

그림 4. Cyrus-Beck clipping 알고리즘의 기본 원리
Fig. 4. The basic principle of Cyrus-Beck clipping algorithm

두 번째의 경우인 무한 평면과 직선의 교차점 계산인 경우 무한 평면의 수학적 표현은 여러 방법으로 표시될 수 있지만 벡터식으로 표시해보면 식(2)와 같다.

$$\vec{n} \cdot \vec{q} = 0 \tag{2}$$

여기서 \vec{n} 은 무한 평면의 수직 벡터이고 \vec{q} 는 무한 평면 위에 놓인 벡터이다. 여기에서 다시 \vec{q} 는 $\vec{p} - \vec{p}_1$ 으로 표시할 수 있는데 \vec{p}_1 은 무한 평면위 특징점의 기준점에 대한 위치 벡터(position vector)이고 \vec{p} 는 무한 평면에 존재하는 임의의 한 점에 대한 위치 벡터 [x, y, z]이다. 그러므로 식 (2)를 다시 쓰면 평면식은 (3)으로 표시된다.

$$\vec{n} \cdot \vec{p} = \vec{n} \cdot \vec{p}_1 \tag{3}$$

직선의 parametric 표현은 직선의 시작점을 v_1 , 끝점을 v_2 라 할 때 식 (4)과 같이 표시된다.

$$\vec{\tau}(t) = \vec{v}_1 + t(\vec{v}_2 - \vec{v}_1) \tag{4}$$

무한 평면과 직선의 교차점 계산은 (3)식을 (4)식에 대입 하므로써 t값을 계산할 수 있다. 이때 t가 0과 1사이의 값이면 직선과 무한 평면은 $\vec{\tau}(t_{cross})$ 점에서 교차한다고 하면 이를 식으로 나타내면 식 (5)과 같다.

$$t_{cross} = \frac{\vec{n} \cdot (\vec{p}_1 - \vec{v}_1)}{\vec{n} \cdot (\vec{v}_2 - \vec{v}_1)} \tag{5}$$

이 알고리즘은 정밀한 충돌에 대한 정보를 얻을 수 있으므로 게임 뿐 아니라 애니메이션과 같은 분야에서도 사용 가능하다. 그러나 오목형 타입(concave)의 오브젝트인 경우 적용할 수 없는 단점이 있다.

IV. 제안한 알고리즘

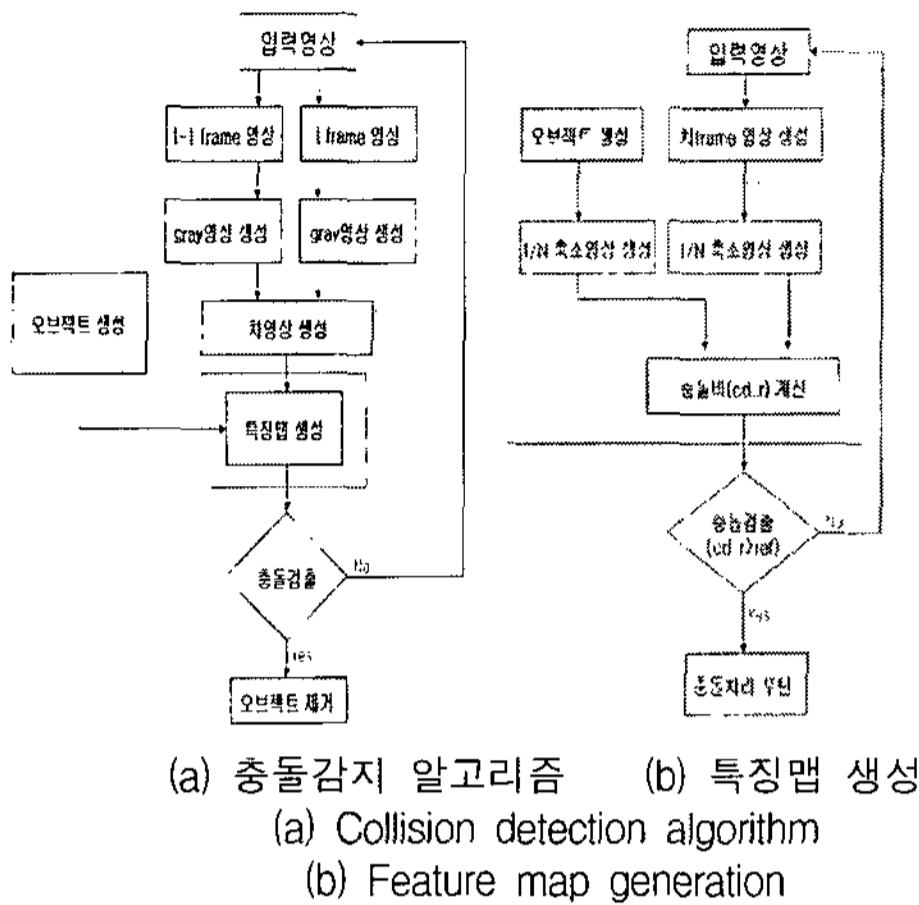


그림 5. 제안한 알고리즘의 구성
 Fig. 5. Flowchart of the proposed algorithm

그림 5는 제안한 알고리즘을 나타낸 것으로 그림 (a)는 입력영상1(t-1 frame)과 입력영상2(t frame)영상의 RGB 값을 그레이 영상 값으로 변환한 후 이들의 차영상을 구하고 이 차영상을 사용하여 특징맵을 구한 다음 가상으로 생성 되는 오브젝트와 충돌 검사를 한다. 이때 사용하는 특징맵(그림 (b))은 오브젝트와 차영상의 크기와 조건에 따라 1/N로 축소한 다음 오브젝트 픽셀과 축소영상의 픽셀 중첩비(충돌비:cd_r)와 충돌확인 변수(ref)와의 대소 구분에 의하여(cd_r>ref) 충돌여부를 판가름한다. 이때 픽셀의 중첩비를 사용하므로 영상의 크기가 1/NN으로 축소된다고 하더라도 불변량적인 속성 때문에 그 크기가 변하지 않는다는 특징을 이용한 것이다. 또한 오브젝트나 차영상의 중첩비를 이용하므로 각각의 좌표정보를 알 수 있으므로 어느 위치가 얼마만큼 중첩(충돌)되었는지도 알 수 있는 장점을 가지고 있다.

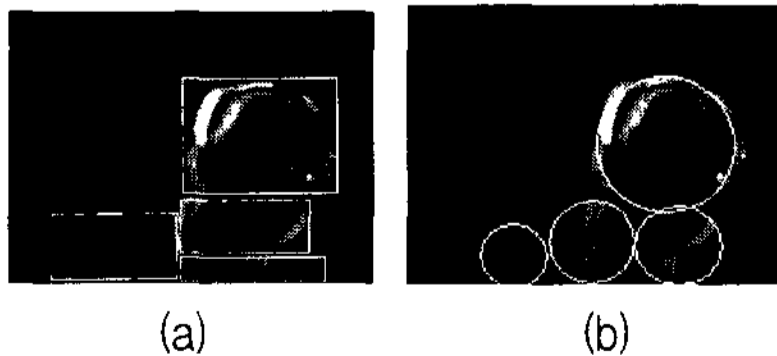


그림 6. 부분 데이터 활용 방법
 Fig. 6. Collision detection using partial shape

그림 6은 게임에서 캐릭터의 충돌 감지를 위하여 비교적 정도가 높은 부분 데이터 활용기법을 사용한 경우

를 나타낸 것이다. 그림(a)는 4개의 사각형을 사용하였으며 그림(b)는 원 4개를 사용한 경우이다. 이 같은 방법의 문제점으로 (a),(b) 모두 감지해야하는 부분을 감지하기 위한 사각형이나 원의 정확한 위치를 자동으로 선정, 계산 할 수 없으며 또한 움직임 위치 데이터 모두를 포함할 수 없기 때문에 정확한 충돌 감지가 어렵다. 특히 변화부분이 많고 불규칙한 경우 그 수가 기하급수적으로 늘어날 수 있으며 정확한 충돌위치 및 충돌정도를 감지하기도 어렵다. 따라서 이 같은 문제점을 캐릭터와 오브젝트간의 픽셀비를 사용하는 제안한 알고리즘을 사용하여 해결 할 수 있다. 그림 5의(b)는 축소 특징맵을 만드는 방법을 나타낸 것으로 예를 들어 320 * 240 크기 영상에서 반지름 50픽셀과의 오브젝트인 경우 화면크기의 경우 전체화면 영상의 픽셀 수 : 76,800 개, 반경이 50인 오브젝트의 픽셀 수 : $3.14 * 50 * 50 = 7,850$ 이며 축소 비율(NN)이 1/2인 경우 전체화면 영상의 픽셀 수 : $76,800/2 = 38,400$ 개 오브젝트의 픽셀 수 : $7,850/2 = 3,925$ 로 입력영상의 경우 38400개의픽셀 수를 둘일 수 있다. 축소 비율에 따른 비교되는 픽셀 수의 변화는 표1과 같다.

표 1. 축소비율에 따른 특징맵 크기
 Table. 1 Feature map size according to reduction ratio

방법	화면픽셀 수(NN=1/2)	화면픽셀 수(NN=1/4)	화면픽셀 수(NN=1/8)
차영상	76,800	76,800	76,800
특징맵	38,400	19,200	9,600
픽셀수 차(개)	38,400	57,600	67,200

표1과 같이 입력영상의 크기와 환경에 따라 축소비율을 조정해 갈 수 있다.

V. 실험

web cam, 가상 볼, 적 캐릭터, 주인공 캐릭터로 구성된 간단한 1인칭 체감형 게임을 만들고 먼저 설정한 점수로 득점한 경우 우승하는 내용으로 제안한 알고리즘을 실험하였다.

5.1 실험 1.

실험 1에서는 제안한 알고리즘을 검증 할 수 있는 환경 구축 실험으로 web cam 입력영상, 가상으로 만든 볼 그리고 적 캐릭터(그림(b) 참조)[11]를 한 화면에 출력하여 보았다. 이때 시간과 점수에 따라 더 많은 볼 생성과 적 움직임을 빠르게 하여 레벨을 조정할 수 있도록 구성

하였으며, 생성된 볼 개수 대 터치 한 볼 수의 비(%)를 점수로 나타내었다.

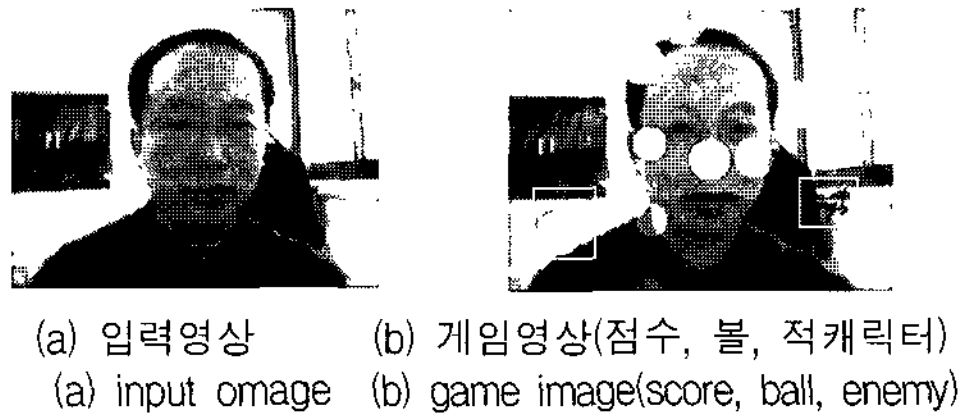


그림 7. 실험 1 영상
Fig. 7. Experiment 1 image

5.2 실험2.

실험2에서는 상체를 움직여 게임하는 경우로 상체와 볼과의 충돌을 제안한 충돌맵 알고리즘을 사용하여 충돌 감지 성능을 확인하였다.

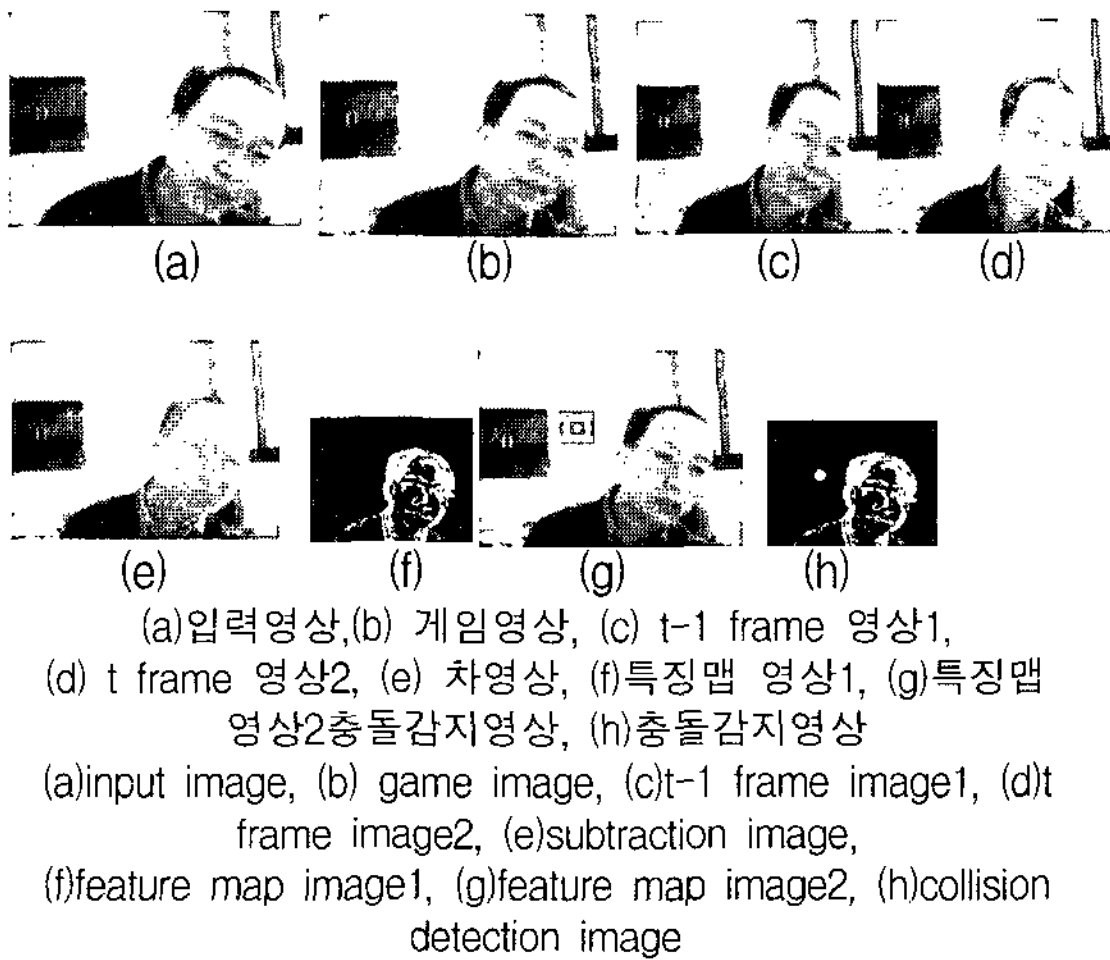


그림 8. 실험 2 영상
Fig. 8. Experiment 2 image

그림(6)의 그림(a)는 입력영상을 그림(b)는 가상의 볼을 생성한 한 것을 나타낸 것이며 그림(c)(d)는 차영상을 만들기 위한 t-1, t 프레임 영상을 나타낸 것이다.

그림(e)는 움직임 데이터 영상이며 그림(f)는 제안한 알고리즘 특징 맵 영상1(축소비 NN=1/2), 그림(g)는 가상 볼 특징맵 영상2(작은 사각형 안의 작은 원(축소비 NN=1/2))를 나타낸 것이며 그 외각 사각형은 원래 원(그림(g))의 영역을 나타내기 위하여 표시된 것이다. 이때 만들어진 2개의 특징맵(그림(f), 그림(g)) 위치 비교를 통

하여 서로 중첩된 픽셀의 비를 구한다. 이 비율은 크기의 영향을 받지 않는 불변량이 되므로 축소비 NN을 입력영상의 크기, 충돌 감지 환경 등을 고려하여 최적 값으로 설정한다. 설정된 NN 값에 따른 충돌값을 기준 충돌비(ref)로 정해진 값과 비교하여 충돌 유무를 결정한다. 그림(h)는 가상 볼과 차영상의 특징맵 영상을 사용하여 충돌감지를 위하여 사용하는 영상으로, 이동 변화정보 영상과 가상 볼 사이에 중첩되는 픽셀이 없으므로 충돌이 발생하지 않은 경우이다.

특징맵을 사용한 충돌감지성능 측정을 위하여 피 실험자 5명이 5회씩 5분 동안 게임을 하였다. 대부분의 경우 발생하는 가상볼을 터치 할 수 있었으며, 시간, 점수가 높아짐에 따라 볼의 속도, 개수가 변경되므로 터치할 수 없는 볼도 발생되었다. 그러나 평균 96.4%의 득점을 하였고, 충돌이 되었는데 감지가 안 된 경우는 없었으므로 이 실험을 통하여 정확한 충돌감지 성능을 확인 할 수 있었다.

5.3 실험 3.

실험3에서는 실험2 영상에 적 캐릭터를 추가하여 게임 실험을 수행하였다. 이때 볼과 적 캐릭터, 플레이어와 적 캐릭터는 단순한 충돌 유, 무만을 확인하기 위한 사각 충돌방법을 이용하였고 게임 플레이어와 가상볼은 특징맵을 사용하였다.

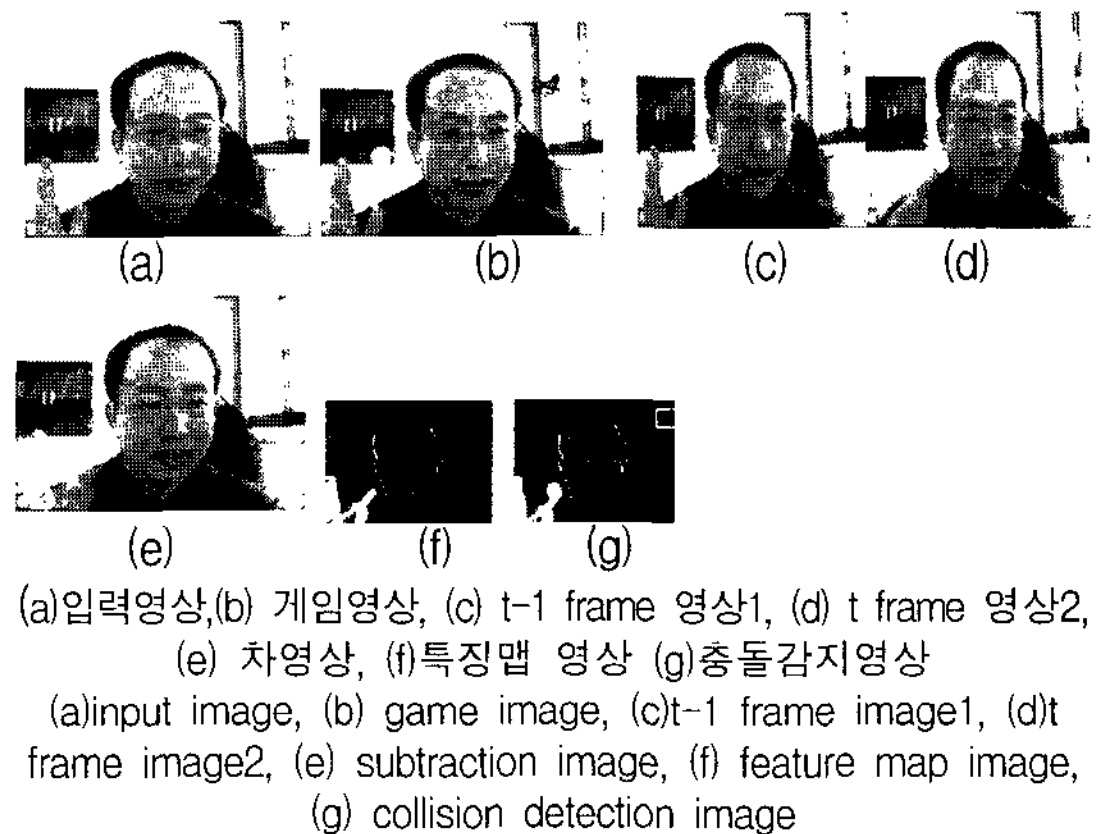


그림 9. 실험 3 영상
Fig. 9. Experiment 3 image

그림 7의 그림(a)는 압력영상, (b)는 가상 볼 오브젝트와 적 캐릭터를 등장시킨 영상이며 그림(c), (d)는 t-1 프레임, t 프레임 영상으로 손의 위치가 달라져 있음을 알 수 있다. 그림(e)는 (c)와 (d)의 차영상으로 움직임 정보를

(붉은 픽셀) 나타낸 영상이며 그림(f)는 차영상을 이용한 특징맵을 나타낸 영상이다. 그림(g)는 충돌감지 영상으로 손의 움직임 데이터와 불이 중첩되어 있으므로 충돌로 감지가 되어 득점을 하고 그 불은 다음 프레임에서 나타나지 않는다. 또한 우측 상단의 사각형은 적 캐릭터의 움직임 영역을 나타낸 것으로 이 영역을 이용하여 가상의 불 혹은 게이머와의 충돌을 감지할 수 있다. 실험2와 마찬가지로 적 캐릭터가 추가된 상황에서 충돌감지 성능 측정을 위하여 피실험자 5명이 5회씩 5분 동안 게임을 하였다.

적 캐릭터가 득점을 위한 경쟁자의 역할을 수행하고 있으며, 가상 불 역시 시간과 득점에 따라 이동 속도와 위치가 변하게 되므로 가상불 터치가 실험 2보다 난이도가 높았으나 평균 88.1%의 득점을 하였다. 이 같이 경쟁자로 인한 난이도 상승과 속도감이 필요한 경우에도 자연스러우며 강인한 충돌감지가 가능하였다.

VI. 결론

체감형 게임에 있어서 효과적인 충돌감지 방법은 꼭 필요한 물리요소이다. 사각형 충돌감지 방법, 원 충돌감지 방법, Cyrus-Beck clipping 알고리즘 등 다양한 방법이 존재하나 변화량과 위치가 불규칙한 특성을 갖는 게임에서는 효과적인 충돌감지 방법으로 적용하기가 어렵다. 따라서 이 같은 문제를 해결하기 위하여 차영상 정보와 오브젝트의 픽셀 정보의 비를 불변량으로 이용 할 수 있다. 이때 계산량을 줄이기 위하여 특징맵 방법을 제안하였으며 이를 사용하여 충돌감지를 실험한 결과 경쟁자가 없는 경우엔 96.41% 경쟁자가 있는 경우엔 88.1%의 득점을 하였다. 즉, 난이도가 낮은 경쟁자가 없는 일반적인 경우와 경쟁자가 있고 난이도가 높고 속도감이 있는 빠른 경우에도 높은 충돌 감지 성능을 나타냈다. 이때 사용하는 특징맵의 크기는 입력영상의 크기와 오브젝트의 크기, 환경 등을 고려하여 차영상 보다 더 작게 만들 수 있으므로 빠른 충돌 감지 구현과 정확한 위치 파악이 가능하다.

참고문헌

[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 Automatic Face and Gesture Recognition*, pp. 536-543, 2000.

[3] 지수미, 체감형 게임의 현장감 증진방안 연구, 광운대 2006 석사논문

[4] 박지영, 김기찬, 박정우, 이준호, "컴퓨터 비전 기술 기반 체감형 복싱게임", *성균관대 논문집 (과학기술편)*, 제53권2호, pp.35-51, 2002

[5] 김혜린, 장혜정, 박승호, "체감형 게임 중신의 텐저블 인터페이스 디자인 연구", 2004 HCI 학술대회 발표 논문집 2권 413-419, 2004

[6] 최삼하, 김경식, 윤성준, "온라인 게임에서의 체감 시스템 적용사례분석", *한국게임학회논문지*, 제4권 2호 pp1-8, 2004

[7] 이영재, "컴퓨터 게임을 위한 충돌 알고리즘", *한국게임학회논문지*, 1권1호, pp42-48, 2001

[8] www.gamasutra.com/features/20000330/bobic_0-1.htm

[9] M.Moore and J. Williams, "Collision Detection and Response for Computer Animation", *Computer Graphics*, vol.22.4, pp 289-298, 1988

[10] 김현준, 경종민, "3차원 컴퓨터 애니메이션을 위한 충돌 검색 및 반응 계산", *전자공학회논문지* 제 30권 a편 제 3호, pp130-138, 1993

[11] 김상형, *Windows API 정복*, 가남사, 2004

저자소개

이 영 재 (Young Jae Lee)



1984 충남대 전자교육공학과(학사)
1994 연세대 전자공학과(석사)
2000 경희대 전자공학과(박사)
2002.3.~현재 전주대학교 문화산업대학 멀티미디어전공 부교수

※ 관심분야: 컴퓨터게임, 영상처리, 컴퓨터비전

이 대 호 (Dae Ho Lee)



1998 경희대 전자공학과(학사)
2001 경희대 전자공학과(석사)
2005 경희대 전자공학과(박사)
2005.9.~현재 경희대학교 학부대학 조교수

※ 관심분야: 컴퓨터비전, 신호처리, 컴퓨터게임, ITS