

# 교육용 증강 현실을 위한 버블 포핑 시스템 구현

이 우 근, 정 다 운, 최 창 열, 김 만 배\*  
강원대학교 IT대학 컴퓨터정보통신공학과  
{wookeun, dujeong, cychoi, manbae\*}@kangwon.ac.kr

## Implementation of a Bubble Popping System for Educational Augmented Reality

Wookeun Lee, Daun Jung, Changyeol Choi and Manbae Kim  
Dept. of Computer and Communications Engineering, IT College, Kangwon National University

### 요 약

가상 현실과 현실 세계가 공존하는 증강 현실은 현재 및 향후 실감미디어 콘텐츠에 중요한 역할을 할 것으로 기대된다. 증강 현실은 게임, 엔터테인먼트, 교육 분야 등에서 응용에 따라 다양한 기술들의 결합이 요구된다. 본 논문에서는 교육적 효과를 주는 버블 포핑 증강 현실 기법을 제안한다. 버블 포핑 시스템은 비디오 입력용 웹캠, 마커 패턴 영상, 그래픽 구 버블 객체, 그래픽 마우스 등으로 구성된다. 일반 마우스의 이동에 따라 그래픽 마우스가 3차원 공간상에서 이동하면서 버블과의 충돌이 발생하게 되면, 충돌시 발생하는 버블의 이벤트로 동적 움직임, 터짐 등의 버블 효과를 구현할 수 있는 방법을 제안한다. 충돌 검출은 카메라와 마커 패턴의 좌표계 및 버블의 3차원 공간위치, 그리고 3차원 모델 데이터를 이용하여 구현한다. 제안 시스템은 ARToolkit을 기반으로 제작되었으며, 시각적인 마커 기반 시스템으로 버블의 움직임과 사용자의 인터랙션을 통해, 버블 포핑의 증강현실을 전달한다. 향후, 현재 개발 중인 포스 피드백 기능이 있는 진동촉각공간 햅틱 마우스와 접목하여 보다 실감적인 증강현실 기술을 개발할 예정이다.

### 1. 서론

실감영상은 3D 및 HDTV 디스플레이의 지속적인 발전과 공급으로 최근 이를 활성화 할 수 있기 위해 실감영상의 제작의 중요성이 대두되고 있다. 실감영상은 2대의 입체카메라를 이용하여 입체영상을 획득할 수 있고, 특수효과 영화에서 자주 사용되는 그래픽 기술을 이용하여 실사와 그래픽의 합성으로 표현되기도 한다. 최근 멀티미디어 기술과 가상/증강현실 기술의 발전으로 사용자에게 몰입감 있는 콘텐츠를 제공하는 여러 시스템들이 제안되고 있다 [1-8]. 증강현실 분야에서 다양한 입력기기를 활용하여, 보다 실감있는 콘텐츠를 제작한다 [1-4].

이러한 증강현실 콘텐츠는 응용분야에 따라서 기술적 접근을 달리하며, 이에 맞추어 전체 시스템이 개발된다. 증강현실은 엔터테인먼트, 게임, 교육 분야의 다양한 분야에서 구현되고 있는데, 본 논문에서는 교육적 효과를 줄수 있는 버블 포핑 (bubble popping) 증강현실을 제안한다. 버블 포핑 시스템은 비디오 입력용 웹캠, 마커 패턴 영상, 그래픽 구 버블 객체, 그래픽 마우스 등으로 구성된다. 일반 마우스의 이동에 따라 그래픽 마우스가 3차원 공간상에서 이동하면서 버블과의 충돌이 발생하게 되면, 충돌시 발생하는 버블의 이벤트로 동적 움직임, 터짐 등의 버블 효과를 구현할 수 있는 방법을 제안한다. 충돌 검출은 카메라와 마커 패턴의 좌표계 및 버블의 3차원 공간위치, 그리고 3차원 모델 데이터를 이용하여 구현한다. 제안 시스템은 ARToolkit을 기반으로 제작되었으며 [9], 시각적인 마커 기반 시스템으로 버블의 움직임과 사용자의 인터랙션을 통해, 버블 포핑의 증강현실을 전달한다.

본 논문의 구성은 다음과 같다. 2절에서는 ARToolkit의 기본 구

조를 설명하고, 전체 시스템 구조는 3절에서 소개된다. 4절에서는 시스템의 각 모듈을 자세히 설명하고, 실험결과는 5절에서 정리한다. 마지막으로 결론 및 향후 연구는 6절에서 정리한다.

### 2. ARToolkit 개요

ARToolKit은 프로그래머들이 손쉽게 증강현실 어플리케이션을 개발할 수 있도록 도와주는 C, C++ 라이브러리이다. 증강현실 어플리케이션을 개발할 때 가장 어려운 부분 중 하나는 가상의 이미지가 사용자의 시각에서는 실제 세상의 객체와 정확하게 맞아떨어지도록 실시간으로 정밀하게 계산되어지게 하는 것이다. ARToolKit은 컴퓨터를 이용한 비전 기술로 실제 카메라 위치와 마커와 연관된 위치를 계산해낸다. 이 마커위에 프로그래머가 가상의 이미지를 형성하게 된다. ARToolKit 에 의해 빠르고 정밀한 트래킹이 제공된다. ARToolKit은 SGI IRIX, Linux, Max OS, Windows등 여러 가지 플랫폼에서 동작이 가능하다.

- 실시간 증강 현실 어플리케이션을 만들기 위한 간단한 프레임 워크
- 멀티 플랫폼 라이브러리(Windows, Linuc, Max OS, SGI)
- 마커에 3D object 오버레이
- 여러 가지 입력 장치 지원 (USB, Firewire, capture card)
- 여러 형식 지원 (RGB/YUV420P, YUV)
- 여러개의 카메라 트래킹 지원
- OpenGL 기반의 빠른 Rendering
- 3D VRML 지원
- 간단한 API모듈

- 다른 언어 지원(JAVA, Matlab)
- 상업적인 용도가 아닌 OpenSource

### 3. 버블 포핑 시스템

그림 1은 본 논문에서 제안하는 전체시스템의 블록도를 보여준다. 기본적으로 ARToolKit 소프트웨어를 기반으로 제작된다. 먼저 USB CAM으로부터 비디오 프레임을 입력 받고 마커패턴의 위치를 설정한다. 버블을 그래픽으로 구현하기 위하여 구(sphere)를 생성하였으며 여러 개의 버블을 제작하였다. 버블 포핑 효과를 얻기 위하여 일반적인 마우스를 이용하였다. 3차원 공간상에서 마우스와 버블과의 충돌이 발생하면 컬러 변경, 버블의 움직임 등의 버블 이벤트를 구현하였다.

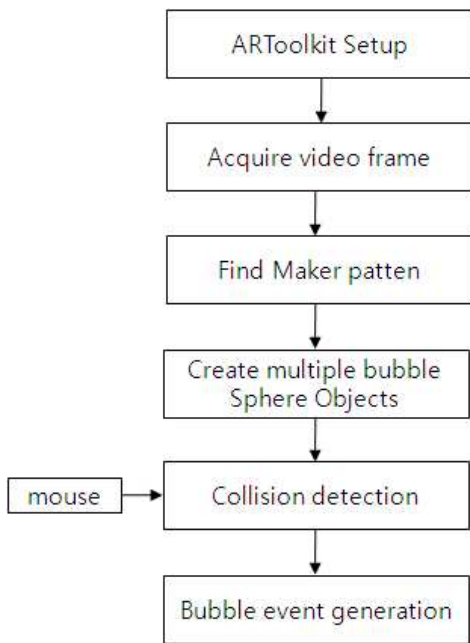


그림 1. 제안시스템의 전체 블록도

### 4. 구현 시스템의 구조

기본적으로 틀에서는 USB Cam과 마커 패턴을 주어진다면, 마커 패턴의 중심을 좌표계의 원점으로 정하며, OpenGL로 렌더링된 객체는 이 원점을 중심으로 3차원 공간에서 보여지게 된다. 따라서 버블 버핑의 주요 기능은 버블 구 객체 생성 및 위치 추적, 가상 마우스 설정, 및 3차원 공간상에서의 마우스 이동, 및 충돌 검출이다. 또한 충돌에 따른 이벤트로 컬러 변환 및 움직임 등을 이용하였다.

#### 4.1 Bubble event generation 알고리즘

구 객체들의 색상을 결정한다. 객체들은 색상코드(color code)을 가지고 있고 마우스와 Hit 할 때마다 색상코드 값이 증가하여 각 객체들은 각기 다른 색을 가지게 된다. 색상코드가 4이상의 값을 가지게 되면 4번의 Hit을 한 것이 되고 이 경우 객체를 그리지 않는다. 그림 2는 이것을 구현한 소스코드이다.

```

for(i=0;i<MAX;i++){
  if (color[i] == 0 && Target_R[i]<25)
  {
    mat_ambient[i][0] = 0.0f;
    mat_ambient[i][1] = 0.0f;
    mat_ambient[i][2] = 1.0f;
    makeball(i);
  }
  else if (color[i] ==0)
  {
    mat_ambient[i][0] = 0.0f;
    mat_ambient[i][1] = 0.0f;
    mat_ambient[i][2] = 1.0f;
  }
  else if (color[i] == 1)
  {
    mat_ambient[i][0] = 0.0f;
    if(mat_ambient[i][1]<=1.0f)
    {
      mat_ambient[i][1] =
      mat_ambient[i][1]+0.1f;
      mat_ambient[i][2] =
      mat_ambient[i][2]-0.1f;
    }
  }
  else if (color[i] == 2)
  {
    if(mat_ambient[i][0]<=1.0)
    {
      mat_ambient[i][0] += 0.1;
    }
    mat_ambient[i][1] = 1.0;
    mat_ambient[i][2] = 0.0;
  }
  else if (color[i] == 3)
  {
    mat_ambient[i][0] = 1.0;
    if(mat_ambient[i][1]>0.0)
      mat_ambient[i][1] -= 0.1;
    mat_ambient[i][2] = 0.0;
  }
  else if (color[i] >=4 && Target_R[i] >=0)
  {
    byeball(i);
  }
  if(Target_R[i]>0)
  {
    glMaterialfv(GL_FRONT,
      GL_AMBIENT, mat_ambient[i]);
    glMatrixMode(GL_MODELVIEW);
    glLoadMatrixd( gl_para );
    glTranslatef( Target_x[i], Target_y[i], Target_z[i]);
    glutSolidSphere(Target_R[i],25,25);
    srand((unsigned)time(NULL));
  }
  glLoadIdentity();
}

```

그림 2. Hit 발생 시 객체의 색상변환 코드

#### 4.2 마우스로 3차원 공간좌표 생성

마우스의 3차원 공간좌표를 구하기 위하여 다음과 같은 방법을 사용하였다. 초기 위치를  $(x_0, y_0, z_0)$  라고 하면 왼쪽 버튼 클릭후 드래그 할 경우 공간상에서  $(\Delta x, \Delta y)$  만큼 이동하여 다음 위치인  $(x_1, y_1, z_1) = (x_0+\Delta x, y_0+\Delta y, z_0)$  좌표를 구한다. 오른쪽 버튼 클릭 후 드래그 할 경우 공간상에서  $\Delta z$ 만큼 이동하여 다음 위치인  $z_1$  좌표를 구한다. 실험에서는  $(\Delta x, \Delta y, \Delta z)$ 를 4로 하였다. 이렇게 얻어진 좌표를 가지고 3차원

공간상에서 마우스 객체를 컨트롤 한다.

### 4.3 collision detection 알고리즘

3차원공간에서 두 sphere 간의 collision detection 알고리즘은 sphere의 중심 점간의 거리를 통하여 구할 수 있다. sphere의 경우 어느 방향에서 충돌하여도 거리가 반지름 R이기 때문이다. 이를 정리하면 다음과 같다. 타겟의 center가  $(x_0, y_0, z_0)$ 이고 반지름 R이 30 인 마우스 sphere와 중점이  $(x_1, y_1, z_1)$  이고 R이 20 인 sphere의 경우 두 중점사이의 거리 F는 다음 식 (1)과 같이 계산된다.

$$F = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (1)$$

따라서, 두 구간의 충돌은 다음 조건을 만족하면 TRUE이다.

$$\text{if}(F \leq R_0 + R_1) \\ \text{HIT!}$$

위의 두 sphere 의 경우  $R_0 + R_1 = 50$  임으로 두 점사이의 거리가 50 이하일 때 collision 되었음을 알 수 있다. 프로그램은 아래와 같이 구현하였다.

```
static void CollisionDetection()
{
    GLfloat Tx,Ty,Tz,F;
    int i;
    for (i=0;i<MAX;i++){
        Tx = Target_x[i] - Mouse_x;
        Ty = Target_y[i] - Mouse_y;
        Tz = Target_z[i] - Mouse_z;
        F = sqrt((Tx*Tx)+(Ty*Ty)+(Tz*Tz));
        if (F < Target_R[i]+Mouse_R && HitState[i] ==0){
            HitState[i] =1;
            color[i]++;
        }
        else
            if(F > Target_R[i]+Mouse_R && HitState[i] ==1)
                HitState[i] =0;
    }
}
```

그림 3. 충돌검출 알고리즘의 구현

## 5. 실험 결과

실험에서는 Lenovo USB Cam을 사용하였다. 입력 비디오는 30 fps로 입력되고, 영상 해상도는 640x480로 설정하였다. 카메라 영상 포맷은 YUV이다.

그림 4는 버블의 그래픽 객체가 마커패턴위에 렌더링 된 결과를 보여주고 있다. 초기 색은 파랑색이다. 또한 마우스 버블객체는 흰색으로 보여준다. 파랑색 버블의 초기위치  $(x, y, z)$ 는  $(x, y) = [-200, 200], z = [0, 200]$ 의 범위 내에서 uniform 랜덤 값을 생성하였다. 각 버블의 반경 R은  $[15, 60]$  범위 내에서 uniform 랜덤 값을 생성하였다. 그림 4의 5개의 버블 구의 생성된 좌표  $(x, y, z)$ 는 각각  $(-144, -8, 80), (178, -124, 178), (82, 113, 10), (-103, 70, 59), (148, -149, 78)$ 이다. 또한 R 값은 46, 32, 23, 55, 41이다. 구 객체들은 마커의 중앙을  $(0, 0, 0)$ 으로

놓고 이를 기준으로 그려진다. 마커는 ARToolKit에서 제공하는 Hiro 패턴을 사용하였다.

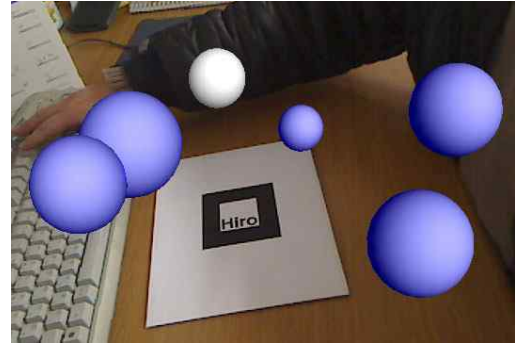


그림 4. 마우스 버블 객체와 구 버블객체

그림 5는 충돌 시 발생하는 객체의 색상의 변화를 보여준다. 그림 5의 좌측그림에서 흰색의 마우스 객체와 구 객체의 충돌 시 색상의 변화를 볼 수 있다. 이 때 구 객체의 color 값이 1증가하며 녹색색상으로 변화한다. 그림5의 우측그림에서는 충돌 횟수에 따라 다르게 변화하는 객체의 색상을 볼 수 있다. color에 따른 색상은 다음과 같다. color=1 일 때 RGB=(0,255,0), color=2 일 때 RGB=(255,255,0), color=3 일 때 RGB=(255,0,0)으로 설정 된다. 각 객체는 color 값을 가지고 있어 충돌 시 마다 값이 변화하게 된다. 1회 충돌 시 color 값은 1이 되고 녹색으로 객체가 칠해진다. 2회 충돌 시 color 값은 2가 되고 노란색으로 객체가 칠해진다. 3회 충돌 시 객체는 적색으로 칠해지고 color 값은 3이 된다. 충돌 시 발생하는 색상 변화 이벤트는 충돌 검출의 성공 여부를 알기 위하여 수행하였다.

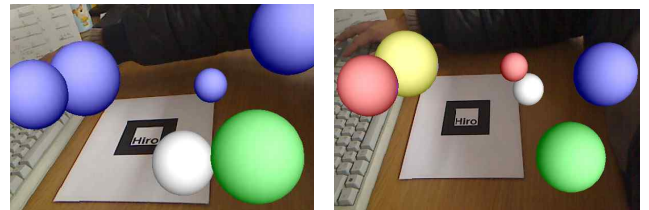


그림 5. 충돌 시 색상의 변하는 이벤트

그림 6는 여러 횟수 충돌 시 구 객체가 사라진 결과를 보여주고 있다. 그림 6에서 보면 5개의 구 객체 중 3개의 구 객체만 남은 것을 볼 수 있다. 구 객체와 마우스 객체가 4회 이상 충돌 하게 되면 구 객체는 사라진다. 이때 색상변화 이벤트에서 사용하는 color값을 가지고 충돌 횟수를 구하여 사용 하는데  $color \geq 4$  일 때는 객체를 그리지 않는 방법을 사용하여 구 객체를 사라지게 한다.

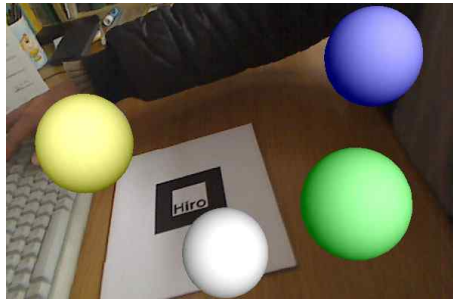


그림 6. 여러 회 충돌 시 사라지는 이벤트

## 6. 결론

증강현실 콘텐츠는 응용분야에 따라서 기술적 접근을 달리하며, 이에 맞추어 전체 시스템이 개발된다. 본 논문에서는 새로운 응용으로 교육적 효과를 가지는 버블 포핑 시스템을 구현하였고, 실시간 처리가 가능하도록 제작되었다.

추후 연구는 입력마우스의 그래픽 객체를 좀더 유사하게 제작하는 것인데, 이는 충돌 검출에서 우수한 알고리즘의 연구가 필요하다. 또한 진동촉각공간마우스와 결합하여, 보다 능동적이고, 실감적인 버블 포핑 시스템을 개발할 예정이다.

## 감사의 글

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업과 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2009-(C1090-0902-0017))

## 참고 문헌

- [1] K. Moustakas, et. al, "A geometry education haptic VR application based on a new virtual hand representation", Proc. of IEEE Virtual Reality, 2005.
- [2] L. Paolis and M. Pulimeno, "A simulation of a billiard game based on a marker detection", IEEE 2009.
- [3] F. Liarokapis, M. White, P. Lister, "Augmented reality interface toolkit", Proc. of the Eighth International Conference on Information Visualization, 2004.
- [4] C. Kirner, et. al, "Case studies on the development of games using augmented reality", IEEE International Conference on Systems, Man, and Cybernetics, Oct. 8-11, 2006.
- [5] 이준훈, 하태진, 류제하, 우운택, "디지로그 북 저작을 위한 펜형 햅틱 사용자인터페이스의 개발", HCI 학술대회, 2009.
- [6] 박선영, 이준훈, 김현곤, 김영미, 최권영, 류제하, "실감책을 위한 시스템 및 저작 도구 기본 프레임워크", HCI 학술대회, 2009
- [7] 증강현실을 이용한 실감형 e-Learning System, 기술전시, HCI 학술대회, 2009.
- [8] T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, S. Lieberknecht, "An Augmented Reality Museum Guide",