

# 물리 기반 모델링을 이용한 증강현실에서의 효과적 객체 상호작용

## Physically-Based Objects Interaction in Augmented Reality Environments

이민경<sup>1</sup>, Minkyong Lee, 김영준<sup>1</sup>, Young J. Kim, 스테판 르동<sup>2</sup>, Stephane Redon,  
1:이화여자대학교 컴퓨터학과, 2:INRIA - Rocquencourt, France

**요약** 본 논문에서는 연속적 충돌검사 방법과 제약 조건 기반의 강체 역학 모델링 기법을 이용하여 마커 기반의 트래킹 환경에서 현실의 객체와 가상의 객체가 물리적으로 현실적이고 안정적으로 상호작용하는 증강현실 방법을 제안한다. 본 논문에서 구현된 증강 현실 시스템은 증강 현실환경상의 현실 객체를 인식하고 트래킹 하는 부분과 증강현실에 등장하는 모든 종류의 객체들 간의 물리적인 상호작용을 시뮬레이션 하는 부분으로 크게 구성된다. 객체 트래킹에 사용되는 일반적인 카메라로는 적은 수의 불연속적인 프레임 밖에 얻을 수 없는 성능의 근본적인 한계에도 불구하고, 본 논문에서는 연속적 충돌검사 방법을 이용하여 객체간의 올바른 충돌 정보를 얻을 수 있었고, 이를 이용하여 제약 조건 기반의 강체 역학 시뮬레이션을 적용하여 안정적이고 현실적인 물리 반응을 생성할 수 있었다. 제안한 방법론은 이러한 트래킹 지연에도 불구하고 본 논문에서 사용된 다양한 벤치마킹 시나리오에서, 안정적으로 현실의 객체와 가상의 객체 사이에 물리적으로 실감나는 인터랙션 결과를 보여주었다.

**핵심어:** 증강현실, 마커 기반 트래킹, 제약 조건 기반의 강체 역학, 가우스 최소 구속 원리, 연속적 충돌검사

### 1. 서론

증강현실이란 사용자가 눈으로 보는 현실세계와 부가정보를 갖는 가상세계를 합쳐 하나의 영상으로 보여주는 가상현실 기법의 하나이다. 증강현실은 원격의료, 방송, 건축설계, 제조공정관리와 같은 다양한 응용분야를 가지고 있고, 이에 대한 연구가 활발히 진행되어 왔다. 하지만 기존의 증강현실 연구는 주로 시각화 및 렌더링에 집중되어, 증강현실상에 등장하는 가상세계의 객체가 현실세계의 객체와 물리적으로 현실감 있는 인터랙션을 하도록 하는 연구는 아직 많이 되어 있지 않고 있다. 이러한 인터랙션이 포함된 증강현실은 트래킹, 태스크 플래닝, 시뮬레이션 등의 분야에 큰 도움이 될 수 있다. 예를 들어, 비용이 많이 들거나 혹은 위험한 과정을 포함하는 훈련, 배치, 조립 작업 등의 애플리케이션에서는 증강현실상의 실감나는 인터랙션을 수행함으로써 실제 수행 결과와 비슷한 효과를 적은 비용으로 안전하게 거둘 수 있다 [9].

이렇게, 현실환경과 가상환경의 물리적인 상호작용이 가능한 증강현실환경을 생성하기 위해서는 현실환경에 등장하는 객체들의 카메라 트래킹과 역학 시뮬레이션이 효과적으로 결합되어야 한다. 하지만 본 논문에서와 같이 현실 환경의 객체를 카메라로 트래킹 할 때는 카메라 장비의 한계로 인하여 불연속적이고 적은 수의 프레임 정보밖에 얻을 수 없는 트래

킹 지연(tracking latency) 문제가 생기고 이는 안정적인 물리 기반 상호작용을 구현하는데 큰 문제가 될 수 있다. 왜냐하면, 올바른 역학 반응을 생성하려면 객체간의 충돌 순간(time of impact)을 효과적으로 알아내야 하는데 트래킹 지연 때문에 객체들의 충돌 순간을 정확히 알아 내지 못하기 때문이다. 예를 들어 충돌 순간이 아닌 객체들간에 이미 침투되어 버린 결과를 얻거나, 프레임 중간에 실제로 충돌이 일어났어도 충돌을 검출하지 못하는 경우가 생긴다. 이러한 올바른지 못한 충돌 정보로는 효과적인 역학 상호작용을 생성할 수 없게 된다.

본 논문에서는 이런 트래킹 지연 문제를 극복하고 강체 역학 시뮬레이션을 올바르게 효과적으로 수행하기 위한 해법으로 연속적 충돌검사(continuous collision detection, CCD) 방법과 제약 조건 기반(constraint-based)의 물리 모델링 기법을 이용하여, 마커 기반의 트래킹 환경에서 현실의 객체와 가상의 객체가 물리적으로 현실감 있게 그리고 안정적으로 인터랙션 하는 증강현실 기법을 제안한다. 제안하는 방법은 크게 현실환경의 객체를 인식하고 트래킹하는 부분과 증강현실에 참여하는 모든 객체들 간의 물리적인 인터랙션을 시뮬레이션 하는 부분으로 구성된다. 또한 본 논문에서 다루는 객체는 모두 강체라고 가정한다.

일반적으로 컴퓨터 그래픽스나 기계공학에서 강체 역학을 시뮬레이션하는 방법으로는 크게 페널티 기반(penalty-

based) 방법, 충격력 기반(impulse-based)의 방법, 제약 조건 기반(constraint-based)의 방법이 있으며, 이 중에서 제약 조건에 기반한 방법은 안정적이고 효과적인 결과를 생성한다고 알려져 있다. 이러한 제약 조건 방법 중에서 본 논문에서는 제약 힘을 시간과 메모리 면에서 효율적으로 구할 수 있는 가우스 최소 구속 원리(Gauss's least constraint principle)를 이용하였다. 또한 이러한 제약 조건 기반의 강제 역학 시뮬레이션에서 가장 중요한 제약 조건은 객체들간의 비침투조건(non-penetration constraint)을 유지하는 것이다. 이러한 비침투조건을 유지하기 위하여 본 논문에서는 연속적 충돌검사 방법(CCD)을 사용하였고, 이는 모션 보간과 Interval Arithmetic(IA)을 이용하여 구현되었다.

## 2. 관련 연구

### 2.1 증강현실에서의 상호작용 연구

지금까지의 증강현실의 연구들은 주로 현실의 영상 이미지로 부터 3차원의 환경과 객체를 인식하고 재구성하여 시각화하고, 재구축한 씬에 부가정보를 오버랩하여 렌더링 하는 기법들이 많이 연구되어왔다. 최근에는 인식된 실제 환경과 가상의 객체가 인터랙션 하도록 하는 기법이 부분적으로나마 연구되어 왔다.

Aliaga[2]의 연구에서는 optical see-through HMD를 통해 실제 환경에 가상의 객체에 대한 그래픽 정보를 오버랩하여 증강현실을 볼 수 있도록 하였다. 시스템은 미리 등록되어 있는 실제 환경의 객체와 hand-held tracker로 속도를 컨트롤 가능한 가상 객체 사이의 충돌을 검출 한 후 가상의 객체가 물리법칙에 따른 적당한 충돌 반응을 보이도록 하여 현실감을 높였다. Lok의 연구에서는 여러 시점에서 얻은 이미지로부터 실시간으로 실제 객체를 visual hull로 재구성[13]하고, 어느 점이 visual hull 내부에 존재하는지에 대한 검사를 하는 접근방법인 volume query mechanism을 이용하여 가상 객체와 아바타 사이의 충돌 검사를 수행하고 충돌 반응을 생성하였다. 또한 이를 이용하여 사용자가 증강현실 환경에서 간단한 조립 테스트를 수행하는 시나리오를 구현하였다[9,10,11]. Leotta[12]는 몇 개의 시점에서 얻은 현실환경 이미지의 피쳐 포인트들로부터 깊이 정보를 얻어 자동으로 현실환경을 3차원 모델로 재구성하는 시스템을 구현하였다. 생성한 씬에는 가상의 객체를 추가하여 실시간으로 실제 객체와 가상 객체간의 물리적인 인터랙션을 시뮬레이션 할 수 있다. 이 방법은 실제 객체에 대한 충분한 피쳐 포인트를 얻기 위하여 현실 환경에 충분한 텍스처가 입혀져 있어야 한다는 제약이 있다.

하지만, 위의 모든 연구들은 카메라 트래킹 지연에 따른 물리 기반의 시뮬레이션의 문제점들을 제대로 해결하지 못하였고, 사용된 충돌검사 방법 및 물리 반응 생성 시뮬레이션들은 상대적으로 단순한 모델들에만 적용 가능한 방법론이었다.

### 2.2 강제 역학 시뮬레이션

강제 역학을 시뮬레이션하는 방법으로는 크게 페널티 기

반(penalty-based) 방법, 충격력 기반(impulse-based)의 방법, 제약 조건 기반(constraint-based)의 방법이 있다. 페널티 기반의 방법은 객체간의 침투를 허용하되 침투한 깊이의 정도에 따라 복원력을 가해 침투한 객체를 떨어뜨리도록 하는 방법이다[14,15,19]. 이에 반해 충격력 기반의 방법과 제약 조건 기반의 방법은 침투를 허용하지 않기 위하여 객체가 충돌하는 순간과 충돌 위치를 구한 후, 실제로 물체가 그와 같이 충돌하였을 때에 마땅히 일어나야 할 작용을 역학 법칙에 의해 계산하여 시스템에 적용한다. Mirtich[1]은 micro-collision을 정의 하여 접촉이나 충돌의 종류를 구분하지 않고 충돌순간의 객체에 impulse만을 가해 비침투 조건을 만족시키는 역학 시뮬레이션 방법을 제안하였다. Baraff[3,4,5]가 제안한 제약 조건 기반의 방법은 객체가 충돌하였을 때 비침투 제약 조건을 선형 상보성 문제(LCP)의 형태로 다루어, 이 제약 조건을 유지하기 위한 정확한 접촉력과 충격력을 계산해 객체에 가해 준다. Redon은 연속적 충돌검사와 가우스 최소 구속 원리를 이용해 제약 조건을 만족시키는 힘을 계산하고 역학 반응을 시뮬레이션 하였다[16,17]

## 3. 제약 조건 기반의 강제 역학

본 논문에서 사용된 제약 조건 기반의 강제 역학 시뮬레이션의 요점은 비침투 제약 조건을 유지하는 것이다. 이를 위해 객체들이 충돌하는 순간과 위치를 검출 후, 침투가 일어나지 않도록 제약하는 힘을 계산하여 가해준다. 본 논문에서는 제약 조건 기반의 역학을 시간과 메모리 면에서 효과적으로 계산하기 위하여 가우스의 최소 구속 원리에 의한 기법[16]과 연속적 충돌 검사기법[17]을 이용하였다.

### 3.1 연속적 충돌검사

일반적으로 사용되는 충돌 검사 방법에는 크게 두 가지 방법이 있는데 그 중 단속적인 방법(discrete method)은 객체의 모션을 일정한 시간 간격마다 추출하여 객체들의 충돌 여부를 검사하는 것이고, 연속적인 방법(continuous method)은 주어진 불연속적인 모션 사이를 임의의 모션으로 보간하여 연속적인 모션을 만들어 그것과 환경과의 충돌을 검사하는 것이다. 본 연구에서는 트래킹 지연에 의한 한정된 수의 프레임 정보로부터 올바른 객체간의 충돌 정보를 얻기 위하여, 또한 정확하고 안정적인 객체간의 물리 반응을 생성할 수 있는 제약 조건 기반의 역학 시뮬레이션을 구현하기 위하여 후자의 기법중 하나인 연속적 충돌 검사 방법(CCD)을 이용한다.

#### 3.1.1 모션 보간법

불연속적인 시간 간격에서, 주어진 위치 데이터만을 가지고 충돌 검사를 하는 방법은 처음 충돌하는 순간을 정확히 알아야 하는 물리 시뮬레이션에는 적합하지 못하다. 또한, 물체의 두께가 얇거나 물체가 빠른 속도로 충돌을 발생한다면 충돌이 일어났음에도 검출하지 못하는 경우도 생긴다. 연속적인 충돌검사 방법에서는 충돌을 놓치지 않고 정확한 처음 충돌 순간을 검출하기 위하여 불연속적인 시간 간격에서

주어진 물체의 위치를 임의의 움직임으로 보간하여 연속적인 이동경로를 만들어 이에 대한 충돌을 검사한다. 이를 위해 선속도와 각속도가 선형이라는 가정과 함께 Screwing 모션을 이용한 Arbitrary in-between motion을 이용한다. 시간 구간  $t \in [0,1]$ 에 대한 Screwing에 기반한 in-between motion은 다음과 같은  $4 \times 4$ 의 homogeneous matrix로 얻을 수 있다.

$$S(t) = P^{-1}V(t)P \quad t \in [0,1] \quad (1)$$

$$V(t) = \begin{pmatrix} \cos(\omega t) & -\sin(\omega t) & 0 & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 & 0 \\ 0 & 0 & 1 & st \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$\mathcal{U}(t)$ 는 z-axis screwing이고  $P$ 는 reference frame에서 screwing local frame으로의 변환이다.  $w$ 와  $s$ 는 시간 구간 동안의 회전과 이동거리를 나타낸다.

### 3.1.2 OBB 기반의 연속적 충돌검사

바운딩 볼륨 계층구조(BVH)는 충돌 검사 연구에서 일반적으로 쓰이는 가속화 방법으로 본 논문에서 사용한 바운딩 볼륨(BV)은 OBB (oriented bounding box)이다[18]. BVH기반의 충돌검사 방법에서는, 두 BV의 충돌을 검사 하여 오버랩이 없을 때는 실제 물체 사이의 충돌이 있을 수 없으므로 불필요한 연산을 제거 할 수 있다. 또한, BV간의 충돌이 있을 경우에는 충돌한 BV를 더 작은 BV의 조합으로 나누어 오버랩을 재귀적으로 반복하여 충돌이 일어난 지점을 근사해 내고, 이 지점에서 정확한 삼각형 수준의 충돌 영역을 검출해낸다.

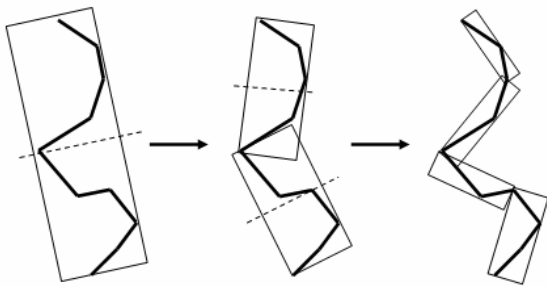


그림 1. OBB를 이용한 바운딩 볼륨 계층구조[18]

OBB의 오버랩을 테스트하기 위해서 분리축 법칙 (separating axis theorem)[18]을 사용한다. 첫 번째 OBB에서의 중심을  $T_A$ , 세 축을  $e_1, e_2, e_3$ , 크기를  $\mathbf{a}$ 라 하자. 두 번째 OBB는 그 중심이  $T_B$ , 세 축이  $f_1, f_2, f_3$ , 크기는  $\mathbf{b}$ 라 하자. 이때, 식 (4)의 15개의 잠재적인 분리축에 대해 부등식 (3)의 테스트를 모두 통과하면 두 OBB는 오버랩 하지 않는다.

$$|\mathbf{a} \cdot \mathbf{T}_A \mathbf{T}_B| > \sum_{i=1}^3 a_i |\mathbf{a} \cdot \mathbf{e}_i| + \sum_{i=1}^3 b_i |\mathbf{a} \cdot \mathbf{f}_i| \quad (3)$$

$$\{\mathbf{e}_i, \mathbf{f}_j, \mathbf{e}_i \wedge \mathbf{f}_j, 1 \leq i \leq 3, 1 \leq j \leq 3\} \quad (4)$$

기본적인 OBB기반의 불연속적인 충돌검사 방법을, 물체간의 충돌 시간을 구하고 OBB들 사이의 연속적인 오버랩을 계산하는 방법으로 전환하기 위하여 Interval Arithmetic(IA)[8]을 사용한다. IA는 식(5)과 같이 정의된 닫힌 실수 구간에 대한 연산이며 식(6)과 같이 벡터화된 구간으로 일반화 될 수 있다. 기본적인 연산은 (7)과 같이 정의 된다.

$$I = [a, b] = \{x \in \mathbf{R}, a \leq x \leq b\} \quad (5)$$

$$I_n = [a_1, b_1] \times \dots \times [a_n, b_n] \\ = \{\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{R}^n, a_i \leq x_i \leq b_i, \forall i, 1 \leq i \leq n\} \quad (6)$$

$$[a, b] + [c, d] = [a+c, b+d]$$

$$[a, b] - [c, d] = [a-c, b-d]$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$1 / [a, b] = [1/b, 1/a] \quad \text{if } a > 0 \text{ or } b < 0$$

$$[a, b] / [c, d] = [a, b] (1/[c, d]) \quad \text{if } c > 0 \text{ or } d < 0$$

$$[a, b] \leq [c, d] \quad \text{if } b \leq c \quad (7)$$

위의 IA를 이용하여 연속적인 시간 구간에서 OBB 테스트를 수행한다. 즉, 부등식 (3)의 각 항은 시간 구간에 대한 함수로 간주되고 IA를 부등식의 양쪽 항에 적용하여, 왼쪽 항의 하한값이 오른쪽 항의 상한보다 크다면 시간 구간에서 오버랩하지 않는다고 확신 알 수 있다. IA 연산은 서로 연산되는 구간끼리의 직접적 관계가 없는 경우만을 고려하기 때문에 실제의 값보다 다소 큰 결과를 줄 수 있지만 수행 속도가 빠르다는 장점이 있어서 효과적인 CCD를 구현할 수 있다.

### 3.2 가우스 최소 구속 원리

일반적으로 제약 조건 기반의 강체 역학 시뮬레이션은 가속도와 제약힘을 연관시킨 선형 상보성 문제(LCP)로 다루어진다. 하지만, 이 방법에서는 구속이 없는 계의 자유도가 불분명하다. 반면에 가우스의 최소 구속 원리는 마찰이 없는 계의 역학 문제를 자유도를 명시적으로 기술하는 알고리즘으로 구성하여 불필요한 계산을 줄일 수 있다. 이 원리에 의하면 제약이 있을 때의 일반화된 가속도(generalized acceleration)  $\mathbf{a}_c$ 는 가능한 일반화된 가속도의 집합인  $\mathbf{a}$ 와 일반화된 질량(generalized mass)  $\mathbf{M}$ 에 대한 다음과 같은 스칼라 함수  $G_c(\mathbf{a})$ 를 최소로 만드는 가속도이다.

$$G_{\alpha}(\mathbf{a}) = \frac{1}{2}(\mathbf{a}-\mathbf{a}_u)^T \mathbf{M}(\mathbf{a}-\mathbf{a}_u) = \frac{1}{2} \|\mathbf{a}-\mathbf{a}_u\|_M^2 \quad (8)$$

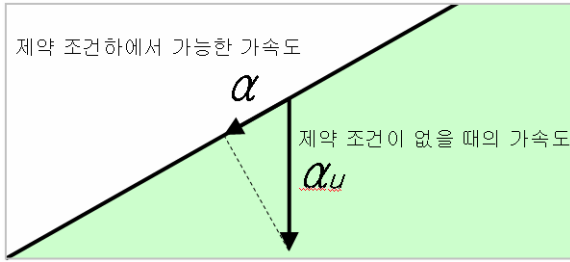


그림 2. 가우스의 최소구속 원리를 이용한 제약 조건하에서의 입자의 가속도 계산. 제약 조건이 없다면 입자의 가속도  $\mathbf{a}_u$ 는 중력가속도  $\mathbf{g}$ 가 될 것이다. 실제 입자의 가속도는 경사면에 의한 비침투 제약 조건에 의해 주어진다. 제약 조건에 의해 입자가 가지는 가속도는 제약 조건하에서 가능한 가속도 중 제약 조건이 없을 때의 가속도에 가장 가까워 질 수 있는 가속도이다

즉,  $\mathbf{a}$ 와  $\mathbf{a}_u$ 의 kinetic distance를 최소로 하는 값, 다시 말해 가능한 모든 제약된 가속도 중에서 제약 조건에 구속되지 않은 가속도에 가장 가까운 가속도를 구한다. 가능한 가속도의 볼록 집합을 정의해 주는 일반적인 제약조건  $\mathbf{J}\mathbf{a} \geq \mathbf{c}$ 에 의해 제약된 가속도는 다음과 같은 식으로 얻어진다.  $\mathbf{J}$ 는 Jacobian 행렬이다[4].

$$\mathbf{a}_c = \operatorname{argmin}\left\{\frac{1}{2} \|\mathbf{a}-\mathbf{a}_u\|_M^2 : \mathbf{J}\mathbf{a} \geq \mathbf{c}\right\} \quad (9)$$

이 식은 임의의 개수의 객체에 대하여 임의의 개수의 제약에 대한 가속도를 효율적으로 구하여 준다.

#### 4. 증강 현실에서의 객체 상호작용

본 논문에서 사용된 증강현실상의 트래킹 기법은, 촬영된 2차원 영상으로부터 실시간으로 공간적인 정보를 파악하고 정확한 사용자의 시점을 알아내는 효과적인 기법인 마커 트래킹 방법을 이용하였다. 프로그램에 미리 등록 되어 있는 그림 3과 같은 마커를 카메라가 관측 가능한 곳에 부착한다. 그리고 한 대 혹은 다수의 카메라에 입력된 영상에서 마커를 검출하여 3차원 공간 내에서의 위치 혹은 운동정보를 추정하였다. 구체적인 시스템 셋업은 그림 4와 같다.

즉, 현실환경의 객체로 이용할 상자와 테이블의 적당한 위치에 마커를 부착하고 상용화된 일반적인 웹캠을 이용하여 이를 트래킹 한다. 트래킹을 통해 매 프레임마다 영상으로부터 얻은 현실환경의 객체 위치 정보는 시뮬레이션 부분에 전달되어 가상의 객체와의 역학 반응이 계산 된다. 이 결과는 비디오 출력 화면이나 HMD에 렌더링 되어 실제의 객체와 가상의 객체가 물리적으로 인터랙션 하는 것을 보여준다. 불연속적이고 적은 수의 프레임 밖에 얻을 수 없는 웹캠 성능의 근본적인 한계로 인한 트래킹 지연에도 불구하고 연속적 충돌검사 방법을 이용하여 올바른 충돌 정보를 얻을 수 있다. 그리고 이를 가지고 제약 조건 기반의 강체 역학 시뮬레이션으로 안정적이고 현실적인 물리 반응을 생성할 수 있었다.



그림 3. ARToolkit에서 트래킹을 위해 사용하는 마커



그림 4. 시스템 셋업. 시스템은 실제 환경을 트래킹하는 카메라와 트래킹을 위한 마커, 그리고 트래킹과 시뮬레이션을 수행하고 결과를 디스플레이 하는 장치로 구성된다. 마커는 증강현실에 참여할 실제 객체의 적당한 위치에 부착된다.

#### 5. 구현 및 결과

본 논문에서 구현된 증강현실 시스템은 Visual C++와 OpenGL 그래픽 라이브러리를 이용한 환경에서 구현되었다. 또한 역학 시뮬레이션을 위하여 연속적 충돌검사와 가우스의 최소구속 원리를 구현한 CONTACT 라이브러리[16,17]와, 트래킹을 위하여 ARToolkit[6]을 이용하였다. 현실환경의 영상입력을 위해서는 최대 해상도 640x480 크기, 30FPS의 성능을 가진 웹캠(Logitech QuickCam Notebooks Pro)을 사용하였다.

객체인식 트래킹, 렌더링 파트에서는 특별한 마커카드를 트래킹하여 실제 세계의 영상에 가상의 이미지를 덮어 씌워 준다. 여기서 사용하는 마커는 그림 3과 같은 모양으로, 검정색 정사각형 모양으로 안에 간단한 기호를 가지고 있으며 정확한 크기와 모양이 프로그램에 미리 입력되어 있다. 구체적인 트래킹 및 렌더링 스텝은 그림 5와 같다. 먼저, 캡처된 영상을 바이너리로 변환 후, 여기서 모든 정사각형을 검출하여 마커로 가정한다. 발견된 각각의 정사각형은 안에 있는 기호에 따라 마커로 확정되고 그 종류가 인식된다. 모양과 크기가 모두 입력되어 있으므로 단 한대의 카메라에 입력된 2차원 이미지에서의 마커의 위치와 모양을 가지고, 3차원 공간에서의 마커 카드에 대한 카메라의 위치와 방향을 계산할 수 있다(a). 이 변환은 행렬로 저장되어 가상 공간에서의 시점을 정하고 가상 공간에서 현실 객체에 대응하는 모델의 위치에 구하는데 이용된다(b). 실제의 카메라 위치와 가상공간의 시점의 위치는 같은 것으로 가정하여, 이 시점을 기준으로 가상의 이미지를 마커 위에 정확히 그릴 수 있다.

가상의 객체는 미리 입력 되어 있는 모델 내에서 시뮬레

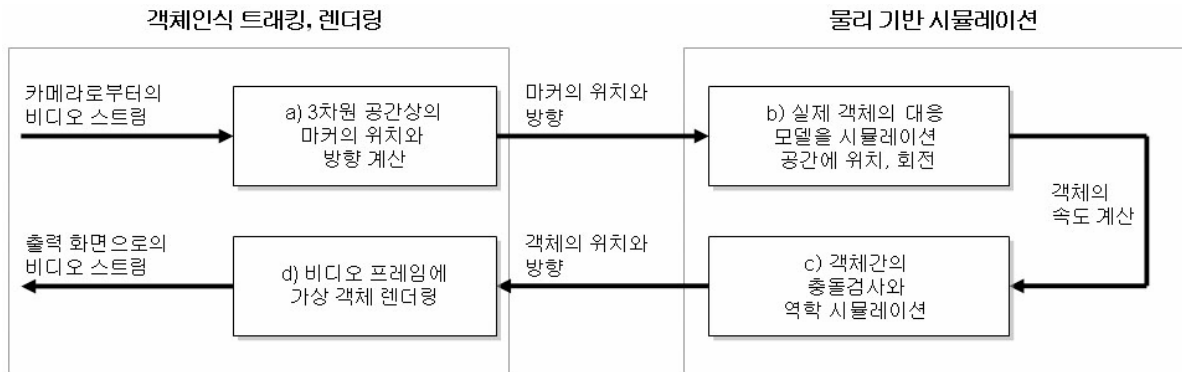


그림 5. 물리기반 인터랙션을 포함한 증강 현실 시스템 다이어그램

이전 중에 사용자 인터페이스로 생성하거나, 시뮬레이션 시작 시에 미리 로드하여 시뮬레이션에 참여시킬 수 있다. 물리기반 시뮬레이션 파트에서는 실제 물체를 위한 객체 모델과 순수한 가상의 물체를 위한 객체 모델이 물리적으로 어떻게 상호작용 해야 할지, 제약 조건에 기반한 강제 역학 시뮬레이션에 의한 물리 반응을 생성하여 매 프레임마다 시뮬레이션 한다. 추가적으로 현실의 객체의 운동이 가상의 객체에 힘을 가하는 dynamics가 있는 경우, 매 프레임간의 변위를 가지고 순간속도를 구하여 시뮬레이션에 반영하여 현실적인 물리 반응이 일어나도록 한다(c). 이 결과는 다시 객체인식 트래킹, 렌더링 파트에 전달되어 실시간으로 비디오 출력 화면이나 HMD에 렌더링되어 사용자에게 보이게 된다(d).

## 5.1 애플리케이션

본 논문에서 제안한 증강현실 기법을 바탕으로 다음과 같은 두 가지 시나리오를 구현하였다.

### 5.1.1 Dynamics가 없는 충돌검출 애플리케이션

첫 번째 애플리케이션은 실제 물체와 가상 물체간의 충돌 정보를 얻는 것이 목적이다. 씬은 두 객체로 구성된다. 하나는 현실에서 사용자가 직접 조작하는 물체로 마커에 의해 트래킹되고, 이 물체는 모바일(mobile) 객체로 간주된다. 다른 하나는 가상의 물체로 스태틱(static) 객체로 간주된다. 그림 6은 기본적인 충돌검출 애플리케이션을 보여준다. 사용자가 조종하는 실제 객체가 가상 객체에 충돌하면 이를 연속적 충돌 검사 방법을 이용하여 검출하고 표시하여 준다. 사용자가 모바일 객체인 스푼을 움직여 가상의 객체인 찻잔 접근 시켜 충돌하게 되면 모바일 객체는 움직임을 멈추고 시뮬레이션은 객체 간의 정확한 충돌 지점을 표시해 준다.

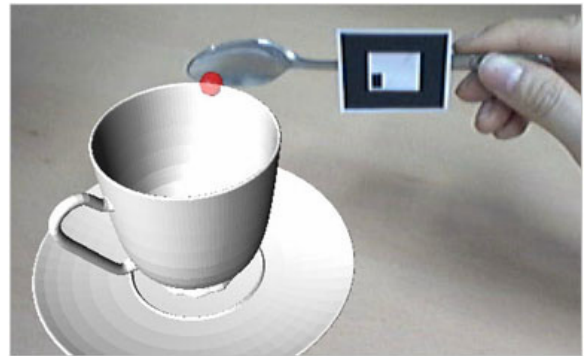


그림 6. 충돌검출 애플리케이션. 사용자가 조종하는 실제 객체가 가상 객체에 충돌하면 이를 검출하고 표시하여 준다. 사용자가 쥐고 움직이는 스푼은 실제 객체이고 찻잔은 가상 객체이다. 스푼에는 트래킹을 위해 마커가 부착되었다.

### 5.1.2 Dynamics에 의한 상호작용을 포함한 애플리케이션

두 번째 애플리케이션은 증강현실에서 역학 시뮬레이션을 수행하여 실제 객체와 가상 객체 간의 물리적인 인터랙션을 생성하는 것이다. 그림 7은 실제 물체인 상자와 가상 객체인 토끼와의 상호작용을 보여준다. 토끼는 역학 법칙에 따라 움직이는 모바일 모델로서 시뮬레이션 시작시에 혹은 시뮬레이션 중간에 유저의 키보드 입력에 따라 임의의 위치와 임의의 속도를 가지고 생성된다. 증강현실에 참여한 가상의 객체는 현실 환경의 테이블 위를 구르거나 테이블 위에 놓인 객체에 부딪혀 비껴가는 등의 현실적인 반응을 보여준다. 반대로 유저가 상자를 조종하여 토끼에게로 움직이면 토끼는 이에 밀려나는 반응을 보인다.



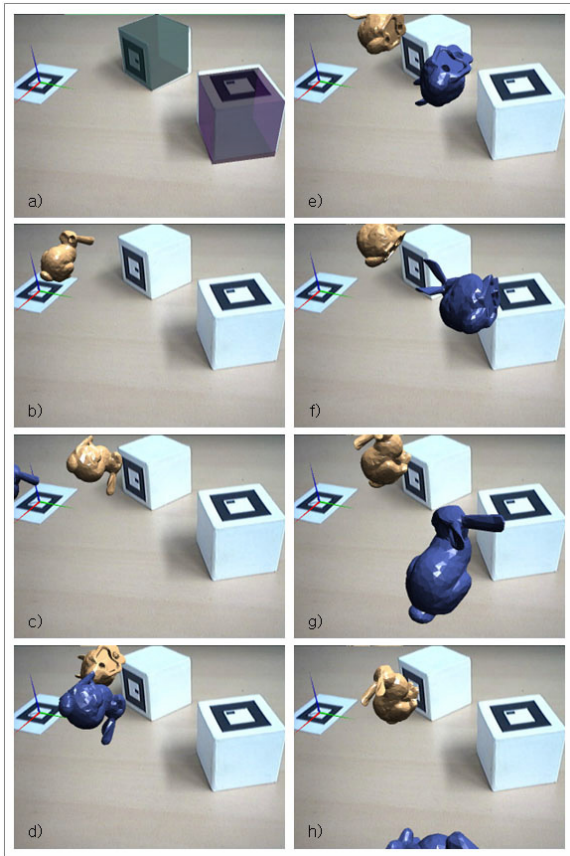


그림 7. 역학 시뮬레이션에 의한 인터랙션을 포함한 어플리케이션. 시뮬레이션 중에 사용자의 키보드 입력에 따라 시뮬레이션 공간의 원점(3차원 좌표축이 디스플레이되는 지점) 부근에서 y축(녹색축) 방향의 속도를 가진 가상 객체인 토끼를 생성하여 증강현실에 참여시켰다. 토끼가 실제 객체인 테이블 위에서 굴러가거나 실제 객체인 박스에 부딪혀 비켜나가는 모습을 a~h에 걸쳐 캡처하였다. 첫 객체를 삽입하기 전의 이미지 a에서는, 시뮬레이션의 디버깅 목적으로, 실제 객체인 상자에 해당하는 모델 위에 물리 시뮬레이션에 사용되는 proxy를 반투명하게 오버랩 하여 렌더링 하였다.

## 6. 결론 및 향후 연구

본 논문은 마커 기반의 트래킹 환경에서 연속적 충돌검사 방법과 제약 조건 기반의 역학 기법으로 트래킹 지연을 보완하여, 증강현실에서 실제 객체와 가상 객체의 현실적이고 안정적인 물리 반응을 생성하는 증강현실 시스템을 제안하였다.

### 6.1 문제점

본 논문에서 구현한 증강현실 시스템에서는 다음과 같은 문제가 있었다. 우선, 비록 역학 시뮬레이션이 안정적인 성능을 보여준다고 하여도, 트래킹의 결과로 얻어지는 마커 위치 정보의 오차로 인하여 매 프레임마다 약간의 차이가 발생하여 시스템의 안정성을 떨어뜨리는 요인이 되기도 하였다. 또한 역학 시뮬레이션에서는 실제의 물체가 가상의 물체와 상호작용 하기 위해서, 매 순간의 위치 정보뿐만 아니라 속도를 알아야 한다. 그러나 마커 트래킹과 컴퓨터의 수치적 예러로 인하여 속도에 오차가 생길 수 밖에 없다. 이러한 오차는 프레임이 지나갈 수록 누적되어 실제 물체의 움직임과 그에 대응 되는 모델의 움직임이 일치하지 않게 된다. 수치적 오차가 생기는 것은 피할 수가 없기 때문에 일정 시간 간격으로 위치 보정을 해줘야 하는데, 이런 인위적인 보정은 상

황에 따라 바로 앞서 설명한 트래킹된 위치의 오차 문제와 같은 상황을 만든다. 즉, 오차 자체 혹은 오차의 누적을 제거하기 위한 보정이 물체들이 오버랩 버리는 잘못된 상황을 만들기도 한다. 이런 문제는 소위 drifting되는 물체를 repositioning하는 문제인데, 이 문제는 객체들간의 global consistency를 필요로 하는 어려운 문제이다.

### 6.2 향후 연구

위에 설명한 트래킹의 오차로 인한 떨림이나 수치적 오차로 인한 예러로 인해 객체들이 침투하는 문제를 보완하기 위하여 본 논문에서 제안한 제약조건 기반 강제 역학 시뮬레이션 방법과 연속적 충돌검사를 이용할 수 있을 것이며 현재 이와 관련된 연구를 진행 중에 있다. 또한 본 논문에서 사용되는 현실 세계의 객체는 미리 등록이 되어야 하는 추가적인 준비 작업이 필요하다. 하지만, 제안한 방법의 주요 응용 목적이 시나리오가 미리 설정되어 있는 가상 훈련이나 가상 실험의 구현이므로 테스트 환경과 모델의 등록은 자연스러운 작업이라 가정 할 수 있다. 그러나 더 일반적이고 확장된 시스템을 구현하기 위하여 임의의 실제 객체를 마커가 없이 증강현실에 참여할 수 있도록 할 계획이다.

### 감사의 글

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업, 한국학술진흥재단의 젊은과학자연구사업(2004-205-D00168), 과학기술부 및 과학재단의 한불 과학 기술 협력기금사업(STAR)의 지원으로 진행되었습니다.

### 참고문헌

- [1] B. Mirtich and J. Canny. Impulse-based Simulation of Rigid Bodies. In Proceedings of Symposium on Interactive 3D Graphics, 1995.
- [2] D. Aliaga. Virtual and Real Object Collisions in a Merged Environment, In Proceedings of VRST'94, 1993.
- [3] D. Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. In Computer Graphics (Proc.SIGGRAPH), volume 23, pages 223-232. ACM, July 1989.
- [4] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. Computer Graphics (Proc. SIGGRAPH), 28:23-34, 1994.
- [5] D. Baraff. Interactive simulation of solid rigid bodies. IEEE Comput. Graph. and Appl. 15, 3, 63-75., 1995.
- [6] B.D. Lucas and T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), April, 1981, pp. 674-679.

- [7] D. Ruspini and O. Khatib. Collision/Contact Models for the Dynamic Simulation of Complex Environments. IEEE/RSJ IROS'97, September 1997.
- [8] J. Snyder. Interval analysis for Computer Graphics. *Computer Graphics*, 26(2), pages 121-130, July 1992.
- [9] Lok, B. Online model reconstruction for interactive virtual environments. In *Proceedings of the 2001 ACM Symposium on Interactive 3D Graphics*. ACM Press, New York, 2001, 69-72.
- [10] Lok, B., Naik, S., Whitton, F., and Brooks, F. Incorporating dynamic real objects into immersive virtual environments. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*. ACM Press, New York, 2003, 31-34.
- [11] Lok, B.C. Towards Merging Real and Virtual Spaces, *Communications of the ACM*, August 2004, 47.8: 48-53.
- [12] Matthew Leotta, Kristin Boyle. Plausible physics in augmented images. *ACM SIGGRAPH 2005 Poster*
- [13] Matusik, W., C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-Based Visual Hulls. In *Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*, July 2000, pp. 369-374.
- [14] Moore, M. and Wilhelms, J., Collision detection and response for computer animation, *Computer Graphics (Proc. SIGGRAPH)*, vol. 22, pp. 289-298, 1988.
- [15] Platt, J.C. and Barr, A.H., Constraint methods for flexible models, *Computer Graphics (Proc. SIGGRAPH)*, vol. 22, pp. 279-288, 1988.
- [16] S. Redon, A. Kheddar, and S. Coquillart. Gauss' least constraints principle and rigid body simulations. In *Proc. of IEEE International Conference on Robotics and Automation*, 2002.
- [17] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. *Proc. of Eurographics (Computer Graphics Forum)*, 2002.
- [18] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *SIGGRAPH 96 Conference Proceedings, Annual Conference Series*. ACM SIGGRAPH, Addison Wesley, August 1996.
- [19] Terzopoulos, D., Platt, J.C., and Barr, A.H., Elastically deformable models, *Computer Graphics (Proc. SIGGRAPH)*, vol. 21, pp. 205-214, 1987.