

가상 환경에서의 손동작을 사용한 물체 조작에 대한 시각적 피드백 시스템

서웅^o 권상모 임인성^{*}

서강대학교 컴퓨터공학과

{wng0620, ksm2012, ihm*}@sogang.ac.kr

Visual Feedback System for Manipulating Objects Using Hand Motions in Virtual Reality Environment

Woong Seo^o Sangmo Kwon Insung Ihm^{*}

Department of Computer Science and Engineering, Sogang University

요 약

최근 다양한 종류의 가상 현실 기기가 개발되면서 고전적인 사용자 입력 방식 이외에도 사용자의 신체 동작을 인식하여 현실감을 높이려는 기술에 관한 연구가 활발하게 진행되고 있다. 이러한 장치 중의 하나인 립모션 컨트롤러는 사용자의 손동작을 인식하여 가상 현실 환경에서 사용자의 손 모양을 실시간으로 나타낼 수 있다. 하지만 인식한 사용자의 손을 사용하여 가상 현실의 물체를 조작하면 종종 손이 물체를 통과하게 되는데, 이러한 상호작용은 실제 현실과는 거리감이 있다. 본 연구에서는 가상 현실에서 사용자의 손과 물체 간의 상호작용 현실감을 높이기 위한 시각적 피드백 시스템의 구축 방법을 제시한다. 가상의 현실에서 사용자의 손과 가상 물체가 충돌하는지 광선 추적법을 활용하여 정밀하게 검사하고, 충돌하였을 때 부호 거리 장과 역기구학을 통해 물체 내부로 들어간 사용자의 손가락 말단의 위치를 물체 표면 밖으로 이동시켜 사용자의 손을 재구성하는 과정을 통해 시각적 피드백을 준다. 이를 통해 실시간으로 가상 현실에서 현실감 있는 상호작용을 할 수 있다.

Abstract

With the recent development of various kinds of virtual reality devices, there has been an active research effort to increase the sense of reality by recognizing the physical behavior of users rather than by classical user input methods. Among such devices, the Leap Motion controller recognizes the user's hand gestures and can realistically trace the user's hand in a virtual reality environment. However, manipulating an object in virtual reality using a recognized user's hand often causes the hand to pass through the object, which should not occur in the real world. This study presents a way to build a visual feedback system for enhancing the user's sense of interaction between hands and objects in virtual reality. In virtual reality, the user's hands are examined precisely by using a ray tracing method to see if the virtual object collides with the user's hand, and when any collision occurs, visual feedback is given through the process of reconstructing the user's hand by moving the position of the end of the user's fingers that enter the object through sign distance field and reverse mechanics. This enables realistic interaction in virtual reality in real time.

키워드: 가상 현실, 손동작, 충돌 검사, 부호 거리장, 역기구학, 사용자 인터페이스.

Keywords: Virtual reality, Hand motion, collision detection, signed distance field, inverse kinematics, user interface.

*corresponding author: Insung Ihm/Sogang University(ihm@sogang.ac.kr)

1. 서론

가상 현실(virtual reality) 기술에 관한 관심이 높아지고 다양한 가상 현실 응용 장치가 보급되면서, 게임 콘텐츠와 같은 엔터테인먼트 분야뿐만 아니라 가상 현실에서 시뮬레이션을 통해 실제로 경험하기 힘든 상황을 훈련하거나 장소의 제약을 받지 않는 체험적 교육 등에 폭넓게 활용되고 있다. 이러한 가상 현실에서 사용자가 몰입감 높은 체험을 하려면 가상 공간이 실제 환경처럼 느껴지도록 가상 현실과 상호작용하는 피드백 시스템이 중요하다. 이때 가상 현실과의 상호작용 피드백을 위해 햅틱(haptic) 장비를 사용하면 촉감이나 물체를 조작할 시 실제 물리적인 피드백을 통해 가상 현실에 대한 몰입감을 높일 수 있다[1,2,3,4]. 하지만 아직 시스템 구축 비용 대비 효율이 낮다는 점에서 산업용을 제외하고 일반 사용자가 그러한 장비를 이용하기 힘들다는 문제가 있다.

현재 가상 현실에서 햅틱 장비가 아닌 립모션 컨트롤러(Leap Motion controller)[5]나 밸브 인덱스 컨트롤러(Valve Index controller)[6]와 같이 사용자의 손동작을 인식하는 장비는 비교적 저가로 활용할 수 있어 일반 사용자들이 쉽게 가상 현실과 상호작용을 할 수 있다. 그러나 손가락을 추적하는 장치를 사용한 가상 현실에서의 물체 조작은 사용자의 손을 인식한 가상의 손이 물체 내부로 겹쳐 들어가는 것을 막는 피드백 시스템이 없어, 실제 현실과는 달리 사용자들이 물체를 잡을 시 손가락이 물체를 통과하면서 잡게 된다[7]. 또한, 인식한 가상의 손과 다수의 삼각형으로 구성된 정교한 물체와의 정밀한 충돌 검사를 실시간으로 수행하는 것이 쉽지 않아, 립모션 컨트롤러를 사용하더라도 가상 현실에서의 현실감이 떨어져 사용자의 몰입감을 떨어뜨리는 요인이 된다.

본 논문에서는 가상 현실에서 다양한 작업을 수행하기에 적합한 사용자의 손동작과 물체와의 상호작용을 기반으로 한 시각적 피드백 시스템을 제시한다. 이 시스템은 립모션 컨트롤러로 사용자의 손을 인식한 가상의 손과 물체 사이의 충돌 검사를 수행하고, 이를 기반으로 추가적인 가상 손동작을 재구성하여 시각적으로 피드백한다. 이를 위하여, 첫째, 가상 손과 물체 사이의 충돌 여부는 GPU 기반의 광선 추적법에 기반한 물체-물체 충돌 검사(object-object collision detection) 기술을 적용하여 가속시키는 방법을 통해 실시간으로 판단하였다.

둘째, 손가락 끝부분을 물체 표면으로 얼마나 이동시켜야 하는지를 효과적으로 계산하기 위해 부호 거리 장(signed distance field)을 사용하였다. 마지막으로, 물체 밖으로 이동시킨 손가락 끝부분을 기준으로 역기구학(inverse kinematics) 기법을 적용하여 각 손가락 관절의 각도를 추정하여 재구성하였다.

2. 관련연구

Nasim 등은 립모션 컨트롤러를 사용하여 가상의 환경에서 사용자가 물체를 잡을 때 손이 물체를 통과하는 문제를 시각적으로 피드백해주는 시스템을 제안하였다[8]. 이 방법을 사용하면 시각적으로 손이 물체를 통과하지 않고 잡을 수 있는데 이때 실시간으로 충돌 검사를 위해 손의 3D 모델이 아닌 단순화된 형태의 3D 모델과 NVIDIA사의 PhysX 물리 엔진[9]을 사용하였다. 이 방법은 가상의 손을 교차가 일어나는 순간의 상태로 고정하여 교차한 상태에서 움직이는 손을 나타내지 못하고, 손을 단순화된 형태의 3D 모델을 사용하여 정밀한 교차 탐지를 하지 못한다는 단점이 있다. 마찬가지로 Höll 등이 제안한 상호작용 방법[10]에서도 가상 공간에서 근사화된 손 모델과 물체 간의 교차를 고려하므로 손과 물체 간의 정밀한 교차 탐지에서 한계가 있다.

본 논문에서 제안하는 시스템에서 적용한 광선 추적법을 활용한 물체 간의 충돌 검사는 두 물체를 이루고 있는 모든 삼각형 쌍에 대해 충돌하는지 검사를 수행하는 것을 의미한다. 하지만 모든 삼각형 쌍에 대해 충돌하는지 확인하는 것은 비효율적이다. 이런 문제를 보완하기 위해 octree, bvh, kd-트리 등 다양한 구조의 공간가속 자료구조[11]가 연구되어왔으며 본 논문에서는 이중 비교적 성능이 높다고 알려진 kd-트리를 사용하여 충돌 검사 할 삼각형 쌍을 감소시켜 실시간으로 물체 간 충돌 검사를 수행하였다. 또한, 가상현실 환경에서 실시간의 성능을 위하여 공간 가속 자료구조의 탐색 및 광선-삼각형 교차 검사에 CUDA 기반의 GPU 가속 기법을 적용하였다.

부호 거리는 특정 점으로부터 다각형 모델로 표현된 물체의 표면에 가장 가까운 거리이고, 해당 점이 물체의 내부에 있는 경우 음수 값을 가지고 물체의 외부에 있을 경우 양수 값으로 표현된다. 부호 거리 장은 물체를 둘러싸고 있는 공간을 3차원 그리드(grid) 형태로 나눈 후, 분할된 각 정점에서 물체의 표면까지 가장 가까운 거리인 부호 거리들의 집합을 의미한다. 본 논문에서는 가상의 손과 물체가 충돌하였을 때 가상의 손가락 끝이 이동해야 할 물체 표면의 위치를 찾기 위하여 Chang 등이 제안한 기법[12]을 사용하여 전처리로 부호 거리 장을 생성하여 사용하였다.

손가락 끝부분의 위치가 정해지면 역기구학을 통해 나머지 손가락 관절의 각도를 계산한다. 역기구학 문제는 Buss가 소개한 계산 방법[13]에 기반을 둔다. 역기구학이란 기구학 중 매니퓰레이터(manipulator)의 말단장치(end-effector)의 위치가 주어졌을 때, 관절각(joint angle)을 구하는 문제이다. 이는 주로 로봇학에서 사용하며, 비선형 방정식을 계산하여 구한다.

3. 시스템 구성 및 설계

본 논문에서는 톱모션 컨트롤러로 인식한 사용자의 손을 통해 물체를 조작할 시 발생하는 가상 손의 물체 통과 문제를 해결하기 위하여 추가적인 가상 손의 재구성으로 시각적인 피드백을 주어 현실감을 높이는 새로운 가상 물체 잡기(object grasping) 시스템을 제시한다. 시스템은 크게 두 부분으로 구성된다. 첫째, 실시간으로 손의 메쉬 모델과 물체와의 충돌 검사를 수행하는 부분과, 둘째, 충돌이 발생하였을 때 손가락을 역기구학을 통해 새롭게 재구성하는 부분으로 나뉜다.

3.1 광선 추적법을 활용한 충돌 검사

광선 추적법을 기반으로 한 충돌 검사를 두 물체를 이루고 있는 모든 삼각형 쌍에 대해 수행해야 한다. 이때 두 물체를 이루고 있는 모든 삼각형 쌍에 대해 수행한다면 연산 시간이 오래 걸려 실시간으로 수행하기 어렵다. 따라서 실시간으로 충돌 검사를 수행하기 위해 전처리 과정으로 Wald 등이 제안한 기법[14]으로 kd-트리를 생성하였다. 충돌 검사를 수행할 물체에 대해서 kd-트리를 미리 생성하여, 광선이 지나가는 경로상의 분할 공간 안에 존재하는 삼각형들만 충돌 검사를 하여 불필요한 연산을 줄였다. 성능 향상을 위하여 추가하여 삼각형 쌍에 대한 충돌을 검사하기 전에, 먼저 충돌 검사할 물체를 둘러싸고 있는 AABB(axis-aligned bounding box)와 사용자의 손에 해당하는 모델의 AABB가 교차하는지 검사한다. 두 개의 AABB가 서로 교차하지 않는다면 해당 삼각형이 물체와 충돌하지 않는다고 판단한다. AABB가 교차한다면 Figure 1과 같이 실제 손을 추적하는 때 프레임에 대해서 손 모델을 구성하는 각 삼각형의 정점들로 삼각형 하나당 3개의 광선을 생성한다. 그리고 광선마다 전처리로 만들어 둔 물체의 kd-트리를 탐색하여 광선이 지나가는 공간에 존재하는 삼각형들과 충돌 검사를 수행한다.

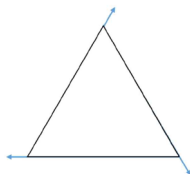


Figure 1: Ray generation for object-object collision detection.

광선과 삼각형들의 충돌 검사는 무게중심좌표 검사(barycentric coordinate tests)의 개선된 방법으로 Wald가 제안한 최적화된 투영 방법(projection method)을 사용한다 [15]. 무게중심 좌표 검사는 광선이 삼각형을 포함하고

있는 평면과 만나는 교점을 구한 뒤, 구한 교점이 삼각형 내부에 존재하는지 확인하는 방법이다. 투영 방법은 무게중심 좌표 검사 방법을 개선한 것인데, 이는 무게중심 좌표와 삼각형을 다른 평면에 투영하여도 중심 좌표가 변경되지 않는 사실을 이용한 방법이다. 따라서 무게중심 좌표와 삼각형을 2차원 좌표 평면인 XY, XZ, YZ 중 투영하는데, 이때 투영된 삼각형의 면적이 가장 큰 평면으로 투영한다. 또한, 추가하여 연산량을 줄이기 위하여 각 삼각형에 대해서 광선과 상관없이 교차 검사 때 필요한 정보를 미리 연산하여 저장함으로써 실시간 연산 시 성능 최적화를 이루었다.

두 물체를 이루고 있는 삼각형 쌍에 대한 충돌 검사는 3D 모델을 구성하는 삼각형 수가 많을수록 연산량이 많아 실시간으로 처리하기 어렵다. 따라서 실시간으로 두 물체 간의 충돌 검사를 위해 CUDA기반의 GPU 병렬처리로 가속하였다. 이를 위해 충돌 검사에서 기준이 되는 물체의 kd-트리, kd-트리의 말단 노드에서의 삼각형 번호 리스트, 정점 정보를 미리 GPU 텍스처 메모리 저장하였다. 따라서 실시간으로 충돌 검사를 수행할 때, GPU에 저장된 정보를 참조하는 방법을 통해 CPU와 GPU 간의 통신을 최소화하여 불필요한 데이터 복사 연산을 피했다. 그리고 추적하고 있는 사용자 손 모델에 대한 정점 정보와 기준이 되는 물체의 좌표계로 옮겨주는 변환 행렬을 GPU 메모리로 복사하고, 이를 사용하여 앞서 설명한 물체 간의 충돌 검사 방법으로 구현된 커널을 호출해 실시간으로 두 물체 간의 충돌 여부를 확인하고 충돌 여부 결과만을 GPU에서 CPU로 전송하였다. 그리고 유니티 엔진에서 실시간 충돌 검사 함수를 사용하기 위해 CUDA로 구현한 충돌 검사 방법을 동적 링크 라이브러리(dynamic link library: DLL)로 구성하여 유니티 엔진에 이식하는 방식으로 구현하였다. 이는 Figure 2와 같다.

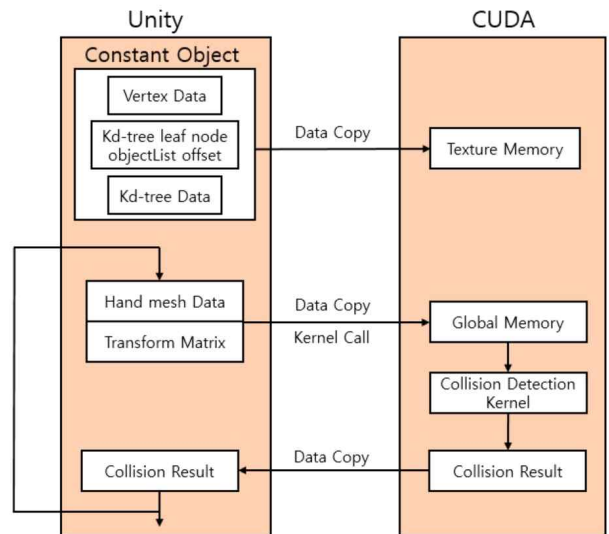


Figure 2: Ray-tracing-based collision detection system.

3.2 역기구학으로 재구성한 가상 손

립모션 컨트롤러로 인식한 사용자의 손을 역기구학으로 재구성하기 위해 말단장치가 도달해야 할 목표 위치가 필요하다. 사용자의 손은 물체를 통과하여 잡고 있기 때문에 손가락의 끝부분을 말단장치라고 하면 도달해야 할 목표 위치는 물체의 표면이 된다. 이를 위해 전처리로 해당 물체에 대해서 부호 거리 장을 생성하여 사용자가 손을 사용하여 물체를 조작할 때, 손가락의 말단장치는 물체 내부로 들어가게 된다. 이때, 부호 거리 장을 활용하여 손가락의 말단장치에서부터 물체 표면까지의 최단거리를 알 수 있고 부호 거리 장의 기울기(gradient)를 계산하여 방향을 알 수 있다. 따라서 손가락의 말단장치의 위치에서 계산한 기울기 방향으로 부호 거리만큼을 이동시킨 위치는 물체 표면에 존재하고 있고, 이를 목표 지점으로 사용하여 역기구학을 통해 사용자의 손가락을 재구성하였다.

앞서 기술한 바와 같이 역기구학이란 기구학 중 매니퓰레이터의 말단장치의 위치가 주어졌을 때, 관절각을 구하는 문제이다. 매니퓰레이터는 관절에 의해 연결된 링크로 구성된다. 매니퓰레이터에 n 개의 관절이 있을 때, 관절각은 열 벡터 $\theta = (\theta_1, \dots, \theta_n)^T$ 로 나타낼 수 있다. 그리고 링크의 어떤 지점을 말단 장치로 정의하였을 때, 말단장치의 위치를 열 벡터 $\vec{s} = (s_1, \dots, s_n)^T$ 로 나타낼 수 있다. 말단장치가 도달해야 할 목표 지점은 열 벡터 $\vec{t} = (t_1, \dots, t_n)^T$ 로 표현가능하고, t_i 는 i 번째 말단장치의 목표 위치이다. 말단장치가 목표 지점에 도달하기 위하여 이동해야 할 변화량은 $\vec{e} = \vec{t} - \vec{s}$ 이다. 말단장치의 위치는 관절 각의 함수이고 이를 $\vec{s} = \vec{s}(\theta)$ 로 표현할 수 있다.

역기구학 문제는 다음 식에서 θ 값을 찾는 것이 된다.

$$\vec{t} = \vec{s}(\theta)$$

위 식은, 함수 \vec{s} 를 자코비안 행렬 J 로 선형근사화하여 반복적인 방법을 통해 계산할 수 있다. 이때, 자코비안 행렬 J 는 다음과 같이 정의된다.

$$J(\theta) = \frac{\partial \vec{s}}{\partial \theta}$$

이를 정기구학을 위한 기본 식을 말단장치의 속도로 나타내면 다음과 같이 정리할 수 있다.

$$\dot{\vec{s}} = J(\theta)\dot{\theta}$$

그리고 관절각 θ 는 $\Delta\theta$ 에 의해 갱신된다.

$$\theta := \theta + \Delta\theta$$

또한, θ, \vec{s}, \vec{t} 를 알고 있다고 가정하면 $J = J(\theta)$ 를 계산할 수 있다. 관절각의 변화에 대한 말단장치의 변화는 변화는 다음과 같다.

$$\Delta \vec{s} \approx J \Delta \theta$$

위 식이 한 번 수행되어 관절각을 갱신하면, 말단장치는 목표 지점 근처에 도달하게 된다. 말단장치가 목표 지점에 충분히 근접할 때까지 반복하여 계산한다. 이 방법은 말단장치의 변화 \vec{s} 가 근사적으로 \vec{e} 와 같아지도록 $\Delta\theta$ 값을 선택해야 한다. 이는 다음과 같은 식으로 나타낼 수 있다.

$$\vec{e} = J \Delta \theta$$

역기구학은 말단장치의 변화로 관절 각의 변화를 계산하는 문제이므로 다음과 같이 자코비안 행렬의 역행렬을 사용하여 구할 수 있다.

$$\Delta \theta = J^{-1} \vec{e}$$

하지만 대부분의 경우 자코비안 행렬이 정사각 행렬이 아니거나 가역 행렬이 아닐 수 있고, 가역 행렬이라도 특이 행렬에 가까우면 문제가 될 수 있다. 따라서 특이점이 발생하는 역행렬을 사용하는 것이 아닌 자코비안의 전치행렬을 사용하는 방법을 사용한다.

$$\Delta \theta = \alpha J^T \vec{e}$$

이때, 가중치 α 를 적절하게 선택하여 수렴 속도를 빠르게 한다. α 값을 선택하는 방법은 새롭게 갱신된 벡터 \vec{e} 의 값을 최소화하는 것이다. 이 방법을 위해 말단장치의 변화가 정확하다고 가정하고, α 값은 벡터 \vec{e} 에 가능한 근접하게 해주는 값으로 선택한다. 이는 다음과 같다.

$$\alpha = \frac{\vec{e} \cdot J J^T \vec{e}}{J J^T \vec{e} \cdot J J^T \vec{e}}$$

사용자의 손가락과 물체 간의 충돌이 발생하였을 시, 사용자의 손가락은 물체를 통과하기 때문에 사용자의 손가락의 말단장치를 물체 표면 밖으로 이동시켜 재구성하였다. 이는 Figure 3과 같다.

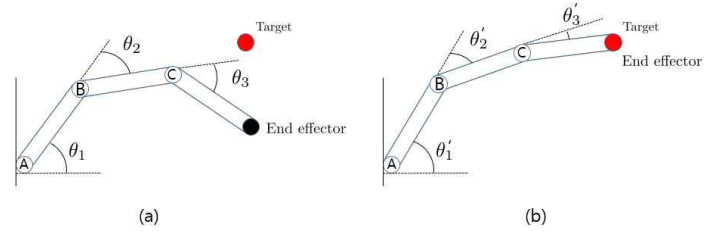


Figure 3: New finger position adjusted via inverse kinematics.

Figure 3(a)의 경우 말단장치를 목표지점으로 옮기기 전 상태이다. 이를 역기구학으로 Figure 3(b)와 같이 옮기기 위해, 먼저 자코비안 행렬을 정의해야 한다. 자코비안 행렬을 관절의 위치와 회전축으로 계산하면 다음과 같다.

$$J = \begin{bmatrix} (R_A \times (E - A))_X & (R_B \times (E - B))_X & (R_C \times (E - C))_X \\ (R_A \times (E - A))_Y & (R_B \times (E - B))_Y & (R_C \times (E - C))_Y \\ (R_A \times (E - A))_Z & (R_B \times (E - B))_Z & (R_C \times (E - C))_Z \end{bmatrix}$$

여기서 A, B, C 는 각 관절의 위치, R_A, R_B, R_C 는 각 관

절 A, B, C 의 회전축을, E 는 말단장치의 위치를 의미한다. 구한 자코비안의 전치행렬과 말단장치가 목표 지점에 도달하기 위해 이동해야 할 변화량을 나타낸 벡터를 곱하여 각 손가락 관절 각들의 변화를 구하였다. 구한 관절 각의 변화를 사용하여 순기구학(forward kinematics)으로 손가락 관절들의 위치와 각도를 갱신하는데, 이때 앞서 기술한 가중치 α 를 사용하여 수렴 속도를 빠르게 하였다. 말단장치가 목표 지점에 충분히 근접할 때까지 반복 계산하여 손가락을 재구성하였다.

4. 시각적 피드백 시스템

4.1 시각적 피드백 시스템 구현

시각적 피드백 시스템은 앞서 기술한 바와 같이 먼저, 각 물체에 대한 kd-트리와 부호 거리 장을 전처리로 미리 생성한다. 이를 시각적 피드백 시스템을 시작할 때, 물체들의 정점 정보와 미리 생성해둔 물체들에 대한 kd-트리, kd-트리의 리프 노드(leaf node)에서의 삼각형 변호 리스트 정보를 GPU 텍스처 메모리에 저장한다. 또한, 물체들에 대한 부호 거리 장 정보는 메인 메모리에 저장하고, 부호 거리 장의 기울기를 계산하여 메인 메모리에 저장한다.

그리고 실시간으로 립모션 컨트롤러로 추적하고 있는 사용자의 손과 조작할 물체 간의 충돌 검사를 수행한다. 이때, “Bounds.Intersects”라는 유니티 엔진에서 제공하는 함수로 사용자의 손을 둘러싸고 있는 바운딩 박스와 물체를 둘러싸고 있는 바운딩 박스 간의 충돌을 검사한다. Figure 4는 사용자의 손과 물체의 바운딩 박스를 나타낸 것이다.

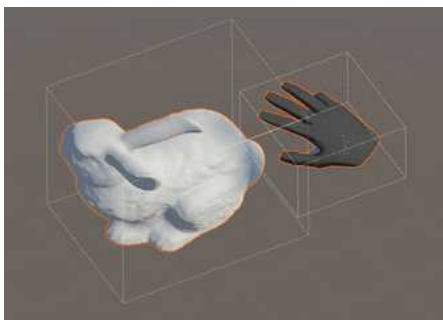


Figure 4: Bounding boxes of hand and 3D object.

사용자의 손과 물체의 바운딩 박스 간의 충돌이 발생한 경우, 앞서 기술한 광선 추적법을 기반으로 한 사용자의 각 손가락과 물체 간의 충돌 검사를 수행한다. 사용자의 손과 물체의 좌표계를 일치시켜주기 위해 사용자의 손의 지역 좌표계에서 세상 좌표계로 변환하고, 세상 좌표계에서 물체의 지역 좌표계로 변환해주는 변환행렬

과 사용자의 손가락의 정점 정보를 GPU 메모리에 저장한다. 이를 사용하여 광선 추적법을 기반으로 한 충돌 검사를 수행하는 CUDA 커널을 호출하여 사용자의 손과 물체의 삼각형들간의 충돌 검사를 수행한다. Figure 5(a)는 사용자의 모든 손가락과 물체와의 충돌이 일어나지 않았을 경우의 모습이다. Figure 5(b)는 사용자의 검지 손가락과 물체 간의 충돌이 일어나 손 전체의 색깔이 반투명한 빨간색으로 나타났고, 손가락이 충돌한 물체 안으로 들어간 부분은 초록색으로 나타내어 충돌이 발생함과 손가락이 물체 안으로 들어간 정도를 시각적으로 알 수 있도록 구현하였다.

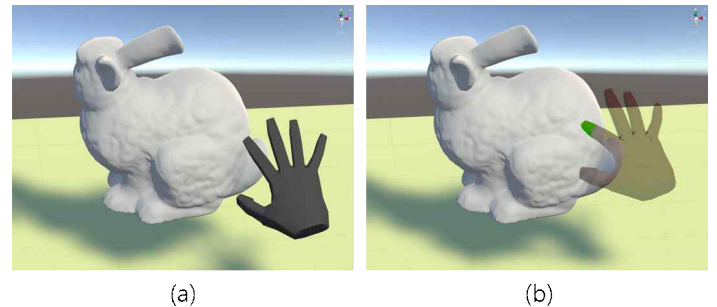


Figure 5: Collision detection between hand and object.

사용자의 손가락이 충돌이 발생한 경우, 물체 내부에 들어가 있는 손가락의 말단장치의 위치를 물체의 부호 거리 장의 해상도를 기준으로 정규화하여 부호 거리 장의 인덱스를 계산하여 부호 거리와 부호 거리의 기울기 정보를 가져온다. 그리고 손가락 말단장치의 위치에서 기울기 방향으로 부호 거리만큼 밀어낸 위치를 계산한다. 이는 물체의 표면에 위치하게 되고, 손가락의 말단장치가 도달해야 할 목표위치가 된다. 앞서 설명한 역기구학을 이용하여 손가락의 말단장치의 위치를 목표위치로 옮기기 위해 비선형 방정식을 풀어 손가락의 관절각들을 계산하여 손가락을 재구성한다. 이는 Figure 6과 같다.

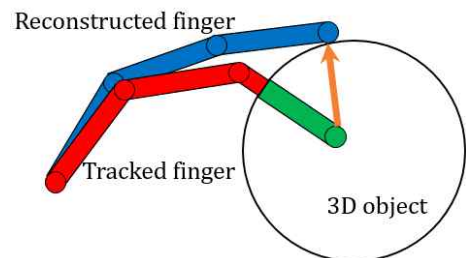


Figure 6: Finger movement in signed distance field.

시각적 피드백 시스템에서 움직이는 사용자의 손과 물체와의 충돌을 검사하면서 시각적 피드백을 위한 손을 보여주기 위해 립모션 컨트롤러로 실시간으로 추적하는 가상 손과 이와 같은 데이터를 받는 재구성을 위한 손을 각각 생성하였다. 실시간으로 추적하는 가상 손은 물체와의 충돌을 검사할 때 사용하고, 재구성을 위한 손은 가상 손이 물체와 충돌이 발생하였을 시 역기구학을 통

해 손가락의 관절각을 계산하여 시각적 피드백을 주는 손으로 사용하였다. 이는 Figure 7과 같다. 손가락과 물체가 충돌이 발생하여 손이 반투명한 빨간색이 되고, 물체 내부로 들어간 손가락 부분은 초록색이 된다. 그리고 재구성한 손은 파란색으로 나타나 시각적 피드백을 할 수 있게 하였다.

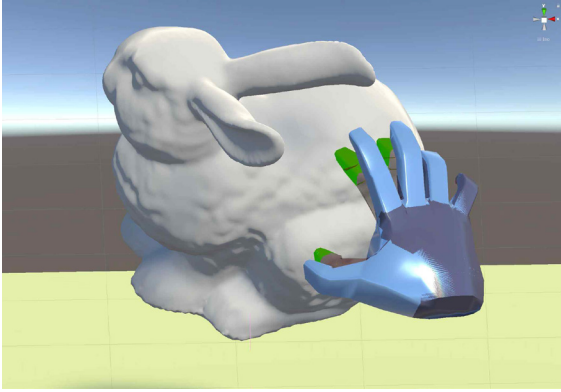


Figure 7: Adjustment and visualization of user hand.

구현한 손을 사용한 물체 조작에 대한 시각적 피드백 시스템이 실시간으로 올바르게 작동하는지 확인하였다. 실제 립모션 컨트롤러로 추적하고 있는 사용자의 손은 반투명한 빨간색을 띠게 하였고, 물체 안으로 들어간 손가락 부분을 초록색을 띠게 하였다. 그리고 재구성한 가상 손은 파란색을 띠게 하여 시각적 피드백 시스템의 결과를 확인하였다. Figure 8, 9는 가상 물체에 대한 조작을 하였을 때의 시각적 피드백 결과이다. 물체 안으로 들어간 사용자의 손가락에 대해 시각적 피드백이 올바르게 작동하는 것을 확인하였다.

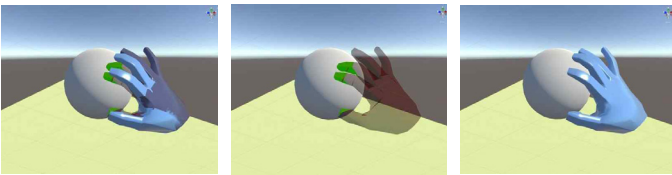


Figure 8: Experimental result for a sphere object.

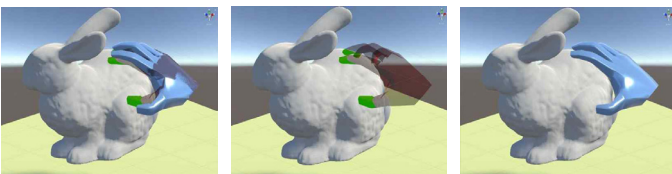


Figure 9: Experimental result for a bunny object.

4.2 날카로운 물체에 대한 처리

Figure 9에서 토끼 물체의 몸통을 잡으면 앞서 확인한 공 물체와 같이 시스템이 제대로 동작하지만, Figure 10과 같이 토끼 물체의 귀의 경우에는 손가락이 귀를 뚫고

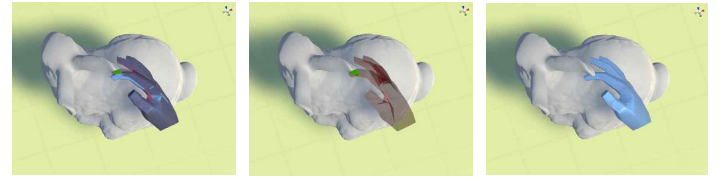


Figure 10: Problems with thin part of object.

지나가는 문제가 발생한다.

문제가 발생하는 경우는 물체에서 얇은 부분을 조작하였을 때 발생하는데 Figure 11과 같이 물체의 얇은 부분 내부에 존재하는 부호 거리의 해상도가 부족하여 조금만 손가락이 통과하여도 표면에서 가장 가까운 곳의 방향이 반대로 향하게 되고, 손가락이 물체를 통과하는 방향과 같은 방향으로 밀어내게 되어 올바르지 못한 결과가 나온 것이다.

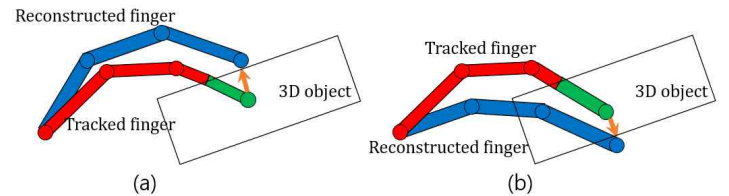


Figure 11: Reason for causing problems with thin objects.

얇은 부분에서의 문제를 해결하기 위해 Figure 12와 같이 손가락이 물체를 통과하는 방향(\vec{u})과 손가락 말단 장치의 위치를 사용하여 가져온 부호 거리 장의 방향(\vec{v})의 내적을 사용하였다. 손가락이 물체를 통과하는 방향과 부호 거리 방향간에 내적을 취하여 얻은 값의 부호가 음수라면 두 벡터의 사잇각은 $90^\circ \sim 180^\circ$ 또는 $-90^\circ \sim -180^\circ$ 이므로 올바른 시각적 피드백의 결과를 얻을 수 있다고 판단하여 기존의 방법처럼 손을 재구성하였고 또한 그때의 부호 거리 장의 인덱스를 저장해두었다. 만약 두 벡터를 내적하여 얻은 값의 부호가 양수라면 두 벡터의 사잇각은 $0^\circ \sim 90^\circ$ 또는 $0^\circ \sim -90^\circ$ 이므로 올바르지 못한 결과를 얻는다고 판단하여 기존의 방법이 아닌 올바른 결과를 얻었을 때 저장해둔 부호 거리 장의 인덱스를 가지고 손가락 말단장치의 위치가 이동해야 할 새로운 목표 위치를 계산하고, 기존의 역기구학 방법을 사용하여 사용자의 손을 재구성하여 시각적 피드백을 하였다.

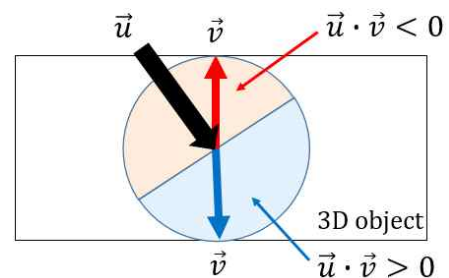


Figure 12: Improved method for thin objects.

5 실험 결과

5.1 실험 환경

본 논문에서 실험에 진행한 PC의 사양은 아래와 같다. 가상 현실 HMD(head mounted display)는 HTC사의 Vive Pro를 사용하였으며 Figure 13과 같이 립모션 컨트롤러를 HMD 전면부에 장착하여 실험하였다.

- CPU : Intel Core i7-5960X(3.00GHz)
- GPU : NVIDIA GeForce GTX 1080



Figure 13: Leap Motion controller on a VR HMD.

5.2 피드백 시스템 적용 결과

Figure 14, 15의 각각의 그림에서 왼쪽의 그림은 실시간으로 추적되는 손과 제안 시스템으로 재구성한 손을 동시에 나타낸 것이다. 중간의 그림은 실제 추적하고 있는 사용자의 손(빨간색)과 물체 안으로 들어간 손가락 부분(초록색)을 나타냈다. 오른쪽의 그림은 재구성한 가상 손만을 렌더링한 결과이다.

실험 과정에서 물체 표면에서 손이 움직일 때, 손가락이 불안정하게 떨리는 문제가 발생하였다. 이는 부호 거리 장의 해상도가 토끼 물체에 대해서는 $128 \times 128 \times 104$, 폰 물체에 대해서는 $76 \times 128 \times 76$ 과 같이 낮을 때 발생하였다. 물체 내부를 통과하여 들어간 손가락 말단장치의 위치를 사용하여 부호 거리 장의 인덱스를 계산하게 되는데, 해상도가 낮으면 물체 내부의 부호 거리의 개수가 적어 부호 거리값 간의 차이가 커지고 결국 손가락 말단장치가 이동해야 할 목표 위치를 계산할 때 오차가 커져 시각적 피드백이 올바르게 못한 결과가 나왔다. 이와 같은 문제는 해상도를 높이면 위와 같은 문제가 발생하지 않았다. 이는 물체 내부에 부호 거리의 개수가 많아 촘촘한 부호 거리 장을 가져 부호 거리값의 차이가 크지 않아 시각적 피드백의 결과가 올바르게 나오고 실험 과정에서 불안정하게 떨리는 문제가 발생하지 않았다.

5.3 얇은 물체에 대한 개선 방법 적용 결과

날카로운 물체에 대해 발생하는 문제를 개선한 방법으로 실험을 진행하였다. Figure 16은 부호 거리 장의 해상도가 $512 \times 508 \times 400$ 인 토끼 물체의 귀에 대한 개선된 시각적 피드백 결과이다. Figure 17은 부호 거리 장의 해상도가 $216 \times 512 \times 264$ 인 의자 물체에 대한 개선된 시각적 피드백 결과이다.

기존의 방법의 경우, Figure 16(a)와 Figure 17(a)의 왼쪽 그림의 빨간색 원 부분을 보면 토끼 물체의 귀나 의자 물체와 같이 얇은 부분을 손으로 조작하였을 때의 시각적 피드백의 결과로 재구성한 손이 올바른 결과를 내지 못하고 손가락이 반대쪽으로 나가는 문제가 발생하였다. 이와 달리 개선한 방법의 경우, Figure 16(b)의 왼쪽 그림의 빨간색 원 부분을 보면 토끼의 귀를 조작하였을 때, 손가락이 반대쪽으로 나가지 않고 그 자리에 유지되는 것을 확인하였다. 또한, Figure 17(b)의 왼쪽 그림의 빨간색 원 부분을 보면 의자를 조작하였을 때, 시각적 피드백이 올바른 결과를 내는 것을 확인하였다.

표면이 고르지 못한 물체에 대해 시각적 피드백 시스템의 실험을 진행하였다. Figure 18은 부호 거리 장의 해상도가 $480 \times 512 \times 508$ 인 바이러스 구조 물체에 대한 시각적 피드백의 결과이다. Figure 19는 부호 거리 장의 해상도가 $512 \times 512 \times 512$ 인 기하학적 구조 물체에 대한 시각적 피드백의 결과이다. 바이러스 구조 물체의 돌기 부분이 아닌 매끄러운 표면에 대해 손을 사용하여 물체 조작을 수행하여 올바른 결과를 얻을 수 있었지만, Figure 18과 같이 물체 표면에 있는 돌기 부분에 대해 물체를 조작하였을 때, 빨간색 원 부분을 보면 재구성한 손가락이 돌기를 통과하여 시각적 피드백이 올바르게 되지 않았다. 또한, Figure 19에서도 표면에 주름이 많고 돌기와 같은 부분이 많은 기하학적 구조 물체에 대해서도 동일한 문제가 발생하였다. 이는 시각적 피드백을 위한 부호 거리 장의 정보를 가져올 때, 물체의 표면이 고르지 않은 부분에 대한 부호 거리 장 정보의 변화가 커 문제가 발생하였다.

6. 결론

본 논문에서는 가상 현실에서 사용자의 몰입감과 현실감을 높이기 위하여 립모션 컨트롤러를 사용하는 사용자의 손을 사용한 물체 조작에 대한 시각적 피드백 시스템을 제시하였다. 시각적 피드백 시스템을 설계하기 위해 광선 추적법을 기반으로 한 물체 간의 충돌 검사를 수행하였고, 물체의 부호 거리 장을 사용하여 물체를 통과하여 들어온 손가락의 말단 부분의 위치에서 가장 가까운

물체 표면의 위치를 계산하였다. 계산한 물체 표면의 목표 위치를 토대로 역기구학을 통한 사용자의 손가락의 관절 각들에 대해 비선형 방정식을 풀어 손가락을 재구성하였다. 또 실시간으로 시각적 피드백을 위해 가장 수행 시간이 큰 광선 추적법을 기반으로 한 물체 간의 충돌 검사에 대해 GPU 가속을 진행하였다.

본 논문에서는 CUDA 기반으로 구현된 광선 추적법으로도 실시간 성능으로 충돌 검사를 수행하였으나 만약 필요에 따라 대용량 데이터에 대해서 실시간으로 충돌 검사를 수행해야 한다면 향후 Nvidia 사의 RTX 시리즈와 같이 광선 추적 하드웨어가 탑재된 GPU를 활용 하여 하드웨어로 광선 추적 연산을 수행한다면 연산 비용이 추가적으로 감소할 것으로 예상된다.

본 논문에서 구현한 시각적 피드백 시스템은 동적으로 움직이는 사용자의 손과 정적인 물체를 고려하고 있으나 시스템 응용에 따라서 향후 가상 물체도 기하 정보가 동적으로 변한다면 물체의 기하 정보가 변할 때마다 광선 추적법을 위한 공간 가속 자료구조의 재구축 및 역기구학을 적용하기 위한 부호 거리 장의 재구축이 필요하다. 따라서 동적 물체에서 효과적으로 사용할 수 있는 BVH (bounding volume hierarchy)와 같은 자료구조를 적용하거나 실시간으로 부호 거리 장을 생성하는 알고리즘을 고려할 필요가 있다.

감사의 글

이 성과는 정부 (과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2020R1A2C2011709).

References

[1] C. W. Borst and A. P. Indugula, "Realistic virtual grasping," in *Proceedings of 2005 IEEE Conference on Virtual Reality(VR)*, pp. 91-98, March 2005.

[2] J. Blake and H. B. Gurocak, "Haptic glove with MR brakes for virtual reality," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 5, pp. 606-615, 2009.

[3] M. Prachyabrued and C. W. Borst, "Effects and optimization of visual-proprioceptive discrepancy reduction for virtual grasping," in *Proceedings of 2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 11-14, 2013.

[4] I. Choi, H. Culbertson, M. R. Miller, A. Olwal and S. Follmer, "Gravity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality," in

Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology(UIST '17), pp. 119-130, 2017.

[5] Ultraleap Ltd., "Leap Motion Controller," Available: <https://developer.leapmotion.com> [Online accessed 24-Apr.-2020].

[6] Valve Corporation, "Valve Index Controllers," Available: <https://store.steampowered.com/valveindex> [Online accessed 24-Apr.-2020].

[7] M. Chessa, G. Maiello, L. K. Klein, V. C. Paulun, and F. Solari, "Grasping objects in immersive virtual reality," in *Proceedings of 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1749-1754, March 2019.

[8] K. Nasim, and Y. J. Kim, "Physics-based interactive virtual grasping," in *Proceedings of HCI Korea 2016*, pp. 114-120, Jan. 2016.

[9] NVIDIA Corporation, "PhysX SDK," Available: <https://developer.nvidia.com/physx-sdk> [Online accessed 24-Apr.-2020].

[10] M. Höll, M. Oberweger, C. Arth, and V. Lepetit, "Efficient Physics-Based Implementation for Realistic Hand-Object Interaction in Virtual Reality," in *Proceedings of 2018 IEEE Conference on Virtual Reality(VR)*, pp. 175-182, March 2018.

[11] I. Wald, W. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. Parker, and P. Shirley, "State of the art in ray tracing animated scene," *STAR - State of The Art Report*, Eurographics 2007, 2007.

[12] B. Chang, D. Cha, and I. Ihm, "Computing Local Signed Distance Fields for Large Polygonal Models," *Computer Graphics Forum (EuroVis 2008: Eurographics/IEEE Symposium on Visualization)*, Vol. 27, No. 3, pp. 799-806, May 2008.

[13] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," Technical report, Dept. Math. Univ. of California, San Diego, CA, USA, 2004.

[14] I. Wald, V. Havran, "On building fast kd-trees for ray tracing, and on doing that in $O(N \log N)$," in *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing*, pp. 61-69, 2006.

[15] I. Wald, Realtime Ray Tracing and Interactive Global Illumination, PhD thesis, Saarland University, 2004.

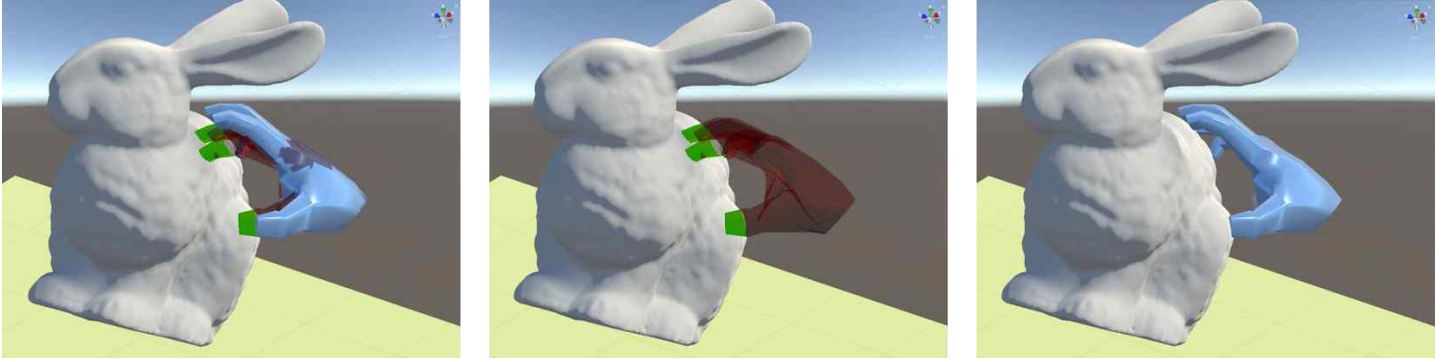


Figure 14: Hand adjustment using the proposed method (Bunny model).

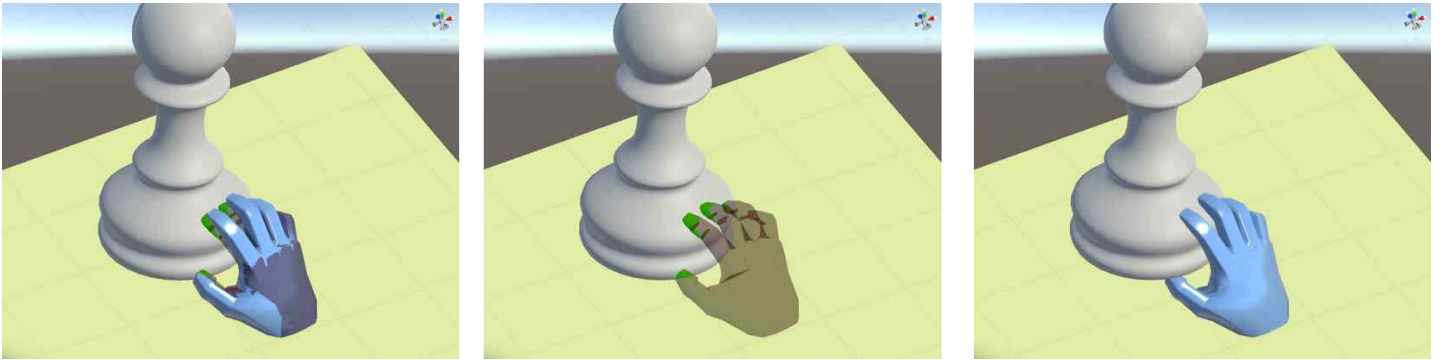
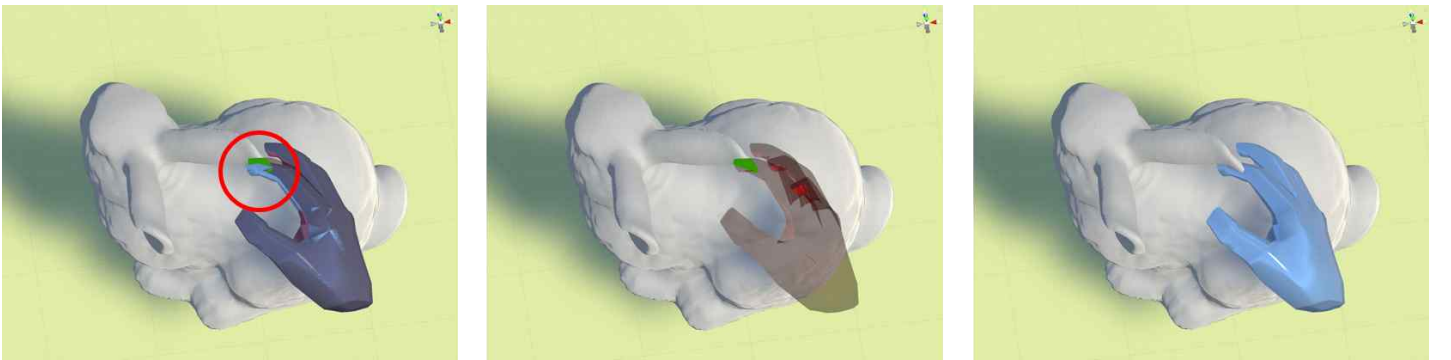
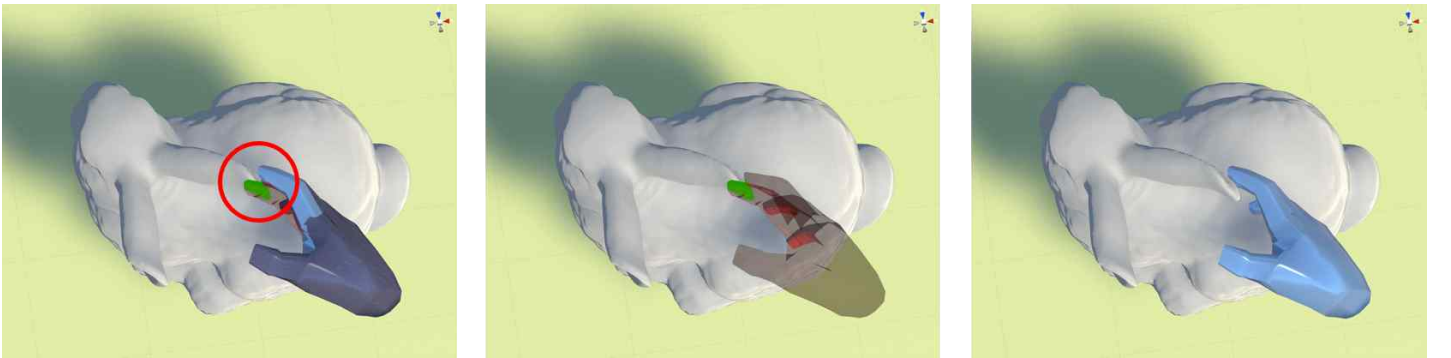


Figure 15: Hand adjustment using the proposed method (Pawn model).

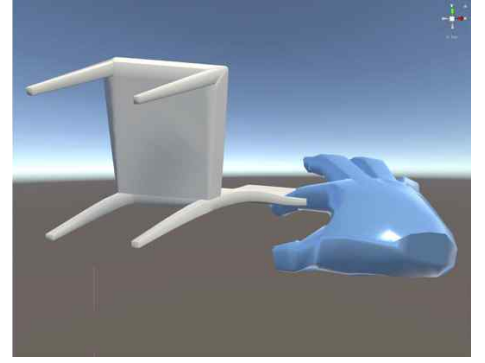
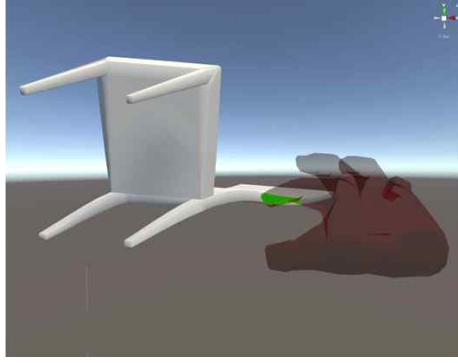
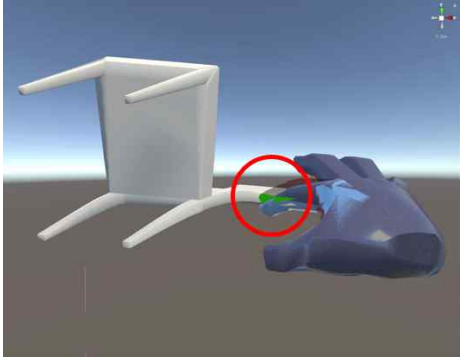


(a) Problematic result

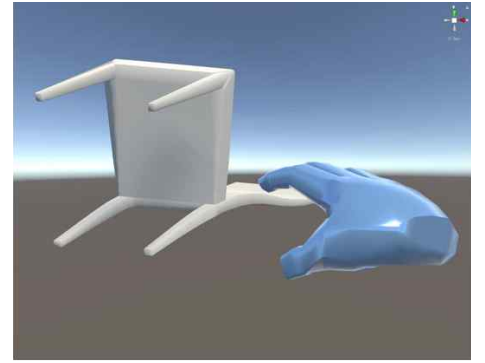
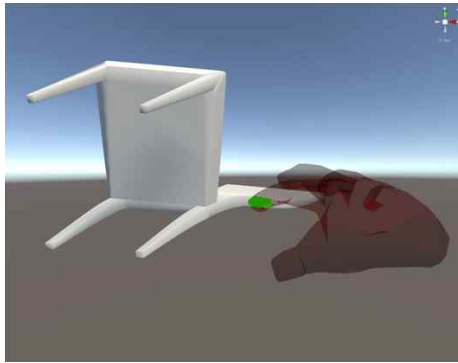
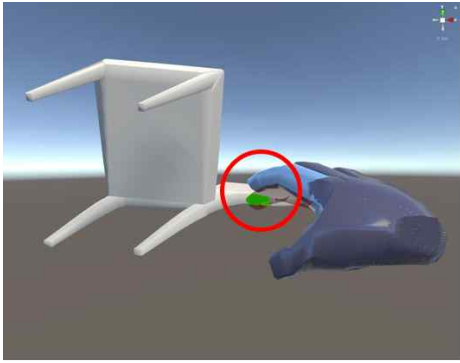


(b) Improved result

Figure 16: Hand adjustment using the proposed method on thin parts (Bunny model).



(a) Problematic result



(b) Improved result

Figure 17: Hand adjustment using the proposed method on thin parts (Chair model).

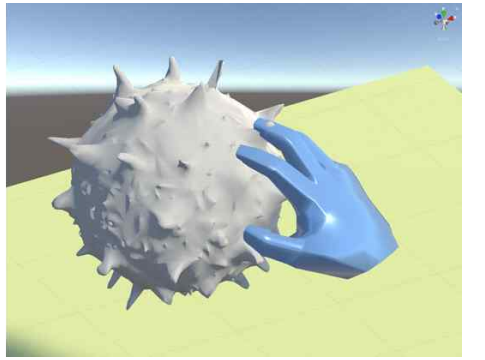
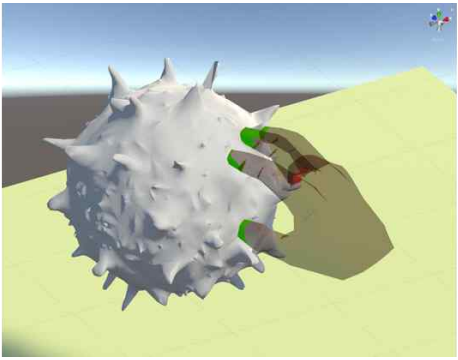
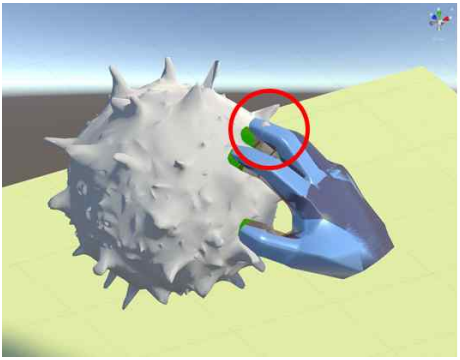


Figure 18: Experiment with intractable 3D object 1.

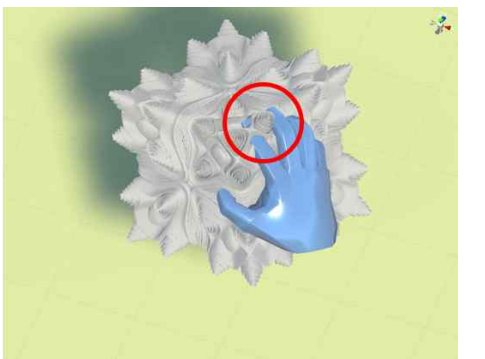
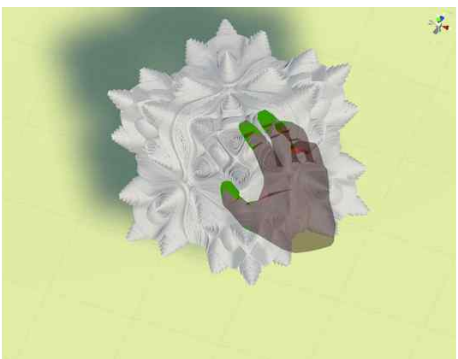


Figure 19: Experiment with intractable 3D object 2.

〈 저 자 소 개 〉



서 옹

- 2012년 한국외국어대학교 전자공학과 학사
- 2014년 서강대학교 컴퓨터공학과 석사
- 2014년~현재 서강대학교 컴퓨터공학과 박사과정
- 관심분야 : 컴퓨터 그래픽스, 모바일 렌더링, GPGPU
- <https://orcid.org/0000-0001-8272-9169>



권 상 모

- 2018년 서강대학교 컴퓨터공학과 학사
- 2020년 서강대학교 컴퓨터공학과 석사
- 관심분야 : 컴퓨터 그래픽스, 가상 현실, 3D 게임 엔진
- <https://orcid.org/0000-0001-8989-6488>



임 인 성

- 1985년 서울대학교 자연과학대학 계산통계학과 학사
- 1987년 Rutgers 대학교 컴퓨터과학과 석사
- 1991년 Purdue 대학교 컴퓨터과학과 박사
- 1993년~현재 서강대학교 공과대학 컴퓨터공학과 교수
- 관심분야 : 컴퓨터 그래픽스, 고성능 계산, 과학적 가시화
- <http://orcid.org/0000-0002-5611-925X>