

GPU 기반의 부채꼴 요소법을 이용한 햅틱 도자기 모델링 시스템

(A Haptic Pottery Modeling System Using GPU-Based Circular Sector Element Method)

이 재 봉 [†] 한 갑 중 ^{**} 최 승 문 ^{***}
(Jaebong Lee) (Gabjong Han) (Seungmoon Choi)

요약 본 논문에서는 사용자가 실제와 같은 촉감을 느끼면서 가상 도자기의 물체 성형을 체험할 수 있는 E-Learning 시스템을 제안한다. 원통형으로 대칭을 이루는 도자기 모양의 특징에 착안하여 부채꼴 모양을 가진 요소의 집합으로 3차원 도자기를 모델링하였다. 부채꼴 요소법에 최적화된 충돌 처리와 인접 요소간 상호작용 알고리즘을 고안하였으며, GPU 기반의 빠른 햅틱 모델과 시각 모델의 동기화를 구현하였다. 성능 평가 결과 부채꼴 요소법은 기존의 변형체 렌더링 기법에 비해 훨씬 더 조밀한 도자기 모델의 현실적인 실시간 햅틱 렌더링이 가능한 것을 확인하였다. 우리가 구현한 시스템을 도자기와 관련된 교육적인 콘텐츠와 잘 결합한다면 초등학교생들을 대상으로 한 E-Learning 시스템으로 성공적인 활용이 가능할 것으로 예상된다.

키워드 : 도자기 모델링, 가상 찰흙, 햅틱 렌더링, 변형체 렌더링, GPU 기반 렌더링

Abstract This paper presents an efficient modeling system of virtual pottery in which the user can deform a body of virtual clay with a haptic tool for E-learning. We propose a Circular Sector Element Method (CSEM) which represents the virtual pottery with a set of circular sector elements based on the cylindrical symmetry of pottery. Efficient algorithms for collision detection and response, interactions between adjacent elements, and GPU-based visual-haptic synchronization are designed and implemented for the CSEM. Empirical evaluation showed that the modeling system is computationally efficient with finer details and provides convincing model deformation and force feedback. The developed system, if combined with educational contents, is expected to be used as an effective E-learning platform for elementary school students.

Key words : Pottery modeling, Virtual clay, Haptic rendering, Deformable object rendering, GPU-based rendering

· 본 논문은 한국연구재단의 도약연구사업(No. R0A-2008-000-20087-0) 및 정보통신산업진흥원의 ITRC 연구센터 지원사업(NIPA-2010-C1090-1011-0008)의 연구비 지원에 의해서 연구되었습니다.

[†] 비 회 원 : LG전자 DS연구소

jbong.lee@lge.com

^{**} 비 회 원 : POSTECH 컴퓨터공학과

hki84@postech.ac.kr

^{***} 정 회 원 : POSTECH 컴퓨터공학과 교수

choism@postech.ac.kr

논문접수 : 2010년 1월 21일

심사완료 : 2010년 6월 21일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제37권 제8호(2010.8)

1. 서론

도자기 체험 학습은 역사, 문화, 정서적인 교육 목적으로 자주 선택하는 주제이다. 하지만 장소나 장비 등의 제약으로 인해 직접 물레를 사용한 도자기 성형을 체험하기란 쉽지 않다. 이 연구의 목적은 가상 현실 상에서 실제 도자기를 만드는 것과 비슷한 느낌을 받으며 도자기 성형을 체험할 수 있는 시스템을 구축하는 것이다. 이와 같이 가상 현실을 이용하여 시스템을 구축하면 얻을 수 있는 몇 가지 장점이 있다. 먼저 기본적인 시스템 구축 비용 이외의 추가적인 비용이 거의 들지 않는다. 또한 특정 장소나 환경에 구애 받지 않기 때문에 학습자가 보다 쉽고 편리하게 접근할 수 있다. 가상 현실 공간 속에서의 도자기 체험이라는 것 자체가 색다른 재미

를 선사할 수 있으며, 컴퓨터와 인터넷을 통해 관련된 교육적인 자료와 쉽게 결합할 수 있으므로 교육적인 효과를 극대화할 수 있다.

본 연구에서는 원통형으로 대칭을 이루는 도자기 모양의 특징에 착안하여 부채꼴 모양을 가진 요소의 집합으로 3차원 도자기를 구성하여 일반적인 성능의 컴퓨터에서도 조밀한 도자기 모델의 실시간 햅틱 렌더링이 가능하도록 하였다. 우리가 개발한 부채꼴 요소법에 적합한 충돌 감지 및 반응 알고리즘을 고안하였으며, 이를 이용해 실제 E-Learning에 사용할 수 있는 시스템을 구현하였다. 성능 측정 결과 매우 조밀한 3차원 도자기 모델의 1kHz 햅틱 렌더링이 가능했으며, 다양하고 현실적인 도자기 모델을 쉽게 디자인 할 수 있었다. 이를 통해 본 연구의 시스템이 가지는 연구적인 의의는 다음과 같다.

- 변형체 모델에 관한 기존 연구들에 비해서 대칭성에 기반한 보다 조밀한 표현이 가능
- 조밀한 모델을 사용하면서 1 kHz이상의 햅틱 렌더링 성능을 달성
- 그래픽 가속 성능을 활용하여 모델을 시각 및 햅틱으로 동시에 표현

2. 관련 연구

가상 찰흙이나 도자기 모델링 시스템에 관한 초기 연구에서는 진동 피드백을 통해 가상 물체와의 접촉 느낌을 제공하였다[1,2]. 이후 직접 물체를 만지는 것과 유사한 느낌을 주기 위해 힘을 전달할 수 있는 햅틱 장치를 사용하기 시작하였다[3-6]. Massie는 PHANToM 장치와 복셀(voxel) 모델을 이용한 햅틱 찰흙 렌더링의 가능성을 보였다[3]. Chai는 가상 찰흙을 NURBS 모델의 자유 변형(free-form deformation)을 통해 모델링하고, ISU 햅틱 외골격 장치를 통해 찰흙과 상호작용하는 힘을 렌더링하였다[4]. McDonnell은 유한 요소법(finite element method)을 단순화하여 가상 찰흙을 모델링 하였다[5]. Han은 도자기 모델에 적합한 형태로 모델을 단순화 한 원통 요소법을 제안하고 이를 통해 고해상도의 도자기를 렌더링 하였으며, 이 방법은 본 논문에서 제시한 도자기 모델링 방법의 기초가 되었다[6].

가상 찰흙이나 도자기의 햅틱 렌더링에 관한 연구는 일반적인 변형체(deformable object)를 다루는 연구의 특수한 경우로 볼 수 있다. 변형체를 렌더링 하는 방법에는 크게 자유 변형과 유한 요소법의 두 가지 접근 방법이 있다.

자유 변형은 미리 정의된 제어점을 기하학적인 관점에서 처리하여 물체를 변형한다[7]. 자유 변형은 상대적으로 처리 속도가 빠르지만, 물체 내부의 상호작용과 관

련한 물리적인 요소를 고려하지 않는 단점이 있다. 자유 변형은 계속해서 발전해 왔으며[8-10], 앞서 언급한 것과 같이 가상 찰흙 모델링 방법으로도 활용되었다[1,2,4].

이와 반대로 유한 요소법은 물리적인 상호작용을 고려하여 현실적이고 정확한 렌더링을 추구하는 방법이다[11]. 하지만 유한 요소법은 계산량이 많아 1kHz 정도의 렌더링 속도를 요구하는 실시간 햅틱 렌더링에 활용하기에는 제약이 따르므로 모델을 단순화하기 위한 다양한 방법이 개발되었다. 예를 들어, Mendoza와[12] Conti는[13] 간단한 질량과 스프링, 댐퍼 모델을 표현하는 1차 방정식을 사용해 햅틱 렌더링을 하였다. 또한 계산하는 요소의 개수를 줄이는 접근 방법으로 표면에 있는 요소들의 상호작용을 고려하는 방법(Boundary Element Method)[14], 긴 요소들의 2차원 집합을 사용하는 방법(Long Element Method)[15,16] 등도 제시되었다. Kim은 볼륨 데이터에서의 빠른 햅틱 렌더링을 위해 인접한 요소들의 상호작용을 고려하는 방법(Shape-retaining Chain Linked Model)을 제시하였다[17]. 또한, 실시간으로 요소의 복잡성을 조절하는 방법도 많이 사용되고 있다[18-21].

3. 도자기 모델의 구조

도자기는 대부분 원통형으로 대칭적인 모양을 가지고 있으며 제작 과정에서는 보통 회전하는 중심축 방향으로만 변형이 된다. 따라서 그림 1과 같이 부채꼴 모양 요소의 집합으로 도자기 모델을 구성하고 각 요소의 길이만 변형하면 실제 도자기를 충분히 잘 표현할 수 있다. 이와 같은 아이디어는 LEM(Long Element Method)과 비슷한 것으로, LEM에서는 직교 좌표계에서 한 축 방향으로만 막대기 형태의 요소를 변형하지만, 부채꼴 요소법에서는 원통 좌표계에서 r 축 방향으로만 요소를 변형한다. 또한, LEM에서는 내부 모델의 부피를 고려한 형태의 변형이 이루어지지만, 부채꼴 요소법에서는 도자기 모델의 부피를 고려하지 않은 요소의 변형이 이루어진다는 차이가 있다. 이와 같이 부채꼴 요소법은 3차원 도자기를 2차원 모델로 표현하고 각 요소의 변형 방향을 제한하여 조밀한 모델의 효율적인 처리를 가능하게 한다.

그림 2에는 앞으로 논문에서 사용할 좌표축과 기호를 정의하였다. 그림에서 y 축을 중심축으로 한 원통은 도자기 모델을 나타내고 오른쪽의 구는 HIP(Haptic Interface Point)를 나타낸다. 각 부채꼴 요소는 원통 좌표계에서 요소 끝 점의 좌표 $S^i = (S_r^i, S_\theta^i, S_h^i)$ 를 통해 나타내며 혼동의 여지가 없을 경우 인덱스 i 는 빼고 표현하기도 한다. 우리가 사용하는 도자기 모델은 이미 내부가 비어있는 상태를 가정하고 있으므로 이렇게 요소

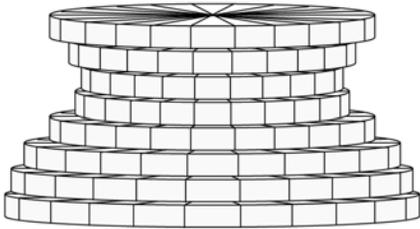


그림 1 부채꼴 모양의 요소로 이루어진 도자기 모델의 구조

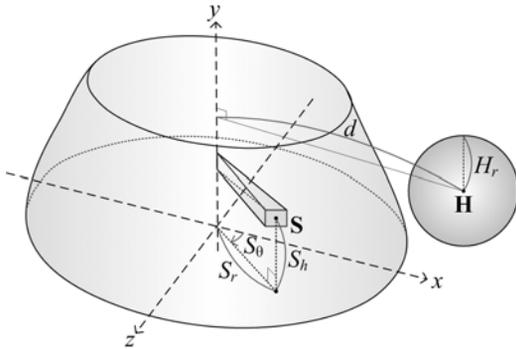


그림 2 부채꼴 요소법에서 사용하는 좌표축과 기호의 정의

의 끝 점 만으로도 충분한 모델링이 가능하다. HIP의 위치는 벡터 H , 반지름은 H_r , y 축과의 거리는 d 로 표현한다. 직교 좌표계에서 위치를 표현할 때에는 $X = (X_x, X_y, X_z)$ 와 같이 표현한다. 이 절에서 설명한 기호와 표현 방법은 앞으로 논문 전체에 걸쳐서 추가적인 설명 없이 사용하도록 한다.

4. 충돌 감지

부채꼴 요소법에서는 안에서 바깥 방향으로 변형(넙힘 모드)할 수도 있고, 밖에서 안쪽 방향으로 변형(좁힘 모드)할 수도 있다. 큰 차이는 없으나 각각의 경우에 사용하는 수식이 조금씩 달라지므로 먼저 좁힘 모드에 대해 자세히 설명한 후, 넙힘 모드에서 달라지는 점을 간단히 언급하도록 한다.

충돌 감지는 먼저 HIP와 충돌할 가능성이 있는 요소를 빠르게 찾아 내고, 찾아낸 요소에 적용할 충돌 감지 방법을 결정 한 후, 실제 정확한 충돌 깊이를 계산하는 순서로 이루어진다.

충돌 가능성이 있는 요소는 그림 3과 같이 수직 방향과 수평 방향으로 나누어 찾는다. 수직 방향으로는 조건 (1)을 만족하는 요소를 찾고, 수평 방향으로는 그림 3의 θ 가 충분히 작다고 가정하면 H_r 이 호 HP의 길이와 근사하므로 조건 (2)를 만족하는 요소를 찾으면 된다. 실제 구현에서는 모든 요소가 θ 와 h 방향으로 균등한

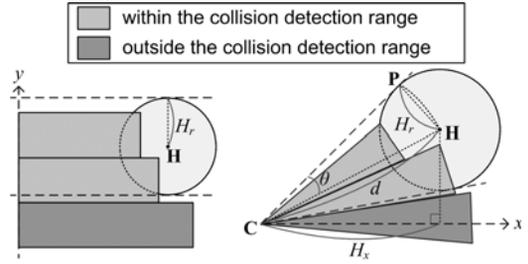


그림 3 수직 방향과(좌) 수평 방향에서(우) 충돌 가능성이 있는 요소를 찾는 방법

간격으로 배치되어 있으므로 조건 (1)과 (2)를 요소의 인덱스 범위로 변환하여 계산하면 $O(1)$ 시간에 충돌 가능성이 있는 요소를 모두 찾을 수 있다.

$$S_h^i \in [H_y - H_r, H_y + H_r] \quad (1)$$

$$S_\theta^i \in \left[\cos^{-1}\left(\frac{H_x}{d}\right) - \frac{H_r}{d}, \cos^{-1}\left(\frac{H_x}{d}\right) + \frac{H_r}{d} \right] \quad (2)$$

충돌 감지 방법에는 두 가지가 있다. HIP가 요소의 끝 점에서부터 충돌하는 일반적인 경우를 점 충돌 감지, HIP가 요소의 측면에서부터 충돌하는 경우를 선 충돌 감지라고 한다. 그림 4에서 볼 수 있듯이 점 충돌이 일어나면 충돌한 요소는 충돌 방향과 같은 방향으로 변형되지만, 선 충돌이 일어나면 충돌 방향과 수직 방향으로 변형되게 된다. 그림 5에서와 같이 S_r 이 d 보다 작으면 점 충돌 감지를 수행하며, 반대의 경우 선 충돌 감지를 수행한다. 즉, 그림 5의 경우 요소 S^1 은 점 충돌, 요소 S^2 는 선 충돌로 분류한다.

충돌 감지 방법을 정했으면 그에 따라 각 요소의 정확한 충돌 깊이를 계산한다. 그림 5는 현재 충돌 감지 중인 요소의 y 좌표에서 y 축에 수직인 평면으로 잘라 그 단면을 위에서 내려다 본 것이다. 여기서 중심축인 y 축과 평면이 만나는 점을 C , 잘린 HIP 단면의 반지름을 r 이라고 한다. 각 요소에서 C 와 S 를 연결하는 직선을 그으면, 이 직선은 HIP의 단면과 두 점에서 만나게 된다. 두 교점 중 C 에 더 가까운 점을 E_1 이라 하고 더 멀

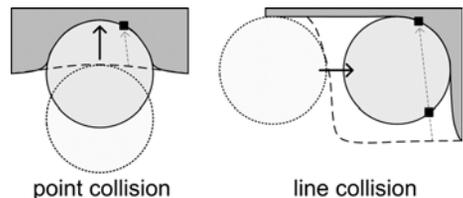


그림 4 점 충돌 감지와(좌) 선 충돌 감지(우)에서 HIP의 충돌 방향(진한 실선 화살표)과 요소의 변형 방향(얇은 점선 화살표)

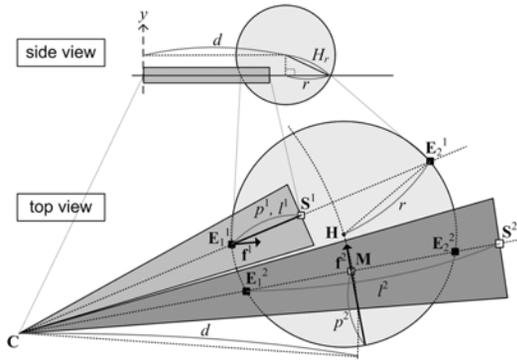


그림 5 점 충돌과 선 충돌 처리 알고리즘

리 있는 점을 E_2 라고 한다. 그림 5에서 S^1 과 같은 점 충돌 감지의 경우 C에서 E_1 사이의 거리와 요소의 길이 S_r 을 비교하여 충돌을 판정할 수 있고, S^2 와 같은 선 충돌 감지의 경우 E_1 과 E_2 사이의 중점인 M이 HIP의 단면 안에 들어가면 충돌한 것으로 본다. 즉, (3)을 이용하여 충돌 깊이 p 를 계산하고 p 가 0보다 크면 충돌한 것으로 판정한다.

$$p = \begin{cases} S_r - \|\mathbf{E}_1 - \mathbf{C}\| & \text{(for point collision)} \\ r - \|\mathbf{H} - \mathbf{M}\| & \text{(for line collision)} \end{cases} \quad (3)$$

확장 모드에서 충돌 감지를 할 경우 충돌 감지 방법을 결정하는 방법과 충돌 깊이를 계산하는 방법이 조금씩 달라진다. 충돌 감지 방법을 결정할 때는 좁힘 모드와 반대로 S_r 이 d 보다 크면 점 충돌로 판정하고 그렇지 않을 때는 선 충돌로 판정한다. 충돌 깊이를 계산하는 식은 선 충돌의 경우 좁힘 모드와 동일하며, 점 충돌의 경우 (4)를 이용하여 계산한다.

$$p = \|\mathbf{E}_2 - \mathbf{C}\| - S_r \quad (4)$$

5. 충돌 반응

이 절에서는 HIP와 충돌한 요소들을 처리하는 알고리즘을 다룬다. 충돌 반응 처리는 충돌한 각 요소의 길이를 변형하는 부분, 충돌에 의해 발생한 힘을 인접 요소로 전달하는 부분, 사용자에게 전달할 힘을 계산하는 부분의 세 부분으로 나눌 수 있다.

5.1 요소의 길이 변형

HIP와 충돌한 요소의 길이 변형은 요소에 작용한 변형력이 요소의 점성과 시간에 대한 요소 변형량과 비례한다는 간단한 물리적인 법칙을 이용한다. 만약 요소 S가 HIP와 충돌한다면 그림 5에서 교점 E_1 까지 요소의 길이를 줄여야 한다. 즉, 변형하여야 할 길이를 그림 5와 같이 l 이라고 하면 l 은 요소에 작용한 변형력과 비례한다고 볼 수 있다.

$$l = S_r - \|\mathbf{E}_1 - \mathbf{C}\| \quad (5)$$

$$\Delta S_r = -\frac{l}{c_\eta} \quad (6)$$

그러므로 l 은 (5)를 통해 계산할 수 있으며, 단위 시간당 요소의 변형량 ΔS_r 은 요소의 점성을 의미하는 계수 c_η 에 대해 (6)으로 계산할 수 있다.

5.2 인접 요소로 힘 전달

보다 현실적인 도자기 변형을 렌더링하기 위해서는 반드시 요소간의 상호작용을 고려해야 한다. 요소간 상호작용에는 여러 가지가 있지만 부채꼴 요소법에서는 내부가 비어있는 모양의 도자기를 가정하므로 전체적인 상호작용의 영향은 크지 않다. 따라서 HIP와 충돌에 의한 힘이 가까운 인접 요소로 전달되는 상호작용만 고려하였다.

중복된 계산을 줄여 계산 속도를 빠르게 하기 위해 먼저 HIP와 충돌한 영역의 경계에 있는 요소를 찾는다. 충돌한 영역이 볼록하다고(convex) 가정했을 때 경계에 있는 요소는 그림 6과 같이 구할 수 있다. (1), (3)과 같이 y 축 방향으로 첫 줄과 마지막 줄에서는 충돌 요소 전부를 찾고, (2)와 같이 중간에 있는 줄에서는 양 끝의 충돌 요소를 찾으면 쉽게 경계에 있는 요소를 찾을 수 있다. 4절에서 설명한 충돌 감지는 각 요소의 인덱스 순서로 이루어지므로 충돌한 영역의 경계에 있는 요소는 추가적인 계산 없이 충돌 감지 단계에서 바로 찾아 낼 수 있다. 다만, 충돌한 영역이 볼록하지 않을 경우 몇 개의 경계 요소가 빠질 수도 있으나 구 형태의 HIP를 사용하므로 현실적으로 이런 경우는 거의 발생하지 않는다.

실제 도자기 반죽에서는 충돌에 의해 발생한 힘이 원형으로 전달되지만 원형 범위 안의 요소를 찾는 것은 많은 계산이 필요하므로 속도를 높이기 위해 사각형 형태로 힘을 전달하도록 하였다. 충돌한 요소는 그림 7과 같이 사각형 형태로 힘을 전달하며 이 때 전달하는 힘은 거리가 멀어질수록 작아지게 된다. 이와 같이 사각형

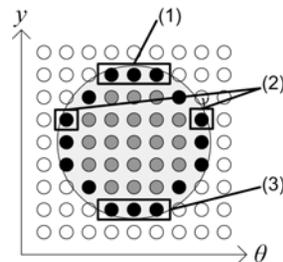


그림 6 y 축 방향으로 맨 위와 아랫줄 (1, 3) 그리고 그 사이의 줄에서 (2) 경계 요소를 찾는 방법

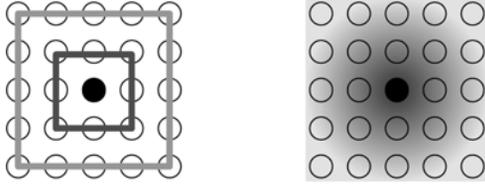


그림 7 인접 요소로 힘 전달. 짙은 색깔은 보다 큰 힘이 전달되었음을 나타낸다. 이것은 선으로 그릴 수도 있고(좌), 음영을 넣어 나타낼 수도 있다(우).

$$\mathbf{f}^i = \begin{cases} \frac{\mathbf{H} - \mathbf{E}_i}{\|\mathbf{H} - \mathbf{E}_i\|} & \text{(for point collision)} \\ \frac{\mathbf{H} - \mathbf{M}^i}{\|\mathbf{H} - \mathbf{M}^i\|} & \text{(for line collision)} \end{cases} \quad (9)$$

$$\mathbf{F}_v^i = c_v p^i \mathbf{f}^i \quad (10)$$

물체의 회전에 의해 발생하는 마찰력은 c_r 을 운동 마찰 계수, \mathbf{t} 를 회전 방향의 단위 벡터라 할 때 (11)과 같이 최대 충돌 깊이를 사용하여 계산할 수 있다.

$$\mathbf{F}_r = c_r \max_i(p^i) \mathbf{t} \quad (11)$$

확장 모드에서는 (9)에서 E_1 만 E_2 로 바꾸고 나머지는 모두 동일한 방법으로 사용자에게 전달하는 힘을 계산할 수 있다.

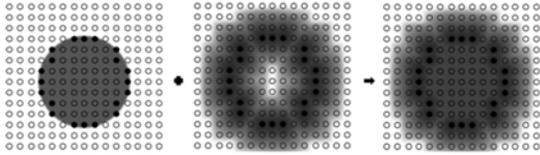


그림 8 사각형 모양 힘 전달에 의해 인접 요소가 자연스럽게 변형되는 단계

으로 힘이 전달되는 형태를 단순화하였지만 모든 경계 요소에서 인접 요소로 전달한 힘이 누적된 최종 결과를 보면 그림 8과 같이 자연스러운 모양이 되는 것을 확인할 수 있다.

힘이 전달된 사각형 범위 안에 있는 요소는 전달된 힘에 의해 추가로 변형이 일어나게 되며 이때 힘 전달에 의한 추가 변형량 l_p 는 (7)을 이용해 계산할 수 있다. 여기서 ΔS_r 은 (6)에서 계산한 충돌 요소(그림 7의 검은색 점)의 단위 시간당 변형량, c_r 는 힘 전달 계수, δ 는 인접 요소간 거리이다. 이 때, δ 는 θ 방향의 해상도인 u_θ 요소의 길이 S_r , 사각형의 일련 번호 n 을 통해 근사적으로 계산하게 된다.

$$l_p = \frac{\Delta S_r}{(c_r \delta)^2 + 1} = \frac{\Delta S_r}{(c_r u_\theta S_r n)^2 + 1} \quad (7)$$

5.3 사용자에게 전달하는 힘

사용자에게 전달하는 힘 \mathbf{F} 는 (8)과 같이 HIP와 요소의 충돌에 의해 발생하는 저항력 \mathbf{F}_v^i 와 물체의 회전에 의해 발생하는 마찰력 \mathbf{F}_r 을 모두 더하여 계산한다.

$$\mathbf{F} = \sum_i \mathbf{F}_v^i + \mathbf{F}_r \quad (8)$$

HIP와 충돌한 각 요소는 충돌 반대 방향으로 저항력을 준다. 충돌 반대 방향의 단위 방향 벡터 \mathbf{f}_i 는 (9)를 이용해 계산할 수 있다(그림 5 참고). 그러면 각 요소에서 발생하는 저항력 \mathbf{F}_v^i 는 요소의 저항 계수가 c_v 일 때 변형 길이와 힘 사이의 비례 관계를 이용하여 (10)과 같이 계산할 수 있다.

6. GPU 기반의 햅틱 모델과 시각 모델 동기화

4절과 5절에서 설명한 바와 같이 부채꼴 요소법은 빠른 충돌 감지 및 반응 처리를 하기 위해 원형으로 대칭적인 도자기 모양의 특징을 활용하여 원통 좌표계에서 모든 계산을 수행한다. 하지만 실제 화면에 도자기 모델을 출력하기 위해서는 모든 요소를 직교 좌표계로 변환하고 법선 벡터를 계산하여 시각 도자기 모델로 변환하는 과정을 거쳐야만 한다. 일반적으로 부채꼴 요소법에서 도자기 모델은 십만 개 이상의 요소로 이루어지고 햅틱 프레임마다 보통 수백 개에서 많게는 천 개 이상의 요소가 HIP와 충돌하기 때문에 이와 같은 햅틱과 시각 모델 동기화 과정에는 상당한 시간이 소요되게 된다. 우리는 [22,23]와 비슷한 방식으로 GPU의 병렬 처리 기능을 활용하여 동기화 과정이 빠르게 이루어질 수 있도록 하였다(그림 9 참고).

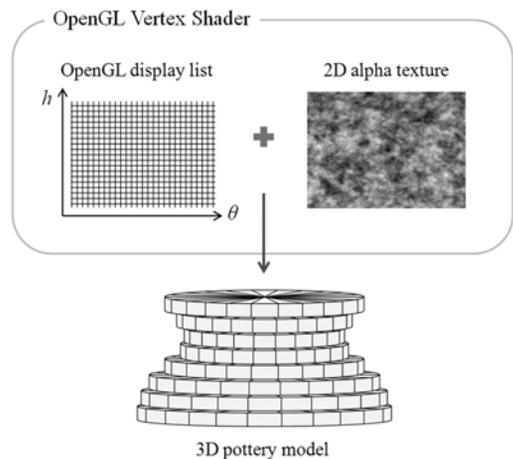


그림 9 GPU 기반의 햅틱 모델과 시각 모델 동기화

3절에서 언급한 것과 같이 부채꼴 요소법에서 각 요소의 좌표 중, 요소의 길이를 나타내는 S_r 만 변형되고 S_θ 와 S_h 는 변하지 않는다. 또한 모든 요소는 θ 와 h 방향으로 균등하게 배치되어 있다. 따라서 S_θ 와 S_h 는 OpenGL display list를 통해 2차원 격자 형태로 메모리에 고정시키고 S_r 은 2D alpha texture의 alpha 값을 이용해 저장할 수 있다. 그러면 OpenGL vertex shader는 display list와 alpha texture에서 각각의 값을 읽어와 직교 좌표계로의 좌표 변환과 법선 벡터 계산을 병렬 처리를 통해 빠르게 수행한다.

실제 좌표 변환 코드의 예를 들어보자. Display list와 alpha texture에는 정규화된 S_r , S_θ , S_h 값을 저장하므로 texColor.a , gl_Vertex.x , gl_Vertex.y 를 적절히 스케일링 한 후 다음과 같은 코드를 통해 각 요소의 좌표를 직교 좌표계로 변환할 수 있다.

```
v = vec3(texColor.a * cos(gl_Vertex.x),
        gl_Vertex.y,
        texColor.a * sin(gl_Vertex.x));
```

7. 모델링 시스템의 구현

이제까지 설명한 부채꼴 요소법을 바탕으로 그림 10과 같이 실제 햅틱 도자기 모델링 시스템을 구현하였다. OpenHaptics Library를 사용하여 구현하였기 때문에 PHANToM Omni를 비롯한 Sensable사의 PHANToM 계열의 모든 햅틱 장비를 사용할 수 있다. GPU 연산을 사용하는 버전과 사용하지 않는 두 가지 버전을 모두 제작하여 그래픽 카드가 따로 장착되어 있지 않은 시스템에서도 사용이 가능하다. 또한 처음 프로그램을 실행할 때, 18,000개의 요소로 이루어진 저해상도의 모델부터 576,000개의 요소로 이루어진 고해상도의 모델까지 총 8 단계의 도자기 해상도 레벨을 선택할 수 있으므로 자신의 컴퓨터 성능에 맞는 활용이 가능하다.

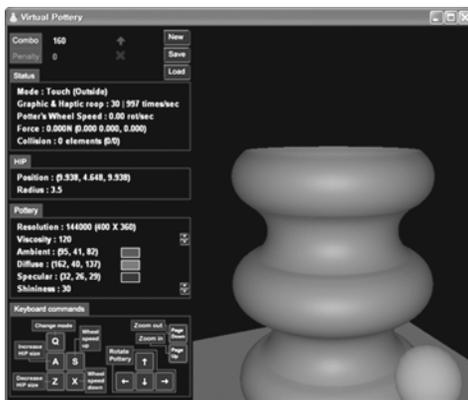


그림 10 구현한 햅틱 도자기 모델링 E-Learning 시스템

다음 절의 성능 평가 결과를 보면 자세히 알 수 있듯이 매우 효율적인 부채꼴 요소법을 사용하여 구현하였기 때문에 비교적 낮은 성능의 PC에서도 도자기 모델의 실시간 햅틱 렌더링이 충분히 가능하다. 실제 Intel Pentium 4 3.2GHz CPU, 2GB RAM, NVIDIA Geforce 8800 GTS 그래픽 카드를 장착한 시스템에서 288,000개 요소를 가진 도자기 모델을 부드럽게 렌더링할 수 있었으며, Intel Pentium M 1.7GHz, 512MB RAM, NVIDIA Geforce go5200 그래픽 카드를 장착한 노트북 환경에서도 18,000~36,000개 요소를 가진 도자기 모델의 실시간 햅틱 렌더링이 가능했다.

사용자는 햅틱 장비를 잡고 가상환경에서 도자기를 성형한다. 키보드를 이용해 도자기를 보는 시점을 이동할 수 있으며, HIP의 크기도 조절할 수 있다. 도자기 모델의 점성 계수도 조절할 수가 있어서 무른 반죽과 단단한 반죽의 느낌을 선택하여 도자기 성형을 할 수 있다. 또한 도자기에 자기가 원하는 색깔을 입힐 수 있어서 더 다양한 도자기의 제작이 가능하다. 그리고 아이들이 흥미를 가지고 올바른 도자기 성형을 할 수 있도록 간단한 점수 시스템도 도입하였다. 부드럽게 도자기를 성형하면 계속 누적해서 콤보(combo) 점수가 올라가며, 순간적으로 무리한 힘을 주거나 잘못된 동작을 할 때에는 페널티(penalty) 점수가 올라가 스스로 결과를 확인하며 도자기 성형 체험을 할 수 있게 하였다.

그림 11은 다양한 형태의 도자기를 모델링 해 본 결과이다. 왼쪽 위는 대접과 같이 넓은 형태, 오른쪽 위는 상감 청자와 같은 형태, 왼쪽 아래는 비대칭 적인 무늬를 넣은 병, 마지막으로 오른쪽 아래는 시스템의 유연성

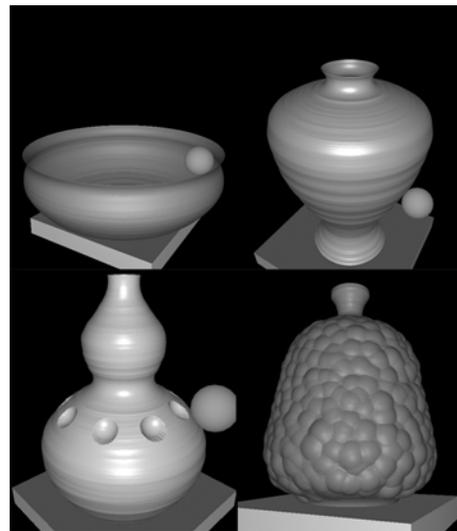


그림 11 다양한 도자기 모델링 결과

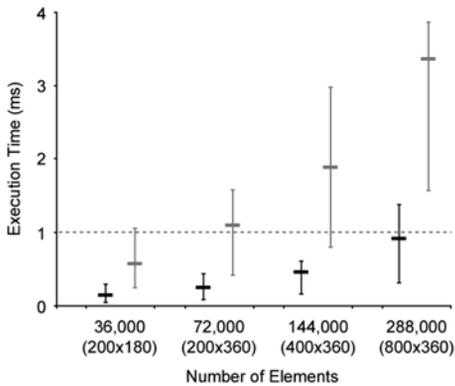


그림 12 다양한 해상도에서 GPU 기반의 동기화를 적용한 시스템과(진한 검은색) CPU로만 계산을 수행한 시스템의(열은 회색) 렌더링 속도 측정 결과

을 보여주기 위해 조금 극단적인 모양을 만들어 본 것이다. 이 예를 통해 매우 다양한 형태의 도자기를 원하는 대로 쉽게 디자인할 수 있음을 알 수 있다.

8. 성능 측정

8.1 렌더링 속도 측정

부채꼴 요소법을 통해 구현한 햅틱 도자기 모델링 시스템의 효율성을 검증하기 위해 렌더링 속도를 측정하였다. Pentium 4 3.2GHz CPU, 2GB RAM, NVIDIA Geforce 8800 GTS 그래픽 카드를 장착한 일반적인 성능의 데스크탑 시스템에서 Sensable사의 PHANToM Omni 장비를 사용하여 테스트를 수행하였다.

그림 12는 GPU 기반의 동기화를 적용한 것과 적용하지 않은 햅틱 도자기 모델링 시스템을 사용해 4가지 해상도의 도자기를 생성했을 때 1회 렌더링에 걸린 시간을 보여준다. 여기서 도자기의 해상도는 '총 요소 수 (수직 방향의 요소 수 × 수평 방향의 요소 수)'와 같은 형식으로 표시하였다. 일반적으로 정해진 크기의 HIP에 대해 수직이나 수평 방향의 요소 수가 증가함에 따라 계산 속도도 증가한다. 또한 계산에 걸리는 시간은 중심축과 HIP 사이의 거리에 반비례하므로 중심축에 HIP가 가까워 질수록 속도는 느려진다. 그림 12에 구간을 사용하여 최대값과 최소값을 표시하고, 중간에 진한 선으로 평균값을 그려 이와 같은 변화를 표현하였다. 1 ms를 기준으로 그린 점선은 안정적인 1 kHz 햅틱 렌더링을 위해 필요한 렌더링 속도를 나타낸다.

평균값을 보면 GPU 기반의 동기화를 적용했을 때, 테스트한 환경에서 288,000개의 요소로 이루어진 도자기 모델의 1 kHz 실시간 햅틱 렌더링이 가능하며, 144,000개 이하의 요소로 이루어진 모델은 매우 안정적으로 처

리할 수 있음을 알 수 있다. 288,000개 요소로 이루어진 도자기의 해상도는 7.2 cm 높이의 모델을 수직 방향 0.09 mm, 수평 방향 1° 간격으로 뿔뿔이 채운 것으로 PHANToM Omni의 위치 정확도가 0.055 mm인 것을 고려하면 매우 조밀함을 알 수 있다.

GPU 기반의 동기화를 적용한 것과 하지 않은 것의 결과를 비교하면, CPU만 사용하여 계산한 경우 평균적으로 3.9배의 계산 시간이 더 걸려, GPU의 빠른 병렬 처리를 좌표 변환과 법선 벡터 계산에 활용한 것이 상당한 도움이 되는 것을 알 수 있다. CPU만 사용해 계산했을 경우에는 테스트 환경에서 72,000개의 요소를 가진 도자기 모델의 실시간 햅틱 렌더링이 가능하며, 36,000개의 요소를 가진 도자기 모델의 경우 안정적인 처리를 할 수 있음을 알 수 있다.

8.2 인접 요소로 힘 전달의 효과

5.2절에서 설명했듯이 부채꼴 요소법은 복잡한 계산에 비해 그 영향이 크지 않은 전체적인 상호 작용 계산은 무시하고 인접 요소에 대한 상호 작용만 고려하여 계산 속도를 최대한 높이면서도 현실적인 모델링을 할 수 있도록 하였다. 실제 인접 요소에 대한 상호 작용을 적용한 것이 적용하지 않은 것에 비해 어느 정도 도자기 모델의 현실성을 높여주는지 비교해 보았다.

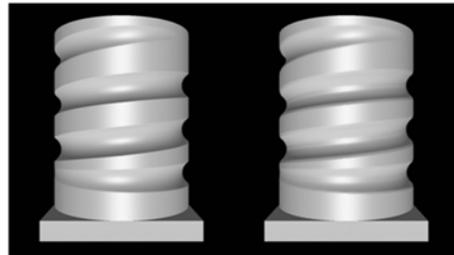


그림 13 인접 요소로 힘 전달 적용하지 않은 결과와(좌) 적용한 결과의(우) 비교

그림 13의 좌측 그림은 인접 요소간 상호 작용을 고려하지 않은 것이고, 우측 그림은 이를 구현하여 적용한 것이다. 좌측 그림에는 비현실적으로 매끈한 표면과 날카로운 모서리가 많이 나타나지만 우측 그림에는 보다 현실적인 도자기 표면의 질감이 부드럽게 잘 표현되어 있음을 알 수 있다.

8.3 햅틱 렌더링의 안정성

실제 부채꼴 요소법을 구현하면 이산적인 모델의 특성으로 인해 현재 렌더링 시점에서 충돌했던 요소들 중 일부가 바로 다음 렌더링 시점에는 충돌하지 않고 다음 단계에서는 다시 충돌하는 것이 반복되는 현상이 발생한다. 때문에 그림 14에서 평균 충돌 깊이와 이에 따른

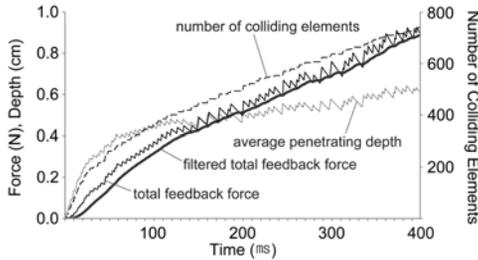


그림 14 햅틱 렌더링의 안정성 측정 결과

사용자에게 전달하는 힘이 불안정하게 떠는 것을 볼 수 있다. 우리는 10Hz 컷 오프 주파수를 가진 Butterworth 저역 차단 필터를 이용해 이 문제를 해결하였으며 그 결과 그림 14에서 볼 수 있는 것처럼 부드럽고 안정적인 힘을 렌더링 할 수 있었다.

8.4 사용자 평가

우리가 구현한 햅틱 도자기 모델링 E-Learning 시스템에 대한 통계적인 사용자 분석은 수행하지 못했으나 비공식적인 사용자 평가를 통해 긍정적인 반응을 얻을 수 있었다. 지난 2년 정도 꾸준히 연구실에 견학 차 방문한 수십 명의 중고등학생들에게 햅틱 도자기 모델링 시스템을 체험할 수 있는 기회를 제공해 주었는데 대부분의 학생들이 상당한 흥미와 재미를 느끼며 도자기 성형에 임하는 것을 관찰할 수 있었다.

9. 결론

본 논문에서는 가상 환경에서 도자기 성형을 체험할 수 있는 시스템을 효율적으로 구현하기 위해 부채꼴 요소법이라는 새로운 렌더링 방법을 제안하였다. 부채꼴 요소법은 원통형으로 대칭적인 도자기 형태의 특징을 활용하여 현실적이고 효율적인 실시간 햅틱 도자기 렌더링이 가능하도록 하였다. 제시한 알고리즘을 이용해 실제 E-Learning 시스템을 구축하였으며 다양한 관점에서 부채꼴 요소법을 이용한 햅틱 도자기 모델링 시스템의 우수성을 검증하였다.

지금 HIP의 모양이 구로 한정되어 있어 평평한 면을 가진 도자기를 만드는 것과 같은 성형에 어려움이 있다. 차후에 HIP의 모양을 다양하게 바꿔 도자기를 성형할 수 있게 확장한다면 보다 다양한 형태의 도자기를 쉽게 제작할 수 있을 것이다. 또한 texture등을 잘 활용한다면 완성된 도자기 표면에 그림을 그리는 것과 같은 확장된 도자기 체험 환경도 제공할 수 있게 될 것이다. 처음에 소개하였던 것처럼 다양한 역사, 문화, 교육적인 콘텐츠와 우리가 제작한 햅틱 도자기 모델링 E-Learning 시스템을 잘 결합하면 실제 교육 현장에서 활용할 수 있는 유용한 기술이 될 것이다.

참고 문헌

- [1] K. Kameyama, "Virtual Clay Modeling System," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp.197-200, 1997.
- [2] K. Korida, H. Nishino, and K. Utsumiya, "An Interactive 3D Interface for a Virtual Ceramic Art Work Environment," in *Proceedings of the International Conference on Virtual Systems and MultiMedia*, pp.227-234, 1997.
- [3] T. Massie, "A tangible goal for 3D modeling," *IEEE Computer Graphics and Applications*, vol.18, pp.62-65, 1998.
- [4] Y. Chai, G. R. Luecke, and J. C. Edwards, "Virtual Clay Modeling Using the ISU Exoskeleton," in *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp.76-80, 1998.
- [5] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk, "Virtual Clay: A Real-Time Sculpting System with Haptic Toolkits," in *Proceedings of the Symposium on Interactive 3D Graphics*, pp.179-190, 2001.
- [6] G. Han, J. Kim, and S. Choi, "Virtual Pottery Modeling with Force Feedback Using Cylindrical Element Method," in *Proceedings of the International Conference on Next-Generation Computing*, pp. 125-129, 2007.
- [7] T. W. Sederberg and S. R. Parry, "Free-Form Deformation of Solid Geometric Models," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 151-160, 1986.
- [8] S. Coquillart, "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," in *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 187-196, 1990.
- [9] S. Coquillart and P. Jancene, "Animated Free-Form Deformation: An Interactive Animation Technique," in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 23-26, 1991.
- [10] W. Hsu, J. Hughes, and H. Kaufman, "Direct Manipulation of Free-Form Deformations," in *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 177-184, 1992.
- [11] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Z. Zhu, *The Finite Element Method*, Elsevier/Butterworth-Heinemann, 2005.
- [12] C. Mendoza, C. Laugier, I. R. A. Zirst, and F. Saint Martin, "Simulating Cutting in Surgery Applications Using Haptics and Finite Element Models," in *Proceedings of IEEE International Conference on Virtual Reality*, pp.295-296, 2003.
- [13] F. Conti, O. Khatib, and C. Baur, "Interactive Rendering of Deformable Objects Based on a

- Filling Sphere Modeling Approach," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp.3716-3721, 2003.
- [14] D. L. James and D. K. Pai, "A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics," in *Proceedings of International Conference on Computer Graphics and Interactive Techniques*, pp.141-153, 2005.
- [15] I. F. Costa and R. Balaniuk, "LEM - An Approach for Real Time Physically Based Soft Tissue Simulation," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2337-2343, 2001.
- [16] K. Sundaraj, C. Laugier, and I. F. Costa, "An Approach to LEM Modeling: Construction, Collision Detection and Dynamic Simulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2196-2201, 2001.
- [17] S.Y. Kim, J. Park, D. S. Kwon, "Real-time Haptic Rendering of High-Resolution Volumetric Deformable Object in a Collaborative Virtual Environment," *Advanced Robotics*, vol.19, no.9, pp.951-975, 2005.
- [18] S. Jun, J. Choi, and M. Cho, "Physics-based s-Adaptive Haptic Simulation for Deformable Object," in *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp.477-483, 2006.
- [19] O. R. Astley and V. Hayward, "Multirate Haptic Simulation Achieved by Coupling Finite Element Meshes Through Norton Equivalents," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp.989-994, 1998.
- [20] J. Kim, S. De, and M. A. Srinivasan, "An Integral Equation Based Multiresolution Modeling Scheme for Multimodal Medical Simulations," in *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp.221-228, 2003.
- [21] G. Debunne, M. Desbrun, M. P. Cani, and A. H. Barr, "Dynamic Real-Time Deformations Using Space & Time Adaptive Sampling," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pp.31-36, 2001.
- [22] M. de Pascale, G. de Pascale, D. Prattichizzo, and F. Barbagli, "A GPU-friendly method for haptic and graphic rendering of deformable objects," in *Proceedings of EuroHaptics*, pp.44-51, 2004.
- [23] Cristian J. Luciano, P. Pat Banerjee, and Silvio H. R. Rizzi, "GPU-based elastic-object deformation for enhancement of existing haptic applications," in *Proceedings of the Annual IEEE Conference on Automation Science and Engineering*, pp.146-151, 2007.



이재봉

2008년 포항공과대학교 수학과 졸업(학사). 2010년 포항공과대학교 컴퓨터공학과 졸업(석사). 2010년~현재 LG전자 DS연구소 연구원



한갑중

2006년 포항공과대학교 컴퓨터공학과 졸업(학사). 2006년~현재 포항공과대학교 컴퓨터공학과 석박사 통합과정 재학중



최승문

1995년 서울대학교 제어계측공학과 졸업(학사). 1997년 서울대학교 제어계측공학과 졸업(석사). 2003년 퍼듀대학교 전기컴퓨터공학부 졸업(박사). 2003년~2005년 퍼듀대학교 박사후 연구원. 2005년 포항공과대학교 컴퓨터공학과 조교수. 2010년~현재 포항공과대학교 컴퓨터공학과 부교수