

점 기반 곡면으로 이루어진 복잡한 가상 물체와의 효율적인 햅틱 상호작용

Efficient Haptic Interaction for Highly Complex Object Generated by Point-based Surfaces

이범찬, Beom-Chan Lee*, 김덕봉, Duck-Bong Kim**, 박혜신, Hyeslin Park*,
김종필, Jong-Phil Kim*, 이관행, Kwan Heng Lee**, 류제하, Jeha Ryu*
*광주과학기술원 정보기전공학부 인간-기계-컴퓨터 인터페이스 연구실,
**광주과학기술원 지능 설계 그래픽스 연구실

요약 본 논문은 연결정보(connectivity) 및 미리 계산된 계층적 데이터 구조(hierarchical data structure)를 이용하지 않는 그래픽 및 햅틱 렌더링 알고리즘을 제안한다. 제안된 알고리즘은 점 기반 그래픽 표현(point-based graphic representation) 기법을 이용하여 3 차원 자유 곡면을 생성한다. 생성된 점 기반 곡면 물체와의 햅틱 상호작용을 위해 그래픽 하드웨어(GPU)에 접근하여 점 기반 곡면에서 생성된 깊이 이미지(depth image)를 이용하여 햅틱 상호작용에 필수 요소인 충돌검출(collision detection) 및 반력 연산(contact force computation)을 수행한다.

핵심어: Haptic rendering, Point-based rendering

1. 서론

햅틱(Haptic)은 고대 그리스어인 'haptikos/haptesthai'라는 '만지다'의 뜻으로부터 유래되었으며, 햅틱 감각은 손을 사용하여 느끼는 모든 지각을 의미하는 용어로 정의되어 사용되었으나, 최근 인체의 모든 촉각 기관을 사용하여 느끼는 지각으로 의미가 넓혀지고 있다. 햅틱 인터페이스를 통해 제시되는 역/촉각 정보는 다양한 환경을 실감 있게 인지하도록 도와주며, 사용자의 몰입감을 증가시켜 환경내의 객체들을 효과적으로 느끼고 조작할 수 있도록 도와주기 때문에 그래픽 렌더링과 함께 의료, 군사, 산업, 교육 및 게임 등에 적용되어 발전해왔다.

다양한 역/촉각제시 알고리즘 가운데 형상기반 햅틱 렌더링 알고리즘(geometry dependent haptic rendering algorithm [1])은 객체의 기하학적 정보를 사용자에게 역감을 통해 제공하는 알고리즘으로 충돌 검출과 반력 계산 등의 과정을 포함한다. 이것은 객체의 형태를 기술하는 방법에 따라 크게 곡면(surface) 기반 방법[2-6]과 볼륨(volume) 기반 방법[7-10]으로 분류될 수 있다. 전자는 삼각 메쉬(triangular mesh), 다각형 (polygon), 음함수(implicit) 곡면 등으로 표현된 곡면을 사용하며, 후자는 복셀(voxel) 등으로 표현된 볼륨 데이터를 사용한다.

최근 3차원 물체의 데이터를 획득하는 기술이 발전함에 따라 기하정보뿐만 아니라 색상 정보를 빠르고 정확하게 획득할 수 있게 되었다. 획득된 데이터는 주로 연결정보가 없으며 불규칙하게 분배된 점 형태로 표현이 된다. 따라서 기

준의 햅틱 상호작용 알고리즘에 적용하기 위해서는 연결정보 생성 또는 계층적 데이터 구조를 미리 구성해야 한다. 그러나 이러한 Pre-processing은 데이터의 용량이 많은 데이터에 대해서는 시간이 오래 소요되며 햅틱 렌더링이 polygon 수에 비례하는 방법론 같은 경우는 햅틱 상호작용 성능이 떨어질 가능성이 있다.

최근 이러한 단점을 극복하기 위해 Proxy graph algorithm[8]이 제안되었는데 이 알고리즘은 대용량 지형도(topology map) 데이터와의 햅틱 상호작용 시 발생하는 polygon을 탐색을 최적화 하였다. 또한 K. Potter et al[9]은 대용량 지형도를 부분적인 bi-linear patches를 이용하여 탐색 시간을 줄이는 노력을 하였다. 그러나 앞의 두 방법론은 데이터의 형태가 규칙적으로 분배된 점 데이터에 효과적이며 그래픽 햅틱 렌더링을 위해 미리 구성된 연결정보가 요구된다. 따라서 미리 계산된 연결정보를 필요로 하지 않는 햅틱 렌더링 알고리즘이[10] 최근 제안이 되었다. 이 알고리즘은 2.5D Depth 카메라로 획득된 물체 데이터를 연결 정보 없이 햅틱 상호작용이 가능하도록 제안되었다.

본 논문은 가상 물체의 연결 정보 및 계층적 데이터 구조가 요구되지 않는 햅틱 상호작용 알고리즘을 제안한다. 본 알고리즘은 불규칙하게 분배된 점으로 획득된 데이터를 시각, 촉각적으로 렌더링 하는 것으로, 그래픽 렌더링을 위해 점 기반 표현(point-based representation) 알고리즘을 제안하여 자유곡면을 생성하였으며 그래픽스 하드웨어 내부에서 스무딩(smoothing) 및 셰이딩(shading)을 수행함으로써 획득된 데이터를 사실적으로 시각화 하는 알고리즘을 개발하였다.

햅틱 상호작용에 있어서는 그래픽스 하드웨어를 이용하여 접촉하고자 하는 면의 깊이 정보를 이용하여 충돌 검출을 수행하였으며, LOMI(Local Occupancy Map Instance)[11]를 이용하여 접촉 힘을 계산한다.

본 논문은 다음과 같이 구성되었다. 시스템 및 알고리즘 개요를 2장에서 언급하고, 3장에서는 점 기반 시각화 알고리즘에 대해 자세히 기술할 것이다. 4장에서는 점 기반 곡면으로 이루어진 복잡한 가상 물체와의 햅틱 상호작용 알고리즘에 대해 기술하며, 본 논문에서 제안된 방법론의 결과 및 효율성을 5장에서 검증한다. 마지막으로 본 논문의 결론 및 향후 연구계획을 6장에 기술하였다.

2. 알고리즘 및 시스템 개요

제안된 알고리즘은 크게 두 부분으로 구성되었다. 3D 스캐너로 획득된 불규칙하게 분배된 점 데이터를 시각화 하는 그래픽 렌더링 알고리즘 부분과 촉각 상호작용을 위한 햅틱 렌더링 알고리즘으로 구성되어 있다. 그림 1은 알고리즘 및 시스템 개념도를 나타낸다.

불규칙하고 연결 정보가 없는 점 데이터를 가지고 시각화 하는 점 기반 표현 기법을 이용하여 그래픽 렌더링 알고리즘을 구성하였다. 3차원 공간상에 이산적(discrete)으로 분포되어 있는 점 데이터로부터 점 기반 곡면을 생성하기 위해서는 점의 좌표, 색상뿐만 아니라, 노말(normal) 및 크기(size of each point)를 가지고 있는 점 기반 표현 요소인 서피스 스플랫(surface splat)으로 변환 시켜 주어야 한다. 또한 각 서피스 스플랫간의 겹침에 의해 발생하는 시각적 문제점들을 해결하기 위해, 스플랫 간의 혼합(blending) 및 정규화(normalization)를 수행하였다.

햅틱 렌더링 알고리즘은 그래픽 렌더링 된 물체의 국부 깊이 정보를 그래픽 하드웨어(graphics hardware)의 깊이 버퍼를 참조하여 현재의 햅틱 포인트(Haptic Interaction Point)와 충돌이 발생을 하였는지 검색을 하고 LOMI를 구축하여 이동되는 햅틱 포인트에 상응하는 가상의 햅틱 포인트(Ideal Haptic Interaction Point)를 결정하여 힘을 계산하였다.

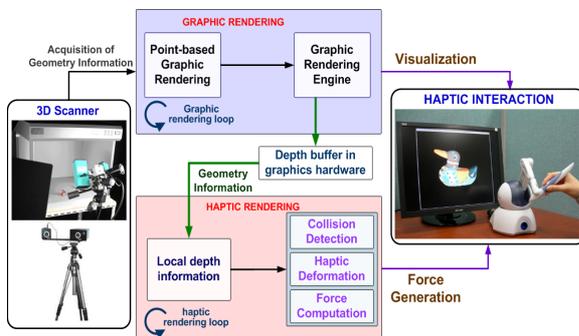


그림 1. 알고리즘 및 시스템 개념도

3. 점 기반 그래픽 시각화

3.1 점 기반 시각화 표현 요소: 서펠(surfel)

3차원 컬러 스캐너로 획득한 점 샘플을 이용하여 점 기반 자

유곡면(point-based surface)을 생성하기 위해서는 그림 2(b)에서 보는 것처럼 좌표, 색상, 노말, 반지름 정보를 가지고 있는 서펠(surfel)[13]을 만들어야 한다. 모델의 효과적인 음영효과(shading effect)를 나타내기 위해 노말 정보가 필요하고, 반지름 정보는 오브젝트 공간에 있는 점들이 이미 지 평면상에 투영시키면서 생길 수 있는 구멍을 피하기 위해서 필요하다. 점 샘플의 노말 값과 반지름 정보를 메쉬 연결 정보(mesh connectivity) 없이 구하기 위해 계층적 공간 분할(hierarchical space partitioning) 기법의 하나인 kd-tree를 이용하여 3D 공간상에 있는 점 데이터를 분할하였으며, 각 정점의 k-nearest 이웃 점을 찾고, 공분산 분석(Covariance analysis)을 이용해서 각 서펠(surfel)의 노말 값과 반지름 값을 구할 수 있다[13].

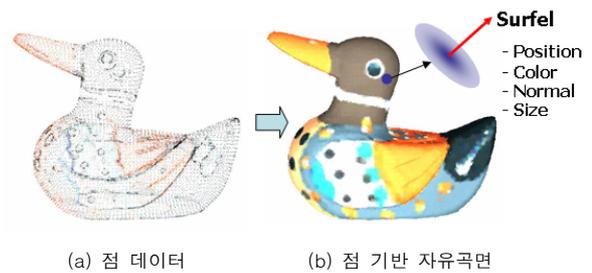


그림 2. 점 기반 자유곡면 생성을 위한 서피스 스플랫의 속성

3.2 GPU 기반의 서피스 스플래팅

그래픽 하드웨어를 기반으로 한 점 기반 자유곡면(point-based surface) 생성에 있어서 몇 가지 고려해야 할 사항은 다음과 같다. 첫 번째 서펠(surfel)의 크기와 형태를 결정해야 한다. 서펠(surfel)의 크기는 이미지 평면상에서 홀(hole free)을 피하기 위한 것이고, 서펠(surfel)의 형태는 이미지 평면상에서 서펠(surfel)이 얼마나 큰 타원형을 나타내는가에 따라 결정지어 진다.

서펠(surfel)의 크기와 형태를 결정한 후 그림 4와 같이 이웃 서펠(surfel)간의 겹치는 시각적 문제점(visual artifact)이 생기게 되는데, 보다 좋은 품질의 점 기반 자유곡면(point-based surface)을 생성하기 위해서는 겹치는 이웃 서펠(surfel) 들간에 블렌딩(blending) 하는 과정이 필요하다. 각 서펠(surfel)의 중심점으로부터 그 서펠(surfel)의 끝으로 갈수록 줄어드는 가우시안 커널(Gaussian kernel)을 이용하여 알파 블렌딩(blending)[13] 해주었다.

그림 3에서 보는 것처럼 한 시점(view point)으로부터 각 서펠(surfel) 들을 블렌딩 하기 위해서는 앞에 있는 서펠과 뒤에 가려지는 서펠들을 블렌딩을 해야 한다. 현재의 그래픽 하드웨어로는 한번에 ϵ -깊이 테스트[13]를 수행할 수가 없기 때문에, 두 번의 과정을 거친 후 점 기반 렌더링(point based rendering)을 한다.

첫 번째 과정에서는 모든 서펠의 깊이 정보가 깊이 버퍼에 저장된다. 그림 3에서 보는 것처럼 ϵ threshold 보다 작을 경우에만 저장이 되고, 그 값보다 크게 되면 깊이 버퍼에 저장하지 않게 된다. 따라서 한 픽셀에는 ϵ threshold 보다 안에 있는 서펠(surfel)만이 저장된다. 이 첫 번째 과정에서는 조명계산만 있을 뿐, 계산 속도를 위해서 알파 블렌딩은 하지 않는다. 두 번째 과정에서는 깊이 버퍼에 ϵ threshold

보다 작은 서펠(surfel)들의 깊이 정보 값만 저장되어 있기 때문에, 각 서펠(surfel)의 실질적인 라이팅 계산 및 알파 블렌딩이 이 과정에서 수행된다.

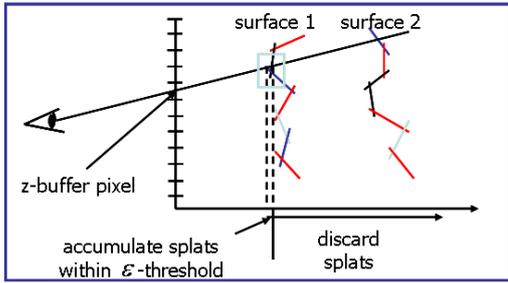
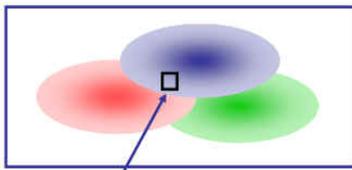


그림 3. ϵ -깊이 테스트

그림 4에서 보는 것처럼 이미지 공간상에 있는 각 픽셀들은 $\sum \alpha_i(rgb)_i$ 의 컬러값을 저장하게 된다. 하지만 각 픽셀에 가중치가 적용된 컬러 값의 합은 시각적 문제점(visual artifact)이 발생하게 되는데[13], 이러한 흠을 제거하기 위해 마지막 과정인 정규화(normalization) 과정이 필요로 한다. 정규화(normalization) 과정에서는 식 (1)과 같이 각 픽셀에 저장된 색상 값들을 각 색상 채널의 가중치의 합(weighted sum)으로 나누어 준다.

$$\sum_i \alpha_i(rgb)_i / \sum_i \alpha_i \quad (1)$$



$$c = \sum_i \alpha_i(rgb)_i / \sum_i \alpha_i$$

그림 4. 정규화 과정

그림 5는 비주얼 품질 비교를 나타낸다. 그림 5(a)는 첫 번째 과정인 서펠(surfel)의 형상과 크기만을 고려하여 렌더링 한 결과를 나타낸다. 이에 반해 그림 5(b)에서 보는 것처럼, 블렌딩(blending) 및 정규화(normalization) 과정을 거친 점 기반 렌더링(point-based rendering)은 단지 첫 번째 과정만 거친 점 기반 렌더링(point-based rendering)보다 더 좋은 비주얼 품질을 나타낸다.

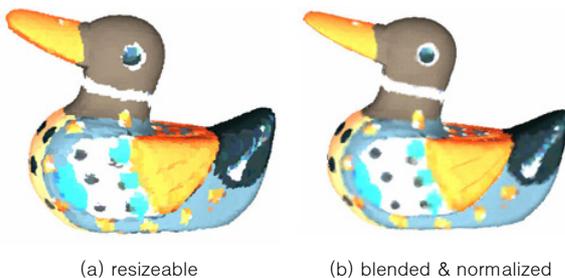


그림 5. 비주얼 품질 비교

4. 햅틱 상호작용

본 장에서는 햅틱 상호작용의 중요한 두 단계인 충돌 검출 및 반력 계산 알고리즘에 대해 상세히 기술한다.

4.1 충돌 검출

충돌 검출은 사용자에 의해 움직이는 햅틱 인터랙션 포인트(Haptic Interaction Point)와 물체와의 충돌을 검출하는 과정이다. 햅틱 상호작용에서의 빠른 충돌 검지 및 반력 계산은 사용자에게 물체와의 상호작용 시 안정한 힘을 제공하기 위해 필수적이다. 따라서 빠른 충돌 검지를 하기 위해 최근 본 연구팀에서 제안한 그래픽 하드웨어를 이용한 충돌 검지 방법을 이용하였다[11].

사용된 충돌 검지 방법은 기하학적 정보를 각 픽셀에 할당된 그래픽스 하드웨어의 깊이 버퍼(depth buffer) 값을 참조하여 수행된다. 가상 물체의 깊이 정보를 얻기 위하여 가상 환경의 6면(left, right, bottom, top)에 위치한 가상 카메라를 위치시키고 6면의 깊이 정보를 획득하는 것으로 시작된다. 이때 형상 정보와 관련 없는 color, texture, light 효과는 제거된다.

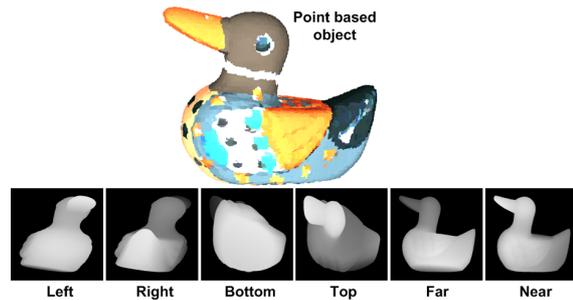


그림 6. 가상 카메라에 의해 획득된 물체의 깊이 정보

그림 6는 점 기반 물체 및 6개의 가상 카메라에 의해 획득된 깊이 정보를 나타내고 있다. 획득된 깊이 정보는 가상 카메라의 상대 위치 파라미터를 통해 객체의 6면에 대한 기하학적 정보로 변환이 되며 변환된 정보를 이용하여 HIP가 가상 물체의 내부에 위치하는 지를 판단하여 쉽게 충돌을 검출할 수 있다.

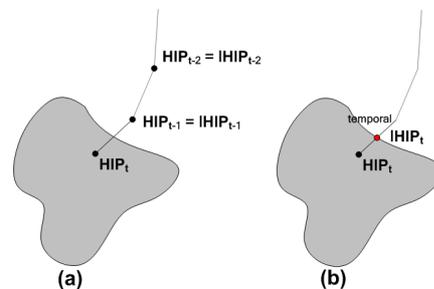


그림 7. 보간 방법에 의한 충돌 검출

앞서 언급한 간단한 충돌 검지는 얇은 물체나 빠르게 움직이는 HIP와 물체의 충돌을 정확히 검출하지 못하는 단점을 갖고 있다. 두 단점을 보완하기 위해 현재 HIP와 이전의

IHIP사이클을 보간 방법을 사용하여 정확한 충돌을 검출 한다. 그림 7(a)는 현 HIP와 이전 IHIP의 연결 선을 보여주고 있으며 이 연결선은 물체와의 충돌을 검출하게 된다. 그림 7(b)에서 보듯이 검출된 점이 임시 IHIP가 된다.

4.2 반력 계산

충돌 검출 후 반력의 크기와 방향을 결정하기 위해 IHIP 주변의 기하학적 정보가 필요하다. 본 논문에서는 앞 장에서 설명한 6개의 가상카메라로부터 얻어진 깊이 정보를 참조하여 구성된 LOMI (local occupancy map instance) [10, 11]를 이용하여 필요로 되는 객체의 기하학적 정보를 충당한다. LOMI는 국부적이고 일시적인 occupancy map으로 육면체의 3차원 격자의 셀들로 구성되어있으며, 각 셀들은 IHIP 주변 객체의 일부분에 대한 내부(interior), 외부(free space), 표면(surface)의 정보를 포함한다. 제안된 LOMI는 전체 가상환경이 아닌 객체의 일 부분에 대해 수행되기 때문에 실시간 연산이 가능하다.

이렇게 LOMI가 구성되면 반력의 크기와 방향을 결정해 주기 위해 god-object, proxy point or surface contact point [4]와 같은 개념의 IHIP를 이용한다. 객체와의 햅틱 인터랙션 과정에서 HIP는 햅틱 인터페이스의 임퍼던스 한계 때문에 객체의 내부로 들어가게 된다. 반면에 이상적인 HIP를 의미하는 IHIP는 가상 객체의 표면에 남아있게 된다. 즉 가상 객체의 외부에 HIP가 위치하면 HIP와 IHIP는 개념적으로 같은 위치를 갖고 있으나 충돌이 검출되면 HIP와 IHIP는 분리되어 두 벡터를 이용하여 반력의 크기와 방향을 계산하게 된다. 제시된 알고리즘에서 IHIP는 HIP에 가장 가까운 surface voxel에 위치된다.

시간 t에서 충돌이 검출되면 HIP_t는 가상 물체의 내부로 들어가는 반면 IHIP_t는 이전 t-1의 IHIP_{t-1}에 의해 생성된 LOMI_{t-1}를 참조하여 LOMI_{t-1}의 셀 중에서 HIP_t와 가장 가까운 거리에 위치한 surface voxel에 남게 된다. 다음 t+1에서 HIP가 새로운 위치 HIP_{t+1}로 이동하게 되면 IHIP_{t+1} 위치는 이전 시간에 생성된 LOMI_t의 가상 물체의 표면을 나타내는 셀들에 의해 결정되고 시간 t+1에서의 반력의 크기와 방향은 3차원 벡터HIP_{t+1}와IHIP_{t+1}에 의해 계산된다. 그림 8은 반력 계산 및 LOMI에 의해 IHIP 위치 결정 과정을 보여준다.

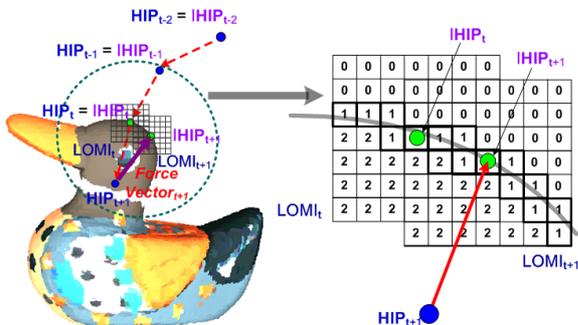


그림 8. LOMI를 이용한 IHIP결정 및 반력 계산

5. 실험 및 결과

본 논문에서 제안하는 알고리즘의 효율성 및 효율성을 위

해 3차원 레이저 스캐너로 획득된 오리 모델 및 다양한 점 기반 물체들을 이용하여 시스템을 구축하였다. 제안된 알고리즘은 PHANToM Omni1.5 햅틱 인터페이스[12]와 팬티엄 3.2GHz CPU, 1G RAM, GeForce 6600그래픽 카드 사양의 컴퓨터에서 구동되었다. 구성된 애플리케이션은 다중 스레드로 이루어졌으며 그래픽 렌더링 루프에 비해 햅틱 렌더링 루프가 높은 우선순위를 갖고 구동된다.



그림 9. 점 기반 곡면으로 이루어진 물체와 햅틱 상호작용

그림 9는 제안된 알고리즘으로 구성된 시스템에서 사용자가 햅틱 장치를 이용하여 점 기반 곡면 표현을 갖는 물체와의 상호작용을 나타내는 그림이다. 사용자의 움직임에 의해 햅틱 장치의 3차원 입력이 결정되고 물체와의 충돌이 발생이 되면 사용자에게 3차원 힘을 계산하여 렌더링 하게 되므로 가상 공간의 물체를 햅틱 장치를 이용하여 만질 수 있다.

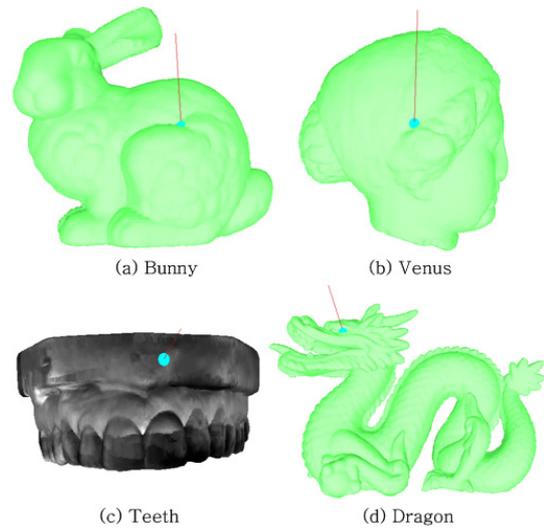


그림 10. 점 기반 곡면으로 이루어진 다양한 물체와 햅틱 상호작용

그림 10은 다른 개수의 점을 가지고 있는 다양한 물체와의 햅틱 상호작용을 나타내는 것으로 하늘색 점은 IHIP의 위치를 나타내며 빨간색 선은 계산된 힘의 크기와 방향을 나타내고 있다.

본 논문에서 제안된 알고리즘의 효율성을 증명하기 위해 각기 다른 점을 가지고 있는 물체에 대한 그래픽 및 햅틱 렌더링 루프의 시간을 측정하였다. 표1은 측정 결과를 나타낸다. 측정 단위는 그래픽 렌더링의 경우 Hz이고 햅틱 렌더링의 경우 KHz이며 500개의 측정 시간 샘플(sample)을 20번

측정하여 평균을 낸 값이다.

표 1 다양한 불규칙 점 기반 물체의 그래픽 및 햅틱 렌더링 시간

모델	Surface Splats 수	그래픽 렌더링 시간	햅틱 렌더링 시간
Duck	16,800	104.15(Hz)	3.30(KHz)
Bunny	35,947	80.33(Hz)	3.28(KHz)
Venus	67,180	68.02(Hz)	3.31(KHz)
Teeth	116,604	61.56 (Hz)	3.25(KHz)
Dragon	437,645	45.98(Hz)	3.30(KHz)

측정 결과에서 알 수 있듯이 많은 점을 갖고 있는 물체의 경우 그래픽 렌더링 시간은 점의 수가 증가함에 따라 효율이 감소하는 반면에 햅틱 렌더링 시간은 빠른 렌더링 시간을 보여주고 있음을 확인할 수 있다. 즉, 제안된 알고리즘의 햅틱 렌더링 시간은 물체를 구성하고 있는 점의 수와 관계없이 빠르게 수행됨을 확인할 수 있다.

본 논문에서 제안된 알고리즘과 다른 방법론으로 구현된 햅틱 렌더링 알고리즘과의 성능을 비교하기 위해 Polygon 기반의 그래픽 햅틱 렌더링이 구현되어 있는 CAID3D Haptic API[14]와 그래픽 햅틱 렌더링 효율을 비교하였다. 표 2는 본 논문에서 제안된 알고리즘과 CAID3D Haptic API와의 그래픽 햅틱 렌더링 측정시간을 나타낸다. 동일한 환경에서의 성능 비교를 위해 위 표 1에서 사용된 Teeth 모델을 이용하여 각 렌더링 시간을 측정하였다. 결과에서 알 수 있듯 햅틱 렌더링을 위해 특정한 데이터 구조(Bounding Box)를 사용하는 CHAI3D의 경우 햅틱 렌더링 시간이 많이 소요되기 때문에 불안정한 햅틱 상호작용(Unstable Haptic Interaction)을 제공하게 되며, 그래픽 렌더링 효율에 있어서도 본 논문에서 제안한 점 기반 곡면 표현 알고리즘이 효율적임을 증명하고 있다.

표 2 제안된 알고리즘과 CHAI3D의 효율성 비교

	그래픽 렌더링 방법론	햅틱 렌더링 방법론	그래픽 렌더링 시간	햅틱 렌더링 시간
본 논문	Point 기반	Local Depth	61.56(Hz)	3.25(KHz)
CHAI3D	Mesh 기반	Bounding Box	61.25(Hz)	0.003(KHz)

6. 결론

본 논문에서는 빠르고 효율적인 그래픽 및 햅틱 렌더링을 위한 점 기반 곡면으로 이루어진 복잡한 물체와의 효율적인 햅틱 상호작용을 제안하였다. 제안된 알고리즘은 미리 계산된 계층적 데이터구조(pre-computed hierarchical data structure)를 필요로 하지 않는 점 기반 곡면 표현 이용한 그래픽 렌더링 알고리즘 및 깊이 버퍼와 LOMI를 이용한 햅틱 렌더링 알고리즘을 통하여 그래픽 및 햅틱 상호작용을 가능하게 하였다. 실험 결과에서 나타났듯이 기존의 Polygon 기반의 햅틱 렌더링 알고리즘 보다 빠르고 안정한 상호작용을 제공할 수 있으며 3차원 스캐너로 획득된 많은 점을 포함하고 있는 물체에 대해서도 효과적으로 그래픽 및 햅틱 렌더링

이 가능함을 증명하였다.

향후 연구과제로는 3차원 스캐너로 획득된 복잡한 물체에 대해 점 기반 곡면 표현법을 이용한 실시간 변형 알고리즘을 개발하는 연구가 진행 될 것이며, 그래픽 및 햅틱 상호작용을 보다 사실적이고 몰입감 있게 하기 위해 그래픽/햅틱 co-location 시스템을 개발이 진행 될 것이다. 아울러 촉감 방송을 위해 실시간으로 방송 콘텐츠 데이터를 획득하고 촉감 상호작용을 할 수 있는 연구로 확장될 것이다.

감사의 글

본 연구는 정보통신부 선도기반기술개발사업(차세대 PC 기술개발), 한국 과학재단 사업의 실감콘텐츠연구센터 및 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(IITA-2006-C1090-0603-0017)의 연구결과로 수행되었음. Wood duck 모델을 제공해 주신 광주과학기술원 실감콘텐츠연구센터 분들께 감사 드립니다.

참고문헌

- [1] K. Salisbury, F. Barbagli and F. Conti, "Haptic Rendering: Introductory Concepts", IEEE Computer Graphics and Applications Magazine, Special issue on Haptic Rendering, April 2004
- [2] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib., "The haptic display of complex graphical environment", In SIGGRAPH 97 conference proceedings, Vol. 1, pp. 295-301, 1997.
- [3] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. "Haptic rendering: programming touch interaction with virtual objects", Symposium on Interactive 3D Graphics, pp. 123-130, 1995.
- [4] C. Zilles and K. Salisbury, "A constraint-based god-object method for haptic display", Conf. Proc. IEEE/RSJ Intelligent Robots and Systems, Vol. 3, pp. 146-151, 1995.
- [5] A. Gregory, M.C. Lin, S. Gottschalk, and R. Taylor, "H-COLLIDE: A Framework for Fast and Accurate Collision Detection for Haptic Interaction", Proc.IEEE Virtual Reality, pp. 38-45, 1999.
- [6] S. Rusinkiewicz and M. Levoy, "QSplat: A Multiresolution Point Rendering System for Large Meshes", Conf. Proc. SIGGRAPH 2000, pp. 343-352, 1999.
- [7] R.S. Avila and L.M. Sobierajski, "A Haptic Interaction Method for Volume Visualization" IEEE Visualization, pp. 197-204, 1996.
- [8] S. Walker and K. Salisbury "Large Haptic Topographic maps: Marsview and the Proxy Graph Algorithm", Proc. ACM Symp. Interactive 3D graphics, pp. 83-92, 2003.

- [9] K. Potter, D. E. Johnson, and E. Cohen, "Height Field Haptics", Conf. Proc. Haptics Symposium, Proc. IEEE Symp. Haptic. Interfaces for Virtual Environment and Teleoperator Systems, pp. 229-235, 2004.
- [10] J.P. Kim and J. Ryu, "Hardware Based 2.5D Haptic Rendering Algorithm using Localized Occupancy Map Instance", Conf. Proc. Artificial Reality and Telexistence (ICAT), 2004.
- [11] J.P. Kim, B.C. Lee, and J. Ryu, "Haptic Rendering with Six Virtual Cameras", Conf. Proc. Human Computer Interaction, pp. 467-472, 2005.
- [12] PHANToM Omni, Sensable Technologies, Inc., <http://www.sensable.com>.
- [13] R.B.Pajarola, M.Sainz, P.Guidotiti. Confetti Object-Space Point Blending and Splatting. IEEE Transaction on Visualization and Computer Graphics, 2004.
- [14] Conti F, Barbagli F, Morris D, Sewell C, "CHAI: An Open-Source Library for the Rapid Development of Haptic Scene", IEEE World Haptics, Demo presented, 2005.