

메쉬 그룹화를 이용한 충돌 검출 알고리즘

박종섭¹ · 장태정^{2*}

¹(주)리얼타임웨이브

²강원대학교 전기전자공학부

Collision detection algorithm by using mesh grouping

Jong-Seop Park¹ · Tae-Jeong Jang^{2*}

¹Department of Electronics and Communication Engineering, Kangwon National University, Chuncheon 24341, Korea, currently working in the Realtimewave Co.

²*Department of Electronics and Communication Engineering, Kangwon National University, Chuncheon 24341, Korea

[요 약]

본 논문에서는 가상 공간에서 객체와의 상호 작용을 위한 빠른 충돌 감지 방법을 제안하였다. 먼저 메쉬 그룹화 단계에서는, 전체 공간을 일정한 크기의 소공간들로 분할하고, 각각의 소공간에 속하는 메쉬들에 대하여 이들을 모두 포함하는 최소의 기본 입체도형(육면체 혹은 구)의 위치와 크기를 정함으로써 그룹화한다. 충돌 검출 단계에서는, HIP(Haptic Interface Point)가 어떤 그룹을 대표하는 입체도형 내부에 들어 있는지를 검사하여 특정 입체도형과의 충돌이 확인되면 해당 그룹의 메쉬들만을 대상으로 실제 충돌이 일어난 메쉬를 찾는다. 본 논문에서 제안한 방법을 적용한 경우와 적용하지 않은 경우의 연산시간을 측정 및 비교함으로써 제안한 알고리즘의 효용성을 확인하였다.

[Abstract]

In this paper, we propose a fast collision detection method for interacting with objects in virtual space. First, in the mesh grouping step, the entire space is divided into small spaces of a predetermined size, and the positions and sizes of the smallest basic stereoscopic shapes (cube or sphere) including all of the meshes belonging to each small space are determined. In the collision detection step, it is checked whether a haptic interface point (HIP) is included in a three-dimensional figure representing a group. When a collision with a specific three-dimensional figure is confirmed, searching is performed only for the meshes in the group to find a mesh on which a possible real collision with HIP occurred. The effectiveness of the proposed algorithm is verified by measuring and comparing the computation time of the proposed method with and without the proposed method.

색인어 : 충돌 검출, 햅틱 렌더링, 메쉬 그룹화

Key word : Collision detection, Haptic rendering, Mesh grouping

<http://dx.doi.org/10.9728/dcs.2018.19.1.199>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 18 December 2017 ; **Revised** 23 January 2018

Accepted 29 January 2018

***Corresponding Author; Tae-Jeong Jang**

Tel: +82-10-2250-6348

E-mail: jangtj@kangwon.ac.kr

1. 서론

햅틱 기술이 처음 소개된 이후 상당 기간에 걸쳐 햅틱 렌더링(haptic rendering)에 관한 기술 개발이 활발히 진행되어 왔다 [1-2]. 햅틱 렌더링은 사용자가 가상 공간에서 가상의 물체와 상호 작용을 할 때 가상 물체 모델과의 접촉에 따른 반작용으로 생성되는 힘 또는 촉감을 햅틱 장치를 통해 적절한 형태로 사용자가 느낄 수 있도록 해 주기 위한 일련의 시뮬레이션 과정이다. 이러한 햅틱 렌더링을 통해 사용자가 가상 물체와 접촉 시 실제와 유사한 촉감을 느끼게 할 수 있으며, 경우에 따라서는 특정한 형태로 촉감을 변형하거나 아예 새롭게 생성한 촉감을 느끼게 할 수도 있다.

가상 공간에서 사용자가 조작하는 햅틱 장치의 끝점 위치에 해당하는 가상 공간 상의 점인 HIP(Haptic Interface Point)와 가상 물체 간의 상호 작용을 시뮬레이션하기 위해서는 HIP와 가상 물체 간의 충돌 여부를 검출해야 하며, 이를 위해서는 HIP와 가상 물체를 구성하는 모든 메쉬들과의 충돌 검출이 반드시 필요하다. 그런데, 일반적으로 촉각 반응은 시각이나 청각 반응에 비해 몹시 민감하여 현실감 있고 안정적인 햅틱 렌더링을 위해서는 비주얼 렌더링에 비해 훨씬 높은 빈도(예: 1kHz)로 새로운 힘 되먹임 값을 계산해 주어야 한다[3-5]. 따라서 최대 1/1000초 밖에 되지 않는 짧은 시간 안에 충돌 검출을 포함한 햅틱 렌더링을 위한 계산을 모두 끝내야만 하기 때문에 빠르고 효율적인 충돌 검출 알고리즘이 필수적이다.

충돌 검출 기법은 가상공간에서 가상물체와 가상물체의 자연스러운 충돌 시뮬레이션, 사용자와 가상물체와의 실감나는 인터랙션 구현 등에서 꼭 필요한 중요한 기술로서 오래전부터 연구되어 왔다[6-10]. 가상공간에서 모델을 생성할 때 육면체의 경우 12개의 삼각형 메쉬면 충분하다. 하지만 구의 경우 구의 모양에 가깝게 모델링 할수록 더 많은 양의 메쉬를 필요로 한다. 이렇게 메쉬가 많아지면 충돌 검출에 있어 과부하를 야기할 수 있다. 이유는 하나의 메쉬가 충돌 조건을 만족하는지 조사하는 과정만 해도 절차가 많은데 수만 개로 이루어진 모델을 일일이 조사한다면 무리가 따를 수밖에 없기 때문이다. 그에 대한 대안으로 그림 1과 같은 바운딩 볼륨(bounding volume)을 이용한다[6-7].

바운딩 볼륨 방법은 복잡한 모델을 감싸는 단순 모델을 생성하여 시각적으로는 원형의 모델을 보여주고 촉각적으로는 단순 모델을 통해 충돌을 검출하는 방법이다. 이러한 바운딩 볼륨 방법에는 x축에 평행한 직육면체 바운딩박스(bounding box)를 이용한 AABB 방식과 모델을 길이 방향으로 최대한 감싸는 직육면체를 이용하는 OBB 방식, 그리고 면의 개수를 늘려 실제 모델과 단순 모델사이의 공백을 최소화한 k-Dops 방식, 그리고 이들을 계층적인 트리 형태로 구성하는 방식 등 다양한 방법이 제안되었다[6-9]. 이러한 방식은 빠르게 모델을 찾아낼 수 있는 장점이 있는 반면 실제 모델에는 접촉하지 않았는데도 충돌한 것으로 판정하게 되는 오류가 발생한다는 단점이 있다. 때문에

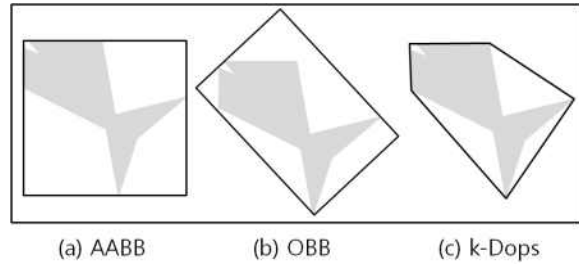


그림 1. 여러 가지 바운딩 볼륨
Fig. 1. Various bounding bolumes

사용자가 조작하는 HIP와 가상물체와의 충돌 시 반력 계산 등을 통한 촉감 생성을 위해 HIP가 충돌한 메쉬가 정확히 어느 것이고 얼마만큼의 깊이로 침투했는지에 대한 값을 필요로 하는 햅틱 렌더링에서는 이렇게 간략화된 바운딩 박스를 쓰는 충돌 검출 방식은 적절하지 않을 수 있다. 단순하고 메쉬가 적은 객체의 경우에는 문제가 없지만 복잡하고 메쉬가 상당히 많은 경우 하드웨어 성능을 높이거나 렌더링을 하는데 있어 특별한 처리가 필요하다.

본 논문은 메쉬가 상당히 많은 객체를 대상으로 하드웨어의 성능을 높이는 대신 특별한 처리를 통해 햅틱 렌더링의 속도저하를 개선해 보기 위한 연구이다. 즉 모든 메쉬들과 HIP(Haptic Interface Point)의 관계를 비교하여 충돌 중인 면을 찾아내는 충돌 검출 과정에서 메쉬들을 그룹화하여 조건에 충족한 그룹만을 대상으로 충돌 검출을 수행하는 방법이다. 또한 그룹화 방식에 있어 육면체의 형태로 그룹화 하는 방식과 구의 형태로 그룹화 하는 방식을 비교하여 연구하였다.

II. 메쉬 그룹화

2-1 육면체를 이용한 메쉬 그룹화

육면체를 이용한 메쉬 그룹화는 다음 네 단계로 이루어진다. 그림 2는 설명의 용이함을 위해 2차원으로 표현하였다.

그림 2의 (1)은 첫 번째로 모델의 메쉬를 모두 포함할 수 있는 Voxel Data Set 자료구조를 구성하는 그림이다. 일단 모델을 모두 감싸는 최대 크기의 육면체를 계산해 원하는 적절한 개수의 셀로 분리하고 순차적으로 인덱스를 부여한다[11].

두 번째로 그림 2의 (2)와 같이 내부에 메쉬를 하나도 포함하지 않는 셀을 조사해 제거한다. 또한 제거한 셀을 감안하여 인덱스를 변경한다. 이렇게 하면 후에 셀을 검색할 때 불필요한 연산을 피할 수 있다.

세 번째로 셀의 공간정보를 이용한 메쉬 분리 작업을 수행하는데, 메쉬가 어느 한 셀에 조금이라도 포함된다면 그 메쉬는 셀의 인덱스와 같은 ID를 부여 받는다. 그림 2의 (3)는 이 과정에서 1번 셀과 1번 셀을 통해 동일한 ID로 그룹화 된 메쉬에 대한 그림이다.

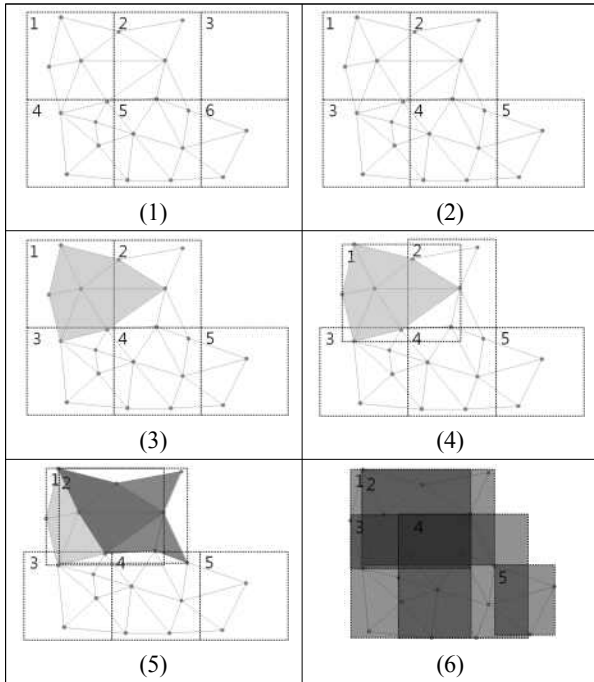


그림 2. 육면체를 이용한 메쉬 그룹화 과정
 Fig. 2. Mesh grouping process using a hexahedron

네 번째로 셀의 크기를 메쉬그룹을 이용해 변형하는 과정이다. 즉 두 번째 과정에서 생성된 그룹의 각 축 상에서 최댓값과 최솟값을 계산해 최소한으로 그룹을 모두 감싸는 셀로 변형한다. 그림 2의 (4)와 (5)는 이러한 과정에 대한 그림으로써 1, 2번 셀의 변형상태를 보여주고 있다. 이렇게 모든 변형과정을 거치면 마지막 그림인 그림 2의 (6)과 같은 형태가 된다. 이제 변형된 셀 내부에는 해당 그룹의 메쉬만 존재하며 HIP를 통해 충돌 메쉬를 검색하는 작업에서 메쉬 대신 현재 HIP를 포함하는 셀을 먼저 탐색 후 탐색된 셀 내부에 존재하는 메쉬만을 이용해 충돌을 검출한다.

2-2 구를 이용한 메쉬 그룹화

구를 이용한 메쉬 그룹화는 육면체를 이용한 메쉬 그룹화 과정을 재사용하여 생성하면 편리하다. 그림 5는 그 과정에 대한 그림이다.

그림 3의 (1)은 육면체를 이용해 그룹화한 셀 데이터로 셀마다 음영을 달리하지 않은 점만 빼면 그림 2의 (6)번과 완전히 동일하다. 이 데이터를 이용해 이 셀을 인덱스의 순서대로 구의 형태로 변형한다. 변형 방법은 셀의 중심점을 계산해 구의 중심으로 이용하고 해당 그룹의 메쉬를 모두 감싸야 하므로 중심점에서 가장 멀리 있는 정점(vertex) 정보를 조사해 두 점 사이를 반지름으로 하는 구를 생성하여 준다. 그림 3의 (2)와 (3)은 그 과정에 대한 그림이며 r은 위에서 설명한 반지름이다. 이러한 과정을 모든 셀에 순차적으로 수행해 시각적으로 표현하면 그림 3의 (4)와 같이 비누거품과 비슷한 형태가 된다.

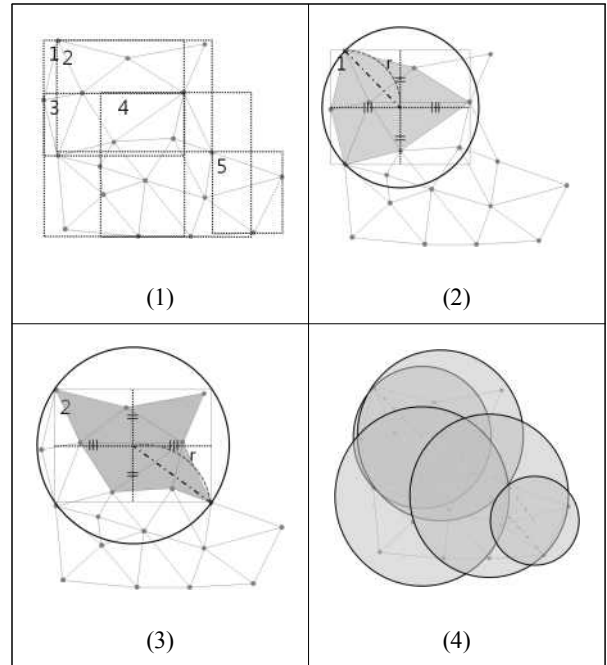


그림 3. 구를 이용한 메쉬 그룹화 과정
 Fig. 3. Mesh grouping process using a sphere

육면체를 이용해 메쉬를 탐색할 때엔 육면체의 x, y, z축 각각의 범위를 이용하지만 구를 이용해 탐색할 때엔 중심점과 반지름을 이용한다. 즉 HIP가 어느 구에 포함된 것으로 판단되면 해당 구에 소속된 메쉬만을 활성화시켜 충돌 검출에 이용하면 된다.

III. 충돌 검출 실험

3-1 실험 방법 및 환경

본 연구를 구현하고 실험하기 위한 시스템으로 Core i5-250 CPU @ 3.33Ghz와 AMD Radeon HD 6450의 그래픽 카드, 그리고 8.00GB의 메모리를 내장한 PC를 사용하였고, 햅틱 디바이스로는 SensAble사의 PHANToM Omni를 사용하였다. 그리고 그림 4은 실험에 사용된 원형의 Stanford bunny 모델로서 9400개의 polygon 메쉬로 이루어져 있다.

실험은 HIP를 통해 충돌 검출에 소요되는 시간을 측정하는 방법으로 진행하였다. 즉 그룹화한 모델 표면을 HIP를 통해 굽는 과정을 5초간 연속하여 수행해 충돌 메쉬를 검색하는데 까지 소요되는 평균 연산시간을 측정하였다. 또한 그룹의 개수를 달리하여 그에 따른 연산시간의 차이를 비교하고, 육면체를 이용해 그룹화한 방법과 구를 이용하여 그룹화한 방법도 측정된 연산 시간을 이용해 비교하였다.

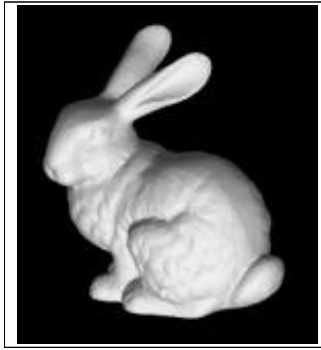


그림 4. 원형 Stanford bunny 모델
Fig. 4. Original Standford bunny model

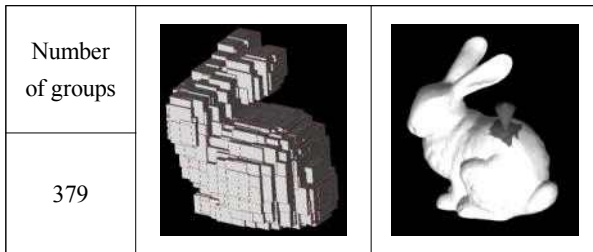


그림 5. 육면체를 이용한 메쉬 그룹(좌)과 활성화된 그룹 내 메쉬(우)
Fig. 5. Mesh groups by using hexahedron (left) and meshes in an active group (right)

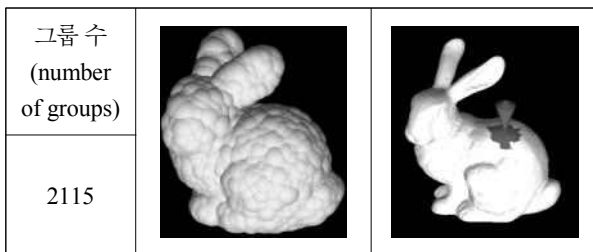


그림 6. 구를 이용한 메쉬 그룹(좌)과 활성화된 그룹 내 메쉬(우)
Fig. 6. Mesh groups by using sphere (left) and meshes in an active group (right)

3-2 실험 결과

그림 5의 왼쪽 그림은 379개의 육면체를 이용하여 그룹화한 bunny 모델을 시각화한 모습이며 오른쪽 그림은 HIP를 통해 탐색 조건을 만족한 그룹 내 메쉬만을 활성화 하여 시각화한 모습이다. HIP주위 진한색의 메쉬를 제외한 모든 메쉬는 시각적으로만 보일뿐 충돌 검출에는 이용되지 않는다.

그림 6은 2115개의 구를 이용해 그룹화한 모습에 대한 그림이다. 왼쪽은 그룹화 된 구들의 모습을 보여주며 오른쪽은 그림 5과 마찬가지로 탐색 조건을 만족한 그룹 내 메쉬를 활성화시켜 시각화한 모습이다.

그림 7은 앞서 설명한 실험 방법으로 측정한 결과 값을 보여주는 그래프이다. 연산시간은 ms단위로 나타내었으며 그룹의 개수 별로 구와 육면체에 대한 연산 시간을 표시하였다. 그룹의 개수를 임의로 정할 수는 없으며 전처리 과정에서 자동으로 생성된 개수이다. 즉 구조화한 복셀에서 한 셀의 크기에 따라 개수가 결정된다. 작게 할수록 셀이 포함할 수 있는 메쉬의 양이 적어지며 그룹의 개수는 증가한다.

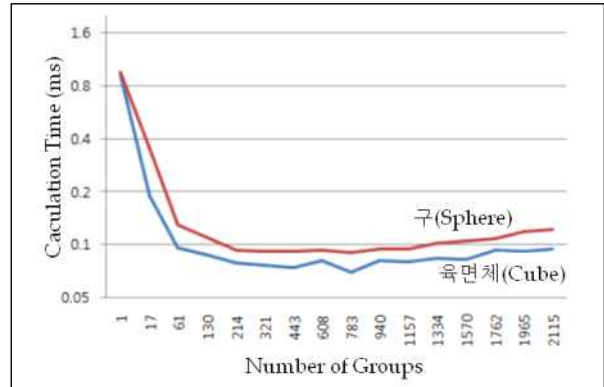


그림 7. 그룹 수에 따른 충돌 검출 연산 시간
Fig. 7. Calculation time for collision detection according to number of groups

그래프를 분석해 보면, 구와 육면체 모두 그룹의 개수가 하나인 경우, 즉 그룹화하지 않은 경우 약 1ms의 연산시간을 가지는데 반해 그룹의 개수가 증가함에 따라 점차적으로 연산시간이 줄어들어 약 61개의 그룹 이상이 되면 1/10정도 단축된 시간을 가졌다. 그리고 그룹이 약 783개 이상으로 증가하면서 부터는 오히려 연산시간이 조금씩 늘어남을 알 수 있다. 이것은 그룹 내의 메쉬는 줄었지만 그룹의 개수가 많아져 그룹 탐색시간이 증가했기 때문이다. 그렇기 때문에 전처리 시간을 고려한다면 실험에 사용한 원본 bunny 모델의 경우 214개 정도의 육면체로 그룹화되었을 때 가장 적절한 그룹화가 이루어졌다고 할 수 있다. 또한 구를 이용해 구의 중심과 반지름을 이용한 탐색과 육면체가 가지는 x, y, z축의 범위 값을 이용한 탐색 방법의 비교에서는 근소하게 육면체의 경우가 연산시간을 절약하는데 유용했다. 원인을 분석해본 결과 구를 이용한 탐색의 경우 구의 중심과 HIP사이의 거리를 계산하는 과정이 육면체의 범위에 HIP가 포함되는지 검사하는 과정보다 복잡했기 때문이었다. 그 그래프를 분석해 보면 구와 육면체 모두 그룹화하지 않은 경우와 동일한, 그룹의 개수가 하나인 경우 약 1ms의 연산시간을 가지는데 반해 그룹의 개수가 증가함에 따라 점차적으로 연산시간이 줄어들어 약 61개의 그룹 이상이 되면 1/10정도 단축된 시간을 가졌다. 그리고 그룹이 약 783개 이상으로 증가하면서 부터는 오히려 연산시간이 조금씩 늘어남을 알 수 있다. 이것은 그룹 내의 메쉬는 줄었지만 그룹의 개수가 많아져 그룹 탐색시간이 증가했기 때문이다. 그렇기 때문에 전처리 시간을 고려한다면

실험에 사용한 원본 bunny 모델의 경우 214개 정도의 육면체로 그룹화되었을 때 가장 적절한 그룹화가 이루어졌다고 할 수 있다. 또한 구를 이용해 구의 중심과 반지름을 이용한 탐색과 육면체가 가지는 x, y, z축의 범위 값을 이용한 탐색 방법의 비교에서는 근소하게 육면체의 경우가 연산시간을 절약하는데 유용했다. 원인을 분석해본 결과 구를 이용한 탐색의 경우 구의 중심과 HIP사이의 거리를 계산하는 과정이 육면체의 범위에 HIP가 포함되는지 검사하는 과정보다 복잡했기 때문이었다.

IV. 결 론

과거에 컴퓨터 그래픽스가 발전하기 전과 달리 최근에는 상당히 복잡하고 섬세한 3차원 모델링이 가능해졌다. 이를 이용해 가상훈련이나 수술 시뮬레이션 등에서 시각적으로 보다 효과적인 시스템 구성이 이루어졌고 따라서 햅틱 렌더링 또한 보다 섬세한 반력을 사용자에게 전달해 줄 필요성이 생겼다. 이러한 햅틱 렌더링에서 몰입도를 높이고 사실감을 향상시키기 위해서는 하드웨어적 성능과 연산량에 따른 연산시간의 충분한 보장이 필요하다.

복잡한 모델에 관하여 연산시간 확보를 위해 바운딩볼륨 등을 통해 지나치게 간략화한 모델을 사용하는 방법은 시각적인 모델과의 차이가 있어 몰입감이 떨어진다. 또한 모델에 변형을 추가적으로 수행해야 할 때 이러한 방법은 효과적이지 않다. 따라서 메쉬의 변형 없이 시각적 렌더링과 햅틱 렌더링의 차이가 발생하지 않는 메쉬 그룹화를 통한 햅틱 렌더링은 실감 면에서 유리할 수 있다. 결과에서 보듯이 충돌 처리에서 이용되는 연산시간을 약 1/10로 단축시켰다. 때문에 나머지 자원을 이용해 보다 안정적인 시뮬레이션과 추가적인 기능구현에서 부담을 줄일 수 있을 것이다.

참고문헌

- [1] Kenneth Salisbury, Francois Conti, and Federico Barbagli, "Haptic Rendering: Introductory Concepts," *IEEE Computer Graphics and Applications*, Vol. 24, Issue 2, pp. 24-32, Mar. 2004.
- [2] Laehyun Kim, "Haptic Rendering Technology," *Journal of the KSME*, Vol. 47, No. 2, pp. 36-43, Feb. 2007 (in Korean).
- [3] C. A. Mendoza and C. Laugier, "Realistic Haptic Rendering for Highly Deformable Virtual Objects," in *Proc. of IEEE Conf. on Virtual Reality*, pp. 264-269, Yokohama, Japan, Mar. 2001.
- [4] Yongmin Zhong *et al.* "Haptic deformation modelling through cellular neural network," *I. J. of SIMULATION*, Vol. 7, No. 8, pp. 3-20, 2006.
- [5] Carlos Garre and Miguel A. Otaduy, "Haptic rendering of objects with rigid and deformable parts," *Computers & Graphics*, Vol. 34, Issue 6, pp. 689-697, Dec. 2010.
- [6] James T. Klosowski, *et al.* "Efficient collision detection using bounding volume hierarchies of k-DOPs," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 4, Issue. 1, pp. 21-36, Jan-Mar 1998.
- [7] Christer Ericson, *Real-Time Collision Detection*, Morgan Kaufmann Pub., 2005.
- [8] Dirk Bartz, James T. Klosowski, and Dirk Staneker, "K-dops as tighter bounding volumes for better occlusion performance," *ACM SIGGRAPH Visual Proc.*, p. 213, 2001.
- [9] König, Henry, Thomas Strothotte, and O. von Guericke. "Fast Collision Detection for Haptic Displays Using Polygonal Models," *Proc. of the Conf. on Simulation and Visualization*, pp. 289-300, Ghent, Belgium, 2002.
- [10] In-kyeong Jo and Hwa-jin Park, "Study on collision processing among objects by 3D information of real objects extracted from a stereo type method in AR," *Journal of Digital Contents Society*, Vol. 11, No. 2, pp. 243-251, June 2010.
- [11] Sarah F. Gibson, "Beyond volume rendering: visualization, haptic exploration, and physical modeling of voxel-based objects," *Visualization in Scientific Computing*, Springer Vienna, pp. 10-24, Jan 1995.



박종섭(Jong-Seop Park)

2010년 : 강원대학교 전기전자공학부 전자통신공학전공
2013년 : 강원대학교 대학원 전자통신공학과 (공학석사)

2013년~현 재: (주)리얼타임웨이브 연구원

※ 관심분야 : 햅틱스(Haptics), 컴퓨터그래픽스(Computer Graphics) 등



장태정(Tae-Jeong Jang)

1986년 : 서울대학교 제어계측공학과 (공학사)
1988년 : 서울대학교 대학원 제어계측공학과 (공학석사)
1994년 : 서울대학교 대학원 제어계측공학과 (공학박사-제어계측)

2000년~2001년: Boston University, USA, Research Assistant Professor

2010년~2011년: Massachusetts Institute of Technology, USA, Visiting Researcher

1995년~현 재: 강원대학교 전기전자공학부 교수

※ 관심분야 : 햅틱스(Haptics), 가상현실(Virtual Reality), 컴퓨터그래픽스(Computer Graphics)