

A Ram Choi, Sung Min Kim, Mee Young Sung. Controlling the contact levels of details for fast and precise haptic collision detection. *Frontiers of Information Technology & Electronic Engineering*, 2017, **18**(8): 1117-1130.

<https://doi.org/10.1631/FITEE.1500498>

Controlling the contact levels of detail for fast and precise haptic collision detection

Key words: Collision detection; Haptic rendering; Bounding sphere; Clustering; Contact levels of detail

Contact: Mee Young SUNG

E-mail: mysung@inu.ac.kr

 ORCID: <http://orcid.org/0000-0001-8078-624X>

Motivation/Main ideas

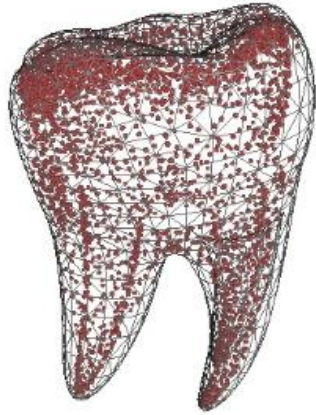
➤ Motivation

Creating bounding spheres in respect to the contact levels of detail (CLOD), which can fit objects while maintaining the balance between high speed and precision of haptic collision detection.

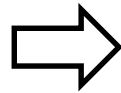
➤ Main ideas

Summary of our method for collision detection	
Method	Specification
Bounding Spheres formation	Creating sphere <ul style="list-style-type: none">• Find centroids• Generate central axis• Produce centers of spheres
	Creating sphere <ul style="list-style-type: none">• Discard meaningless centers• Cluster centers through single-link clustering
Two-level collision detection	Fast detection of sphere <ul style="list-style-type: none">• Calculate the distance• Check for overlap
	Precise detection in sphere <ul style="list-style-type: none">• Project the haptic pointer• Take outer product and inner product of vectors• Check distance

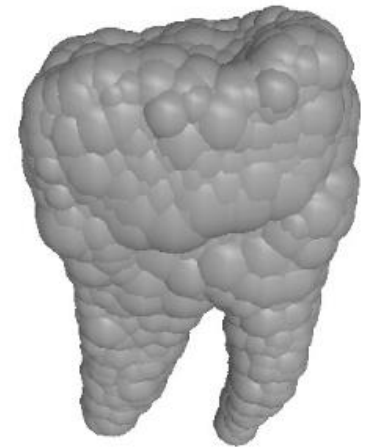
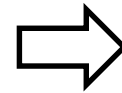
Method : Bounding sphere formation



Finding centroids



Clustering centers



Creating bounding spheres

Method : Two-level collision detection

1. Fast detection of sphere

```
1. GPU: for each spherei do  
2.   difference = hapticCursor.pos - spherei.pos  
3.   distance = difference.length  
4.   sumOfRadius = hapticCursor.radius + spherei.radius  
5.   if distance < sumOfRadius then  
6.     collision detected  
7.   end if  
8. end for
```

2. Precise detection in sphere

```
13. for each trianglej do  
14.   for each trianglej do  
15.     isPositive = true  
16.     vector1 = (vertexa, vertexb)  
17.     vector2 = (vertexa, vertexc)  
18.     vector3 = (vertexa, T)  
19.     crossResult1 = vector1 × vector3  
20.     crossResult2 = vector3 × vector2  
21.     if (crossResult1 · crossResult2) < 0 then  
22.       isPositive = false  
23.       break  
24.     end if  
25.   end for  
26.   if isPositive then  
27.     index = i  
28.     break  
29.   end if  
30. end for
```

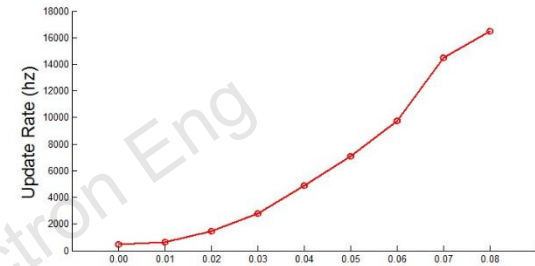
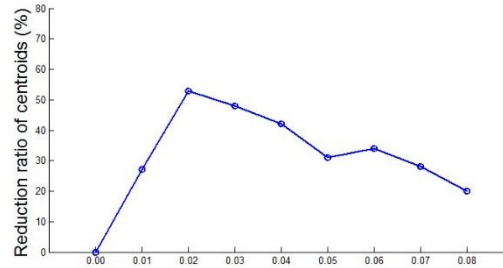
Method : Contact levels of detail



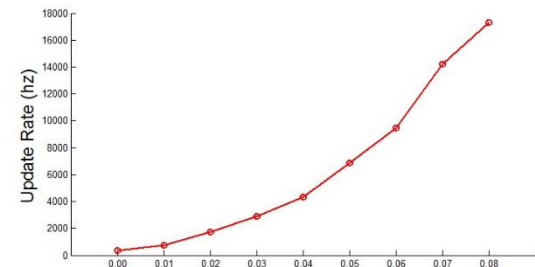
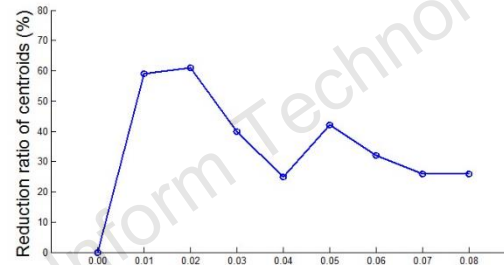
Contact levels of detail controlled by variable distances ($d=0-0.11$) of single-link clustering, where d indicates the side length of the clustering cube, and the numbers in the parentheses denote the number of spheres.

Major result(I)

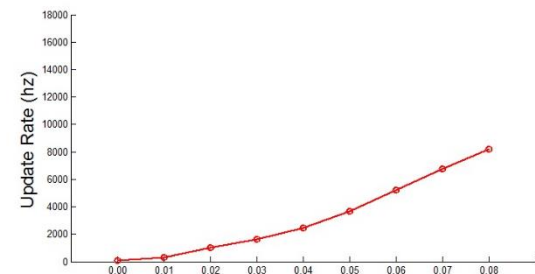
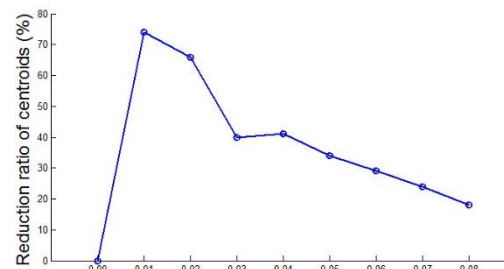
Reduction ratio for tooth.
 (a) Step reduction ratio.
 (b) Update rate.



Reduction ratio for clover.
 (a) Step reduction ratio.
 (b) Update rate.



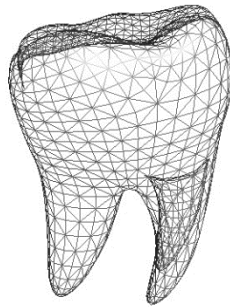
Reduction ratio for hat.
 (a) Step reduction ratio.
 (b) Update rate.



Major result(II)



(a)



(b)



(c)



(d)

Depiction of a tooth.

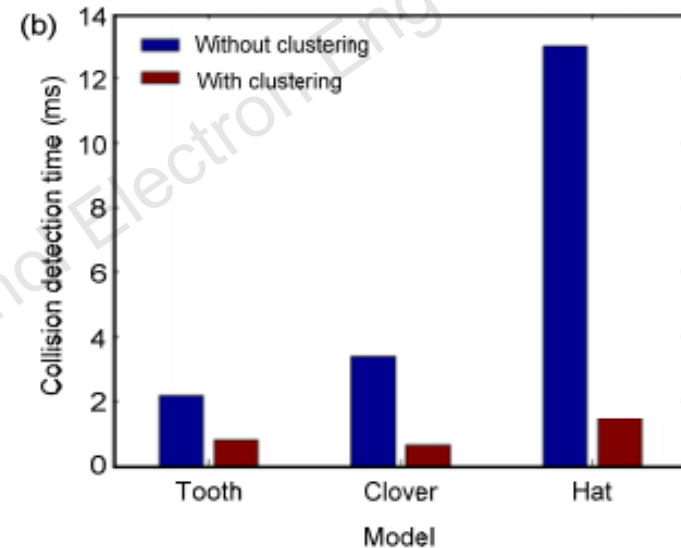
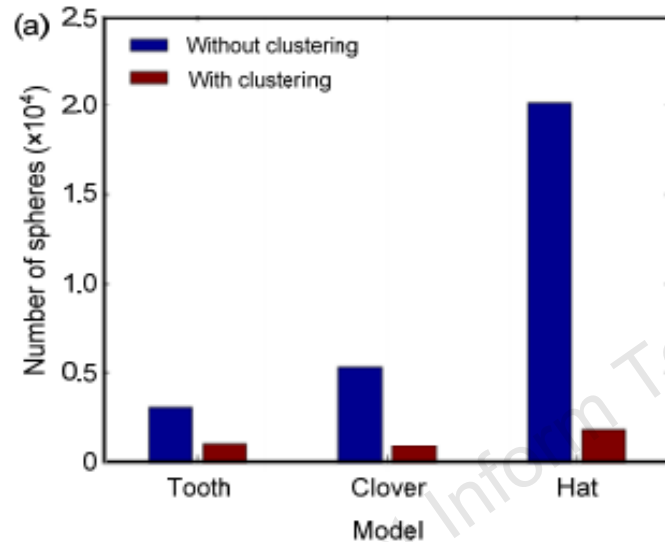
(a) Model.

(b) Polygons.

(c) Bounding spheres without clustering.

(d) Bounding spheres with the optimal clustering distance of 0.02.

Major result(IV)



Performance comparison.

(a) Number of spheres without and with clustering (clustering distance = 0.02).

(b) Comparison of collision detection time without and with clustering (clustering distance = 0.02).

Conclusions

- Fast enough for haptic rendering, using spherical structure for bounding volume.
- Precise enough to the voxel level.
- Variable clustering distances for controllable CLOD.
- Less overhead for maintaining bounding sphere hierarchy.
- Smaller radii of bounding spheres using centroids of triangular polygons.