Wen Sheng
Holly Rushimier
February 12, 2016

CGAL Code

1. Minkowski & Convex Hulls Comparison

```
Sheng-Wen:Minkowski_sum_2 shengwen$ ./sum_triangle_triangle_ch data-triangle
run time: 6.11047
Sheng-Wen:Minkowski_sum_2 shengwen$ ./sum_triangle_triangle data-triangle
run time: 36.1466
```

Using large file (data-triangle), the results . In terms of time, Minkowski takes more time than convex hull as the above image shows as calculate the sums is O(n*n) efficiency which slows down the process.

2. Half-edge data structure

According to subdivision method,

the new vertices number = the old vertices number + the old faces number

the new faces number = the old faces number * 4, as the example shows below:

```
OFF
8 6 0
-1 -1 1
-1 1 1
1 1 1
1 -1 1
-1 -1 -1
-1 1 -1
1 1 -1
1 -1 -1
4 3 2 1 0
4 0 1 5 4
4 6 5 1 2
4 3 7 6 2
4 4 7 3 0
4 4 5 6 7
```

After subdivision,

```
OFF
14 24 0
```

```
# 14 vertices
# -----------------------------------------


-0.62963 -0.62963 0.62963
-0.62963 0.62963 0.62963
0.62963 0.62963 0.62963
0.62963 -0.62963 0.62963
-0.62963 -0.62963 -0.62963
-0.62963 0.62963 -0.62963
0.62963 0.62963 -0.62963
0.62963 -0.62963 -0.62963
0 0 1
-1 0 0
0 1 0
1 0 0
0 -1 0
0 0 -1

# 24 facets
# -----------------------------------------

3  12 3 8
3  12 0 9
3  11 6 10
3  11 2 8
3  10 1 8
…
```

3. Polyhedron Cut Cube explanation and Modification

`(CGAL-4.7/examples/Polyhedron/polyhedron_cut_plane_3.cpp)`

This program firstly generate a cube with vertices: (1 0 0), (0 0 1), (0 0 0), (0 1 0), (1 0 1), (0 1 1), (1 1 0), (1 1 1) and define a plane using 3 points. Then the main function invokes the CGAL::polyhedron_cut_plane_3 function and print out the result(the plane intersects with the cube).

I modified the source code to save the result into .off file. The modified file is in **polyhedron_prog_cut_cube_off.cpp**

The running command is: **./polyhedron_prog_cut_cube_off ./test.off**