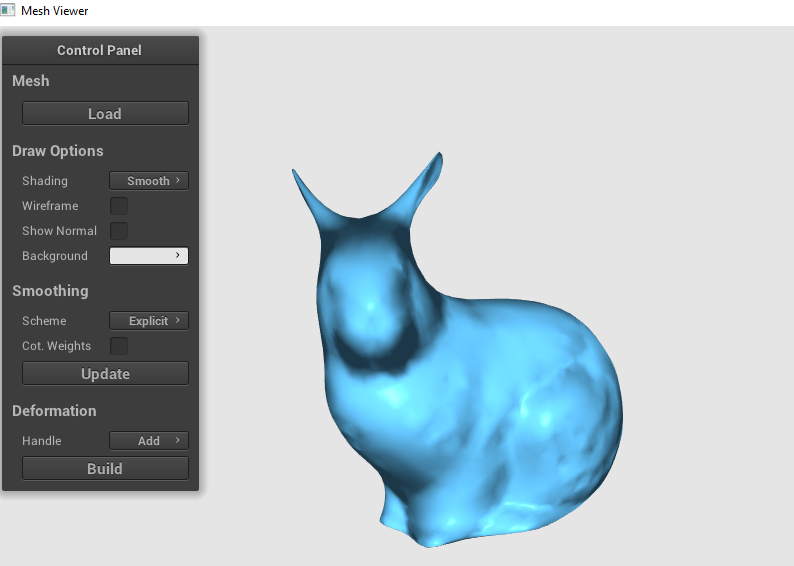
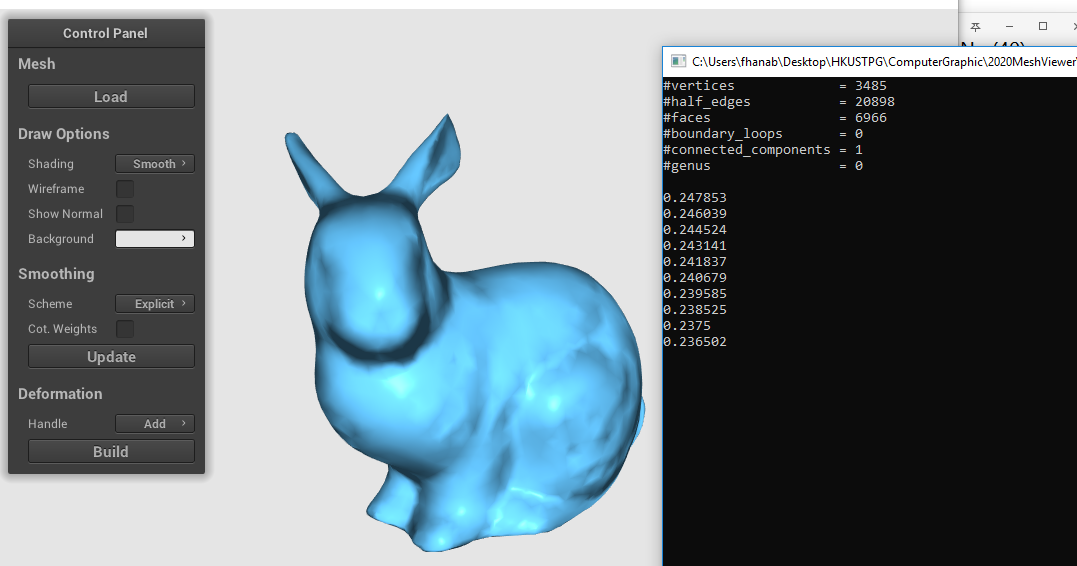
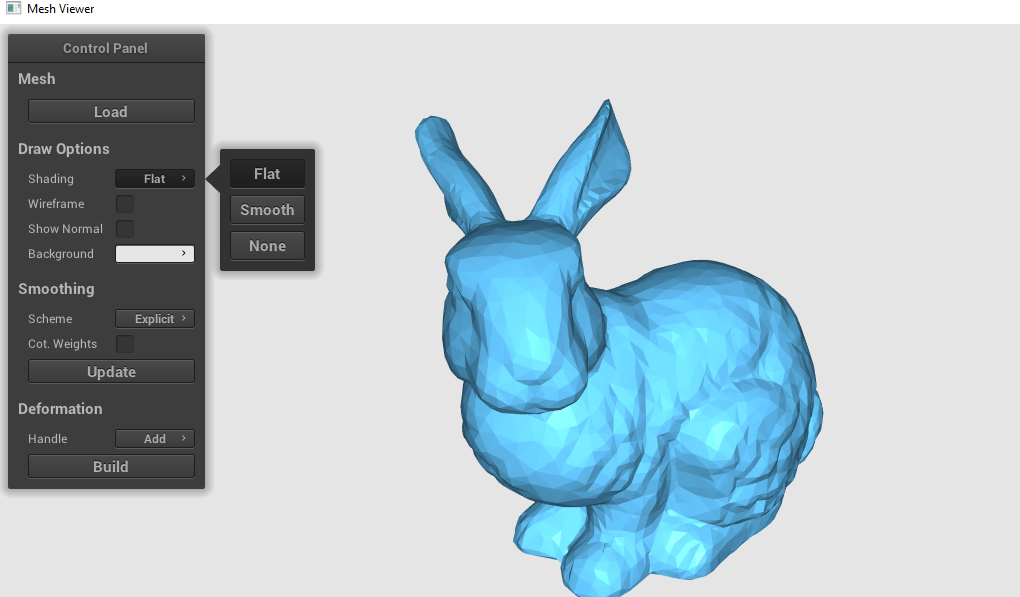
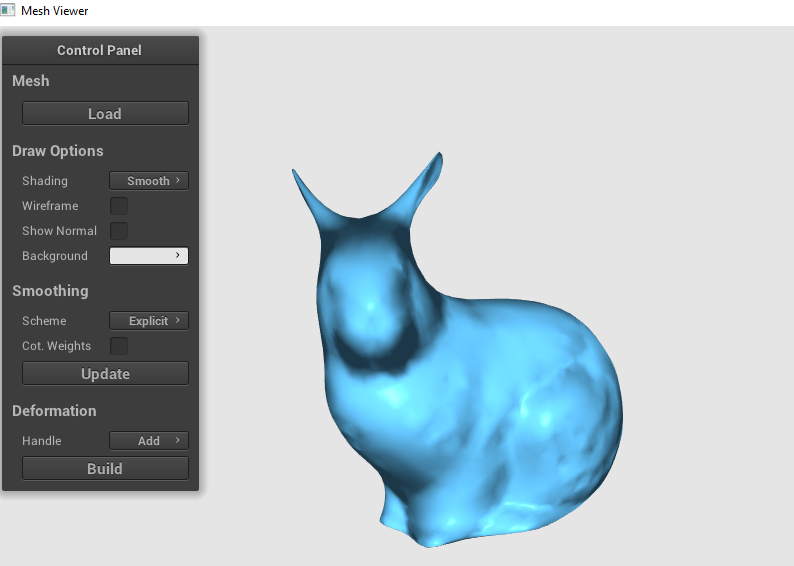
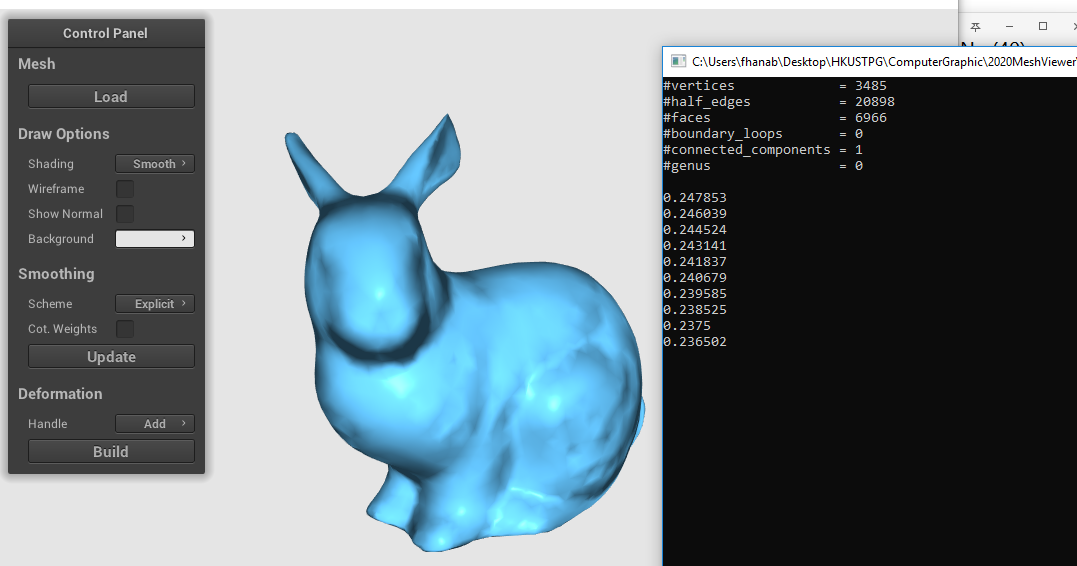
FENG HAN 20407369

1. Explicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30

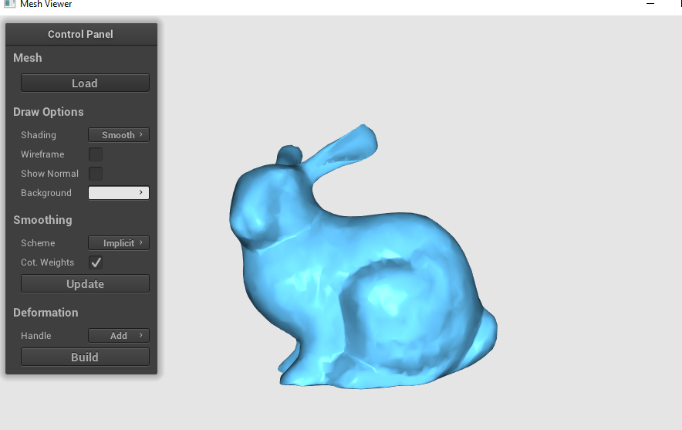
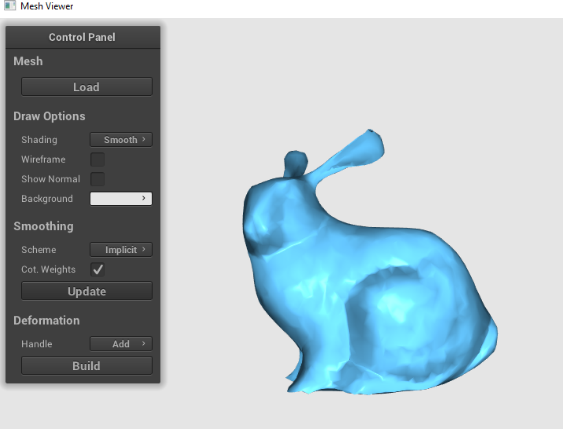


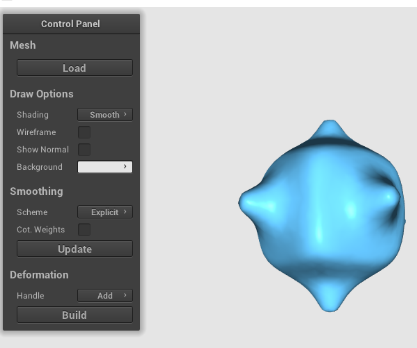
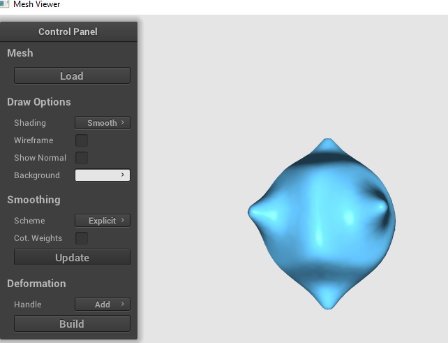
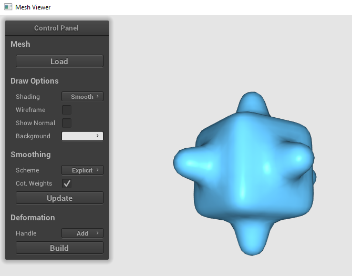
Explicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30



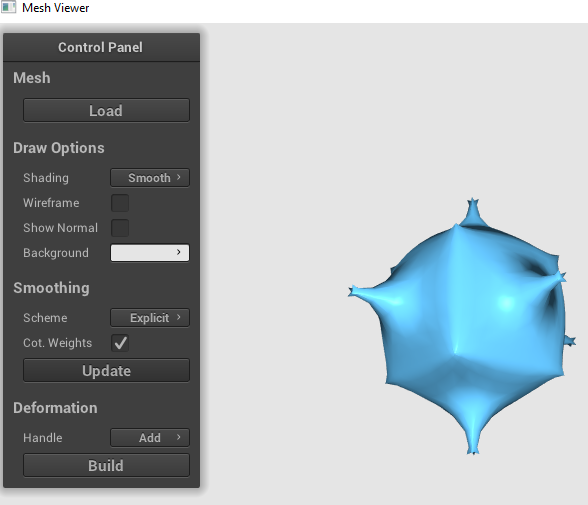
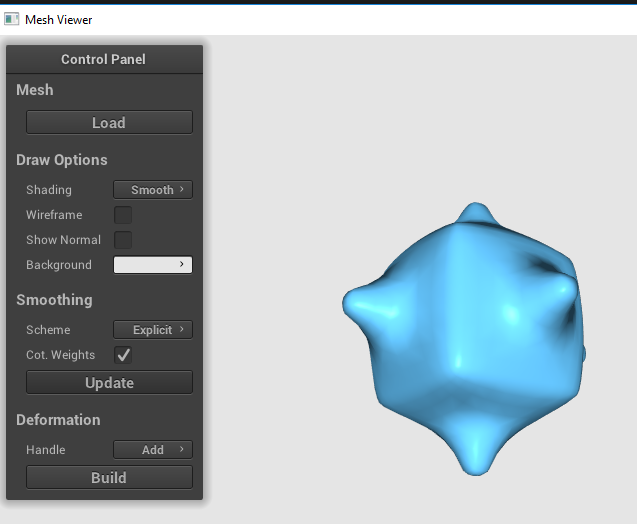
1. Implicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30

Implicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30



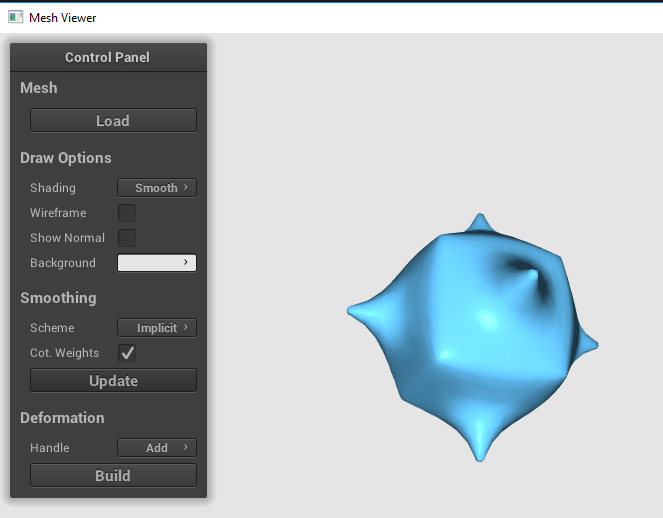
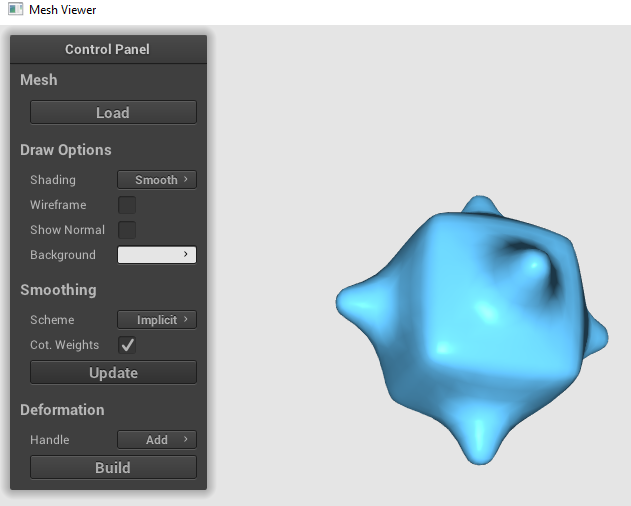
1. Explicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30
2. 

Explicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30

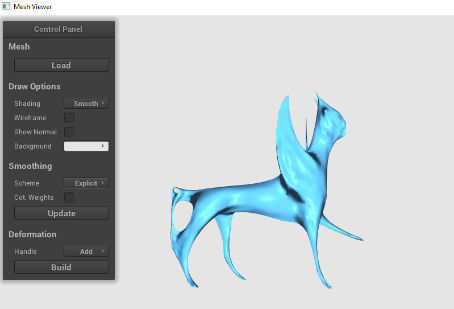
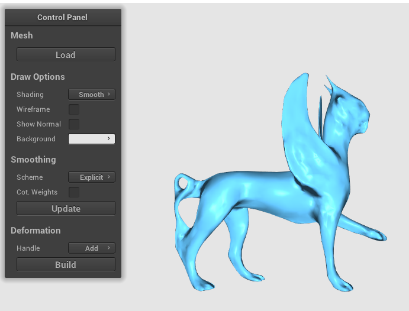
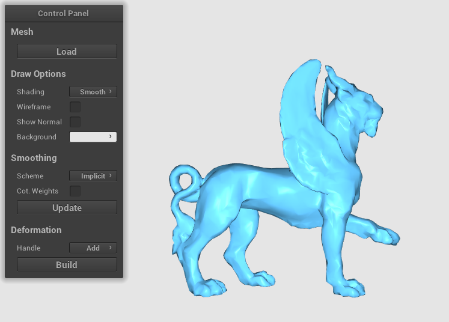


Implicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30

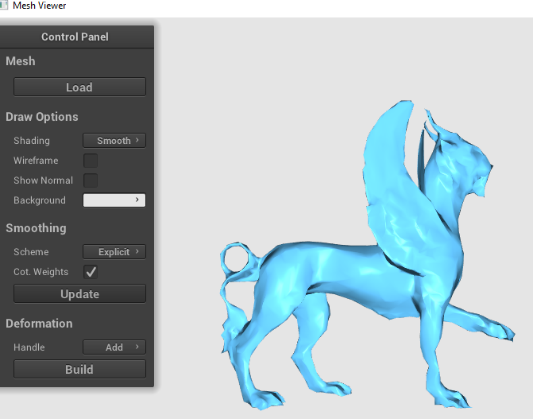
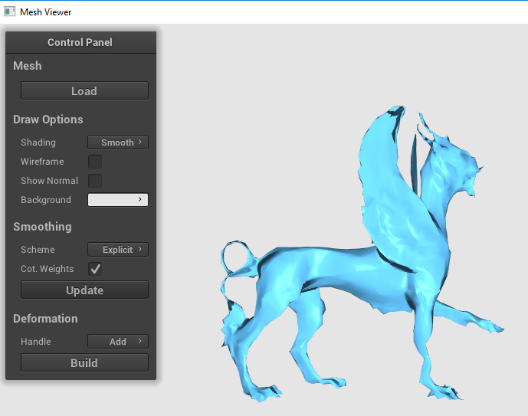
Implicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30



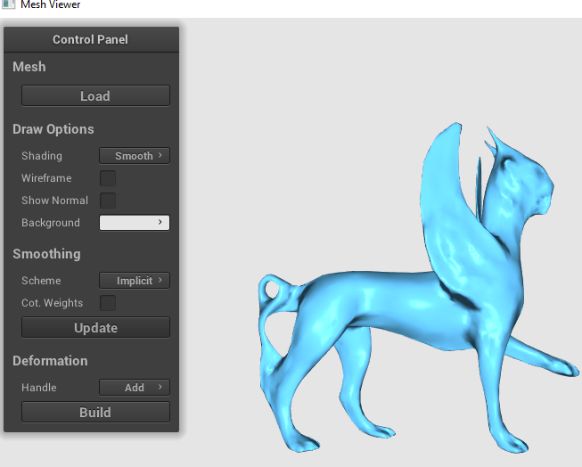
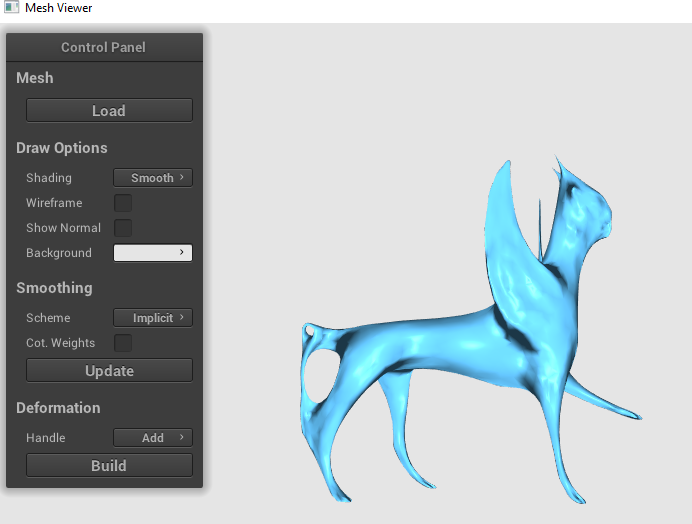
1. Explicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30



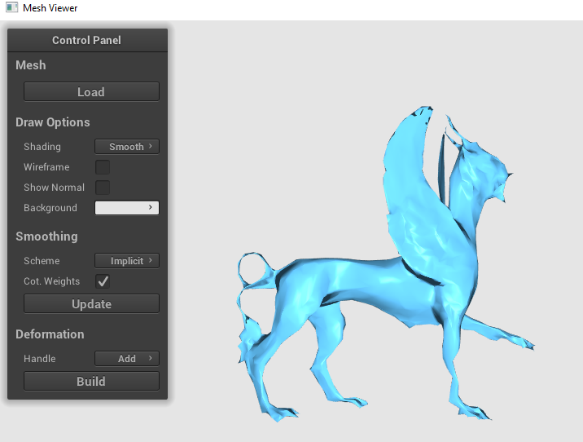
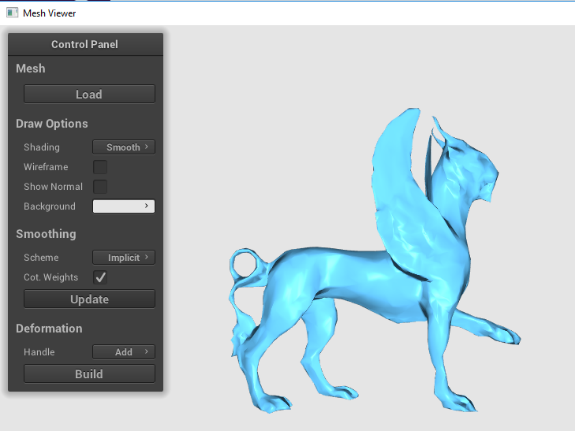
Explicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30



1. Implicit Umbrella Smooth: Uniform-weight Laplacian iteration 0, 10, 30



Implicit Umbrella Smooth: Cot-weight Laplacian iteration 0, 10, 30



I feel that the implicit methods cost less time for the same smoothing results.  
If lambda is bigger, then the smoothing process will become faster. However, it will  
become easier to shrink to a point

After smoothing for some number of times, both of the explicit and implicit Laplacian smoothing schemes are  
able to smooth the surface of the objects. Compared with uniform weight smoothing, cotangent weight  
smoothing seems to perserve the shape of the objects more, which may also introduce more minor sharp places  
after smoothing for many times.  
Implicit smoothing, compared with explicit smoothing, seems to be better in make the sharp detail more sleek,  
which can be seen from bumpy cube smoothed with cotangent weight.  
The value of lambda seems to have effects on how much the vertices move, the larger the lambda is, the more  
significant shape changing could be observed.  
The linear system solver seems to be able to get good result like error smaller than 1e-7 for a low number of  
iterations.

Using uniform weight and cotangent weight will both cause obvious shrinkage after applying Laplacian smoothing a number of times. Using cotangent weight helps to reserve the shape to some extent. In the implicit scheme, the smoothing process is more stable. Implicit methods are more time consuming than explicit methods because they solve vertex positions iteratively. The maximum iteration is set to 200 but increasing this value will not give an obviously better result. In the experiment the difference is about 0.01 second, which is not noticeable for human. The sparseness of Laplacian matrix makes implicit methods efficient enough. Running time depends on the complexity of objects. For example, cube\_bumpy.obj (with 1250 vertices) has the least running time, while the feline.obj (with 4176 vertices) has the most running time.