Virtual Material Deposit and 3D Volume Preserving Smoothing

CS 6491 Project 2: Phase 3

By: Anil Rohatgi
Gtg618b
Georgia Institute of Technology

Introduction:

The purpose of this phase of project 2 was to modify the 3d triangle mesh engine created in the previous phases, to implement a virtual material deposit with 3D volume preserving smoothing. The user should be able to click on a region of a flat triangular mesh, locally subdivide the region, and raise the selected area in a smooth manor. The more the user clicks, the more triangles are added, and the smoother and higher the region gets. However, the smoothing performed on these raised sections is written so the interior volume of each material deposit is not affected by the smoothing. Using this technique, a "terrain" like model can be obtained easily, as demonstrated in figure 1 below.

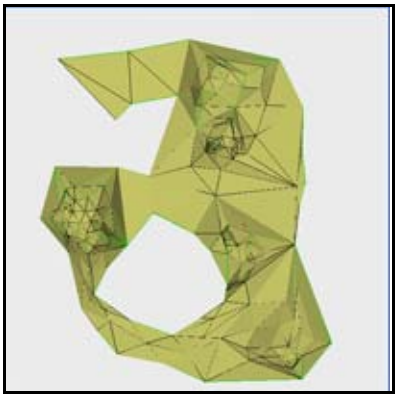

Figure 1. Top view of terrain model

Relaxation:

The first step in producing this program was to take the subdivision abilities from phase 2, and make them applicable to a 3D environment. A relaxation method was used on the 2D flat triangular mesh, before any rising could be done, in an attempt to make the subdivision uniform over the local region. This way, the lift could perform over a relatively homogenous region. The relaxation in 2D was done using a laplacian smoothing algorithm, where an interior vertex would move a fraction of the distance towards the average of its neighbors with every iteration. The results of this relaxation algorithm can be shown below.

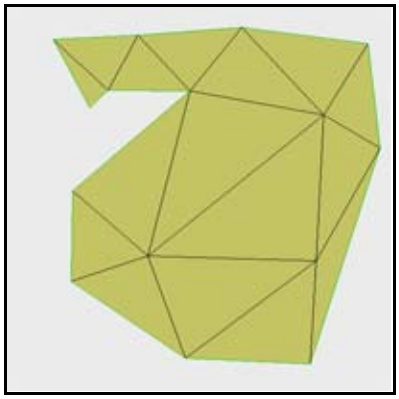

Figure 2 Mesh before subdivision and relaxation. Note how uneven the triangles are in the upper right hand quadrant

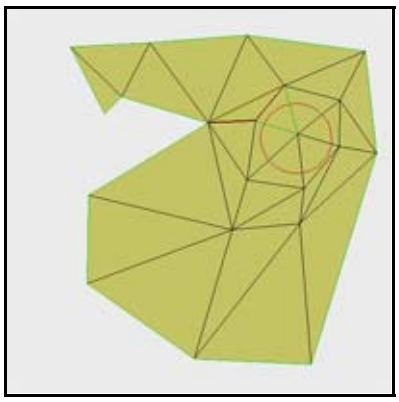

Figure 3. Subdivided mesh. Note how evenly sized the subdivided triangles have become after relaxation

It should also be noted that this technique tends to shrink border vertices of the mesh very quickly. It is for this reason, that the border vertices were moved using the area preserving smoothing algorithm designed on project one.

Raising and Smoothing:

Once the relaxation of the mesh has taken place, the vertices can be raised in a more uniform fashion. The first attempt at this additive technique was nothing more than adding a constant value to the Z coordinate of each vertex within the control region. However, this created unpleasant steep sharp edges between raised vertices and non-raised vertices. It is for this reason that a smoothing algorithm was used to create a more gradual and pleasing elevation.

The first level of smoothing simply extends the two dimensional relaxation technique into three dimensions. The elevation component of the mesh is moved toward the average of its neighbors every iteration, so the mesh tends to level out. Figure 4 demonstrates this. However, as in the case with the 2D triangle mesh, this operation tends to shrink the mesh, and causes it to lose volume. A new technique presented below is an attempt to smooth the mesh, while maintaining a relatively constant volume.

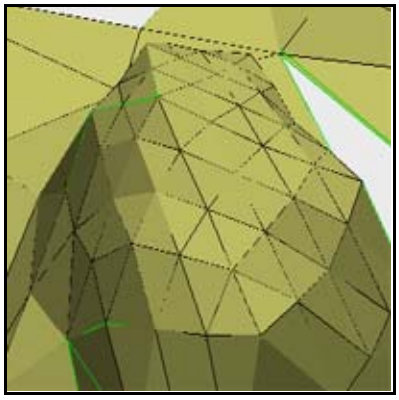

Figure 4. Smoothed material deposit

Volume Loss Calculation:

The final challenge in getting the virtual material deposit to work was to devise a way to smooth the raised regions, without compromising the internal volume of the shape.

The first thing that needed to be done is to calculate the volume lost by each vertex in the smoothing attempt. The configuration of a moving vertex toward the average of its neighbors is shown below in figure 5.

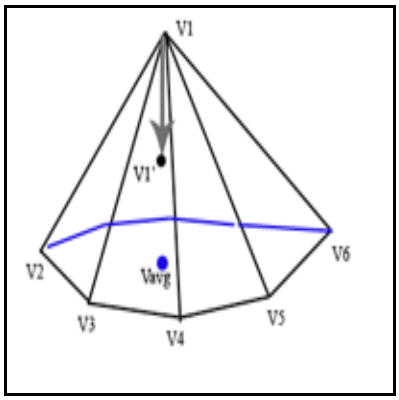

Figure 5. Typical vertex configuration for smoothing. Note V1' is the new position of a vertex after moving toward its neighbor's average

Using this configuration, a rough estimate of the volume lost by smoothing can be calculated. This is done by taking each adjacent triangle and calculating the volume lost as the vertex moves toward its new position. This volume can be thought as a swept out space as the triangle's vertex moves from one point to another. The 3d volume that describes this loss is a tetrahedral. Figure 6 shows this configuration.

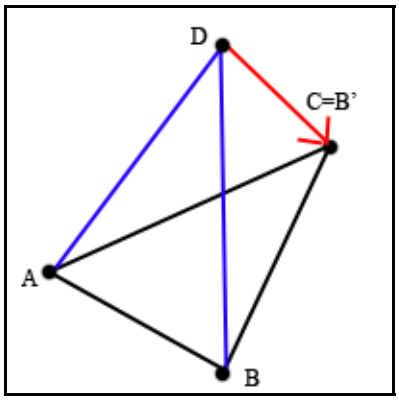

Figure 6. Configuration for a single tetrahedral.

In the figure above, the vertex D is the vertex that is moving, and A and B are the two neighbors that define the blue triangle. Vertex C is the new position of D after it has been moved by the Bi-laplacian. The volume lost as vertex D moves to position C, is the same as the volume of the tetrahedral shown above. The equation to calculate this is included below.

1/6*(AB X AC) * AD = volume

Equation 1.

Each tetrahedral contributes to a fraction of the total volume lost, so these differential volumes are calculated and summed up for every triangle incident upon vertex D.

Volume Loss Restoration:

Now that the volume lost is known, measures must be taken to compensate for the loss. The method proposed is as follows. The volume lost within each tetrahedral is calculated and stored. The original volume bounded by the triangle, the original vertex position, and the average of the vertex neighbors is also stored. A ratio can be found between these two volumes as

(new bounded volume)/ (old bounded volume).

Next, this same ratio is applied to move vertex B in a direction pointing away from the average position of vertex D's neighbors. This operation is analogous to increasing the radius of a cone as the height decreases to compensate for the volume loss. Vertex B was chosen, because moving its position only effects vector AB in equation 1 and therefore by shifting its position based upon the volume ratio directly compensates for the volume lost. Figure 2 shows this procedure graphically. A pseudo code algorithm is also provided below.

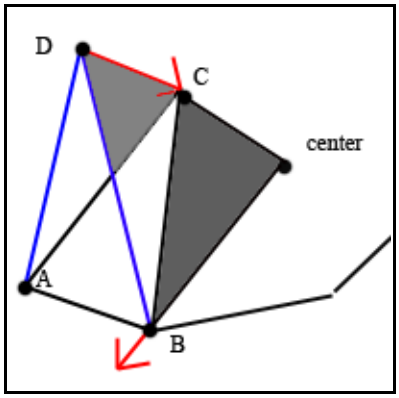

Figure 7. Graphical representation of the volume preserving algorithm.

```
For each incident Triangle
{
        OV = Calc Original Vol;
        VL =Calc Vol Lost;
        Ratio = (OV-VL)/OV
        Vertex B+= Ratio/2* (Center-B);
}
```

Table 1. Pseudo Code for Volume Preserving Algorithm

The results of this operation cause the regions of deposits to bulge slightly to compensate for their contraction due to the smoothing algorithm. Iteratively this approach causes a reaction counter-reaction scenario between smoothing and bulging to try and maintain the interior volume of the shape.

Results and Figures:

Included below are a couple examples of the program in action.

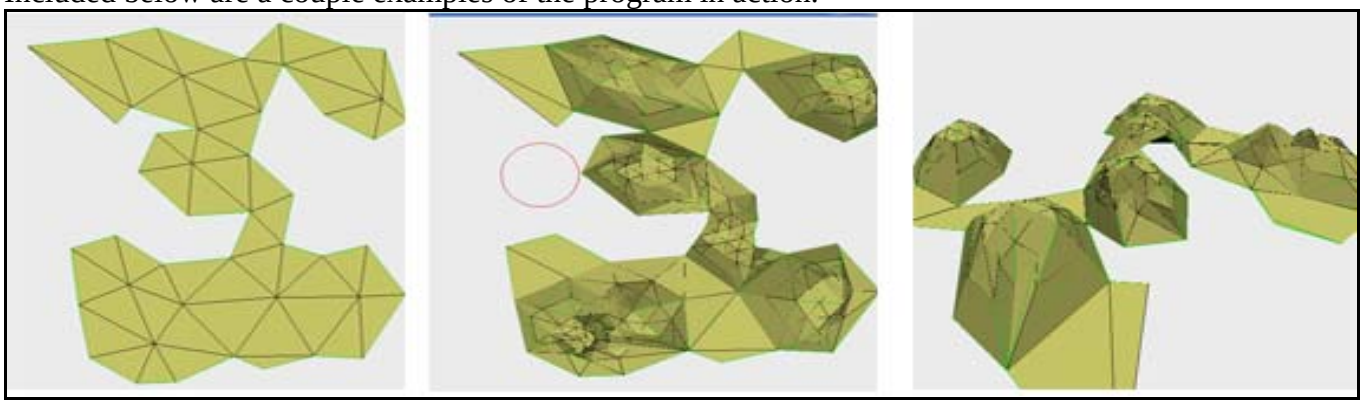

Figure 8. A demonstration of the blob deposit modeling. Image 1 is the flat triangle mesh. Image 2 is after material deposit. Image 3 is a perspective view of the 3D model.

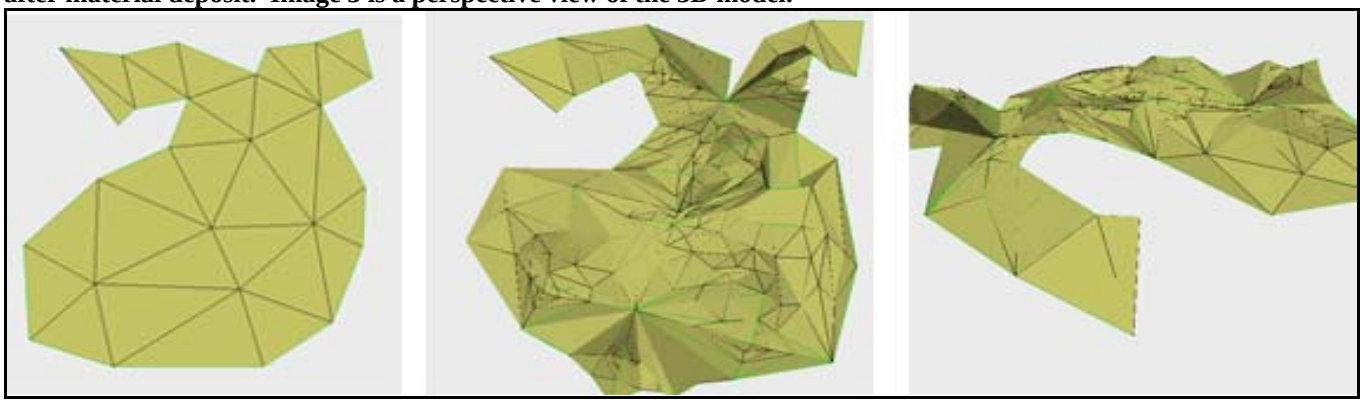

Figure 9.Another demonstration of the blob deposit modeling. Image 1 is the flat triangle mesh. Image 2 is after material deposit. Image 3 is a perspective view of the 3D model.

Problems encountered and Improvement Suggestions:

Again, the majority of the problems implementing this algorithm did not lie within the thought process, but rather within the code. The array index issues I have encountered since phase one has remained and the consequences are becoming more severe. One major issue with my implementation is after a certain number of clicks upon the mesh, sections of it will disappear. My belief is that this is due to an array indexing error where the program points to a null entry and does calculations based upon it. However, this problem only arose very late in the game, and despite many hours of debugging and code searching, I was unable to locate the source of this error. Given more time, this error could be fixed, and the algorithm would be fully functional.

As far as the algorithm is concerned, a further investigation could have been in using not only vertex B to compensate for the volume loss, but to use a combination of both vertex A and B to more evenly distribute the volume change. This implementation would require much more math, because changing vertex A affects all of the vectors in equation 1.

Another place that could be improved in my program is a way to interactively change parameters such as radius of influence and iteration numbers. However, again, due to time constraints I was not able to add these features.