3D Displacement Map based on Surface Curvature

Hanyoung Jang † and JungHyun Han ‡

College of Information and Communications, Korea University, Seoul, Republic of Korea

Abstract

The next-generation graphics hardware supports tessellation. As a result, per-vertex displacement mapping integrated with tessellation will be widely used in real-time applications. The traditional displacement mapping techniques based on the height map have limitations in reconstructing complex surfaces such as self-occluded surfaces. Further, in order to combine tessellation and displacement mapping, the degrees of tessellation should be determined by analyzing the features of the original surface. This paper proposes a novel 3D displacement mapping technique based on mean curvatures of surfaces, which overcomes the limitations of the existing techniques. The vertices of the original surface are parameterized to the texture space using the curvature-based algorithm, and then the 3D displacement map is constructed by calculating the displacement vectors from the coarse mesh to the original surface. The map generated by our method has the effect of the adaptive tessellation with no additional stage for determining LODs. The method can reconstruct self-occluded surfaces and high-quality silhouettes. The proposed algorithm runs as fast as 1D displacement mapping.

1. Introduction

Displacement mapping has been widely used for representing the surface details. The most popular texture used for displacement mapping is the height map. This paper proposes to use a *3D displacement map*, which contains the 3D displacement vectors instead of the 1D height values. A notable advantage of 3D displacement mapping is that highly-detailed and complex surfaces can be reconstructed using a coarse mesh and moderate tessellation. For example, a quad is used as the coarse mesh for reconstructing the complex surface in Figure 1.

The method proposed in this paper generates a 3D displacement map by analyzing the features of the original surface, and restores the features using a small number of polygons without a complex distance error metric. The method does not suffer from the stair-stepping artifacts shown in 6, which can be easily observed in the traditional displacement mapping techniques.

Compared with existing algorithms for displacement mapping, the method proposed in this paper has the following strengths in expressing highly-detailed surfaces.

Figure 1: *Self-occluded objects.*

- **High performance**: Unlike ordinary surface reconstruction algorithms based on the expensive adaptive tessellation, the proposed algorithm simply uses a fixed low-LOD tessellation. The surface features such as sharp edges and silhouettes are recorded in the displacement map, and they are neatly restored in the surface reconstructed using the fixed LOD.

† jhymail@gmail.com
‡ jhan@korea.ac.kr

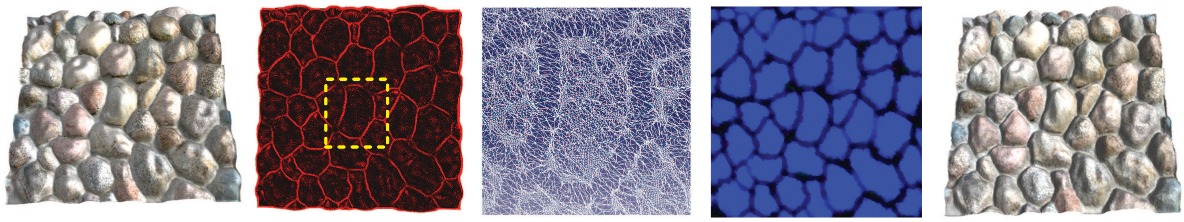

Figure 2: *The left-most figure shows the input surface. The second figure illustrates the curvature values, where a lighter red color represents a higher curvature. The third figure shows the parametrization result for the yellow rectangular area of the second figure. The fourth figure shows the 3D displacement map. The right-most figure is the reconstructed surface through 3D displacement mapping.*

- **Effectiveness**: The proposed method can reconstruct a complex surface from an extremely coarse mesh such as a quad or a box. Such a coarse mesh alleviates the tedious work at the modeling stage, and also is free from the problem of the limited bandwidth between CPU and GPU.

2. Algorithm Overview

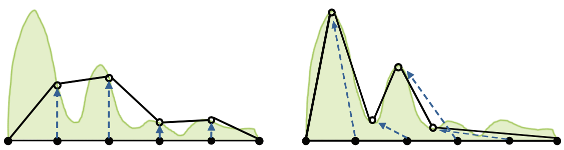

Figure 3: *1D displacement mapping often fails to reconstruct the original surface unless a high degree of tessellation is supported. In contrast, 3D displacement mapping with a 3D vector map can reconstruct the surface with a moderate degree of tessellation.*

Our goal is to reconstruct the original highly-detailed surface with a moderate degree of tessellation with the aid of curvature-based 3D displacement map, as illustrated in Figure 3. This section overviews the proposed approach.

The input to our system is the original mesh of a highly-detailed surface. The algorithm proposed in this paper creates the 3D displacement map through three stages: vertex-weight estimation, parametrization, and map generation(Figure 2).

The first stage computes the weight of a vertex using the surface curvature of the vertex. In general, the vertices of high curvatures define the main features of the surface. Therefore, it is important to restore those vertices for reconstructing the original surface. Section 2.1 gives a detailed description on this issue.

In the second stage, a mapping function from the original surface to the texture domain is calculated through parametrization. In the process of parametrization, the position of each vertex is changed in order to possess the area proportional to its surface curvature. The parametrization result in Figure 2 shows that the vertices located between rocks have high curvatures and therefore are sparsely positioned. Section 2.2 discusses the parametrization process in detail.

In the third stage, the mapping function from the coarse mesh to the original surface is computed, i.e. for each texel of the displacement map, a displacement vector is calculated at the tangent space of the coarse mesh. Section 2.3 presents this stage.

2.1. Vertex Weight Estimation

Our goal is to assign a weight value to a vertex based on the visual importance of the vertex. The vertex weight depends on the Laplacian of the surface. Note that the mean curvature value is proportional to the Laplacian of surface. Therefore, we adopt the mean curvature.

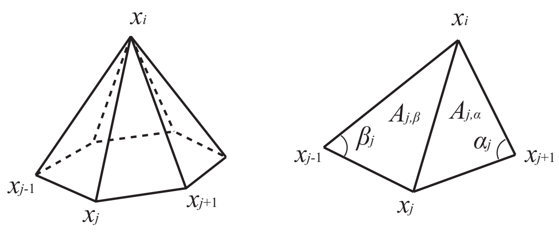

Figure 4: *1-ring neighbors and notations.*

To derive the mean curvature in a triangular mesh, the 1-ring neighbors of each vertex are considered. Figure 4 shows notations needed for estimating the mean curvature of x_i.

$$\bar{\kappa}\mathbf{n} = \frac{\nabla A}{2A} = -\frac{1}{4A}\sum_{j\in N_1(i)}(\cot\alpha_j + \cot\beta_j)(x_j - x_i). \quad (1)$$

The mean curvature is calculated using Eq. (1) which is called *curvature normal equation* [DMSB99]. In the equation, $\bar{\kappa}$ stands for mean curvature, $\mathbf{n}$ is the surface normal, A is the sum of the areas of the triangles sharing vertex x_i, and ∇ is the derivative with respect to the (x,y,z) coordinates of x_i.

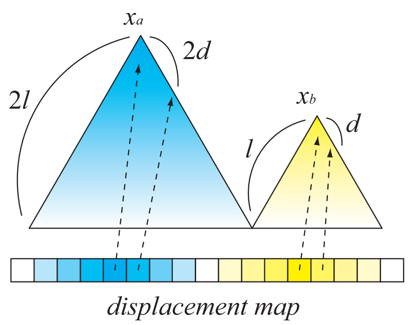

Figure 5: *A pixel-wide error leads to different amounts of geometric error,* 2*d and d.*

We do not use the mean curvature itself as the weight of a vertex because it does not consider geometric error. In Figure 5, the left triangle is scaled twice from the right one. If the *vertex area* depends only on its curvature, x_a and x_b have the same area in the texture space. Suppose that displacement mapping is performed using regular sampling. Then, the same distance error in the texture space results in different amounts of the geometric error, as shown in Figure 5. Therefore, the vertex weight should be determined considering the geometric error that will be produced when the vertex is reconstructed.

In reality, however, it is quite difficult to evaluate the amount of geometric error. Worse, there is no known method to take the geometric error into account when we determine the vertex weight. In our method, we choose an *approximate* solution, i.e. we multiply the mean curvature by $4A$. Note that, as a result, A is removed from Equation (1). It is a reasonable solution since the remaining term ∇A represents the area difference caused by the position errors of x_a and x_b. Finally, the vertex weight m is estimated using the following simplified equation:

$$m_i = 4A\bar{\kappa} = -\mathbf{n}\cdot\sum_{j\in N_1(i)}(\cot\alpha_j + \cot\beta_j)(x_j - x_i). \quad (2)$$

2.2. Parametrization

In order to save the information of the original surface into the 3D displacement map, the one-to-one mapping function from the original surface to the texture space should be defined. One of the most popular parametrization techniques is *angle-preserving parametrization*. It is suitable for texture mapping and regular sampling-based remeshing.

Recall that, in our work, we preserve the vertex area in proportion to its weight. If a vertex from the original surface occupies a large area in the displacement map, many vertices of the tessellated mesh will move toward the vertex position. The angle-preserving parametrization does not serve this purpose. Therefore, we have developed our own parametrization technique which is suitable for our purpose.

As a preprocessing step, we use a traditional parametrization method, such as those presented in [Flo03], to convert the 3D vertex positions of the original surface into 2D points in the texture space. Then, two algorithms are alternately applied until the position differences of vertices are kept under a threshold value.

The first algorithm is based on a simple edge-spring method. Each vertex pushes away the 1-ring neighbor vertices. The repulsive force is in proportion to the vertex weight. The advantages of this method are that it is simple to implement and guarantees fast convergence of the vertex positions.

The second algorithm updates the vertex positions using a triangle-area criterion. Note that, in the ideal texture space, a vertex is assigned an area in proportion to its weight, i.e. the distance between two vertices is proportional to the sum of their weights. The area of the ideal triangle is computed using Heron's formula(Equation (3)).

$$A_{\triangle} = \sqrt{s(s-m_a-m_b)(s-m_b-m_c)(s-m_c-m_a)}, \quad (3)$$

$$\textit{where } s = \frac{m_a+m_b+m_c}{2}.$$

The area of the ideal triangle will be compared with that of the corresponding triangle in the texture map. Note that the texture map's (u,v) coordinates are in the range of [0,1]. Therefore, the area of the ideal triangle is normalized.

The second algorithm compares the normalized area of the ideal triangle with the area of the current triangle in the texture map. If the area of the current triangle is larger, it has to be made smaller. In the current implementation, the vertex on the opposite of the shortest side is moved along the longest edge. It makes the triangle converge to an equilateral triangle, and eventually helps to prevent the reconstructed surface from being distorted.

2.3. 3D Displacement Map Generation

Through the parametrization stage, the mapping function from the original surface to the texture domain has been derived. Let us call by f its inverse function. At the modeling stage, the coarse mesh is usually generated by using mesh simplification algorithms [GH97]. The mapping function from the coarse mesh to the texture domain is also defined at the modeling stage when each vertex is assigned the texture coordinates. Let us call by g its inverse function.

Given f and g, the mapping function h from the coarse mesh to the original surface is directly obtained, i.e. the 3D displacement map is generated by evaluating the vectors from the coarse mesh to the original surface. Note that the displacement vectors should be defined in the tangent space of the coarse mesh.

3. Implementation and Conclusion

The proposed algorithm has been implemented in Direct3D 10 with shader model 4.0 on a PC with 2.4GHz Intel Core2 CPU and 2GB memory. The PC is equipped with NVIDIA GeForce 8800GT which has a 600MHz core, 512MB memory, and PCI-Express 2.0 interface.

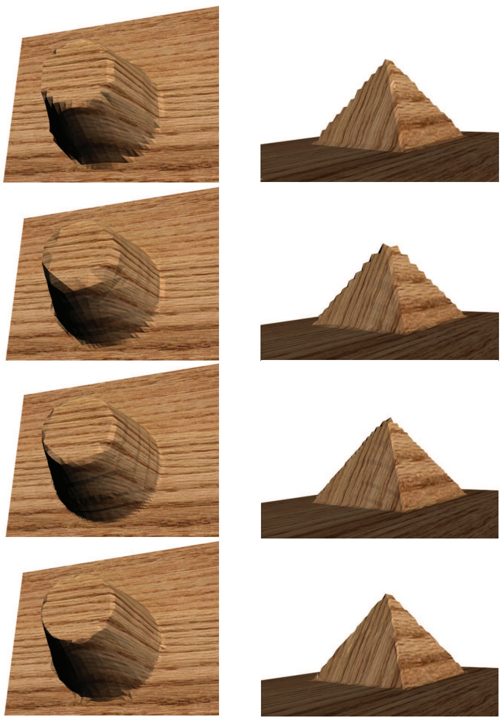

Figure 6: *Displacement mapping results: 1D displacement mapping (first row), conventional 3D displacement mapping (second row), conventional 3D displacement mapping with a highly-tessellated mesh (third row), 3D displacement mapping based on the curvature-based parametrization (fourth row).*

Figure 6 compares the proposed method with other techniques. The first row shows the results of 1D displacement mapping. The resolution of the height map is 512×512, and the tessellated mesh consists of 12K triangles. The second row shows the results of the conventional 3D displacement mapping [Flo03], where the 3D displacement map is generated using so-called mean value coordinates. The tessellated mesh also consists of 12K triangles, and the stair-stepping artifacts are observed. The third row is obtained using 50K triangles. The quality of the result is improved at the cost of performance decrease.

In contrast, the last row shows the results of our method, curvature-based 3D displacement mapping. The tessellated mesh also consists of 12K triangles, and the reconstructed mesh is quite smooth. The smooth edges are obtained because the vertices of the tessellated mesh are displaced towards the edges of high curvature.

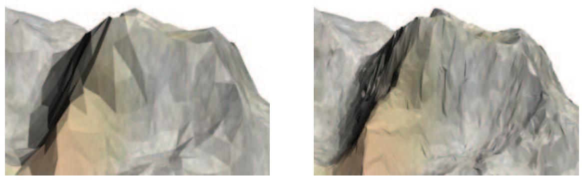

Figure 7: *Zoomed-in views of shape edges.*

Figure 7 shows the results of applying the proposed 3D displacement mapping to a terrain model. Even though the higher resolution(right figure) produces the better result, the high-frequency regions of high curvatures are well reconstructed for low-LOD tessellations(left figure).

Acknowledgement

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency (NIPA-2009-(C1090-0902-0046)).

References

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 317–324.

[Flo03] FLOATER M. S.: Mean value coordinates. *Comput. Aided Geom. Des. 20*, 1 (2003), 19–27.

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 209–216.