Research Report:
Simultaneous minimization of both distance and derivative errors

Hirokazu Yagou

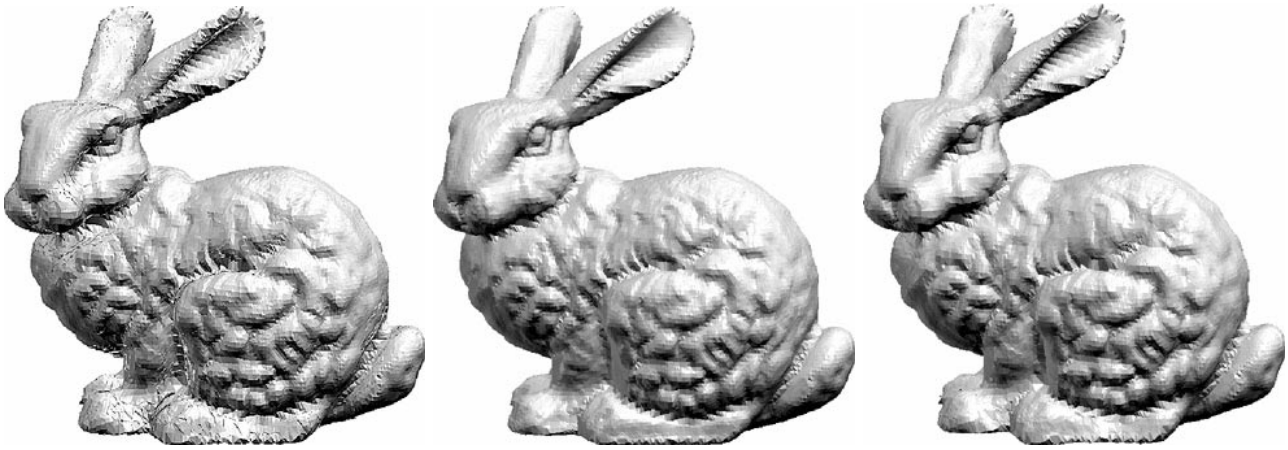


Figure 1: Left: distance error minimization. Middle: derivative error minimization. Right: a combination of the two error minimization techniques

1 Introduction

When I tested unsharp masking applied to mesh normals, tried to update vertex positions using three kinds of error minimization techniques.

1. Distance error minimization,
2. Derivative error minimization,
3. A combination of the two error minimization methods.

The new error minimization method combines the best properties of the previous methods: the distance and the derivative ones. The minimization of distance errors is computationally cheaper than the one of derivative errors, but triangle flipping occurs at the minimization process. Conversely, the minimization of derivative error requires much enough computation. However, the minimization technique restrains induction of the triangle flipping at its process. The developed minimization scheme is computationally cheaper than the derivative error minimization, and also the triangle flipping is restrained at updating vertex positions.

2 Experimental Results

Figure 1 shows the results of unsharp masking processes based on the three minimization methods. The left figure shows the masking result in a case of the distance error minimization. Vertices are set to optimal positions, but the triangle flipping appears on the surface of the mesh model. The middle figure shows the result based on the derivative error minimization. the triangle flipping does not appear on the surface. However, vertices does not reach the optimal positions. The right figure shows a unsharp masked model

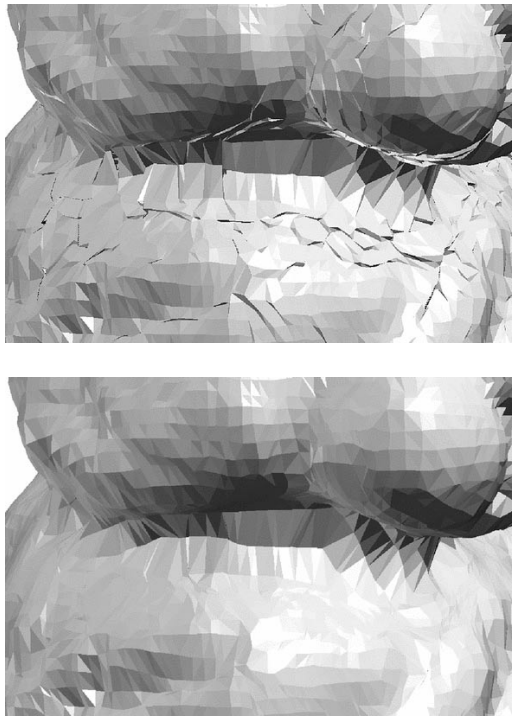


Figure 2: Top: the distance error minimization. Bottom: the new technique.

updated its vertices using the new minimization technique. The vertices reach the optimal positions, and the triangle flipping is restrained. Figure 2 shows enlarged pictures of the bunny's neck part. The top figures shows the masking results based on the distance error minimization. Flipped triangles can be seen. The bottom figure shows the masking results based on the developed new technique. No flipped triangles are shown there.

3 Implementation Details

A updated vertex is defined as P. Let R be a mesh triangle adjacent to P, $C = C(R)$ be the centroid of R, and $\mathbf{m}(R)$ be the filtered mesh normal of R. The distance error at P

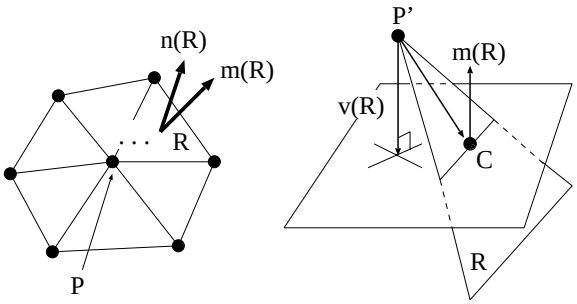


Figure 3: Left: the 1-ring neighborhood of the P. Right: a visual representation of Equation (3)

is given as the following form:

$$E_d = \sum A(R)\big(\mathbf{m}(R)\cdot(P-C)\big)^2 \qquad (1)$$

where the sum is taken over all triangles R adjacent to P, and $A(R)$ denotes the area of R. The new position of the vertex P is computed based on the standard gradient descent method for minimizing the error:

$$P' \longleftarrow P + \frac{1}{\sum A(R)}\sum A(R)\mathbf{v}(R) \qquad (2)$$

$$\text{with} \quad \mathbf{v}(R) = \big(\mathbf{m}(R)\cdot(C-P)\big)\mathbf{m}(R). \qquad (3)$$

Next, I further updated the position of P' using the derivative error minimization. The derivative error at P' is defined as the area weighted sum of squared differences of the original normals $\mathbf{n}$ and the filtered normals $\mathbf{m}$ on incident triangles at P':

$$E_n = \sum A(S)(\mathbf{n}(S)-\mathbf{m}(R))^2 \qquad (4)$$

where S is a triangle on a mesh constructed by vertices P'. In this minimization process, The partial derivative of Equation (4) with respect P' is written as follows:

$$\frac{\partial E_n}{\partial P'} = \sum 2\left(\frac{\partial A(S)}{\partial P'} - \frac{\partial A(S')}{\partial P'}\right) \qquad (5)$$

where S' is a triangle which is obtained by projecting S onto the plane defined by $\mathbf{m}(R)$. The gradient of the triangle area $A(S)$ is computed by the following equation:

$$\frac{\partial A(S)}{\partial P'} = -\frac{1}{2}\big((P_1-P')\cot\alpha + (P_2-P')\cot\beta\big). \qquad (6)$$

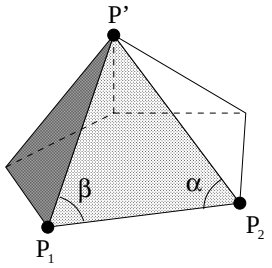


Figure 4: A triangle, S, whose vertices are $\{P', P_1, P_2\}$ and its angles.

The triangle S has three vertices $\{P', P_1, P_2\}$.

The standard gradient descent flow minimizing E_n is defined by

$$P'' \longleftarrow P' - \lambda\sum\left(\frac{\partial A(S)}{\partial P'} - \frac{\partial A(S')}{\partial P'}\right). \qquad (7)$$

According to my experiences, the flow is stable by setting $\lambda = 0.1$.

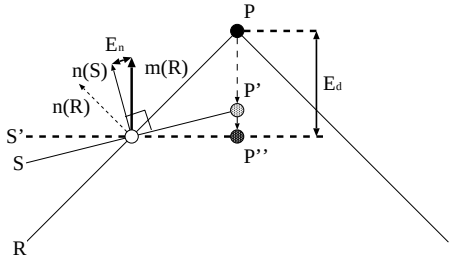


Figure 5: A conceptual figure of the updating process.

4 Discussion

The initial vertices are quickly set to optimal positions using Equation (2) and (3). However, several vertices induce the triangle flipping. Those vertices are regulated by further updating their positions via the derivative error minimization. Consequently, flipped triangles are suppressed.

This method is computationally cheaper than the minimization of derivative errors. To check processing speeds of the new method and the derivative error minimization, I carried out an experiment on the speed benchmark. In the experiment, the Stanford bunny model was used. The number of vertices is 34834, and the number of triangles is 69451. A personal computer that is equipped with a 1.7-gigahertz CPU (Pentium 4) and a 512-megabyte RAM was used in the benchmark test.

The derivative error minimization requires $10 \times k$ iterations to set mesh vertices to optimal positions; k is the number of filtering iterations to diffuse mesh normals. When we use the new technique, it requires only k iterations to set vertices to the optimal positions. The processing time of the derivative error minimization with 10×100 iterations was 128.953 second. The time of the new technique with 100 iterations was 15.157 second. It is apparent that the new technique quickly set vertices to optimal positions with restraining the triangle flipping.