3D Mesh Enhancement via Filtering Face Normals

Hirokazu Yagou

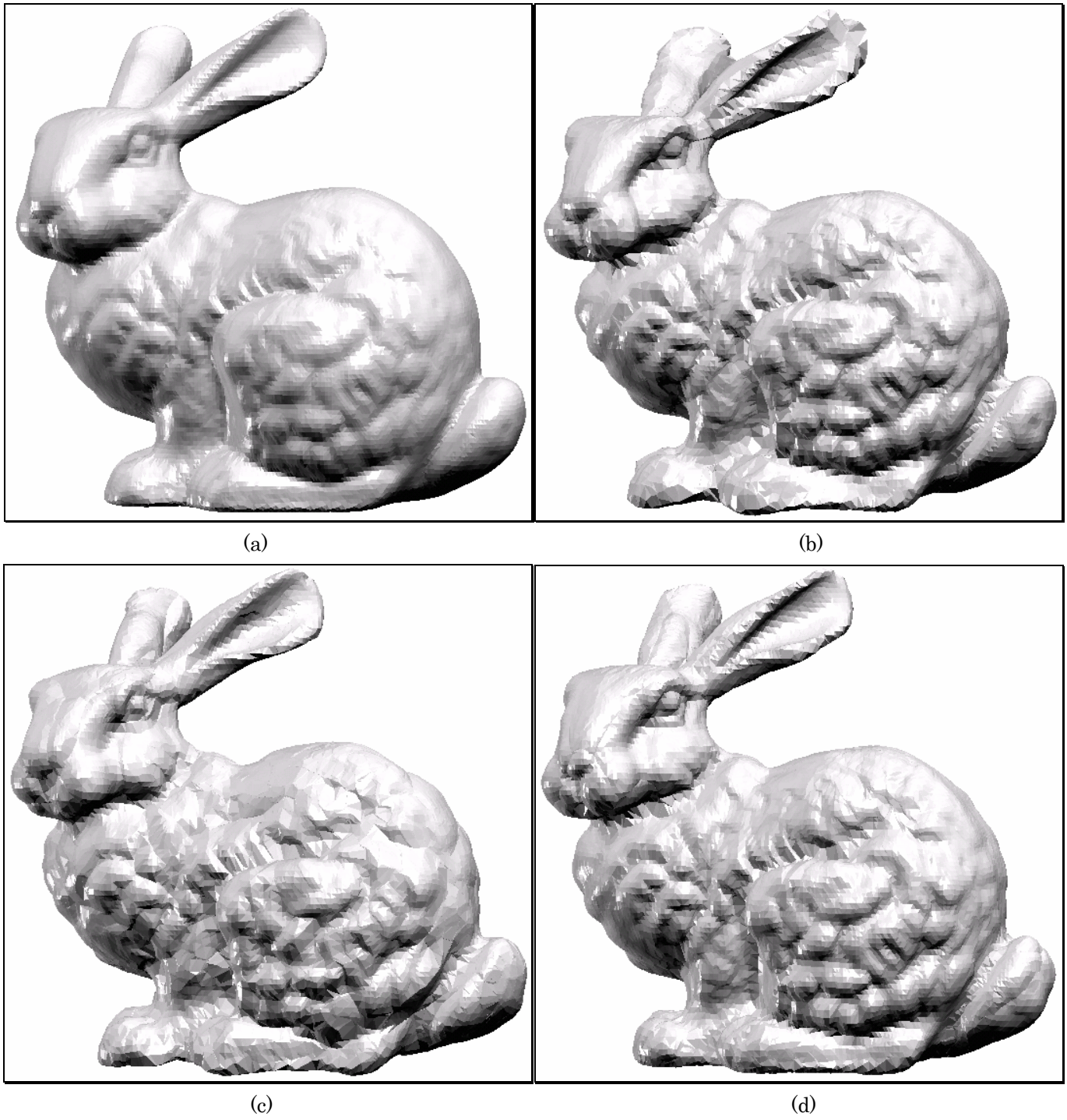


Figure 1. Experimental results of mesh enhancement techniques. (a) Original model. (b) Enhanced by iterative mean filtering. (c) Enhanced by iterative median filtering. (d) Enhanced by nonlinear diffusion.

1. Algorithm

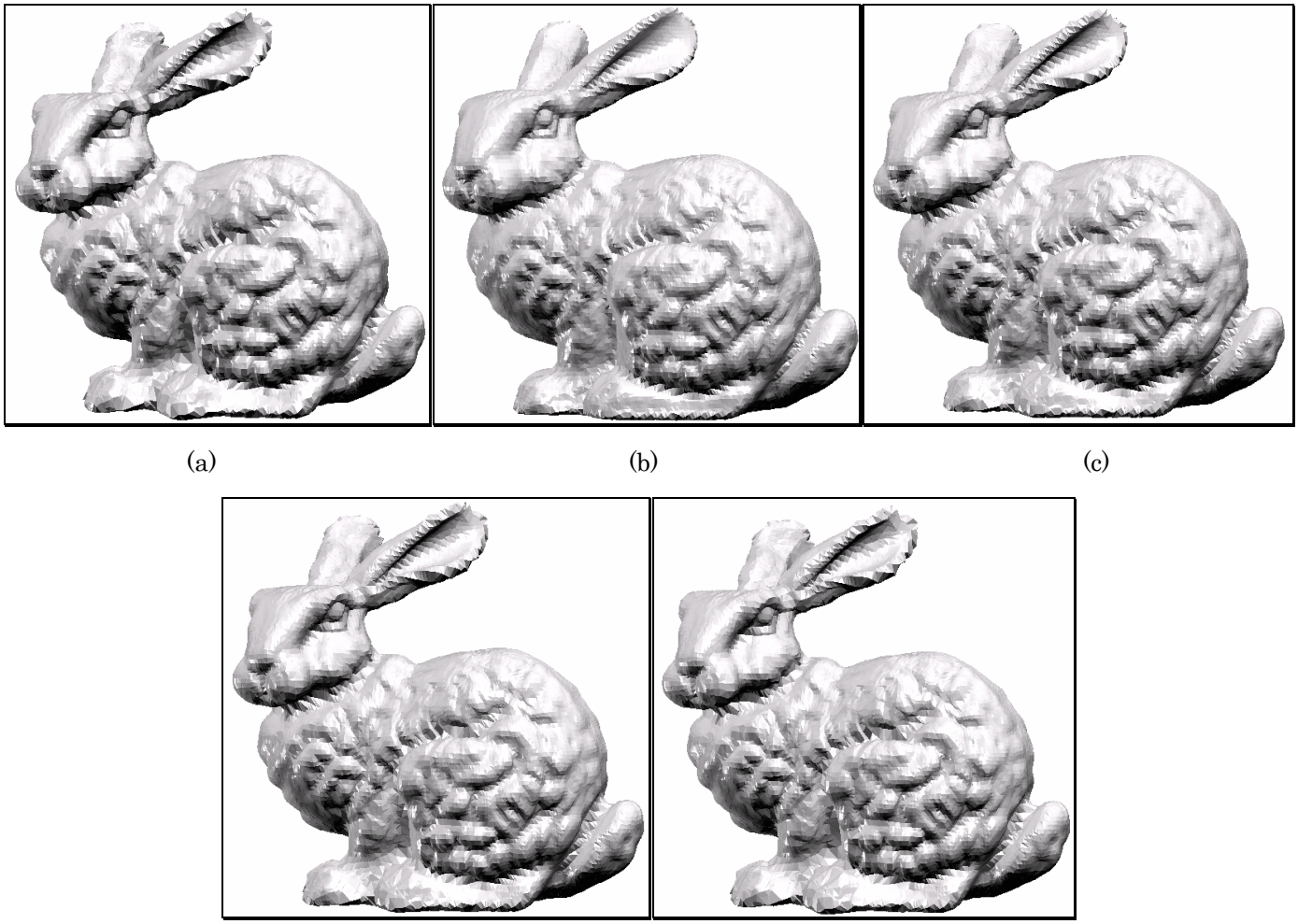
The face normal field $\{N\}$ on a triangle mesh was enhanced by the following rule:

$$M = \frac{(1+a)N - aN^n}{\left\|(1+a)N - aN^n\right\|} \quad (1)$$

where $\{M\}$ is a new face normal field; $\{N^n\}$ is a face normal field smoothed by n iterations; and a is a boost threshold. Vertex positions were updated by the derivative error minimization:

$$\min \frac{\partial E_n}{\partial P}, \qquad E_n = \sum A(T)\big(N(T) - M(T)\big)^2. \quad (2)$$

$A(T)$ is the face area of a triangle T. $N(T)$ and $M(T)$ are the original face normal and a new face normal produced by the rule (1), respectively. For setting vertices to proper positions, the updating is performed by $10 \times n$ iterations. When the derivative error minimization was used for updating vertex positions, such many iterations was required in experiments. This is shown by Fig. 2. The updating by the distance error minimization induced the edge flipping.



(a) (b) (c)

(d) (e)

Figure 2. Updating vertex positions. (a) Only filtering face normals (100 iterations). (b), (c), (d), and (e) are results of updating vertex positions and again computing face normals from those updated vertices. (b) Updating by 100 iterations. (c) Updating by 300 iterations. (d) Updating by 500 iterations. (e) Updating by 1000 iterations.

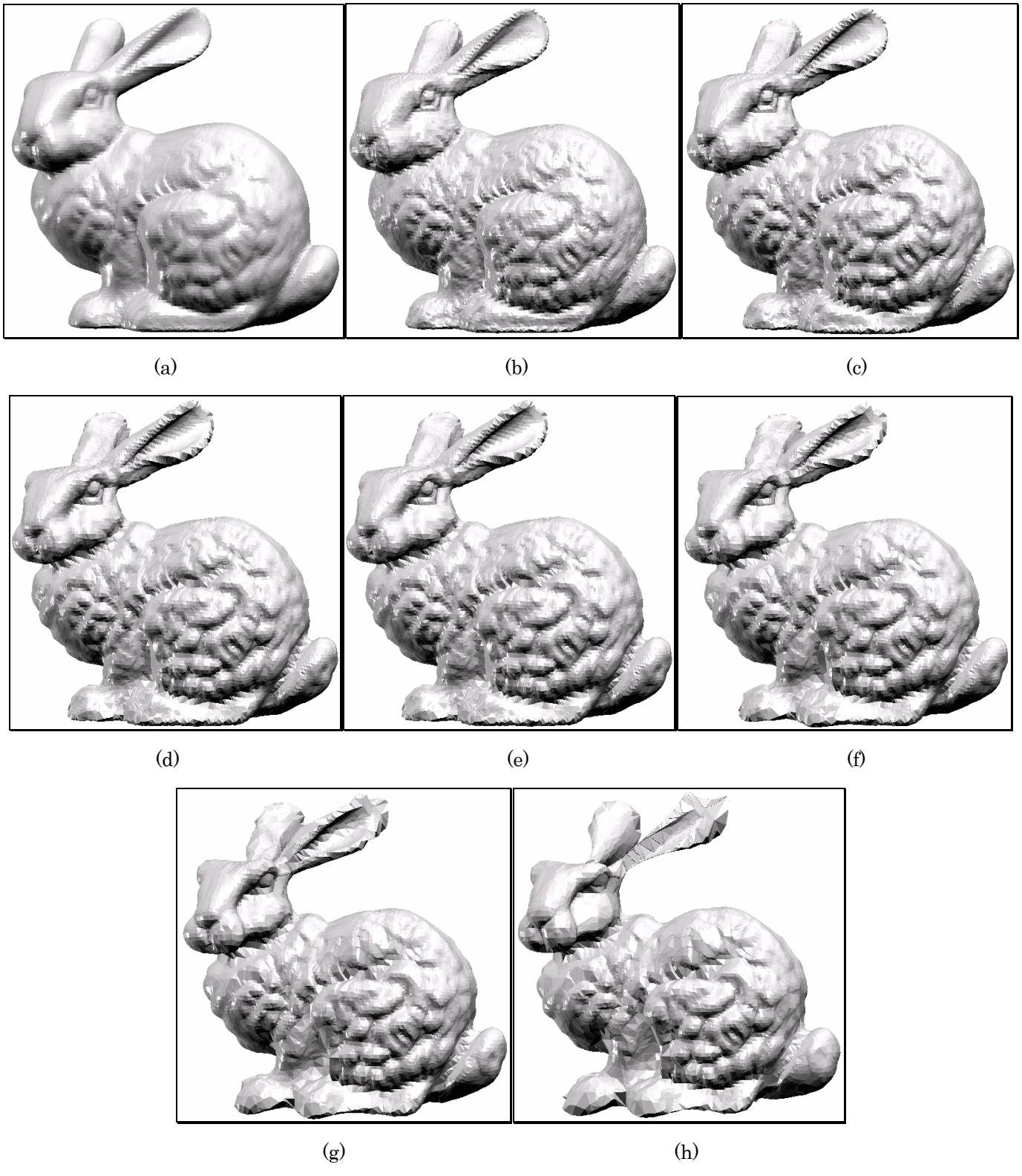
2. Experimental Results

The following experiments were performed:

1. Fix the boost threshold and change enhancing iteration,
2. Change the boost threshold and fix enhancing iteration,
3. Enhancement by three filtering methods,
4. Recover smoothed features by enhancing operation.

At experiment 1, 2, and 4, the iterative mean filtering were used.

2.1 Fix the boost threshold and change enhancing iteration



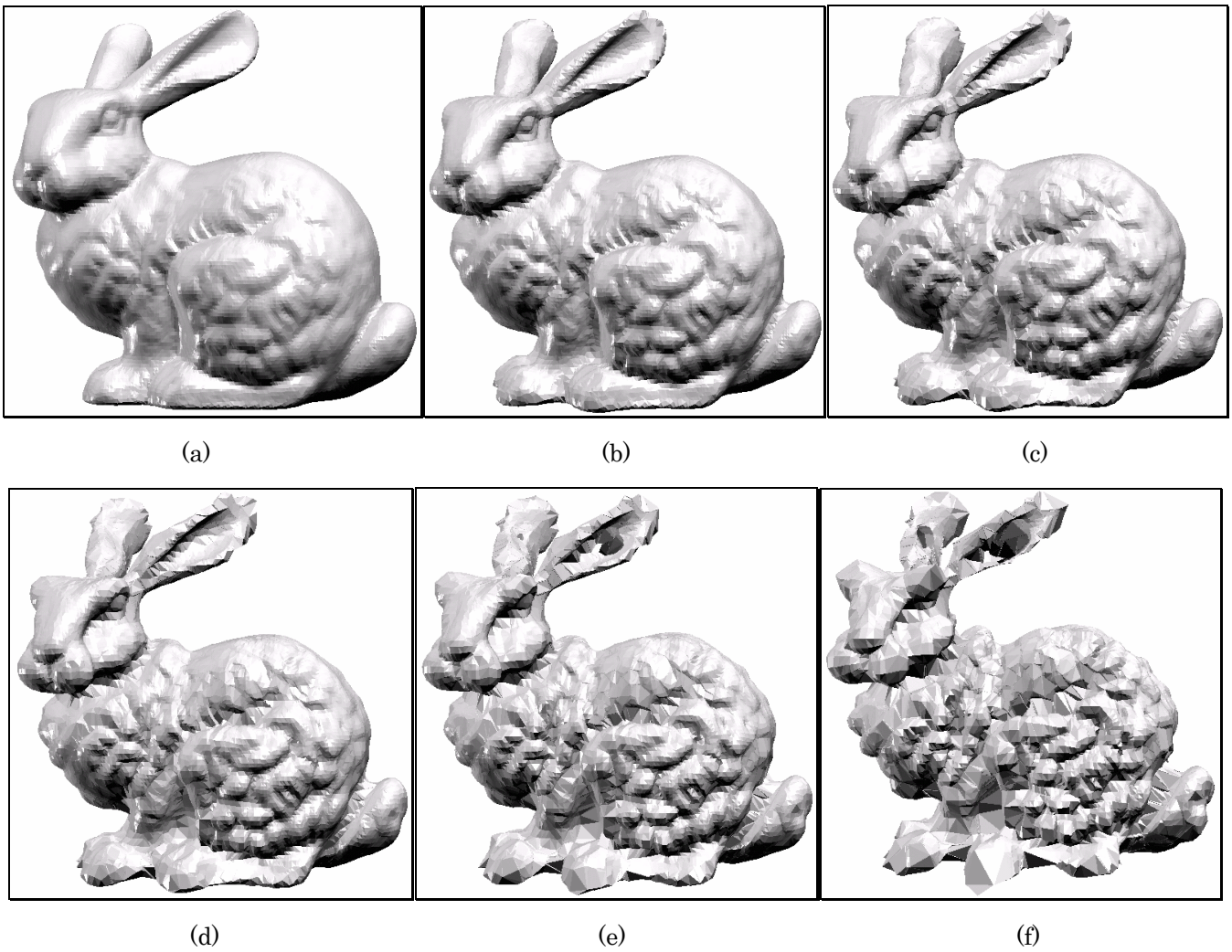
(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 3. The boost threshold was 1.5 for all cases. (a) Original model. (b) Enhanced by 5 iterations. (c) 10 iterations. (d) 20 iterations. (e) 50 iterations. (f) 100 iterations. (g) 200 iterations. (h) 500 iterations.

2.2 Change the boost threshold and fix enhancing iteration



(a) (b) (c)

(d) (e) (f)

Figure 4. The enhancing iteration is fixed to be 200 iterations.

(a) Original model. The boost threshold: (b) 0.5, (c) 1.0, (d) 1.5, (e) 2.5, and (f) 5.0.

2.3 Enhanced by three filtering methods

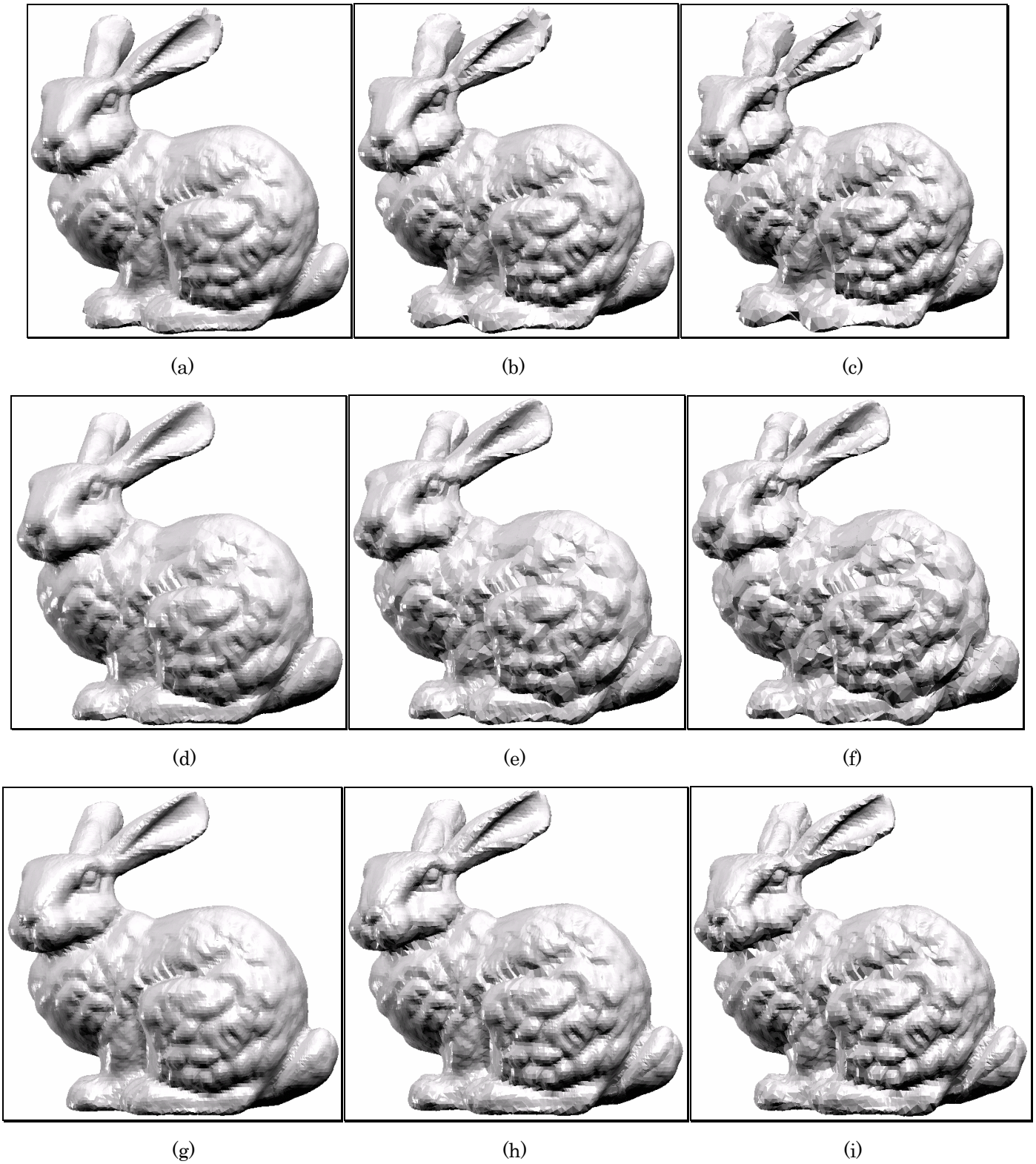


Figure 5. The enhancing iteration is 200 for all cases.

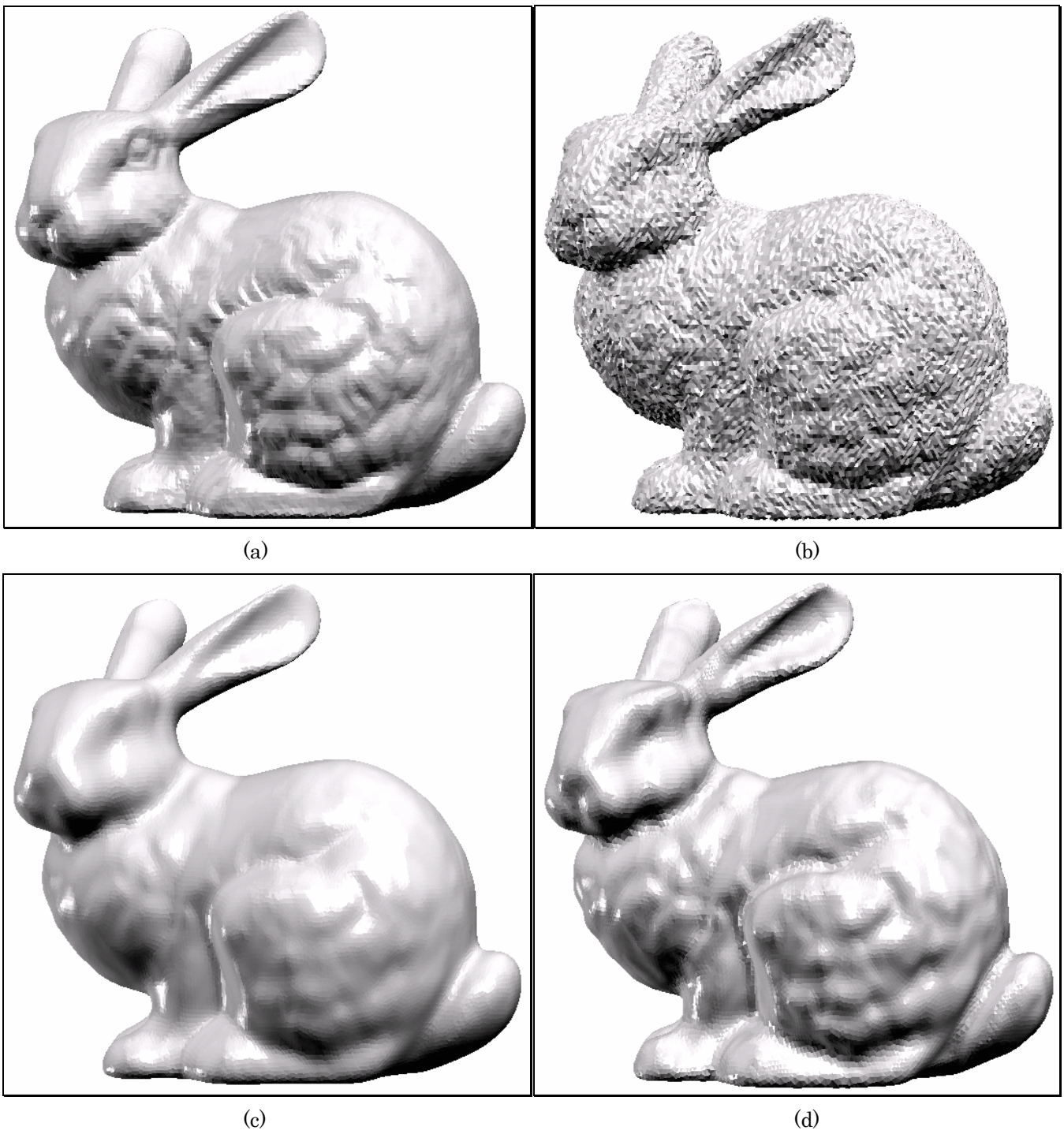
(a), (b), and (c) were enhanced by iterative mean filtering. The boost threshold: (a) 0.5, (b) 1.0, and (c) 1.5.

(d), (e), and (f) were enhanced by iterative median filtering. The boost threshold: (d) 0.5, (e) 1.0, and (f) 1.5.

(g), (h), and (i) were enhanced by nonlinear diffusion. The boost threshold: (g) 0.5, (h) 1.0, and (i) 1.5.

When the nonlinear diffusion was used, the weight function $\frac{1}{1+(k/T)^2}$ was selected (T = 1.0).

2.4. Recover smoothed geometric features by enhancing operation



(a) (b)

(c) (d)

Figure 6. (a) Original model. (b) Noise is added. (c) Smoothed by the Laplacian smoothing (10 iterations) (d) Enhanced by iterative mean filtering (20 iterations, the boost threshold = 1.5).

3. Left Work

The following enhancement rule, described in Geometric Surface Processing via Normal Map, did not well work. However, I check my implementation and again try to experiment.

$$N'_{(i)} = \frac{(1+a)N^{n}_{(i)} - aN^{n+1}_{(i)}}{\left\|(1+a)N^{n}_{(i)} - aN^{n+1}_{(i)}\right\|} \quad (3)$$