한국게임학회 논문집
제2권 제2호 2002년 11월

Mesh Simplification using New Approximate Mean Curvatures

새로운 근사 평균 곡률을 이용한 메쉬 단순화

곽재희Jae-Hee Kwak1), 이은정Eun-Jeong Lee2) 유관희Kwan-Hee Yoo2,3)

1) 충북정보산업고등학교 Chungbuk Information Industrial High School, kw094@hanmail.net
2) 충북대학교 정보산업공학과 대학원 Department of Information Industrial Engineering of a Graduate School, Chungbuk National University, dolphin@cgs.chungbuk.ac.kr
3) 충북대학교 컴퓨터교육과 및 컴퓨터정보통신연구소 Department of Computer Education and Institute of Computer and Information Communication, Chungbuk National University, Chungbuk, 361-763, KOREA, khyoo@cbucc.chungbuk.ac.kr

요 약

일반적으로 삼각형 메쉬는 가상 게임 캐릭터와 같은 기하학적 객체를 모델링하기 위해 사용되고 있다. 아주 조밀한 메쉬는 복잡한 객체를 세부적으로 표현하는 장점은 있지만 객체를 저장, 전송 및 렌더링하는데 많은 비용을 요구한다. 그러므로 세밀한 객체를 질 좋게 근사시킬 수 있는 기법, 즉 삼각형 메쉬의 단순화가 연구되어왔다. 본 논문에서는 주어진 메쉬를 단순화하기 위해 사용될 수 있는 정점과 에지에 관한 근사 평균 곡률이라는 새로운 측정치를 제시한다. 에지 평균 곡률은 이웃한 에지를 고려하게 계산되고 정점 평균 곡률은 부속된 에지의 평균 곡률의 평균으로 정의된다. 그리고 제안된 측정치를 토끼, 용 및 치아와 같은 모델에 적용한다. 결과로, 제안된 평균 곡률이 주어진 모델에 더 좋은 근사를 제공하기 위한 좋은 기준치로 사용될 수 있음을 알았다.
키워드:메쉬 단순화, 평균곡률, 재삼각화, 에지 병합, 정점제거

Abstract

In general, triangular meshes have been used for modeling geometric objects such as virtual game characters. The dense meshes give us considerable advantages in representing complex, highly detailed objects, while they are more expensive for storing, transmitting and rendering the objects. Therefore, several researches have been performed for producing a high quality approximation in place of detailed objects, that is, a simplification of triangular meshes. In this paper, we propose a new measure with respect to edges and vertices, which is called an approximate mean curvature and is used as criteria to simplify an original mesh. An edge mean curvature is computed by considering its neighboring edges, and a vertex mean curvature is defined as an average of its incident edges' mean curvatures. And we apply the proposed measure to simplify the models such as a bunny, dragon and teeth. As a result, we can see that the mean curvatures can be used as good criteria for providing much better approximation of models.
Keywords: mesh simplification, mean curvatures, re-triangulation, edge collapse, vertex removal

1. Introduction

Recently, the mesh expression of objects such as virtual game characters in the field of computer graphics and computer games has been generalized by the appearance of three-dimensional scanners. The mesh is represented by a lot of triangles that are formed of connecting three vertices. However, the dense meshes give us considerable advantages in representing complex, highly detailed objects, while they are more expensive for storing, transmitting and rendering the objects. Therefore, many researches have been preformed for producing a high quality approximation in place of detailed objects, that is, a simplification of triangular meshes.

In general, a mesh simplification is done through the following two steps[3]. The first step calculates the criteria such as curvatures to simplify a given mesh and, in the next step, the simplification of the mesh is accomplished according to criteria. One of the methods of mesh simplification was proposed by Schroeder *et.al*[10]. They analyzed the features of a vertex as curvatures and then performed the re-triangulation by removing the vertex with curvature below cut-off criteria. Hoppe *et.al*[5] proposed the edge collapse. They first calculated the curvature of edge connecting two vertices and then simplified meshes by merging the two vertices of the edge, which has curvature below cut-off criteria, into one vertex. The approximate curvature method was used as the criterion of mesh simplification. Rosenfeld and Jonston[9] suggested a method to calculate the approximate curvature using an interval angle at adjacent vertices. Turk[11] thought the edge curvature can change into a curvature radius and then suggested the approximate curvature. Andrew *et.al*[1] refer to approximation of the mean edge curvature and the Gaussian vertex curvature. Kim *et.al*[7] calculated the direct curvature using approximation of the mean curvature and Gauss curvature. And then they executed the mesh simplification by this criterion. An essential condition for the mesh simplification is to calculate the curvature at the three-dimensional mesh model. We propose an easy method that approximates the mean curvatures. In order to approximate mean curvatures, in this paper, we first compute the curvature by considering neighboring three edges about each edge in the mesh. Next, the vertex curvature was defined as the mean curvatures about the adjacent edges. The mesh simplification was performed with respect to vertices and edges on the basis of the mean edge and vertex curvatures, respectively.

The composition of this paper is as follows: We propose the newly defined mean curvatures that are used as the criterion of mesh simplification in Section 2. In Section 3, we discuss the mesh simplification method performed by using the proposed mean curvatures. In section 4, we describe the data structure and algorithm. The results applied to bunny, dragon and teeth of the proposed methods are argued in Section 5. Finally, we will see the research results of this paper and present further research items.

2. The Proposed Approximate Mean Curvatures

Many researchers [1,3,4-11] have proposed criteria to find methods for mesh simplification using the curvatures of curves and curved surfaces. The curvatures [12] can be defined at a specific point on curve or curved surface subject to second differentiation. First of all, consider general concept of curvature on a curve X(s), which is parameterized by arc-length s. If $X'(s)$ is the unit tangent vector function, then the curvature is defined to be the rate at which $X'(s)$ changes with respect to s.

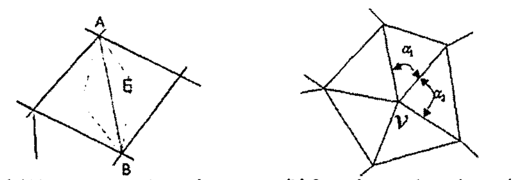
Now, extend the concept to a surface with a surface normal at a point P in three-dimensional space. Normal curvatures are defined as the curvatures of curves on the surface lying in planes including the tangent vector at the given point P. Because of the difficulty of calculation of exact normal curvatures on a surface, the following other curvatures are defined: principal, Gaussian and mean curvatures. The principal curvatures at a point of a surface are the minimum and maximum of the normal curvatures at that point, and the Gaussian curvature is the product of two principal curvatures, and the mean curvature is half the sum of two principal curvatures. Unfortunately, it is very difficult to exactly calculate these curvatures on a surface of mesh modeling objects. Therefore, there are many attempts to obtain approximate curvatures of a point of mesh object's surface[12]. The important factor in approximating the curvatures is how to explain the main features of a mesh model. In order to express them better, a few studies[4-11] proposed methods to calculate more approximate curvatures to the ideal. Andraw and Smith[1] proposed methods for approximating a mean curvature at an edge and a Gaussian curvature at a point on a surface, respectively. A mean curvature C_E of an edge E can be defined as :

$$C_E = \theta \times \frac{L_E}{Area} \qquad (E1)$$

Here θ is an angle between two normal vectors of two faces sharing the edge E, and L_E is the length of E and Area is defined as one third of the area of each triangle shaded as shown in Fig. 1(a). He also defined the Gaussian curvature K_v at a vertex v of a surface as the following:

$$K_v = \frac{2\pi - \sum \alpha_i}{N_v} \qquad (E2)$$

Here N_v is the number of surfaces adjacent to the vertex v and α_i is an angle inbetween two edges which are adjacent to v and in a face (see Fig. 1(b)).



(a) Mean curvture at an edge (b) Gaussian curvture at a vertex

Fig. 1. The mean curvatures approximation of [5]

There are some problems in calculating mean curvatures due to the violation of the curvature's concepts. Therefore, it is required to obtain mean curvatures explaining its concepts in three-dimensional space. Now we explain the new method to calculate the mean curvature approximation of a polygon model. To do it, we first calculate the mean curvatures with respect to edges and then obtain the mean vertices those using the previous processed edge mean. To begin with, calculate the mean curvature CE_i for an edge E connecting two vertices P and Q_i.

It is true that there are three vertices including Q_i incident to vertex P since there are adjacent triangles in a mesh model. Let Q_{i-1}, Q_i, and Q_{i+1} be the three vertices. Using a similar method to one in the plane, edge mean curvatures can be obtained. In order to do it, we get three middle points of the three edges PQ_i, PQ_{i-1}, and PQ_{i+1}, respectively. In addition, we define the circle that connects three midpoints. Let rp be the radius of this circle. In a similar fashion, there are three vertices including P incident to vertex Q_i . Let Q_{i-1}, P, and Q_{i+1} be the three vertices. Similarly, we get the circle connecting three middle points of three edges $Q_{i+1}P$, Q_iP and $Q_{i-1}P$, and PQ_{i+1}, respectively. Let r_{qi} be the radius of the obtained circle. The detailed description is shown in Fig. 2.

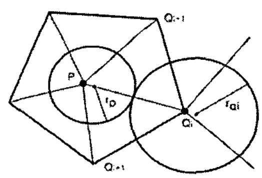


Fig. 2. The proposed mean curvature approximation

Then, the mean curvature CE_i of an edge PQ_i in a mesh model can be obtained like as (E3) equation.

$$CE_i = \frac{r_p + r_{qi}}{2 r_p r_{qi}} \qquad (E3)$$

The CE_i shows the curvature of an edge E_i incident to vertex V_k. And, when n is the number of adjacent vertices about vertex V_k, the mean curvature CV_k for V_k is defined as the mean value that takes an average of the mean curvatures of attached edges like as (E4) equation.

$$CV_k = \frac{1}{n}\sum_{i=1}^{n} CE_i \qquad (E4)$$

3. A Mesh Simplification

In this Section, consider how to remove vertices and edges in the mesh for implementing a simplification of a mesh representing an object in three-dimensional space. We propose the methods to process a vertex removal and an edge collapse for a mesh simplification in Section 3.1 and in Section 3.2, respectively. And we present re-triangulation techniques after a vertex removal and an edge collapse, respectively. The re-triangulation of the mesh has to be done since mesh models are generally composed of a set of triangles in three-dimensional space.

3.1 Vertex removal

In this Section, we introduce the mesh simplification method with removing the vertex. Section 3.1.1 describes how to select a vertex to be removed in mesh models, and Section 3.1.2 discusses how to reconstruct a given mesh model after removing the selected vertices.

3.1.1 Selection of a vertex to be removed

Consider the geometric shapes with respect to a vertex v in a given mesh. According to the arrangement of the faces adjacent to v, the vertex v can be classified into three kinds: simple vertex, complex vertex, and boundary vertex. The detailed information is described as well in Fig. 3.

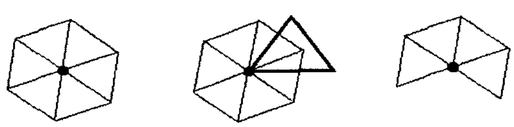


Fig. 3. (a) Simple vertex (b) Complex vertex (c) Boundary vertex

Complex vertices are not deleted from the mesh since the changes of a geometric shape may be occurred. All other vertices become candidates for deletion. Therefore, we calculate the mean curvatures with respect to the candidate vertices by an algorithm described in Section 2. When the curvature of each vertex is produced, the curvatures of these vertices are sorted in ascending order. The vertices with the curvature below a given threshold are selected as those to be removed. Specially, we do ordering of the vertices adjacent to the vertex v in counter-clockwise to efficiently re-triangulate a given mesh after removing the selected vertex.

3.1.2 Re-triangulation

In general, if one vertex from the mesh is eliminated, several triangles can be removed. One or two holes can appear after removing a simple or boundary vertex, and after removing an interior edge vertex, respectively. According to the composition features of the mesh, a geometric shape can be formed convexly or concavely. Therefore, the re-triangulation has to be implemented with other methods individually. The re-triangulation technique of a convex shape uses the "split line" that connects two nonadjacent vertices. First, we select one of those vertices as a pivot vertex. Second, we choose a nonadjacent vertex to the pivot vertex in counter-clockwise. Draw the split line that connects the selected vertex and a pivot vertex. Using the split line, divide the hole in two. The operation is repeated for a newly-created hole until the new hole consists of three vertices (see Fig. 4).

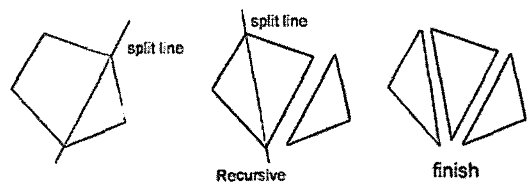


Fig. 4. A re-triangulation process for a convex shape

We set out to find the optimum split line to compose re-triangulation nearest adjacent on the original object shape. The problem can be solved using the aspect ratio. The aspect ratio method means the least of distance from one vertex to the split line. In other words, the distance means one of the vertical line from the vertex to the split line. And the optimum split line is a line connecting a given pivot and the vertex with the largest value of aspect ratio of each vertex.

The re-triangulation technique of a concave shape is implemented by the criterion of the angle of three vertices. When there is the concave mesh as shown in Fig. 5, let n be the number of vertices adjacent to a vertex v_1. Let v_2 be a voluntary start vertex in the set of the adjacent vertices ordered in counter-clockwise. At this time, let θ be the angle of three vertices v_2, v_3, and v_4. If $180/n \leq \theta \leq 360/n$, we generate the triangle consisting of the three vertices v_2, v_3, and v_4. Otherwise, we exclude v_3 in the list of adjacent vertices. Then, clearly, the number of the adjacent vertices is a decrease of one. We again designate the start vertex with v_3, and then implement repeatedly until the number of the adjacent vertices becomes three. In this way, the new re-triangulated triangle is tacked onto the array of the faces. By doing the vertex removals to the level degree required by the users, the mesh simplification can be achieved.

3.2 Edge collapse

In order to collapse an edge E in a mesh mode, let i and j be two vertices incident to the edge, respectivly. If the edge E has to be collapsed, it is clear that the mean curvature for E has one below a given threshold. In this case, we compare the curvatures of the two vertices, and then one of both is moved to another with bigger curvature. For instance, if the curvature of a vertex i is a bigger than the vertex j, the vertex j is moved to the vertex i. In this process, the faces including E will be eliminated and the number of triangles in the mesh model will be decreased (see Fig. 6). By doing the edges collapse to the level degree required by the users, the mesh simplification can be achieved.

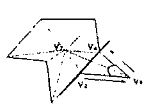


Figure 5. A re-triangulation of a concave shape

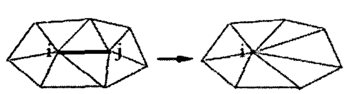


Figure 6. An edge collapse

4 Implementation

4.1 Data Structures

A triangular mesh has been used as data structures for expressing an object. A triangular mesh is composed of a set of triangles connecting three vertices. Each vertex has x, y, and z coordinate axes. All vertices and faces are preserved in each array, respectively. A part of a triangular mesh is shown in Fig. 7 in which a vertex v_1 and a triangle T1 has (x_1, y_1, z_1) and three vertices (v_1, v_8, v_2), respectively, and a list of triangles using a vertex v1 contains seven triangles (T_1, T_2, T_3, T_4, T_5, T_6, T_7).

4.2 The curvature approximation

Using above mentioned data structures, the vertices adjacent to a vertex can be easily obtained because they just keep the data of current vertices and faces in the corresponding array. A face consists of vertices, and the order of the vertices makes clockwise. As in Fig. 8, take the vertices adjacent to a vertex A_2. Let $Face_A$ be the first face designated as vertex A_2 as shown in Fig. 8. Clearly, A_0 can be searched as the next vertex of A_2, A_1 the next one of A_0, and A_2 the next one of A_1. Then the face has an order beginning on A_0 and ending on A_2 in sequence. Now, search the face beginning on A_2 and ending on A_0. Clearly, it will be the face $Face_B$ containing an edge B_2B_0 as shown in Figure 8. Here, B_2 is registered to the surrounding vertex of vi, that is, A_0 from $Face_A$. Then, the next vertex of A_2 can be searched from $Face_B$ again. By repeatedly applying the above procedure, all faces surrounding vertex A_2 can be

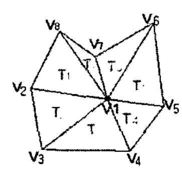


Fig. 7. Data Structure

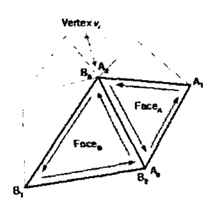


Fig. 8. The expression of a face using vertices

directly taken. Secondly, we get the mean curvature with respect to edges and vertices by using the two equations E_1 and E_2, described in Section 2.

4.3 An implementation of a mesh simplification algorithm

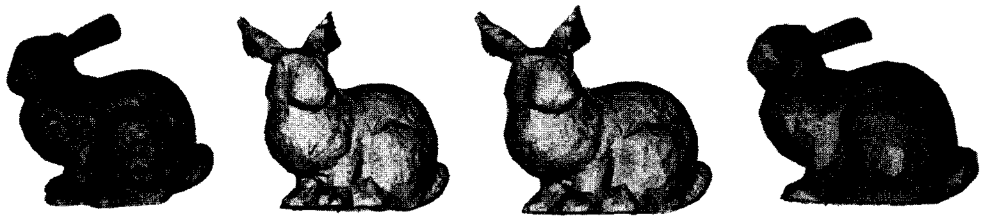
After getting the curvature of each vertex, we line up the set of vertices with the criterion of curvatures. And, we put the flag operator called "Removal" in the data structures of vertices to eliminate the vertices with a curvature below a given one. After determining false as the initial value, the flag is modified to true for the removal value if they come under an excluding vertex. And then the composed faces, which are the front and back those of the vertex having the biggest perpendicular and horizontal value, are added to an arrangement of faces. In the process, the re-triangulation method in Section 3 is applied repeatedly. The algorithm discussed until now is described in Fig. 9.

5 Experiment Results

Fig. 10(a) shows the original bunny model has the number of vertices at 35,947, and the number of surfaces at 69,451. Fig.10 (b) shows a mesh model simplified by edges' collapse proposed in this paper. The simplified mesh model has 7,591 vertices. Fig.10 (c) also shows a mesh simplification model by vertices' removal, which has 1,512 vertices. Fig. 10(d) appears a mesh simplified by quadric error metrices, which consists of 1,262 vertices. Fig. 11(a) shows the original dragon model has the number of vertices at 437,645, and the number of surfaces at 871,414. Fig. 11 (b) and (c) are a mesh simplified by edges' collapse and one simplified by vertices' removal which has 3,000 vertices and 1,512 vertices, respectively. Fig. 12 describes a mesh simplification for teeth models. Fig. 12 (a) appears an original model with 78,387 faces, Fig. (b) shows a mesh simplified by edges' collapse of (a). The resulting mesh has 22,327 faces. Fig. (c) and (d) show an original model with 140,811 faces and a mesh simplified by edges' collapse (48,822 faces) of (c), respectively. Finally, Fig. (e) shows an original model with 167,746 faces. Fig. (f) and (g) show two other views of a mesh simplified by edges' collapse (48,822 faces) of (e).

```
Procedure NewMeanCurvature()
  Begin // v_i : an ith vertex, adj_k : the kth vertex adjacent to the vertex v_i (in counter clockwise)
   while ( i < the total number of vertex )
      while (k < the vertex number neighboring v_i )
         vector_1 ← ( adj_{k+1} – v_i ) / 2; vector_2 ← ( adj_k – v_i ) / 2; vector_3 ← ( adj_{k-1} – v_i ) / 2
         radious_1 ← a radius of circle that makes three vertices v_i + vector_1, v_i + vector_2, v_i + vector_3
         edgeCurvature_1 = 1 / radious_1
         vector_1 ← ( adj_{k+1} – adj_k ) / 2; vector_2 ← ( v_i – adj_k ) / 2;   vector_3 ← ( adj_{k-1} – adj_k ) / 2
         radious_2 ← a radius of circle that makes three vertices adj_k+vector_1, adj_k+vector_2, adj_k+vector_3
         edgeCurvature_2 = 1 / radious_2
         edgeCurvature += (edgeCurvature_1 + edgeCurvature_2 ) / 2
       end while
       v_i.curvature = edgeCurvature / k //k: the number of the neighboring vertices to v_i
   end while
end ProcedureNewMeanCurvature()
Procedure SimplifiedMesh()
Begin
  NewMeanCurvature(); // compute mean curvatures w.r.t. each vertex and each edge
  while (the number of vertices > the wanted number)
    for (the vertices with a curvature below a give threshold value)
       remove the vertices or collapse the edges according to the rules in Sections 3.1 and 3.2
     re-triangulate the given mesh and recomputed the curvature for new vertices
End proceduresimplifiedmesh()
```

Fig. 9. An algorithm for calculating the mean curvatures for edges and vertices

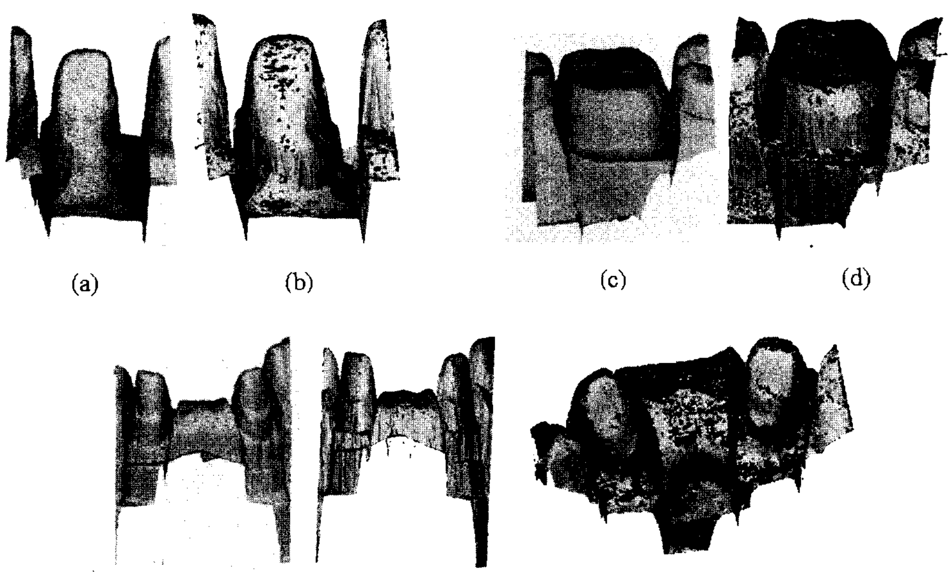


(a) (b) (c) (d)

Fig. 10. A mesh simplification for a bunny model:
(a) the original model with 35,947 vertices and 69,451 faces; (b) a mesh simplified by edges' collapse (results: 7,591 vertices); (c) a mesh simplification by vertices' removal (results: 1,512 vertices); (d) a mesh simplified by quadric error metrics(results:1,262 vertices)

(a) (b) (c)

Fig. 11. A mesh simplification for a dragon model:
(a) the original model with 437,645 vertices and 871,414 faces; (b) a mesh simplified by edges' collapse of QEM (results: 3,000 vertices); (c) a mesh simplification by vertices' removal (results: 1,512 vertices)



(a) (b) (c) (d)

(e) (f) (g)

Fig. 12. A mesh simplification for teeth models:
(a) the original model with 78,387 faces; (b) a mesh simplified by edges' collapse (22,327 faces) of (a); (c) the original model with 140,811 faces; (d) a mesh simplified by edges' collapse (48,822 faces) of (c); (e) the original model with 167,746 faces; (f)-(g) mesh simplified by edges' collapses (48,822 faces) of (e)

6. Conclusions and Discussions

We presented a mesh simplification that employs mean curvature approximation about the vertex and edge of an original mesh as criteria of mesh simplification. We have introduced two methods. One method is the vertex elimination that removes the vertices below standard of a regular curvature and then make a re-triangulation. The other method is an edge collapse that finds edges below the standard of the regular curvature and then renews the face data by collapsing two vertices into one. These methods are applied to the bunny, dragon and teeth models. These results, which are modeled according to the vertices after removal, approach the geometric features of the original object using the new mean curvature approximation more than the existing mesh simplification algorithms.

The advantage of the proposed method is that it uses the definition about the real vertex curvature as it is, so the application of the value about the curvature is correct. On the other hand, the disadvantage is that the speed of execution is quite slow because the adjacent faces for each vertex have to be searched and the curvature about each vertex is computed. We need to improve the speed of modeling by applying better data structure to the program. In addition, a problem awaiting solution is a method of implementing mesh simplification about a strip at which several edges connect, and simplify the mesh about a surface in which two triangles are attached.

Acknowledgements

The bunny and dragon models are courtesy of the Large Geometric Models Archive (http://www.cc.gatech.edu/projects/large_models/). This work was supported by the Korea Research Foundation Grant (KRF-2002-03-D00309).

References

[1] Andrew, D. and C. Smith, The folding of the human brain: from shape to function, University of London, Ph.D Dissertations, 1999.

[2] Arvo, J., GRAPHICS GEMS II, A PPROFESSIONAL, Academic Press, pp 10-13, 1991.

[3] Cignoni, P., C. Montani and R. Scopigno "A comparison of mesh simplification algorithm" Computers and Graphics, 1997.

[4] Garland, M. and P.S. Heckbert, "Surface simplification using quadric error metrics," ACM Computer Graphics(prof. Of SIGGRAPH '95), pages 209-216, 1997

[5] Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. "Mesh optimization", Computer Graphics (SIGGRAPH '93 Proceedings), pages 19-26, August 1993.

[6] Jang, J.-H. and K.-H. Yoo, "Geometric features extraction for reconstructing surface of artificial teeth", Proceedings of the Korea-Japan Computer Graphics Conference 2001, pp.96-105, 2001.

[7] Kim, S.-J., S.-W. Hong, and C.-H. Kim, "Mesh simplification with discrete curvature error metric", Proceedings of HCI 2001, pp. 449-504, Korea, February 2001.

[8] Ronfard, R. and J. Rossignac. " Full-range approximation of triangulated polyhedra", Computer Graphics Forum, 15(3):C67?C76, C462, September 1996.

[9] Rosenfeld, A., and E. Johnston, "Angle detection in digital curves", IEEE Transactions on Computers, 22:875-878, 1973.

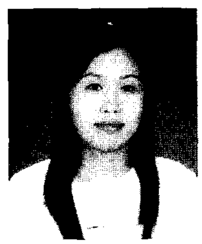
[10] Schroeder, W.J., J. A. Zarge, W. E. Lorensen, "Decimation of triangle meshes", Computer Graphics (SIGGRAPH '92 Proceedings), volume 26, pages 65-70, July 1992.

[11] Turk, G., "Re-tilling polygonal surfaces", Computer Graphics (SIGGRAPH '92 Proceedings), Vol.26,

No.2, pp.55-64.
[12] Yamaguchi, F.,Curves and surfaces in computer aided geometric design, Springer-Berlag, 1988

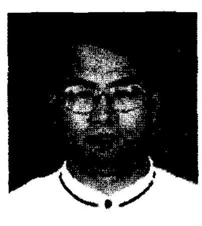
곽재희

Email: kw094@hanmail.net
1994:서원대학교전자계산학과 (학사)
2002:충북대학교교육대학원 전자계산교육전공(석사학위취득)
1997-2000:충북보은상업고등학교 근무
2000-현재:충북정보산업고등학교 근무
주관심분야: 컴퓨터그래픽스, 메쉬 단순화



이은정

Email:dolphin@cgs.chungbuk.ac.kr
2002:충북대학교컴퓨터과학과(학사)
2002-현재:충북대학교 정보산업공학과(석사과정 재학중)
주관심분야:컴퓨터그래픽스, 메쉬 단순화, 워터마킹



유관희

Email: khyoo@cgs.chungbuk.ac.kr
1985:전북대학교전산통계학과(학사)
1988:한국과학기술원 전산학과(석사학위취득)
1995:한국과학기술원 전산학과(박사학위취득)
1988-1997: (주)데이콤 종합연구소 선임연구원
1997-현재: 충북대학교 컴퓨터교육과, 정보산업공학과 및 컴퓨터.정보통신연구소 교수
2002-현재: (주)시멘텍 기술이사
주관심분야: 컴퓨터 그래픽스, 네트워크 게임, 인공치아 모델링