An Improved Laplacian Smoothing Approach for Surface Meshes

Ligang Chen, Yao Zheng, Jianjun Chen, and Yi Liang

College of Computer Science, and Center for Engineering and Scientific Computation,
Zhejiang University, Hangzhou, Zhejiang, 310027, P.R. China
{ligangchen,yao.zheng,chenjj,yliang}@zju.edu.cn

Abstract. This paper presents an improved Laplacian smoothing approach (ILSA) to optimize surface meshes while maintaining the essential characteristics of the discrete surfaces. The approach first detects feature nodes of a mesh using a simple method, and then moves its *adjustable* or *free node* to a new position, which is found by first computing an optimal displacement of the node and then projecting it back to the original discrete surface. The optimal displacement is initially computed by the ILSA, and then adjusted iteratively by solving a constrained optimization problem with a quadratic penalty approach in order to avoid inverted elements. Several examples are presented to illustrate its capability of improving the quality of triangular surface meshes.

Keywords: Laplacian smoothing, surface mesh optimization, quadratic penalty approach.

1 Introduction

Surface triangulations are used in a wide range of applications (e.g. computer graphics, numerical simulations, etc.). For finite element methods, the quality of surface meshes is of paramount importance, because it influences greatly the ability of mesh generation algorithms for generating qualified solid meshes. Since surface meshes define external and internal boundaries of computational domains where boundary conditions are imposed, and thus they also influence the accuracy of numerical simulations.

Mesh modification and vertex repositioning are two main methods for optimizing surface meshes [1, 2]. While mesh modification methods change the topology of the mesh, the vertex repositioning, also termed as mesh smoothing, redistributes the vertices without changing its connectivity. This paper only focuses on smoothing techniques for surface mesh quality improvement.

Despite their popularity in optimizing 2D and 3D meshes [3, 4], smoothing methods for surface meshes present significant challenges due to additional geometric constraints, e.g. minimizing changes in the discrete surface characteristics such as discrete normals and curvature. When improving surface mesh quality by vertex repositioning, changes in the surface properties can usually maintained small by keeping the vertex movements small and by constraining the vertices to a smooth

surface underlying the mesh or to the original discrete surface. One approach commonly used to constrain nodes to the underlying smooth surface is to reposition each vertex in a locally derived tangent plane and then to pull the vertex back to the smooth surface [1, 5]. Another one is to reposition them in a 2D parameterization of the surface and then to map them back to the physical space [6, 7].

In this paper, an improved Laplacian smoothing approach (ILSA) is presented to enhance the quality of a surface mesh without sacrificing its essential surface characteristics. The enhancement is achieved through an iterative process in which each *adjustable* or *free node* of the mesh is moved to a new position that is on the adjacent elements of the node. This new position is found by first computing an optimal displacement of the node and then projecting it back to the original discrete surface. The optimal displacement is initially obtained by the ILSA, and then adjusted iteratively by solving a constrained optimization problem with a quadratic penalty approach in order to avoid inverted elements.

2 Outline of the Smoothing Procedure

The notations used in the paper are as follows. Let $T = (V, E, F)$ be a surface mesh, where V denotes the set of vertices of the mesh, E the set of edges and F the set of triangular faces. f_i , e_i and $\mathbf{v}_i$ represents the i 'th face, edge and vertex of the mesh respectively. $A(b)$ denotes the set of all entities of type A connected to or contained in entity b , e.g., $V(f_i)$ is the set of vertices of face f_i and $F(\mathbf{v}_i)$ is the set of faces connected to vertex $\mathbf{v}_i$, which is also regarded as the local mesh at $\mathbf{v}_i$ determined by these faces. We also use $|S|$ to denote the number of elements of a set S .

The procedure begins with a simple method to classify the vertices of the mesh into four types: *boundary node*, *corner node*, *ridge node* and *smooth node*. The first two types of vertices are fixed during the smoothing process for feature preservation and the last two are referred to as *adjustable nodes*. More sophisticated algorithms for detecting salient features such as crest lines on discrete surfaces can be adopted [8]. Then in an iterative manner, a small optimal displacement is computed for each adjustable node using the ILSA, which accounts for some geometric factors. Moreover, for each smooth node its optimal displacement is adjusted by solving a constrained optimization problem so as to avoid inverted elements. Finally, all those redistributed vertices are projected back to the original discrete surface. The complete procedure is outlined as Algo. 1, of which the algorithmic parameters will be explained later.

3 Classifying the Vertices

The boundary nodes, if they exist, can be identified by examining the boundary edges that have only one adjacent element. For each interior node $\mathbf{v}_i$, let $m = |F(\mathbf{v}_i)|$, we first evaluate its discrete normal by solving the following linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{1}. \quad (1)$$

where $\mathbf{A}$ is an $m \times 3$ matrix whose rows are the unit normals of $F(\mathbf{v}_i)$, and $\mathbf{1} = (1, 1, \dots, 1)^t$ is a vector of length m . Since $\mathbf{A}$ may be over- or under-determined, the solution is in least squares sense and we solve it by the singular value decomposition (SVD) method [9].

Algo. 1. The smoothing procedure.

```

Set the algorithmic parameters: max_global_iter_num,
    max_smooth_iter_num, relax1, relax2,  $\mu$ , wl, wa;
Classify the vertices of the mesh into 4 types;
Initialize the smoothed mesh  $T_{new}$  as the original mesh  $T_{ori}$ ;
for step := 1 to max_global_iter_num do //global iteration
    Compute the normal of each vertex of  $T_{new}$ ;
    Compute the optimal displacement of each ridge node of  $T_{new}$ ;
    Compute the initial displacement of each smooth node of  $T_{new}$ ;
    Adjust the displacement of each smooth node in order to avoid
        inverted elements;
    Project the redistributed position of each adjustable node back
        to the original mesh  $T_{ori}$ , denote this new mesh as  $T'_{new}$ ;
    Update  $T_{new}$  as  $T'_{new}$ ;
end for
Set the final optimized mesh as  $T_{new}$ .

```

The length of the resulting vertex normal has a geometric interpretation as an indicator of singularity of $\mathbf{v}_i$. Let $f_j \in F(\mathbf{v}_i)$, $1 \leq j \leq |F(\mathbf{v}_i)|$, $\mathbf{N}(f_j)$ the unit normal of f_j and $\mathbf{N}(\mathbf{v}_i) = \mathbf{x}$ the unknown vertex normal. The equation corresponding to f_j in Eq. (1) is

$$\mathbf{N}(f_j) \cdot \mathbf{x} = |\mathbf{x}| \cos \angle(\mathbf{N}(f_j), \mathbf{x}) = 1. \quad (2)$$

Now it is obvious that, for some solution $\mathbf{x}$ of (1), the angles between $\mathbf{x}$ and $\mathbf{N}(f_j)$, $1 \leq j \leq |F(\mathbf{v}_i)|$, would be approximately equal. Roughly speaking, if the local mesh $F(\mathbf{v}_i)$ is flat, the angles would be small, otherwise they would be large, consequently the length of the resulting vertex normal would be short and long, and the vertex will be regarded as a smooth node and a ridge node, respectively.

The ridge nodes will be further examined to determine whether they are corner nodes or not. Let e_i be an edge formed by two ridge nodes, if the bilateral angle between two faces attached to e_i is below a threshold angle ($8\pi/9$ in our algorithm), these two nodes are said to be *attached-sharp nodes* of each other. If the number of such nodes of a ridge node is not equal to two, this node is identified as a corner node. The geometric interpretation of the classification is self-evident.

4 Repositioning the Adjustable Vertices

4.1 Computing Displacements by the ILSA

In each global iteration of Algo. 1, the ILSA is employed to compute the initial optimal displacements for both ridge and smooth nodes. The procedure for treating these two types of nodes is similar. The major difference lies in that smooth nodes take all their adjacent nodes' effect into account, while ridge nodes consider only the effect of their two attached-sharp nodes. Algo. 2 illustrates the details.

Here if $\mathbf{v}_i$ is a ridge node, then $n = 2$ and $\{\mathbf{v}_j, j = 1, 2\}$ are two attached-sharp nodes of $\mathbf{v}_i$, otherwise $n = |V(\mathbf{v}_i)|$ and $\mathbf{v}_j \in V(\mathbf{v}_i)$. $\mathbf{d}(\mathbf{v}_j)$ is the current displacement of $\mathbf{v}_j$. Such treatment of ridge nodes tries to prevent the crest lines on surface meshes from disappearing. The adjusting vector $\mathbf{vec}$ takes the lengths of adjacent edges into consideration in order to obtain a smoother result.

4.2 Adjusting Displacements by a Quadratic Penalty Approach

Unfortunately, Laplacian smoothing for 2D mesh may produce invalid elements. When used for surface meshes, there are still possibilities of forming abnormal elements. To compensate for this, we adjust the displacement iteratively for each smooth node by solving a constrained optimization problem.

The idea originates in the minimal surface theory in differential geometry. Minimal surfaces are of zero mean curvature. Their physical interpretation is that surface tension tries to make the surface as "taut" as possible. That is, the surface should have the least surface area among all surfaces satisfying certain constraints like having fixed boundaries [10]. Every soap film is a physical model of a minimal surface. This motivates us, for the local mesh $F(\mathbf{v}_i)$ at a smooth node $\mathbf{v}_i$, to move $\mathbf{v}_i$ to minimize the overall area of the elements of $F(\mathbf{v}_i)$ in order to make this local mesh "taut" and thus to avoid invalid elements as much as possible. This new position $\mathbf{v}'_i$ is also softly constrained to be on a plane by a quadratic penalty approach.

Let $\mathbf{d}_{cur}(\mathbf{v}_i)$ and $\mathbf{N}(\mathbf{v}_i)$ be the current displacement and the discrete normal of $\mathbf{v}_i$ respectively. Initially $\mathbf{d}_{cur}(\mathbf{v}_i)$ is the result from Algo. 2. Let $\mathbf{x}$ be the new pending position of $\mathbf{v}_i$ and $\mathbf{d}_{new}(\mathbf{v}_i) = \mathbf{x} - \mathbf{v}_i$ the new adjusting displacement. Suppose $\mathbf{v}_j \in V(\mathbf{v}_i)$, $1 \leq j \leq n+1$, $n = |V(\mathbf{v}_i)|$ are the vertices surrounding $\mathbf{v}_i$ in circular sequence and $\mathbf{v}_{n+1} = \mathbf{v}_1$. $s(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ represents the area of the triangle $\Delta \mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3$. Now the optimization problem can be formulated as follows

$$\min_{\mathbf{x}} g(\mathbf{x}) \quad \text{subject to } c(\mathbf{x}) = 0 \quad (3)$$

where

$$g(\mathbf{x}) = w l \frac{1}{n} \sum_{j=1}^n |\mathbf{x} - \mathbf{v}_j|^2 + w a \sum_{j=1}^n \beta_{ij} s^2(\mathbf{x}, \mathbf{v}_j, \mathbf{v}_{j+1}) \quad (4)$$

Algo. 2. ILSA for adjustable nodes.

Initialize the vertex displacement $\mathbf{d}(\mathbf{v}_i)$ of each adjustable node $\mathbf{v}_i$ of T_{new} as the Laplacian

coordinate of $\mathbf{v}_i$: $\mathbf{d}(\mathbf{v}_i) := \frac{1}{n} \sum_{j=1}^n \mathbf{v}_j - \mathbf{v}_i$;

for $k := 1$ to $max_smooth_iter_num$ **do**

for each adjustable node $\mathbf{v}_i$ **do**

Compute a vector $\mathbf{vec}$: $\mathbf{vec} := \frac{1}{S} \sum_{j=1}^n \alpha_{ij} \mathbf{d}(\mathbf{v}_j)$,

where $S = \sum_{j=1}^n \alpha_{ij}$ and $\frac{1}{\alpha_{ij}} = dist(\mathbf{v}_i, \mathbf{v}_j)$ is the distance

between $\mathbf{v}_i$ and $\mathbf{v}_j$;

Update $\mathbf{d}(\mathbf{v}_i)$: $\mathbf{d}(\mathbf{v}_i) := (1 - relaxI) \cdot \mathbf{d}(\mathbf{v}_i) + relaxI \cdot \mathbf{vec}$;

end for

if <condition> **then** // e.g. smooth enough

break the iteration;

end if

end for

and

$$c(\mathbf{x}) = \begin{cases} \mathbf{N}(\mathbf{v}_i) \mathbf{g} \mathbf{d}_{new}(\mathbf{v}_i) & \text{if } \mathbf{d}_{cur}(\mathbf{v}_i) = \mathbf{0} \\ \mathbf{d}_{cur}(\mathbf{v}_i) \mathbf{g} \mathbf{d}_{new}(\mathbf{v}_i) - |\mathbf{d}_{cur}(\mathbf{v}_i)|^2 & \text{if } \mathbf{d}_{cur}(\mathbf{v}_i) \neq \mathbf{0} \end{cases} \quad (5)$$

Here wl and wa are two algorithmic parameters and $\beta_{ij} = 1/s(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_{j+1})$. It can be observed that the constraint $c(\mathbf{x}) = 0$ is used to penalize the deviation of $\mathbf{x}$ from a plane. When $\mathbf{d}_{cur}(\mathbf{v}_i) = \mathbf{0}$, it is the tangent plane at $\mathbf{v}_i$, otherwise it is the plane vertical to $\mathbf{d}_{cur}(\mathbf{v}_i)$ and passing through the node $\mathbf{v}_i + \mathbf{d}_{cur}(\mathbf{v}_i)$. In other words, it tries to constrain $\mathbf{x}$ to be on the current smoothed discrete surface. It is also observed from Eq. (4) that we include another term related to the length $|\mathbf{x} - \mathbf{v}_j|$ and we use the square of area instead of area itself for simplicity. The area of a triangle can be calculated by a cross product $s(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = |\mathbf{v}_2 - \mathbf{v}_1| \times (\mathbf{v}_3 - \mathbf{v}_1) / 2$. The quadratic penalty function $Q(\mathbf{x}; \mu)$ for problem (3) is

$$Q(\mathbf{x}; \mu) = g(\mathbf{x}) + \frac{1}{\mu} c^2(\mathbf{x}) \quad (6)$$

Algo. 3. Adjusting vertex displacement by a quadratic penalty approach.

```

for  $k:=1$  to  $max\_smooth\_iter\_num$  do
  for each smooth node  $v_i$  do
    Compute the adjusting displacement  $d_{new}(v_i)$  by solving problem(3)
    Update the displacement:  $d_{cur}(v_i) := relax2 \cdot d_{new}(v_i) + (1-relax2) \cdot d_{cur}(v_i)$ 
    if <condition> //e.g. tiny change of vertex displacements
      break the iteration;
    end if
  end for
end for

```

where $\mu > 0$ is the penalty parameter. Since $Q(x; \mu)$ is a quadratic function, its minimization can be obtained by solving a linear system, for which we again use the SVD method. This procedure of adjusting vertex displacement is given in Algo. 3.

4.3 Projecting the New Position Back to the Original Mesh

Once the final displacement of each adjustable node is available, the next step is to project the new position of the node back to the original discrete surface to form an optimized mesh. It is assumed that the displacement is so small that the new position of a node is near its original position. Thus, the projection can be confined to be on the two attached-ridge edges and the adjacent elements of the original node for ridge and smooth nodes, respectively.

5 Experimental Results

Two examples are presented to show the capability of our method. The aspect ratio is used to measure the quality of the elements.

The first example is a surface mesh defined on a single NURBS patch. The minimal and average aspect ratios of the original (resp. optimized) mesh are 0.09 and 0.81 (resp. 0.39 and 0.89).

The second example which is obtained from the Large Geometric Model Archives at Georgia Institute of Technology, is a famous scanned object named horse. The original mesh has 96966 triangles and 48485 nodes, and its average aspect ratio is 0.71, which has increased to 0.82 for the optimized counterpart. Note that the poor quality of the original mesh in several parts of the neck of the horse in Fig. 1(a) whose optimized result is given in Fig. 1(b). The details of horses' ears of both initial and optimized meshes have also shown that our surface smoothing procedure is capable of preserving sharp features.

Fig. 1. Details of the neck of the horse for the initial (a) and optimized (b) meshes

6 Conclusion and Future Work

We have proposed an improved Laplacian smoothing approach for optimizing surface meshes. The nodes of an optimized mesh are kept on the initial mesh to avoid the shrinkage problem. A simple but effective procedure is also suggested to identify the feature points of a mesh in order to preserve its essential characteristics. Furthermore,

to avoid the formation of inverted elements, we adjust the initial displacements by solving a constrained optimization problem with a quadratic penalty method.

In the future, this smoothing technique will be integrated into a surface remesher. A global and non-iterative Laplacian smoothing approach with feature preservation for surface meshes is also under investigation.

Acknowledgements. The authors would like to acknowledge the financial support received from the NSFC (National Natural Science Foundation of China) for the National Science Fund for Distinguished Young Scholars under grant Number 60225009, and the Major Research Plan under grant Number 90405003. The first author is very grateful to Mr. Bangti Jin of The Chinese University of Hang Kong for his valuable revision of this paper.

References

1. Frey, P.J., Borouchaki, H.: Geometric surface mesh optimization. *Computing and Visualization in Science*, 1(3) (1998) 113-121
2. Brewer, M., Freitag, L.A., Patrick M.K., Leurent, T., Melander, D.: The mesquite mesh quality improvement toolkit. In: *Proc. of the 12th International Meshing Roundtable*, Sandia National Laboratories, Albuquerque, NM, (2003) 239-250
3. Freitag, L.A., Knupp, P.M.: Tetrahedral mesh improvement via optimization of the element condition number. *International Journal of Numerical Methods in Engineering*, 53 (2002) 1377-1391
4. Freitag, L.A., Plassmann, P.: Local optimization-based simplicial mesh untangling and improvement. *International Journal of Numerical Methods in Engineering*, 49 (2000) 109-125
5. Knupp, P. M.: Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. Part 1 – a framework for surface mesh optimization. *International Journal of Numerical Methods in Engineering*, 48 (2000) 401-420
6. Garimella, R.V., Shashkov, M.J., Knupp, P.M.: Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Computer Methods in Applied Mechanics and Engineering*, 193(9-11) (2004) 913-928
7. Escobar, J.M., Montero, G., Montenegro, R., Rodriguez, E.: An algebraic method for smoothing surface triangulations on a local parametric space. *International Journal of Numerical Methods in Engineering*, 66 (2006) 740-760.
8. Yoshizawa, S., Belyaev, A., Seidel, H.-P.: Fast and robust detection of crest lines on meshes. In: *Proc. of the ACM symposium on Solid and physical modeling*, MIT (2005) 227-232
9. William H.P., Saul A.T., William T.V., Brain P.F.: *Numerical Recipes in C++*. 2nd edn. Cambridge University Press, (2002)
10. Oprea, J.: *Differential Geometry and Its Applications*. 2nd edn. China Machine Press, (2005)