Normal Meshes for Multiresolution Analysis of Irregular Meshes with Boundaries∗

Kyu-yeul LEE**, Seong-chan KANG***
and Tae-wan KIM****

In this paper we present a remeshing algorithm for irregular meshes with boundaries. The irregular meshes are approximated by regular meshes that have topological regularity, which is essential for the multiresolution analysis of the given irregular meshes. Normal meshes are used to reduce the necessary data size at each resolution level of the regularized meshes. The normal mesh uses one scalar value, i.e., normal offset value that is calculated by the regular rule of a uniform subdivision and normal sampling, while other remeshing schemes use one 3D vector at each vertex. Since the normal offset cannot be adopted for the boundaries of meshes, we use a combined subdivision scheme that resolves the problem of the proposed normal offset method at the boundaries. Lastly, an example showing the effectiveness of the proposed scheme to reduce the data size of a mesh model is included.

Key Words: Normal Mesh, Boundary, Remeshing, Irregular Meshes, Subdivision, Multiresolution Analysis, Wavelets

1. Introduction

In computer graphics and geometric modeling area, triangular meshes generally represent the shapes. And with the advent of laser scanning systems, recent progress in acquiring shape from range data permits the acquisition of seamless million-polygon meshes from physical models. Such meshes are notoriously expensive to store, transmit, and render, and are awkward to edit. Multiresolution analysis offers a simple, unified, and theoretically sound approach in dealing with these problems.

A multiresolution model provides the viewer with a handy resolution model that meets the needs of individual viewers. This model can be used to facilitate the calculation of the errors of simplified models, to edit at multiresolution levels, and it can also be used for adaptive visualization and transmission.

But a multiresolution analysis requires a mesh satisfying subdivision-connectivity. However, acquired raw meshes encountered in practice typically do not have this subdivision-connectivity.

In this paper we present a method for overcoming the subdivision-connectivity restriction, meaning that irregular meshes are remeshed to regular meshes with subdivision connectivity, to which classical multiresolution and wavelet algorithms can be applied.

In particular, we use a normal mesh, which is a multiresolution mesh allowing each level to be written as a one dimensional normal offset from a coarser triangle, while other remeshing schemes use one 3D vector at each vertex. Since the normal offset cannot be properly used for the mesh boundaries, we use a combined subdivision scheme that resolves the problem of the proposed normal offset method at the boundaries.

2. Related Works

2.1 Related works on remeshing

A number of researchers have considered algorithms, which remesh an original irregular mesh with the goal of applying classical multiresolution approaches. Efficient representations of irregular
meshes have been pursued by a number of researchers since 1995.

The most widely used remeshing method is not directly remeshing an irregular mesh into a regular mesh, but using subdivision schemes for remeshing. Subdivision schemes create subdivision-connectivity and meshes approximating the originals. Therefore a (semi-)regular mesh is made by recursive subdivisions from irregular mesh like Fig. 1.

As below Fig. 2, first a base domain is found, then the base domain is subdivided, and the wavelets are applied at new midpoints of the old edges, and finally a semi-regular normal mesh is produced to describe the same geometry by successive subdivision of coarse base domain faces. The base domain is the coarsest level of the irregular mesh we simplified.

In 1995, Eck et al.\(^1\) used Voronoi tiling to partition a mesh into a number of triangular regions and use harmonic maps to build a parameterization which affinely map the triangular region onto the corresponding face of the base domain and remesh onto a semi-regular form through remeshing process.

In 1998, Lee et al.\(^3\) proposed an algorithm based on the mesh simplification to find a base domain. Concurrent with simplification, a conformal mapping is constructed which assigns every vertex from the original mesh to a base domain.

In our research, we use the vertex pair contraction of Garland\(^{2,4}\) for mesh simplification to find a base domain, then recursively subdivide the base domain to build a regular mesh like Fig. 3, where the wavelets for multiresolution are calculated by normal samplings.

### 2.2 Related works on normal mesh

In 2000, Guskov et al.\(^5\) and Lee et al.\(^6\) proposed algorithms for normal mesh and displaced mesh respectively. These algorithms found a simpler geometry and a set of normal offsets which together are equivalent to the given original mesh.

Guskov et al. first computes a base point using interpolating Butterfly subdivision as well as an approximation of the normal. This defines a straight line. And then the intersection point with the original mesh is found to get a normal offset. This recursive subdivision and normal sampling build a normal mesh.

Lee et al. obtains an initial control mesh, and globally optimizes the control mesh vertices, and samples the displacement map by shooting rays along the domain surface normals until they intersect the original mesh. At the ray intersection points, the signed displacement is computed.

In the case of the mesh with a boundary, Guskov et al. worked only for surfaces without boundaries, and Lee et al. used vector-based wavelet \((x, y, z)\) in the boundary. We extend Guskov et al.'s works to general meshes where boundary edges are considered, and present an algorithm to use a scalar-based wavelet (normal offset) for the open surface with boundaries.

### 3. Multiresolution Analysis and Normal Mesh

Multiresolution analysis decomposes the original information into a lower level resolution version and detail coefficients (differences), and the original information is transformed into multi-resolution information during this process. The linear B-spline curve of Fig. 4(a) can be represented in another form, like (b), which is a coarser representation of the overall shape, and is made by removing the mid control points, with wavelets that capture the missing information. The curve (a) represents only the high-resolution, whereas the curve (b) represents high-resolution as well as low-resolution, because the high-resolution curve (a) can be recovered by subdividing the low-resolution curve (b) and adding wavelets at

---

Fig. 1 Irregular mesh and semi-regular mesh

Irregular mesh

- Irregular mesh
- Semi-regular mesh

Subdivision

![Irregular mesh and semi-regular mesh](image1)

Fig. 2 Semi-regular mesh through a base domain subdivision

Irregular mesh

- Base domain
- Semi-regular mesh

Mesh simplification

Subdivision

![Semi-regular mesh through a base domain subdivision](image2)

Fig. 3 Remeshing and multiresolution analysis of Bunny model

Original irregular mesh

Base domain

Fractal subdivision

Irregular mesh (505)

Irregular semi-regular mesh (773)

Corner semi-regular mesh (1036)

Subdivision

+ Wavelets (Normal Sampling)

![Remeshing and multiresolution analysis of Bunny model](image3)
new midpoints.

As shown in Fig. 5, the high resolution linear B-spline curve is constructed by subdividing the low-resolution curve (piecewise linear B-spline curve with two control points) into two sub-curves by introducing edge midpoint and by adding the detail information, i.e. a wavelet which is the difference between the curve (a) and curve (b) at the midpoint of every edge of level $j$ generates a new vertex on level $j+1$. When wavelets are added in all the new midpoints for the edges of a face of level $j$, we can build the faces of level $j+1$. There is subdivision connectivity between curves or surfaces of level $j$ and that of level $j+1$. The curves or surfaces can be hierarchically decomposed only if they can be generated through a simple process known as recursive subdivision.

A normal mesh is a multiresolution mesh where each level can be written as a normal offset from a coarser version. Figure 6 shows the difference between a general mesh and a normal mesh.

The general mesh of Fig. 6 (a) is described by the grayed base domain and the wavelet vector $(x_i, y_i, z_i)$ which is the difference between the midpoint $m$ of each edge and vertex $v$, i.e. $v = m + (x_i, y_i, z_i)$. But the normal mesh of Fig. 6 (b) uses normal offset $(s_i)$ as the wavelet, i.e. $v = m + s_i \cdot N$ ($N$ is a unit normal vector).

Figure 7 shows how a normal mesh represents a multiresolution. The vector wavelets are calculated to minimize the error between a lower resolution mesh and a higher resolution mesh, whereas the normal wavelets do not minimize the error, because they are constrained only in the normal direction of the face of the lower resolution mesh. However, if the lower resolution mesh is subdivided recursively, the normal wavelets can represent the original mesh as well.

In the near surface boundaries it is generally difficult for a mesh to be represented by a normal offset. The problem is that the domain surface cannot cover the original surface. A closed surface as shown by Fig. 8 (a), is easily represented by normal offsets on the base domain, whereas an open surface, which has boundaries as shown by Fig. 8 (b), is not easily represented by normal offsets. To represent a surface with boundaries by normal offset, the surface boundaries should be located exactly in the normal direction of the base domain boundaries.

Hence the only way to obtain a normal mesh is to change the base domain. We utilize the combined subdivision scheme to change the base domain to given surface boundaries.
4. Proposed Remeshing Algorithm

4.1 Overview of the proposed algorithm

Figure 9 gives an overview of the proposed remeshing algorithm.

To convert an original irregular mesh into a normal regular mesh, the algorithm performs the following steps as shown in Fig. 10 for a simple case:

1) In the first step, the boundaries of the original irregular mesh are searched and the boundary vertices are denoted with tag.
2) In the second step, the original mesh is successively simplified into a base domain. Simplification is done using a sequence of vertex pair contraction\(^{33,44}\).
3) In the third step, a parameter of the removed vertex is computed together with the mesh simplification. The mapping of the removed vertex on the base domain can be described as barycentric coordinates with respect to the base domain.
4) In the forth step, a normal regular mesh is constructed using parameterization information about the relation between the original vertex and the mapped vertex on the base domain. At first, a coarse semi-regular mesh is found that satisfies the boundary conditions by combined subdivision, and then normal sampling is carried out at each vertex of the coarse semi-regular mesh. These combined subdivisions and normal sampling steps are recursively applied to build a fine semi-regular mesh from the coarse semi-regular mesh.
5) In the final step, the remeshed normal mesh is restored in the normal mesh format.

4.2 The steps of remeshing

4.2.1 Boundary search

As shown in Fig. 11, an edge is called a boundary edge if it is not shared by two faces. A vertex is called a boundary vertex if it belongs to a boundary edge.

There may exist more than one boundary, and the data structure of boundary representation is described in Fig. 12.

In the combined subdivision of the re-parameterization step, the indexes of the boundary to which the boundary vertices belong and the parameter representing the order of the boundary vertex in the boundary list are needed. Therefore, each boundary vertex has the list index of the boundary and the ordering parameter of the boundary vertex list. For example, in Fig. 12, boundary vertex 5 has the list index ‘1’ from the boundary list and parameter ‘3’ from vertex list of the boundary.

4.2.2 Base domain generation

Locally smooth surfaces can be described by a single scalar height function over the tangent plane. But the whole surface cannot be described by a single scalar height function as shown in Fig. 13(a), because the parameterization function that map the vertex of the base domain to the surface cannot assign each element of the base domain to a unique element of the surface. Therefore, the surface should be parameterized over a base domain composed of a number of triangles face as shown by Fig. 13(b).

In order to produce a good base domain made up of a small number of triangles, we use the vertex pair contraction\(^{33,44}\) that uses a sequence of edge collapse transformations prioritized according to the quadric

Fig. 9 Overview of the proposed remeshing algorithm

Fig. 10 Remeshing steps of irregular meshes with boundaries

Fig. 11 Boundary edge and inner edge

Fig. 12 Data structure of boundary edges
error metric, which is the cost of a contraction defined by quadratic form. The mesh shown in Fig. 14(a) is a mesh before vertex pair contraction, and the mesh in Fig. 14(b) is the same mesh with vertex \( v \) and \( p_s \) contracted. The vertex \( v \) is removed, and the vertex \( p_s \) moves to the optimal position, which minimizes the sum of squared distances from the target point in space to the grayed planes sharing vertex \( v \) and \( p_s \).

In the resulting base domain, the region of low curvature is simplified, whereas the region of high curvature is held in reserve. The base domain is now in a suitable form for expressing the original mesh using a normal offset.

### 4.2.3 Parameterization

As is shown in Fig. 14, to maintain an efficient correspondence between the original mesh and the simplified mesh, we have used the MAPS scheme\(^{(2)}\) to track parameterizations of all original vertices on the simplified mesh so far. The removed vertex \( v \) is projected onto the face of the shortest distance from the vertex \( v \) among the grayed faces sharing the vertex \( p_s \). The mapping point \( II(v) \) can be described by Eq. (1), where the triangle \((p_u, p_s, p_i)\) is a face of the base domain and \( a, \beta, \gamma \) are the corresponding barycentric coordinates.

\[
II(v) = a p_u + \beta p_s + \gamma p_i
\]  

(1)

The data structure for the parameterization using vertex pair contraction is as below, Fig. 15.

When an edge \((p_u, p_i)\) is removed as shown in Fig. 16, the grayed face sharing vertex \( p_s \) and \( p_i \) is modified.

Therefore the earlier parameterizations of point \( II(v) \) in the neighborhood are updated onto the resulting neighborhood like Fig. 17.

If the boundary vertex \( p_s \) is removed, the lined faces are removed too, and the vertex \( p_i \), not having a
surface with a boundary as shown in Fig. 19. The combined subdivision takes into consideration the boundary condition only near the boundary, but applies the general subdivision scheme away from the boundary.

The boundary curve may be given in the form of a smooth parametric curve or any other form. In this study, the boundary curve is assumed to be a uniform linear B-spline curve interpolating the boundary vertices found at the boundary search step.

New vertices are generated, as shown in Fig. 19, from a combined subdivision of the base domain (a). Initially, every old boundary edge is subdivided to generate the new boundary vertices $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and inner vertex $\mathbf{v}_4$ and then the boundary vertices $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ are moved to the boundary.

In the case of the vertex $\mathbf{v}_4$, which is generated by subdivision of the edge defined by the vertices $\mathbf{v}_0$, $\mathbf{v}_1$, the position of the vertex $\mathbf{v}_4$ is the point on the boundary curve with the parameter $\frac{u_4 + u_5}{2}$ that is an average value of the parameters of the vertex $\mathbf{v}_0$ and $\mathbf{v}_1$.

As an example of the combined subdivision, Fig. 20 shows the base domain, the boundary, and the parameters of a curved boundary surface. New vertices are generated as in Fig. 20 (b) by the combined subdivision of the base domain of Fig. 20 (a). When the edge composed of the vertices having the parameters 9 and 16 is subdivided, a new vertex having a parameter value of 12 on the boundary curve, is obtained by averaging the values 9 and 16 (if it is not a integer, it will be rounded down).

2) Normal sampling

The normal sampling is performed for new vertices of the semi-regular mesh to determine their positions on the original mesh. It is done to compute the normal of the base domain surface and the signed distance from the vertices to the original surface along the normal at each vertex. The normal vector of the vertex $\mathbf{v}_b$, which is not a boundary vertex, is the average of the normal vectors of the faces sharing the vertex $\mathbf{v}_b$ shown in Fig. 21. For the boundary vertices such as $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ which lie on the boundary of the surface itself, normal sampling is not necessary.

To check where the line defined by the vertex $\mathbf{v}_b$ and normal vector intersects the original irregular mesh surface, as shown in Fig. 21 (a) the parameter coordinates $\Pi(v)$ is used as an initial indicator, which was stored at the parameterization step.

First, a nearest parameter $\Pi(v)$ to the vertex $\mathbf{v}_b$ is searched, and the vertex $\mathbf{v}$ of the original mesh for the parameter $\Pi(v)$ is determined, and then an intersection point between the normal vector of vertex $\mathbf{v}_b$ and a triangle sharing vertex $\mathbf{v}$ is calculated.

If the barycentric coordinate of the intersection point is not between 0 and 1, the intersection point is outside the triangle. The position of a real intersection point is estimated according to the sign of the barycentric coordinate, as shown by Fig. 22.

For example, if the $w$ of barycentric coordinates has a negative value, the real intersection point may be located to the left of the triangle. In this case, the process of finding the intersection point is repeated until every barycentric coordinate is between 0 and 1. For the Bunny model, most intersection points are found within 10 iterations.

Figure 23 illustrates an example of a curved sur-
face that could result in a possible failure to find intersection points.

In the case of the bad intersection point, where the magnitude of the normal vector is extraordinarily large compared to the average value of the normal sampling, the bad intersection point is disregarded. To avoid such intersection points, a good base domain approximating the curved surface is needed.

The normal offset found in the normal sampling step is the wavelet of the vertex $v_b$. In Fig. 24 (a), the vertex $v_b$ is replaced by adding the wavelet. $v_b$ is a point on the original mesh is the final position of the vertex of the semi-regular mesh. Figure 24 (b) shows the next recursive triangular subdivision of the present semi-regular mesh and the normal sampling at the vertex of the new finer semi-regular mesh. This recursive subdivision and normal sampling construct the re-parameterization step.

Figure 25 describes the relationship between multiresolution and remeshing in our algorithm. The wavelets for multiresolution are calculated by normal samplings. The base domain and the determined wavelets at each resolution are able to represent the multiresolution of the original mesh.

4.2.5 Normal mesh format To represent a mesh with 4 faces, shown in Fig. 26, a general mesh format needs the $x$, $y$, and $z$ coordinates of 6 vertices (the geometric information) and 4 connectivities (the topology information) between the vertices. On the other hand, normal mesh format needs the $x$, $y$, and $z$ coordinates of 3 vertices of the base domain with only 3 scalar normal offsets for geometric information and

only 1 connectivity between the vertices of the base domain, because the topological information is already available from the regular subdivision rule.

The method of finding the topological information using the regularity of subdivision is described in Fig. 27. When the base domain is subdivided twice, 15 generated vertices from 0 to 14 are set in order, and this order is determined by the number of subdivision. If a normal offset is stored in this order, more topology information is unnecessary. For example, the vertices that are stored after the order of 3, 5, and 12 are the vertices generated in the 1st subdivision step and the others are the vertices generated in the 2nd subdivision step.

5. Results of Remeshing

We have implemented and tested the algorithms described in the preceding sections. The test model of Fig. 28 is the Bunny model of Stanford University made up of an irregular triangular mesh with 69,451 faces. We generated a base domain with 1,089 faces,
and after 3 successive steps of subdivision and normal sampling, a normal regular mesh with 69,696 faces was reconstructed using the remeshing process.

When the remeshed normal mesh is stored in the normal mesh format, the size of Bunny model is reduced from 2.3M to 0.52M, about one fifth of the original size.

5.1 The face numbers of the base domain

The quality of the remeshing depends significantly on the number of faces in the base domain. A base domain with many faces approximates well to the original mesh. Therefore, normal samplings generally succeed at vertices, and the number of non-normal wavelets and the remeshing errors decrease, but the drawback of this is that the amount of information required increases.

Many subdivisions do not guarantee a good remeshing result, and the process is probably optimal when the remeshed normal mesh contain roughly the same number of triangles as the original mesh, as shown in Eq. (2)

\[
\text{Num. of orginal faces} \approx \text{Num. of base domain} \times 4^{\text{Num. of subdiv.}}
\]  

(2)

Table 1 presents a summary of the comparison between the remeshing qualities of two base domains, one with 1,089 faces and another with 270 faces. In order to evaluate the quality of a normal mesh remeshed by our algorithm, we have chosen a metric that measures the average distance between the remeshed mesh model and the original mesh model. Because a wavelet represents the difference between the remeshed and the original mesh model, an error is simply measured by the magnitude of wavelets. Note that the number of faces of a base domain influences errors and the non-normal percentage, which is a percentage of the failed normal sampling. All computing times were measured on a Pentium 667 MHz with 256 MB of memory. The errors indicate error distances over the diameters of bounding sphere holding all of the data expressed as a percentage and the non-normal indicates the number of non-normal vertices over the number of total vertices of the mesh model expressed as a percentage.

5.2 Other models

We performed a series of experiments in which normal meshes for various models were built. The summary of the results is given in Table 2. All four models, the bunny, the cat, the engine, and the mannequin, have boundaries. In particularly, the engine was composed of several separated domain surfaces with boundaries and demonstrates the advantage of our method in terms of handling irregular mesh with boundaries. Figure 29 illustrates the steps of the algorithm and presents examples of its applications. As more levels of combined subdivision and normal

<table>
<thead>
<tr>
<th>Data set</th>
<th>Bunny</th>
<th>Cat</th>
<th>Engine</th>
<th>Mannequin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input face</td>
<td>69451</td>
<td>6989</td>
<td>11891</td>
<td>15544</td>
</tr>
<tr>
<td>Input size</td>
<td>2,340 KB</td>
<td>231 KB</td>
<td>411 KB</td>
<td>560 KB</td>
</tr>
<tr>
<td>Base domain</td>
<td>1089</td>
<td>109</td>
<td>185</td>
<td>241</td>
</tr>
<tr>
<td>Subdiv. Level</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Output face</td>
<td>69696</td>
<td>6976</td>
<td>11840</td>
<td>15442</td>
</tr>
<tr>
<td>Output size</td>
<td>524 KB</td>
<td>56 KB</td>
<td>97 KB</td>
<td>116 KB</td>
</tr>
<tr>
<td>Error</td>
<td>0.012 %</td>
<td>0.033 %</td>
<td>0.022 %</td>
<td>0.021 %</td>
</tr>
<tr>
<td>Non-normal</td>
<td>0.93 %</td>
<td>1.53 %</td>
<td>0.62 %</td>
<td>0.99 %</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>1.58</td>
<td>0.11</td>
<td>0.18</td>
<td>0.23</td>
</tr>
</tbody>
</table>
sampling are applied, the model becomes a finer semi-regular mesh which approximates the original mesh.

6. Conclusion

In this paper we present a method for overcoming the subdivision connectivity restriction, which means that irregular meshes are remeshed to regular meshes with subdivision connectivity so that classical multiresolution and wavelet algorithm can be employed. Particularly, we remesh an irregular mesh into a normal mesh. A normal mesh is a multiresolution mesh each level of which can be described as a normal offset from a coarser version. Hence the normal mesh can be stored with a single float per vertex, while other remeshing methods use 3 floats per vertex.

To solve the problem that a boundary is difficult to express with normal offset, we used the combined subdivision scheme to take into consideration given boundary conditions. And remeshed normal meshes are restored in normal mesh format, which is effective at reducing the amount of data, because it uses regular rules of uniform subdivision and 1 float per vertex.

Our method as demonstrated in this paper can be effectively used to increase rendering speeds and to allow the rapid transmission of 3D models in network-based applications.

Acknowledgements

This research was supported in part by the Research Institute of Marine System Engineering of Seoul National University, and in part by the Korean Ministry of Science and Technology through the National Research Laboratory for Modeling Kernel Development.

References

(10) Levin, A., Combined Subdivision Schemes, Ph.D’s thesis, School of Mathematical Sciences, TelAviv University, (2000).