An Adaptive Subdivision Scheme On Composite Subdivision Meshes

Anh-Cang PHAN
VinhLong College of
Economics and Finance
VinhLong, Vietnam
pacang@vcef.edu.vn

Romain RAFFIN
Aix-Marseille University
CNRS, LSIS UMR 7296
13009, Marseille, France
Romain.Raffin@univ-amu.fr

Marc DANIEL
Aix-Marseille University
CNRS, LSIS UMR 7296
13009, Marseille, France
Marc.Daniel@univ-amu.fr

ABSTRACT

One of the commonly used techniques in hole filling and mesh joining is the construction of connecting meshes between meshes to generate a new mesh model consisting of the composite subdivision meshes. One problem in subdivision meshes is how to further subdivide this reconstructed mesh model or these composite subdivision meshes to enhance the quality of the surface as needed. In this paper, we propose a new local subdivision method of the composite subdivision meshes. Our method does not alter the surrounding mesh areas, and guarantees that the discrete continuity between these meshes is preserved without the occurrence of cracks or holes between them. We specifically address a local subdivision scheme on a connecting mesh (the mesh area covering hole or crack) suitable for refining only the connecting mesh or the selected mesh areas. Our method can produce a smooth mesh model with a natural shape, and allows approximation or interpolation of surfaces. We implement our method for various triangular meshes and present our experimental results.

Keywords

Subdivision surfaces, Adaptive scheme, Triangular mesh.

1 INTRODUCTION

3D object models with complex shapes are generated by a set of assembled patches or separate mesh areas which may be at different resolution levels, even with different subdivision schemes. Cracks, gaps or holes may appear along the boundary between these patches if we further subdivide them at different resolution levels, even with different subdivision schemes. On the other hand, some research related to mesh connection such as in [Phan12, Phan13a] often lead to the construction of a high quality connecting mesh CM (the mesh area covering hole or crack) between meshes M_1, M_2, and a continuous surface. After the connecting mesh CM is produced, we can further subdivide CM and/or M_1, M_2 of a model to enhance the quality of the surface as needed. Unfortunately, there are two problems posed for subdivision of CM and/or M_1, M_2. If we subdivide M_1, M_2 and CM separately by different subdivision schemes, i.e global subdivision schemes in Butterfly [Dyn90], Loop [Loop87], Kobbelt [Kobbelt96], cracks or holes can also reappear on the surface after subdivision. This prevents some further processing of the mesh and high quality rendering. Therefore, it is not ideal to apply a global scheme for this subdivision. But if we use adaptive subdivision methods, i.e methods available in [Amresh02, Liu04, Pakdel04, Pakdel07, Husain10, Husain11], the neighboring faces around the subdivided area are also subdivided to avoid cracks. Thus, it changes the connectivity and valence of vertices around the subdivided area, produces some extraordinary vertices and long faces that are unavoidable in most adaptive subdivision algorithms. This will not only alter the shape of the limit surfaces, but also reduce its smoothness. Consequently, the challenge is to subdivide the connecting mesh or composite subdivision meshes.

In order to deal with these problems, we have developed a local subdivision on a connecting mesh or composite subdivision meshes suitable for refining only the connecting mesh or the selected mesh areas so that it does not change the surrounding mesh areas, and ensures the surface continuity. Our contributions are as follows: 1) Provide a local subdivision scheme that manages whether or not a given face in a selected mesh needs to be subdivided at the next level of subdivision. 2) Propose an adaptive subdivision method of composite subdivision meshes defined with subdivision surfaces, each mesh being at a different subdivision level to generate a smooth discrete surface with a natural shape and visually fair connectivity.

The remaining of the paper is organized as follows: We briefly review previous and related work for subdivision surfaces in Section 2. Section 3 details the construction of our adaptive subdivision algorithm. We present

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

the continuity of surfaces generated by our method in Section 4. Then we show and compare implementation results of our algorithm in Section 5. Finally, we draw the conclusion in Section 6.

2 PREVIOUS WORK

As mentioned in Section 1, the subdivision schemes of the whole mesh will be inefficient because if a face is subdivided in a subdivision step but its neighboring faces are not, cracks will be created as shown in Fig. 1.

To overcome this problem, incremental adaptive subdivision is sometimes necessary in applications to refine only selected mesh areas of 3D models. Several adaptive subdivision methods [Amresh02, Liu04, Pakdel04, Pakdel07, Husain10, Husain11] had been proposed to determine the areas to be subdivided and handle cracks. These methods refine a subset of the faces of the control mesh and remove cracks so that the surface can be further used in applications. On the other hand, many methods dealing with cracks and holes are introduced in [Jiang, Cas05]. These methods adopt various criteria to compute the planar shape from a 3D surface patch. However, they do not mention the continuity and the progressive change in resolution between meshes after dealing with cracks and holes.

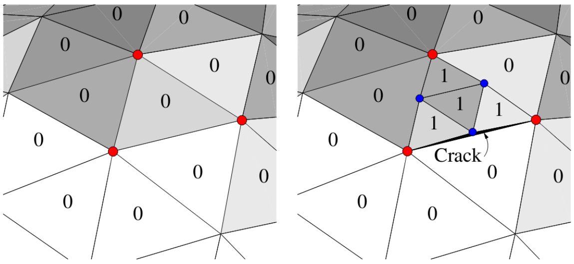

Figure 1: Crack on the mesh is caused by subdividing a face (taken from [Pakdel07]). The numbers represent the subdivision level.

Although these existing algorithms deal with cracks and holes, they provide partial or inefficient solutions for subdivision of a connecting mesh. These methods recommend refining the areas of interest with the same subdivision scheme. The area outside the adaptively subdivided area is also subdivided in the most common case to handle cracks, and therefore they create more faces. On the other hand, removing cracks with the insertion of edges makes a change in the connectivity of vertices. It also modifies the resulting surface of the subdivision process. Consequently, handling cracks is complex and expensive in computational time.

In fact, no existing adaptive subdivision method satisfies all desires for a smooth surface consisting of patches or meshes with different resolutions. In this paper, we introduce a new adaptive subdivision method on composite subdivision meshes or a connecting mesh between two meshes of different resolutions. This subdivision method is based on the *dead* property of vertices which are boundary vertices between triangular meshes to consider if the faces should be subdivided or not at the next subdivision level. It has been designed as an efficient solution for refining a selected mesh area without handling cracks and altering the original mesh areas around the adaptively subdivided area while maintaining the continuity between them. Consequently, it can avoid creating long faces both inside and outside the subdivided area. It permits to follow the junction between two initial triangular meshes along the subdivisions of the meshes. In addition, our subdivision algorithm can be easily applied to triangular subdivision schemes (i.e Butterfly or Loop schemes).

3 OVERVIEW OF THE ALGORITHM

Our method is inspired from the adaptive subdivision methods for Loop subdivision scheme [Amresh02, Liu04]. They are based on the angle of the normal vectors of adjacent faces of a face or a vertex considered as error estimation to decide whether the face needs to be subdivided or not at the next subdivision step. However, the user needs to determine a good threshold for these angles to set the "flatness" property of faces. Our adaptive subdivision method is based on the "*boundary vertices*" of the selected mesh area to define the property of each vertex to be *dead* or *alive* as introduced in Section 3.1. Based on the property of each vertex, our subdivision scheme only refines the connecting mesh or the selected mesh areas.

In order not to alter the mesh areas around the adaptively subdivided mesh area, faces of these areas closest to the original boundaries between meshes must not be subdivided and their boundaries are kept to avoid handling cracks. This problem is overcome by a dead vertex based refinement (DVR) scheme. It is designed to refine faces of the adaptively subdivided mesh such that the connectivity between meshes is preserved. In the following, we will first introduce a definition of *dead* vertices, edges and faces related to our algorithm.

3.1 Definition of dead vertices, edges and faces

Given two triangular meshes M_1 and M_2 joined by CM, an edge is usually shared by two triangular faces. If it is shared by only one, it corresponds to a boundary edge and its end vertices are called boundary vertices. Let us introduce some items used in our algorithm.

- **For vertices**: to keep the boundaries of meshes M_1, M_2 and CM, we will mark boundary vertices as *dead* vertices. In contrast, the remaining vertices are called *alive*. Thus, we can classify a vertex depending on its position.
- **For edges**: to keep boundary edges of meshes M_1, M_2 and CM, we must not subdivide these edges and mark them as *dead* edges. An edge is called *dead*, if its two vertices are *dead*. In all other cases, the edge is called *alive*.

- **For faces**: A face is called *dead*, if its three vertices are *dead*. In all other cases, the face is called *alive*. *Dead* faces must not be subdivided.

  Let n be the degree of deadness of a triangular face which equals to its number of *dead* vertices ($0 \leq n \leq 3$). For instance, if $n = 3$ (all vertices of a face are classified as *dead*), the face is called a *dead* face. In case $n \leq 2$, the face is called an *alive* face. It will be suitably refined with DVR scheme where the *dead* edge connecting two *dead* vertices must not be subdivided to keep the boundaries of meshes. Details of the refinement are presented in the following section. Fig. 2 is an example of *dead* and *alive* vertices, edges and faces.

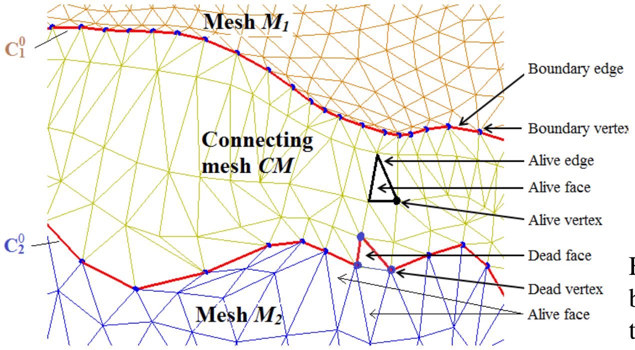

Figure 2: Definition of *dead* vertices, edges and faces.

3.2 Dead vertex based refinement scheme (DVR)

In order to refine a selected mesh area, such as the connecting mesh CM, we determine the property of faces of the model based on *dead* vertices. Our scheme called "Dead vertex based refinement scheme" is illustrated in Fig. 3. The rules of DVR scheme work as follows:

- **Classification of vertices**: For each vertex, we specify its property (*alive* or *dead*) based on definition in Section 3.1 and then mark it.

- **Adaptive subdivision rules**: For each face, we determine n the degree of deadness of a face based on the *dead* property of its vertices. Subdivision is then performed as shown in Fig. 3. There are four cases for a face $F_0 = \{v_1, v_2, v_3\}$ to create new subfaces based on the degree of deadness of F_0:

  - **Case n = 0**: if all vertices of F_0 are *alive*, F_0 is split into four new subfaces based on the regular subdivision method as shown in Fig. 3a. In this case, no new *dead* face is created while four new *alive* faces F_1, F_2, F_3, F_4 are created.

  - **Case n = 1**: if one vertex of F_0 is a *dead* vertex, F_0 is split into four new subfaces consisting of one newly created *dead* face and three newly created *alive* faces. This splitting indicates that *dead* faces spread gradually after each subdivision step. It implies that faces around the boundaries of meshes are gradually refined and thus it progressively changes the resolution between meshes. In Fig. 3b, one new *dead* face F_2 and three new *alive* faces F_1, F_3, F_4 are created.

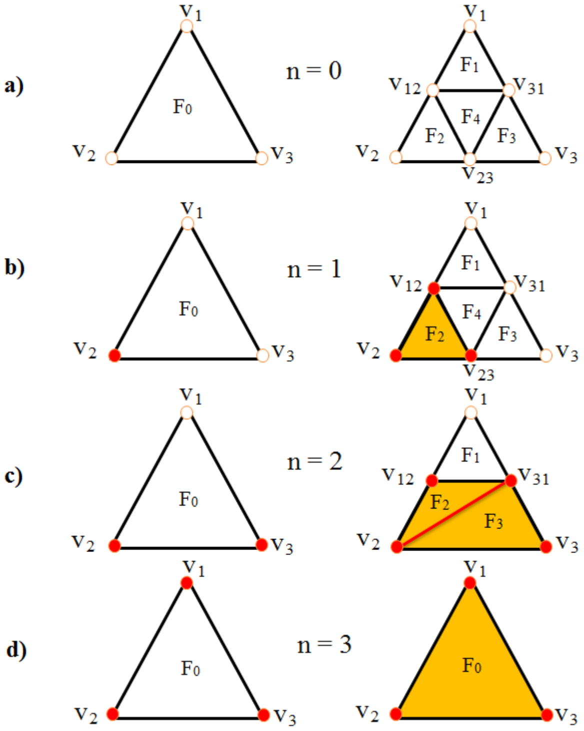

Figure 3: Our DVR scheme: adaptive subdivision rules based on the degree of deadness, where an *alive* vertex is represented as an empty circle; a *dead* vertex is represented as a red circle; an *alive* face is represented as an empty triangle; a *dead* face is represented as a yellow triangle.

  - **Case n = 2**: if two vertices of F_0 are *dead* vertices while one remaining vertex of F_0 is *alive*, F_0 is split into three new subfaces consisting of two newly created *dead* faces and one newly created *alive* face. Because we have two *dead* vertices, the *dead* edge connecting these two vertices must not be split while the remaining edges of F_0 are *alive* edges, and thus they are split. As a result, F_0 is split into three subfaces. In Fig. 3c, a triangular face is split into three smallest subfaces as possible. That is, three subfaces are created by comparing the length d_1 of edge v_2v_{31} with the length d_2 of edge v_3v_{12}. If $d_1 < d_2$, F_0 is split into one new *alive* face F_1 and two new *dead* faces F_2, F_3 to have an optimal split with more compact triangles. Conversely, F_0 is split into new subfaces F_1, (v_3, v_{12}, v_{31}), and (v_3, v_{12}, v_2).

  - **Case n = 3**: if all three vertices of F_0 are *dead* vertices, F_0 will not be split and thus no new subface is created (see Fig. 3d).

The local refinement scheme indicates that faces of CM (or the subdivided area) close to the boundaries are split less than faces far to these boundaries leading to a progressive change in resolution between meshes. It can be also applied adaptively to generate more details in needed areas.

3.3 Our proposed adaptive subdivision method

A process of boundary detection is performed before applying DVR scheme to have the suitable mesh refinements. The adaptive subdivision is performed in two phases (Fig. 4).

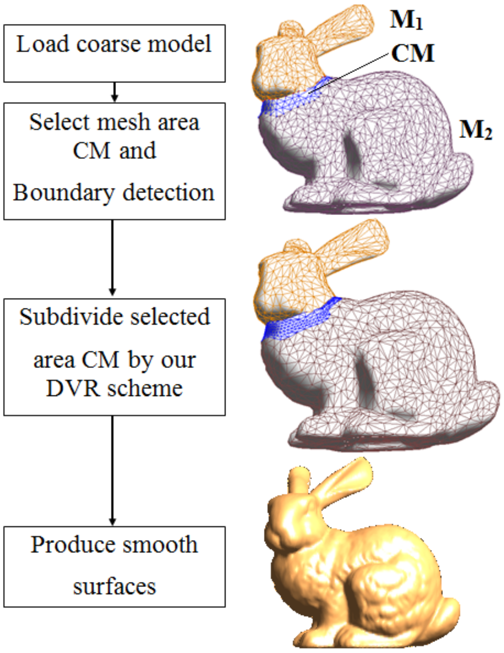

Figure 4: Framework for our subdivision method.

1. **Phase 1. Select mesh area CM and Boundary detection.**

   In this phase, we first read an input model at subdivision level j including meshes M_1 and M_2 joined by a connecting mesh CM (see Fig. 2). Then, we detect and mark boundary vertices of M_1, M_2 and CM.

   In this work, we join M_1 and M_2 by CM2D-TPW method [Phan13a]. CM is constructed by adding triangle strips to each boundaries of M_1 and M_2 until they are close enough to be linked based on the distance between them (smaller than a user specified threshold). This is done by using a tangent plane local approximation to create vertices of CM. Then, a Lifted B-spline wavelet transform is applied for multiresolution analysis of CM to obtain a progressive change in resolution between meshes after joining them together.

2. **Phase 2. Subdivide CM by our DVR scheme.**

   The adaptive subdivision process on CM at the next subdivision level $j+1$ is based on DVR scheme and described as follows:

   - ***Step 1. Classification of vertices and faces***. This step is to determine the *alive* or *dead* property of vertices and faces. Initially, we can consider the marked boundary vertices as a list of *dead* vertices. All the remaining vertices are set to *alive*. Based on definition of *dead* edges and faces in Section 3.1, we obtain the lists of *dead* edges and faces.

   - ***Step 2. Creation and classification of new vertices***.

     - New vertex creation: The vertex creation improves the smoothness of the mesh and the transition between M_1, M_2 and CM. No new vertices will be created for *dead* edges. For each *alive* edge, a new vertex is created by the odd vertex masks of the Linear, Butterfly or Loop subdivision schemes. The choice of these subdivision schemes to apply for our method does not depend on the subdivision schemes for initial meshes M_1 and M_2. Note that in case of the Linear subdivision scheme, each triangular face is divided into four new subfaces by adding new vertices in the middle of each edge. In case of the Loop subdivision scheme, if we use this scheme to create new vertices, *alive* vertices (even vertices) of CM will be repositioned as a linear combination of their neighbors by the even vertex mask.

     - New vertex classification: a newly created vertex of an *alive* edge is classified as a *dead* vertex if one vertex of the edge is a *dead* vertex. Otherwise, it is set to be an *alive* vertex as shown in Fig. 3. After that, we mark the property of the newly created vertices. This implementation is to gradually increase the number of *dead* vertices after each subdivision step to obtain a progressive change in resolution.

   - ***Step 3. Creation and classification of new faces***. For each *dead* face ($n=3$), no new subfaces will be created. For each *alive* face ($0 \leq n \leq 2$), new subfaces will be created by adaptive subdivision rules of DVR scheme (see Fig. 3).

   - ***Step 4. Update a list of** dead **vertices***. We add the newly created vertices which are marked to be *dead* into a list of *dead* vertices consisting of existing *dead* vertices and new *dead* vertices. Finally, we repeat steps 2 through 4 up to a user defined number of iterations.

4 THE CONTINUITY OF SURFACES GENERATED BY OUR METHOD

Because we concentrate on subdivision of triangular faces and the new vertices created by our method are based on triangular subdivision schemes, the convergence of subdivision surfaces built with our method depends on these schemes. For example, if we apply our method combined with the Loop scheme to create new vertices, the convergence analysis of our method resides in the convergence of this scheme presented in [Loop87].

In this work, we implement our subdivision method on Linear, Butterfly, and Loop schemes, but it can be applied to other triangular schemes with at least C^1 continuity. Obviously, it was shown that a subdivided mesh converges to a limit surface with C^1 or C^2 continuity except at boundary vertices if our method is applied to the Butterfly or Loop schemes respectively [Dyn90, Loop87]. A review of the continuity of surfaces for these subdivision schemes can be found in [Loop87, Dyn90, Doo78, Zorin96, Kawa06]. Besides, our method maintains the connectivity between meshes M_1, M_2 and CM because the boundaries of these meshes are kept unchanged during subdivision. On the other hand, our method can also be applied to single surfaces. The experimental results to illustrate our algorithm are shown in Section 5.

5 APPLICATIONS AND RESULTS

We implement our adaptive subdivision method for various triangular meshes of 3D object models. In this section, we give a number of experimental results. Our proposed algorithm has been implemented in Matlab.

5.1 Subdivision surfaces

In this section, the purpose of the application is to show that our method can produce less triangular faces than the others such as Linear, Butterfly, and Loop methods while keeping the surface continuity. We apply our adaptive subdivision method to the Linear, Butterfly or Loop subdivision schemes for refining the Mannequin mesh model. We also give some results of our adaptive subdivision with three these different subdivision methods. Fig. 5 shows the comparison of the Linear subdivision method and our adaptive method using the Linear scheme at level 1, where *dead* vertices consist of original boundary vertices and new *dead* vertices which are represented as red points and yellow points respectively after each subdivision step. It can be seen that our adaptive subdivision does not alter the original boundary of the Mannequin mesh while subdividing faces. The faces close to the boundary can be kept unchanged or be split into three subfaces. Moreover, our method can avoid handling cracks due to a difference in subdivision resolution of neighboring faces.

Some meshes and Gaussian curvature maps of the Mannequin model generated from Butterfly, Loop subdivision schemes and our method are illustrated in Figs. 6, 7. We can see that the quality of the surfaces subdivided by Butterfly, Loop schemes and our method is the same except in the boundary area where our method also applies the Butterfly, Loop schemes to create new vertices during subdivision process. Indeed, based on our subdivision rules and the results of our experiment, if we apply our method to the Butterfly or Loop subdivision schemes and the list of *dead* vertices is empty, our adaptive subdivision scheme is the same with the Butterfly or Loop schemes. However, in our method the numbers of *dead* vertices and *dead* faces increase after each subdivision step as illustrated by yellow vertices in Figs. 6, 7. Consequently, it can efficiently decrease the number of produced faces in every subdivision step as listed in table 1 while keeping the continuity property of subdivision surfaces.

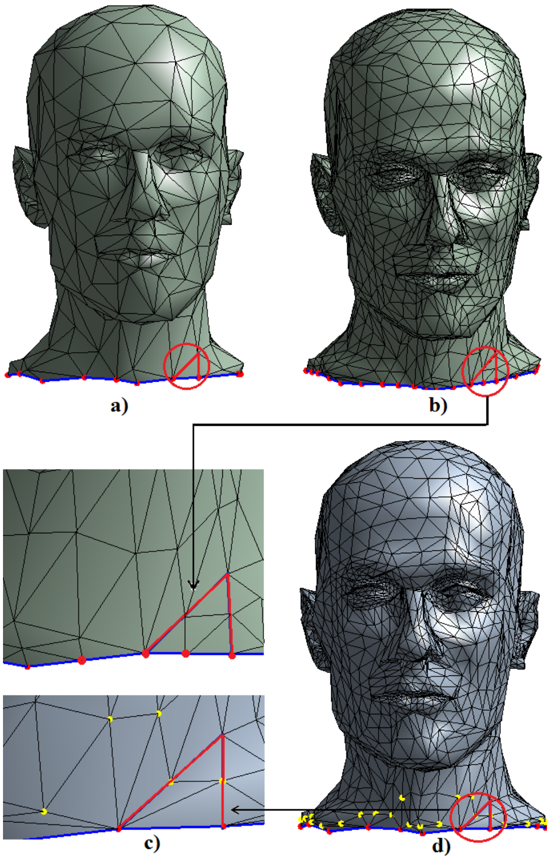

Figure 5: The Mannequin mesh generated by the Linear subdivision and our adaptive subdivision applied to the Linear subdivision scheme: a) Coarse mesh; b) Linear subdivision at level 1; c) Zooms of refined mesh by Linear subdivision and our method; d) Our adaptive subdivision at level 1.

We made some comparisons for the basic Linear, Butterfly, Loop subdivision methods and our method using these schemes at two subdivision levels. Table 1 shows the subdivision steps, the number of vertices (V) and faces (F) of meshes and the mesh reduction rate r for the Mannequin model, where $r = \frac{n_1 - n_2}{n_1} 100$, n_1 is the number of vertices or faces of meshes subdivided by Linear, Butterfly, Loop schemes, n_2 is the number of vertices or faces of meshes subdivided by our method. The more our subdivision process is implemented, the more *dead* vertices and faces are created. Consequently, the number of newly created vertices and faces of the refined mesh will decrease and thus the reduction ratio will increase.

Based on our results and comparisons, it can be seen that our proposed method can improve the basic subdi-

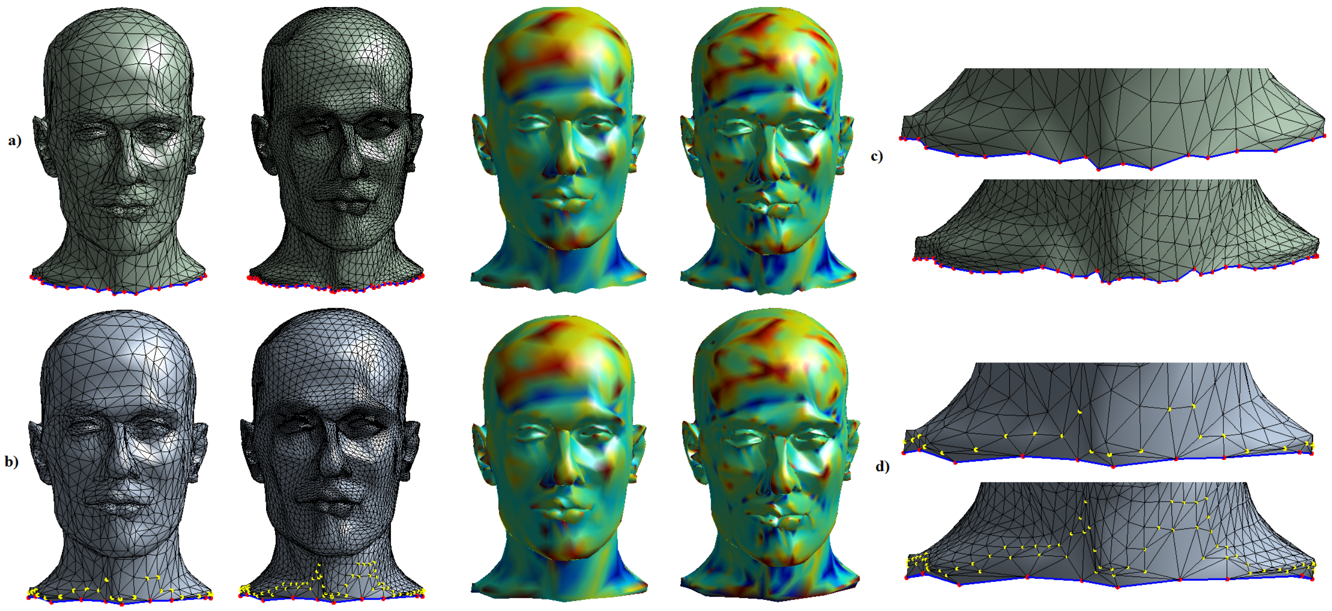

Figure 6: The Mannequin meshes and Gaussian curvature maps are produced by: a) Butterfly subdivision at levels 1 and 2; b) Our adaptive subdivision applied to the Butterfly subdivision scheme at levels 1 and 2; c) Zoom of one of the interesting parts of meshes in Fig. 6a; d) Zoom of one of the interesting parts of meshes in Fig. 6b.

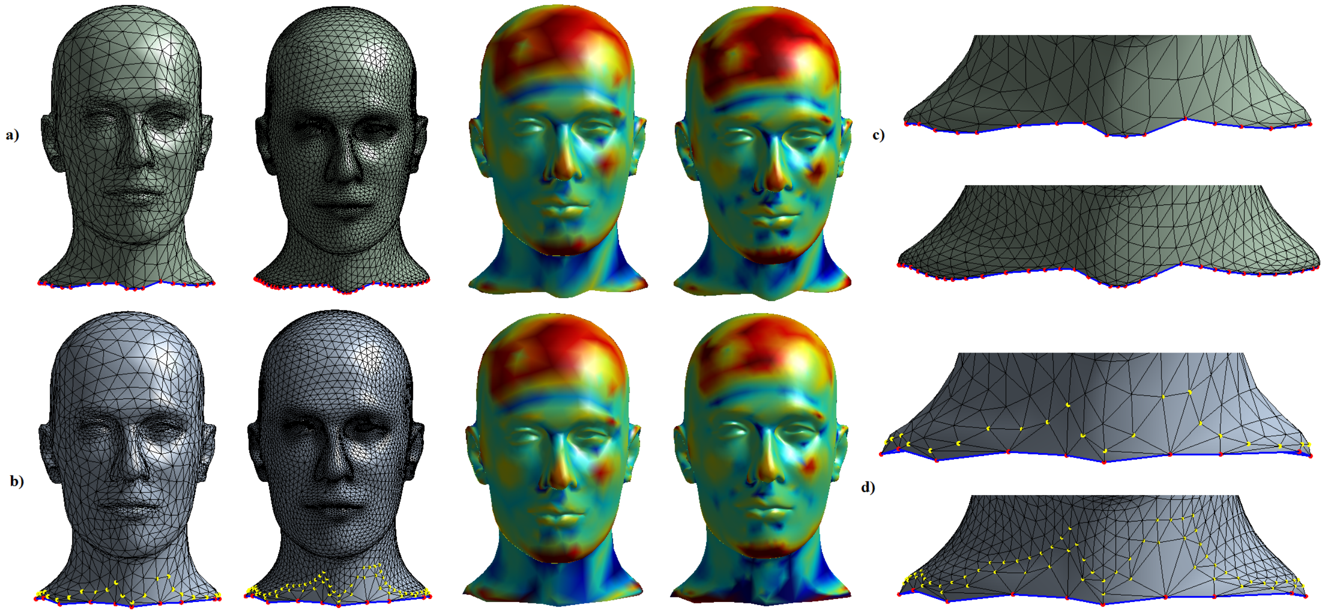

Figure 7: The Mannequin meshes and Gaussian curvature maps are produced by: a) Loop subdivision at levels 1 and 2; b) Our adaptive subdivision applied to the Loop subdivision scheme at levels 1 and 2; c) Zoom of one of the interesting parts of meshes in Fig. 7a; d) Zoom of one of the interesting parts of meshes in Fig. 7b.

Subdivision Method	**Coarse mesh**		**1 iteration**		**2 iterations**	
	V	F	V	F	V	F
1. Linear, Butterfly and Loop	428	839	1694	3356	6743	13424
2.Our adaptive subdivision	428	839	1679	3341	6598	13179
Reduction rate r (%)	0	0	0.89	0.45	2.15	1.83

Table 1: The number of vertices (V) and faces (F) in a Mannequin mesh model refined by Linear, Butterfly, Loop method and our method at different subdivision steps.

vision methods because it can reduce a number of vertices and faces while it guarantees the continuity, and locality of subdivision surfaces without causing cracks.

The continuity of surfaces generated by our method depends on the applied subdivision scheme as presented in Section 4. Consequently, surfaces produced by our method are continuous at least C^1, except at extraordinary vertices because we implement our method on the Butterfly or Loop schemes.

5.2 Subdivision of a connecting mesh on a hole

Next, we introduce another application of our method for refining a connecting mesh on a hole. To further subdivide a connecting mesh CM on a hole, we give a simple example of the Tet model with a hole filled by HF-RBFC method [Phan13b] using a RBF local interpolation and a centroid interpolation of boundary vertices. Then, we apply our adaptive subdivision scheme for the connecting mesh CM of the reconstructed mesh model consisting of the original mesh M and CM. Our method efficiently handles subdivision of these meshes with different levels of resolution, even with different subdivision schemes. We show results in Fig. 8 by applying our method to Loop and Butterfly schemes where the first row represents meshes and the second row represents surfaces. In our method, we designed a locally-controlled subdivision scheme. Thus it allows different tension in different mesh areas of a model, where the details of the local tension are given in [Dyn90]. This is not a disadvantage, because one would often like to have different tension in different mesh areas for design flexibility in manipulating real world objects. For example, if we subdivide many times the two mesh areas of the Tet model as shown in Fig. 8, the tension will not be the same since boundary vertices and edges (also called *dead* vertices and edges) between these two meshes are kept. As a result, we will gain a mesh and surface with different tensions.

We note that the user must control the result and an additional criterion of subdivision is required to control this result when applying our method such that the generated surface has a natural shape as desired. For instance, we cannot subdivide the meshes too much to avoid a too large difference in resolution between them because the resulting surfaces are deformed with undesired shapes as illustrated in Figs.8d-e.

5.3 Incremental subdivision

Another application of our method is incremental subdivision through refining only some selected areas of the mesh. Therefore, we can create surfaces that are densely subdivided in areas of higher curvature or selected areas. As a result, the selected mesh areas become finer while the rest of the mesh is coarse. Furthermore, the refined mesh progressively changes in resolution between coarse and fine areas as shown in Figs. 9, and 10. These figures illustrate examples of incremental subdivision of the right eye area of the Mannequin mesh. This allows us to gradually increase the resolution of the selected areas while keeping the surface continuity. We first apply our method to the Loop subdivision scheme for the right eye area with two resolution levels as illustrated in Figs. 9a-c. To observe the quality of produced surfaces, we plot meshes (top row), zooms of the interest right eye area (middle row), and surfaces (bottom row). The refined mesh and surface quality of the right eye area in Fig. 9c are finer than the ones in Figs. 9a-b because the resolution is higher while still maintaining the continuity of the surface. We then incrementally subdivide the right eye area and the complementary mesh area of the Mannequin model by applying our method to two subdivision schemes, and with different resolution levels as plotted in Fig. 10. Two mesh areas consisting of the right eye area and the complementary mesh area are incrementally subdivided by applying our method to the same Loop scheme but at different subdivision levels as shown in Fig. 10a. As a result, we obtain the refined right eye area with higher resolution. We also test our method with Butterfly scheme and give a result as plotted in Fig. 10b. Besides, we incrementally subdivide two mesh areas with different subdivision schemes, such as Loop and Butterfly schemes, and at different levels as shown in Fig. 10c. We also obtain a finer mesh especially in the right eye area.

5.4 Subdivision of a connecting mesh between meshes

We give some examples of subdivision of a connecting mesh CM generated by our mesh joining method (CM2D-TPW method [Phan13a]). After joining meshes M_1, M_2 by the connecting CM, we can further subdivide CM and/or M_1, M_2 by our subdivision method. Fig. 11 illustrates examples of applying our method to Butterfly or Loop schemes.

We first join two meshes M_1, M_2 to generate the connecting mesh CM using CM2D-TPW method (see Fig. 11b). From the reconstructed model in Fig. 11b, we then subdivide CM by our subdivision scheme to generate a new connecting mesh CM_1 while we do not recompute M_1, M_2 (Fig. 11c). Next, we subdivide M_1, M_2 by our scheme to produce new meshes M_1', M_2' while CM is kept as illustrated in Fig. 11d. Finally, we apply our method to further subdivide all meshes M_1, M_2, and CM. As a result, we obtain new meshes M_1', M_2', and CM_2. Fig. 11e shows that the new mesh CM_2 remains properly join with both new meshes M_1', M_2'.

To see the quality of the resulting meshes easily, Fig. 11 shows the images of the refined meshes and zooms of the corresponding meshes, where a) two initial meshes M_1, M_2 before connecting; b) the connecting mesh CM produced by CM2D-TPW method; c) CM is subdivided by applying our subdivision method to Loop scheme at level 1; d) M_1 and M_2 are subdivided separately by applying our method to Loop scheme at level 2 and Butterfly scheme at level 1, respectively; e) M_1, M_2, and CM are subdivided separately by applying our method to Loop scheme at level 2, Butterfly scheme at level 1, and Loop scheme at level 2 respectively; f)-h) Zooms of the corresponding meshes. According to the experimental results, our adaptive subdivision method is ef-

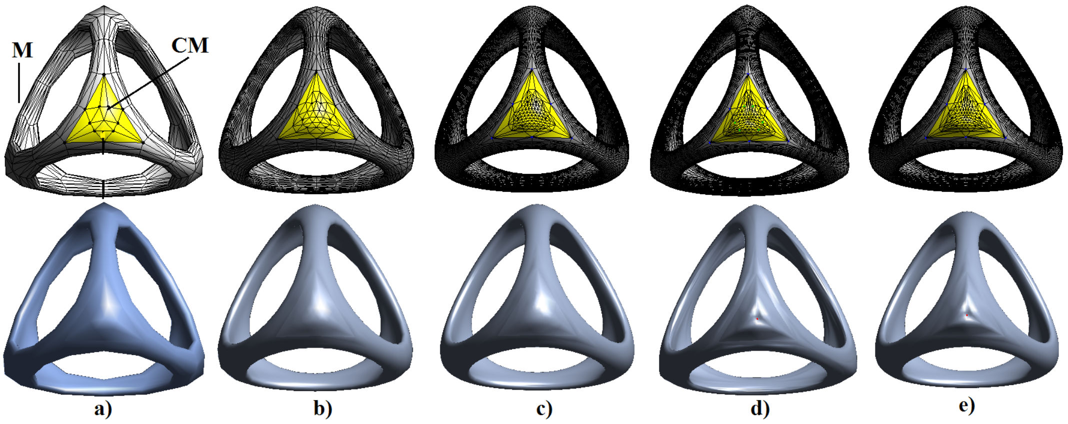

Figure 8: Zoom of subdividing the connecting mesh CM on a hole of the Tet model. The two meshes M and CM are subdivided by applying our method to two schemes respectively: a) Coarse mesh; b) Loop-Loop at level 1; c) Loop-Loop at level 2; d) Butterfly-Butterfly at level 2; e) Loop-Butterfly at level 2.

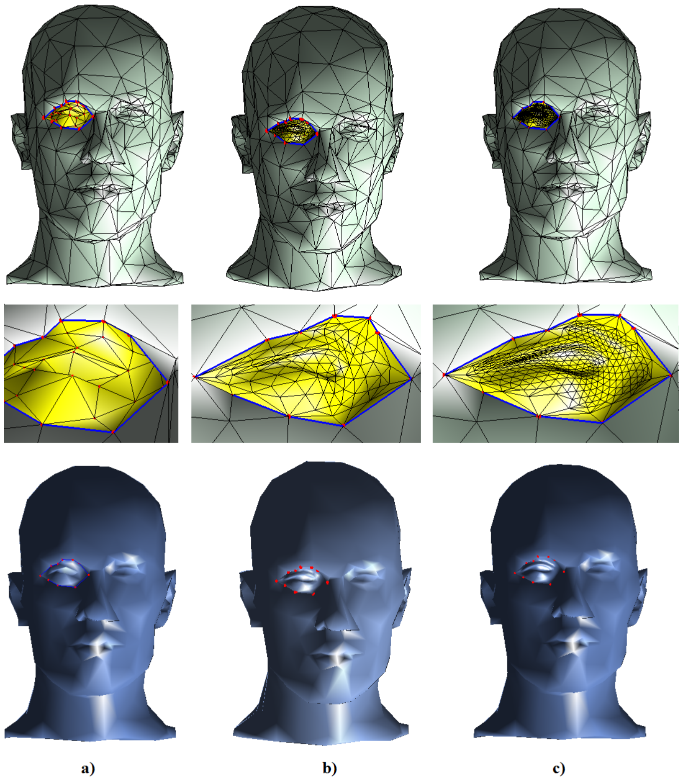

Figure 9: Incremental subdivision of the right eye area of the Mannequin model by applying our method to Loop scheme: a) Coarse mesh; b) at level 1; c) at level 2.

ficient to further subdivide the connecting mesh CM and/or meshes M_1, M_2. It can keep the boundaries of meshes and does not alter the surrounding meshes during subdivision. Therefore, our method can avoid handling cracks. As already mentioned in Section 5.2, the user must pay attention to the number of additional subdivisions. For instance in this example, a criterion to check for flatness of the group of triangles sharing the same vertex could be useful.

6 CONCLUSION

In this paper, we have introduced a new adaptive subdivision method for triangular meshes. Our method can be efficiently applied to the Butterfly or Loop subdivision schemes. It is based on the *dead* property of vertices. New vertices are created by these basic subdivision schemes. Therefore, our adaptive subdivision method produces C^1 and C^2 continuous surfaces, except at extraordinary vertices and boundary vertices. Our adaptive subdivision is a good completion work in filling holes and joining meshes because it can be applied on the connecting mesh CM and/or selected meshes M_1, M_2 so that new CM remains a correct connection with new subdivided meshes M_1 and M_2. This method is simple, reliable for adaptive subdivision to refine some selected mesh areas as needed. It allows us to separately refine selected mesh areas of a model at

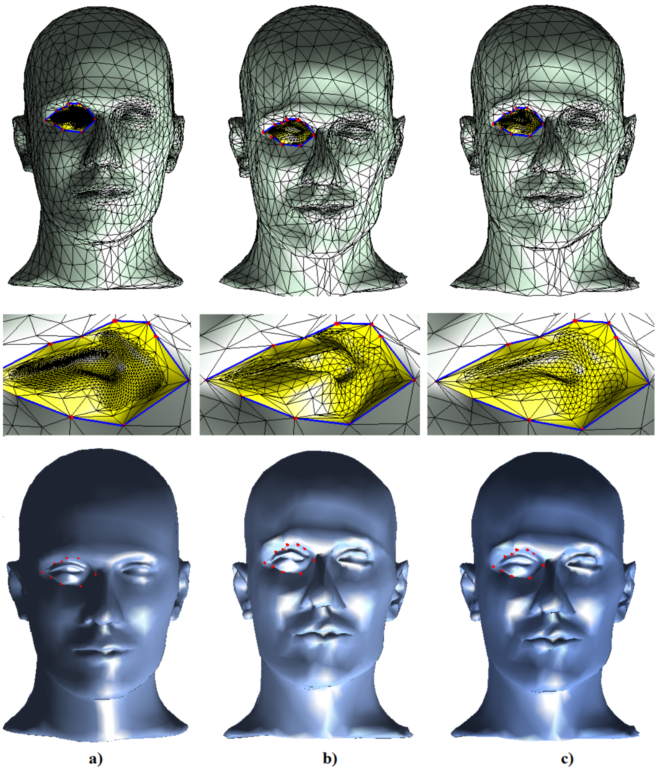

Figure 10: The right eye area and the complementary mesh area of the Mannequin model are incrementally subdivided by applying our method to two subdivision schemes respectively: a) Loop at level 3 for the right eye area and Loop at level 1 for the complementary area; b) Butterfly at level 2 for the right eye area and Butterfly at level 1 for the complementary area; c) Loop at level 2 for the right eye area and Butterfly at level 1 for the complementary area.

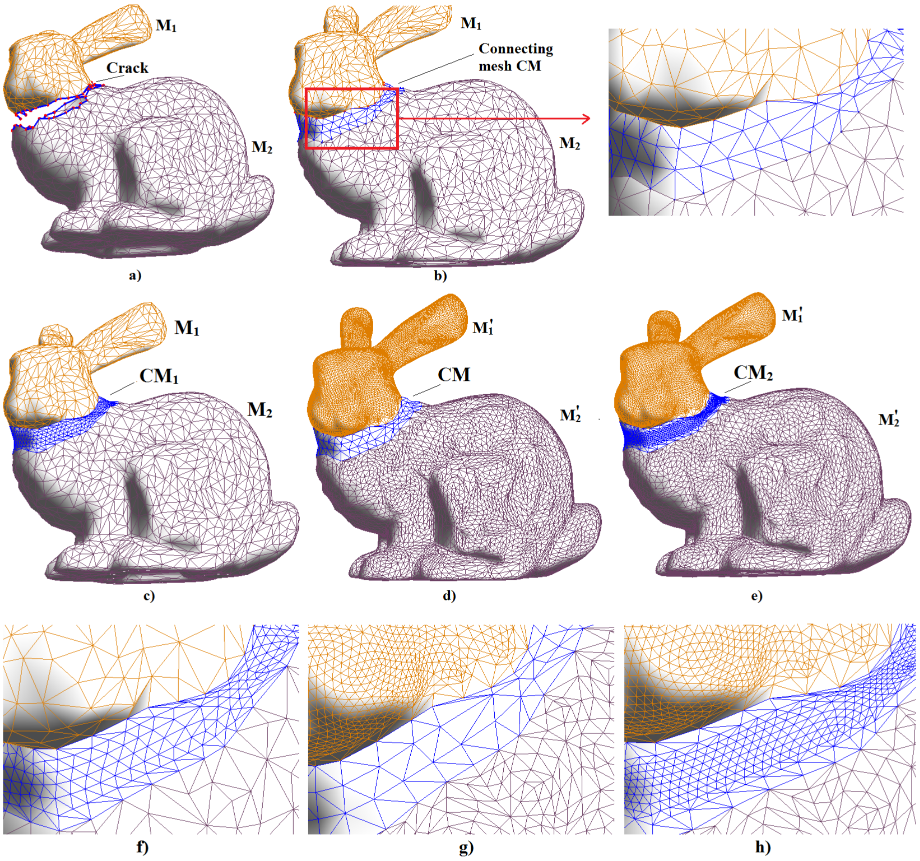

Figure 11: Two meshes M_1, M_2 and CM of the Bunny model are further subdivided by our adaptive scheme.

different resolution levels while maintaining the valid connectivity between meshes. Our method does not change the original mesh areas around the adaptively subdivided mesh during subdivision process. It can be also a drawback as if the original object is coarsely described, the subdivision does not alter these areas and a difference of density of vertices and faces is created. Moreover, our method does not create high valence vertices near the boundaries which can lead to ripple effects on surfaces as some existing algorithms [Zorin97, Seeger01, Amresh02]. Additionally, it can keep the original boundaries between meshes, thus our adaptive subdivision rules can prevent the appearance of cracks without needing to handle them.

As the boundaries are not changed in our adaptive subdivision method, the meshes cannot be subdivided too much and therefore a reasonable number of adaptive subdivisions is advised to avoid an important difference in resolution between meshes. The user must control the result such that the generated surface has a natural shape as desired. In addition, although our method can be applied for meshes CM and/or M_1, M_2, some slim or narrow triangular faces may be formed at the boundaries between meshes. This case happens because original meshes around the adaptively subdivided mesh and the original boundaries between them are kept. In order to suppress this drawback, it could be possible to insert vertices on existing boundary edges and make possible a local re-triangulation yielding to more regular triangular faces without modifying the boundaries. As we focus in the work we presented on keeping the two resulting subdivision surfaces unchanged we did not modify the boundaries. This study will be one of our future work.

7 REFERENCES

[Amresh02] A. Amresh, G. Farin, and A. Razdan. Adaptive Subdivision Schemes for Triangular Meshes. In Hierarchical and Geometric Methods in Scientific Visualization, pp. 319-327, Springer-Verlag, 2002.

[Cas05] G. Casciola, D. Lazzaro, L. B. Montefusco, and S. Morigi. Fast surface reconstruction and hole filling using positive definite RBFs. Numerical Alg., Vol.39, No.1-3, pp. 289-305, 2005.

[Doo78] D. Doo and M. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. In Computer Aided Design, Vol. 10, No. 6, pp. 356-360, 1978.

[Dyn90] N. Dyn, D. Levin, and J. A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. ACM Transactions on Graphics, Vol. 9, pp. 160-169, 1990.

[Husain10] N. A. Husain, A. Bade, R. Kumoi, and M. S. M. Rahim. Iterative selection criteria to improve simple adaptive subdivision surfaces method in handling cracks for triangular meshes. In Pro. of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 207-210, ACM, USA, 2010.

[Jiang] D. Jiang and N. F. Stewart. Reliable joining of surfaces for combined mesh-surface models. In Pro. of 21st ECMS, pages 297-303, 2007.

[Kawa06] H. Kawaharada and K. Sugihara. C^k-continuity of Stationary Subdivision Schemes. Technical reports/Mathematical engineering technical reports, The University of Tokyo, 2006.

[Kobbelt96] L. Kobbelt. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. In Computer Graphics Forum, pp. 409-420, 1996.

[Liu04] W. Liu and K. Kondo. An Adaptive Scheme for Subdivision Surfaces based on Triangular Meshes. Journal for Geometry and Graphics, Vol. 8, No. 1, pp. 69-80, 2004.

[Loop87] C. Loop.Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, 1987.

[Husain11] N. A. Husain, A. Bade, M. Rahim. Iterative Process to Improve Simple Adaptive Subdivision Surfaces Method with Butterfly Scheme. In Pro. of the World Academy of Science, Engineering and Technology 79, pp. 622-626, 2011.

[Pakdel04] H.-R. Pakdel and F. F. Samavati. Incremental Adaptive Loop Subdivision. In Pro. of the ICCSA 2004, pp. 237-246, Springer, 2004.

[Pakdel07] H.-R. Pakdel and F. F. Samavati. Incremental subdivision for triangle meshes. In Journal of Computational Science Engineering, Vol. 3, No. 1, pp. 80-92, July 2007.

[Phan12] Anh-Cang Phan, R. Raffin, and M. Daniel. Mesh connection with RBF local interpolation and wavelet transform. In Pro. of the SoICT '12, pages 81-90, ACM, New York, NY, USA, 2012.

[Phan13a] Anh-Cang Phan, R. Raffin, and M. Daniel. Joining Meshes with a Local Approximation and a Wavelet transform. In Pro. of the WSCG 2013, pages 29-38, Plzen, Czech Republic, 2013.

[Phan13b] Anh-Cang PHAN. Crack removal and hole filling on composite subdivision meshes. Ph.D. thesis, Aix-Marseille University, France, 2013.

[Seeger01] S. Seeger, K. Hormann, G. Hausler, and G. Greiner. A Sub-Atomic Subdivision Approach. In Pro. of the VMV-2001, Stuttgart, Germany, November 21-23, pp. 77-86, Aka GmbH, 2001.

[Zorin96] D. Zorin, P. Schröder, and W. Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. Computer Graphics Proceedings (SIGGRAPH 96), Vol. 30, pp. 189-192, 1996.

[Zorin97] D. Zorin, P. Schröder, and W. Sweldens. Interactive Multiresolution Mesh Editing. Computer Graphics, Vol. 31, pp. 259-268, August 1997.