Joining Meshes with a Local Approximation and a Wavelet Transform

Anh-Cang PHAN
Aix-Marseille University
CNRS, LSIS UMR 7296
13009, Marseille, France
Anh-cang.Phan@univ-amu.fr

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

Constructing a smooth surface of a 3D object is an important problem in many graphics applications. Subdivision methods permit to pass easily from a discrete mesh to a continuous surface. A generic problem arising from subdividing two meshes initially connected along a common boundary is the occurrence of cracks or holes between them. Therefore, we propose a new method for joining two meshes with different resolutions using a tangent plane local approximation and a Lifted B-spline wavelet transform. This method generates a connecting mesh where continuity is controlled from one boundary to the other. This guarantees that the discrete continuity between these mesh areas is preserved and the connecting mesh can change gradually in resolution between coarse and fine areas.

Keywords

Mesh connection, smoothness, local approximation, B-spline wavelets.

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 on their surfaces because of having a difference in subdivision schemes or a difference in resolution levels between adjacent faces.

In order to overcome these drawbacks and particularly cracks, this paper presents a new technique creating a smooth connecting surface linking two meshes with different resolutions and with different subdivision schemes. We aim at constructing a high quality connecting mesh between two selected areas of a model so that we can preserve the "continuity" between these selected mesh areas to generate a smooth surface. It means that the curvatures must be "continuous" on the boundaries, which must be studied in terms of discrete curvatures, the latter being not presented here. The discrete boundary curves of the connecting mesh are created by a linear interpolation, a tangent plane local approximation, and a Lifted B-spline wavelet transform. They respect the local curvatures and change their point densities gradually from coarse to fine and conversely. Our contributions are as follows: 1) Provide a mesh joining solution by constructing a high quality connecting mesh (CM) between two meshes defined with subdivision surfaces, each mesh being at a different subdivision level; 2) Detect and remove cracks on the surface caused by a difference in resolution between neighboring faces; 3) Apply a local tangent plane reconstruction and a wavelet transform to position newly inserted vertices on the expected surface. This enables us to reconstruct smooth connecting surfaces from boundary vertices of the two meshes. Therefore, the continuity between the meshes will be preserved; 4) Allow filling holes, pasting meshes, and joining 3D objects to generate a smooth discrete surface with a natural shape and visually fair connectivity.

The remaining of the paper is organized as follows: Section 2 and 3 detail the previous and related works for mesh connection. Our algorithm is described in section 4 and details are given in section 5. We show and compare results of our algorithm in section 6. Finally, we draw the conclusion in section 7.

2 PREVIOUS WORKS

Subdivision surfaces have been used widely in fields of geometric modeling, computer graphics for shape design, animation, multi-resolution modeling or even as movie production, game engines and many other engineering applications. Today one can find a variety of subdivision schemes for geometric design and graphical applications such as Catmull-Clark [Cat98], Doo-Sabin [Doo78], Butterfly [Dyn90], Loop [Loo87], etc.

In addition, the theory of wavelets has been applied successfully in the areas of computer graphics as surface reconstruction, subdivision, multi-resolution analysis, etc. Many subdivision methods applying wavelet-based multiresolution analysis of an arbitrary surface were introduced in [Sto96, Lou97, Mal98, Zor00], etc. Moreover, subdivision wavelets with the lifting scheme have been developed in [Swe98, Ber04a]. This latter could be an interesting approach for CAD applications, like surface reconstruction.

Commonly, a subdivision of the entire input mesh is not necessary nor desired while one only needs to subdivide some areas of the mesh to add details on the object or obtain a smoother surface. This is important to reduce the number of faces of the mesh as well as the unnecessary subdivision levels, and save the refinement time or the storage space. Some research [Hus10, Hus11, Pak07] are related to the incremental subdivision method with Butterfly, Loop and Catmull-Clark schemes. The main goal of these methods is to generate a smooth surface by refining only some selected areas of a mesh and remove cracks by simple triangulation. However, this simple triangulation has some undesired side-effects. It changes the connectivity, the valence of odd vertices, and produces high valence vertices leading to long faces. This not only alters the limit subdivision surface, but also reduces its smoothness. It creates ripple effects on the subdivision surface. Moreover, these methods recommend adaptively refining the areas of interest using the same subdivision scheme. A surrounding part of the area outside the adaptively subdivided area is also refined in the most common case in order to reduce brutal normal deviation. The error is computed at each step and the subdivision is stopped once this error reaches a user specified threshold. Users need to compute and control over a number of subdivision steps.

On the other hand, such models of 3D objects are formed by assembled patches or meshes. Thus, available methods to join these meshes along boundary curves between them are of immediate practical interest. The main challenge in designing a mesh connection algorithm is to guarantee that the continuity between the meshes is preserved, while the resolution between them is changed gradually. In addition, the algorithm would be simple, and efficient. This problem has been studied extensively and there are several considerable research works relevant to connecting or joining meshes in [Bar95, Fu04]. These methods consist in connecting the meshes of a surface at the same resolution level which adopt various criteria to compute the planar shape from a 3D surface patch by minimizing their differences. They are computationally expensive and memory consuming. Recently, Di Jiang and F. Stewart [Jia07] introduced the joining algorithms based on the use (as a supplement to absolute error criteria) of normal-vector error criteria for the discrepancy between the surface patch and the associated mesh patch. They join given meshes together while maintaining a proxy for the normal-vector error, as well as the absolute error. However, these algorithms adjust the vertices of the input meshes in a way that constrains them to lie in a transfinite interpolation defined by Whitney extension. Therefore, they can produce large changes in the normal direction of triangles near the mesh boundary, even turn the triangles upside down by the joining process. Moreover, the algorithms do not mention the continuity and the progressive change in resolution between meshes after joining them together.

In [Phan12], a mesh connection method based on a RBF local interpolation and a wavelet transform has been introduced. In this paper we propose a more reliable method for mesh connection which allows us to construct a high quality connecting mesh and a continuous surface. The goal is to gain in both time of computation and surface quality. This new method is based on a tangent plane local approximation and a wavelet transform without solving any linear system, handling cracks, modifying the original boundaries of mesh areas of a model and the closest faces around the boundaries during the connecting process. We will compare both methods in section 6.

3 BACKGROUND

3.1 Wavelet-based Multiresolution representation of curves and surfaces

Wavelet is receiving a lot of attention due to the practical interest of 3D modeling in a large range of applications, such as Computer Graphics and CAD [Mal98, Ols08]. Wavelet tool can be used to derive a hierarchical multi-resolution representation of curves and surfaces [Lou97]; a smooth curve at different resolution levels [Ols08]; an overall shape edition of a curve while preserving its details [Suc09]; and a curve approximation [Kho00]. Wavelet analysis provides a set of tools to represent functions hierarchically [Sto96]. The coarse scaling function represents coarse curves or surfaces and encodes an approximation of the function, while the wavelet function represents the difference between coarse and fine curves or surfaces, and encodes the missing details. Many subdivision wavelets have been proposed in [Ber02, Ber04a, Sam04]. They allow a decomposition of curves and surfaces at different resolutions while maintaining geometric details. In addition, the combination of B-splines and wavelets leads to the idea of B-spline wavelets [Ber04a]. B-spline wavelets form a hierarchical basis for the space of B-spline curves and surfaces in which every object has a unique representation. Taking advantage of the lifting scheme [Swe96b], the Lifted B-spline wavelet [Ber02] is a fast computational tool for multiresolution analysis

of a given B-spline curve with a computational complexity linear in the number of control points. They allow representing B-spline curves at multiple resolution levels, editing curves, etc.

The Lifted B-spline wavelet transform includes the forward and the backward B-spline wavelet transform. From a fine curve at the resolution level J, C^J, the forward B-spline wavelet transform decomposes C^J into a coarser approximation of the curve, C^{J-1}, and detail (error) vectors. The detail vectors are a set of wavelet cœfficients containing the geometric differences with respect to the finer levels. The backward B-spline wavelet transform can be used to reconstruct fine curves from a coarse curve and detail vectors. Given a coarse curve at the resolution level $J-1$, C^{J-1}, the backward B-spline wavelet transform synthesizes C^{J-1} and the detail vectors into a finer curve, C^J.

In our approach, we apply the Lifted B-spline wavelet transform for multiresolution analysis of discrete boundary curves of a connecting mesh.

3.2 Tangent Plane Local Approximation for implicit surface reconstruction

Reconstruction of 3D surfaces from point samples is a well-studied problem in computer graphics. It allows fitting of scanned data, filling of surface holes, connecting and remeshing of 3D complex models. Implicit surface methods can directly reconstruct approximating surfaces from 3D scattered data set, such as moving least squares (MLS) [Lev03], implicit surface methods [Car01, Cas05], etc. We can classify the methods as either global or local approaches. Global fitting methods use the whole input data to compute implicit functions. Their disadvantage is that the computational complexity rapidly increases consequently to the data set size. Moreover, they present the well-known feats to discard local details (which can be an advantage or also a disadvantage). Therefore, it is difficult to use these methods directly to reconstruct implicit surfaces from large point sets consisting of more than several thousands of input points. Practical solutions on large point sets involve the local fitting methods to reconstruct the surfaces such as RBF local interpolations [Bra06, Cas05], adaptive RBF reduction and fast RBF methods [Car01]. Both methods require the construction of linear constraints on the control points for each interpolation point and thus the definition of a system of linear equations. The surface reconstruction can be obtained by solving this system. However, without adding off-surface constraints, the linear system may become trivial and we cannot solve it to specify implicit function values.

In order to extrapolate local frames (tangents, curvatures) between two meshes, we need a local approximation method on the points that will be projected [Hop92, Ale04, Lev03]. We choose a tangent plane based local method to approximate the expected surface using a set of local tangent planes computed at each sample point (see in Fig. 1b) because this method does not require such information as methods depending on off-surface constraints. This method enables us to reconstruct smooth implicit surfaces from a set of control points.

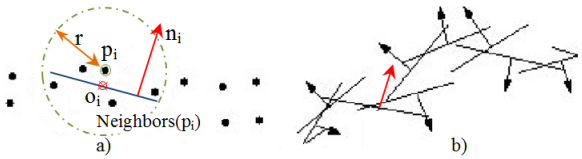

Figure 1: Nearest neighbors for tangent plane estimation.

It first considers subsets of nearest neighboring points to estimate local tangent planes as shown in Fig. 1a, and then defines an implicit function as a signed distance to the tangent plane for the data points. For example, Hoppe's method [Hop92] reconstructs a surface from a set of all unorganized points scattered on or near the surface using a contouring algorithm and the parameters k, ρ, δ defined by the user, where k is the number of nearest neighbors, ρ and δ are the thresholds of the density and noise. Thus, the considered data set is large and can contain noise. Our approach is inspired from this method. However, since our model is already a mesh model and not a set of sparse data points, we can completely determine nearest neighbors based on the connecting edges without using the parameters k, ρ, and δ. Additionally, the connecting mesh is constructed from boundary vertices of meshes and their neighbors. Thus, the cardinality of the data set is reduced. We are also able to compute the approximation error (RMS error).

Given a set of data points $P = \{p_i\} \in \mathbb{R}^3$ of a surface CM, we would like to find a signed distance function $f(p)$ from an arbitrary point $p \in \mathbb{R}^3$ to the surface CM. However, because CM is an unknown surface, the authors approximate the surface using a set of tangent planes computed at each data point as shown in Fig. 1b. The tangent plane and the signed distance function are computed as described in the following section.

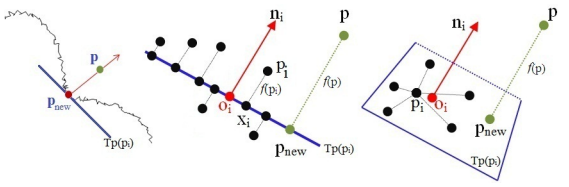

Figure 2: Estimation of a tangent plane and the projection of an arbitrary point onto it.

3.2.1 *Estimation of a tangent plane*

Let $Tp(p_i)$ be the tangent plane corresponding to point p_i and passing through a centroid point o_i. An arbitrary

point p is projected onto tangent plane $Tp(p_i)$ which has point o_i closest to point p. A point p_{new} is an orthogonal projection of point p onto $Tp(p_i)$ as illustrated in Fig. 2.

Tangent plane $Tp(p_i)$ is determined by passing through point o_i with unit surface normal n_i as follows:

- **Find local neighbors of each data point**: For each point $p_i \in \mathbb{R}^3$, we find a set of nearest neighbors of p_i denoted Neighbors(p_i).

- **Compute a centroid point on a tangent plane**: For each point $p_i \in \mathbb{R}^3$, we compute the centroid point o_i based on all nearest neighbors of p_i:

$$o_i = \frac{\sum_{p_j \in Neighbors(p_i)} p_j}{N} \quad (1)$$

  Where N is the number of the neighbors of p_i.

- **Estimate a normal vector of a tangent plane**: The principal component analysis (PCA) method is used to estimate normal n_i of $Tp(p_i)$. The point covariance matrix $CV_i \in \mathbb{R}^{3\times3}$ from the neighbors of p_i is first computed:

$$CV_i = \sum_{p_j \in Neighbors(p_i)} (p_j - o_i)^T (p_j - o_i) \quad (2)$$

We then compute eigenvalues $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3}$ of CV_i associated with unit eigenvectors $v_{i,1}, v_{i,2}, v_{i,3}$. Since normal n_i is the eigenvector corresponding to the smallest eigenvalue, we choose to be either $v_{i,3}$ or $-v_{i,3}$. The choice determines the tangent plane orientation [Hop92].

3.2.2 *Construction of a signed distance function*

Our goal is to find a signed distance function $f(p)$ from an arbitrary $p \in \mathbb{R}^3$ to CM. The function $f(p)$ is the distance between p and the closest point $p_{new} \in$ CM. Since CM is the unknown surface, we find a tangent plane $Tp(p_i)$ which is a local linear approximation of CM and passes through the centroid o_i closest to p. Therefore, the signed distance $f(p)$ to CM is represented as the signed distance $dist(p, p_{new})$ between p and its projection p_{new} onto $Tp(p_i)$ (see in Fig. 2). The function $f(p)$ satisfies the local approximation constraint defined by:

$$f(p) \quad = \quad dist(p, p_{new}) = (p - o_i).n_i \quad (3)$$

and p_{new} by:

$$p_{new} \quad = \quad p - (f(p).n_i) \quad (4)$$

3.2.3 *Evaluation of the error of the tangent plane based local approximation*

The local constraint (3) is too strict. Thus, if the data are noisy, the accuracy of the surface reconstruction is evaluated by the error of the approximation defined by equation (3). Since the approximation is based on fitting a plane to a set of local neighboring points, the minimize least squares (MLS) error [Ale03, Dor97, Lev03] is commonly used to evaluate the local approximation error. The MLS error evaluated at $p_i \in P$ is calculated as the sum of the squared distances from the local neighbors of point p_i to $Tp(p_i)$:

$$E_{MLS}(p_i) = \sum_{p_j \in Neighbors(p_i)} ((p_j - o_i).n_i)^2 \quad (5)$$

Additionally, we can use the root mean square (RMS) error [Sar11] to evaluate the local approximation error. The RMS error evaluated at p_i is:

$$E_{RMS}(p_i) = \sqrt{\frac{\sum_{p_j \in Neighbors(p_i)} ((p_j - o_i).n_i)^2}{N_k}} \quad (6)$$

4 METHOD OVERVIEW

4.1 Notation

In order to lighten notations, we decided not to use vectorial notations for all the notations or equations having vectorial relations. Moreover, we denote the position vector $\overrightarrow{Op}$ of a vertex p by p, where O is the frame origin. Each multiplication of a scalar value and a vector is understood as the vector components multiplied by the scalar value.

Let M_1 and M_2 be two meshes subdivided at different resolution levels, and p_i, q_k their vertices. An edge connecting p_i to q_k is denoted e_i or p_iq_k. An edge is usually shared by two faces. If it is shared by only one, it corresponds to a boundary edge and its end vertices are called boundary vertices. We need to construct a connecting mesh CM between meshes M_1 and M_2 so that the continuity between them can be preserved as illustrated in Fig. 3. First we will introduce the notations relevant to the algorithm:

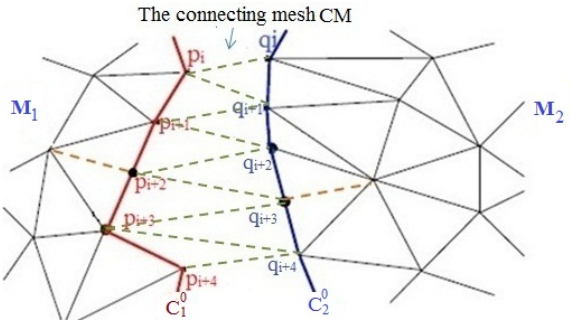

Figure 3: Topology representation of the algorithm.

- s: number of intermediate discrete curves (also called the number of newly created boundary curves) of CM created between M_1 and M_2 (see Fig. 4). It is a user parameter computed based on the distance between two original boundaries of M_1 and M_2 and it controls the resolution of CM.

- j: order number of the decomposition step to create intermediate discrete curves, also called the level. Since two boundary curves between M_1 and M_2 will be created at each level j, j is in $[1, \frac{s}{2}]$.

- C_1^j and C_2^j: two boundary curves of CM at level j. C_1^0 and C_2^0 are the two original boundary curves of meshes M_1 and M_2.

- $N(C_1^j)$: number of vertices of boundary curve C_1^j at level j. It corresponds to the density of vertices of boundary curve C_1^j.

- p_i^j, q_i^j: vertices i on boundary curves C_1^j and C_2^j. ($p_i^0 = p_i$ and $q_i^0 = q_i$)

- L_1^j: list of the boundary vertex pairs (p_i^{j-1}, q_k^{j-1}).

- L_2^j: list of the boundary vertex pairs (q_k^{j-1}, p_i^{j-1}).

4.2 CM2D-TPW algorithm

The idea is to create new boundary curves C_1^j and C_2^j between M_1 and M_2 based on the previously created boundary curves C_1^{j-1} and C_2^{j-1} using the tangent plane local approximation and the Lifted B-spline wavelet transform. After that, we connect each new boundary curve C_1^j to C_1^{j-1}, and C_2^j to C_2^{j-1}. C_1^j is created in a direction from C_1^{j-1} to C_2^{j-1} and conversely for C_2^j. Therefore, ***this algorithm is called the algorithm of connecting mesh in two directions based on the tangent plane local approximation and the Wavelet transform*** (CM2D-TPW). The algorithm consists of the following main steps detailed in the next sections:

- **Step 1**. Boundary detection: read the input geometry model of two meshes M_1 and M_2. Detect and mark boundary vertices of the two boundaries C_1^0 and C_2^0 in M_1 and M_2.

- **Step 2**. Boundary vertex pairs and boundary curve creation: for each level j, we pair the boundary vertices of C_1^{j-1} and C_2^{j-1} based on the distance between them. If this distance is too narrow (smaller than a certain threshold), we go to Step 3 to connect the boundary curve pair (C_1^{j-1}, C_2^{j-1}). In contrast, we create two new boundary curves C_1^j, C_2^j. The boundary curve creation first produces vertices of two new boundary curves from the paired boundary vertices by a linear interpolation, and then projects them onto the expected surface CM using a tangent plane local approximation. It finally refines or coarsens these new boundary curves by applying wavelet transforms and vertex insertion and deletion operations.

- **Step 3**. Boundary curve connection: perform a boundary triangulation for each boundary curve pair (C_1^{j-1}, C_1^j) and (C_2^{j-1}, C_2^j).

- **Step 4**. Repeat steps 2 and 3 until both mesh areas M_1 and M_2 have been connected or patched by all newly created triangles.

5 MESH CONNECTION

5.1 Boundary vertex pairs

In order to create boundary curves between two meshes M_1 and M_2 by interpolating previously created boundary curves, we pair the boundary vertices $p_i^{j-1} \in C_1^{j-1}$ with $q_k^{j-1} \in C_2^{j-1}$ and vice versa based on the distances between them. Since the densities of vertices of both boundary curves are different, we need to create two lists of the closest boundary vertex pairs L_1^j and L_2^j.

Assume that j is the current level, for each boundary vertex $p_i^{j-1} \in C_1^{j-1}$, we search for and insert into L_1^j the corresponding paired vertex $q_k^{j-1} \in C_2^{j-1}$ such that: $\left(\forall q \in C_2^{j-1}, dist(p_i^{j-1}, q_k^{j-1}) \leq dist(p_i^{j-1}, q)\right)$, where the notation $dist(p_i^{j-1}, q_k^{j-1}) = ||p_i^{j-1} - q_k^{j-1}||$ is the Euclidean distance between p_i^{j-1} and q_k^{j-1}. The list of boundary vertex pairs L_2^j is created similarly.

5.2 Boundary curve creation

The basic idea is to create two new boundary curves C_1^j and C_2^j from the paired vertices at each level j. Paired vertices are obtained by the shortest distances between vertices of each boundary. New boundary vertices $p_i^j \in C_1^j$ and $q_k^j \in C_2^j$ are created by boundary vertex pairs $(p_i^{j-1}, q_k^{j-1}) \in L_1^j$ and $(q_k^{j-1}, p_i^{j-1}) \in L_2^j$ respectively. A new boundary curve C_1^j is created in a direction from C_1^{j-1} to C_2^{j-1} and a new boundary curve C_2^j is created in a direction from C_2^{j-1} to C_1^{j-1} as shown in Fig. 4.

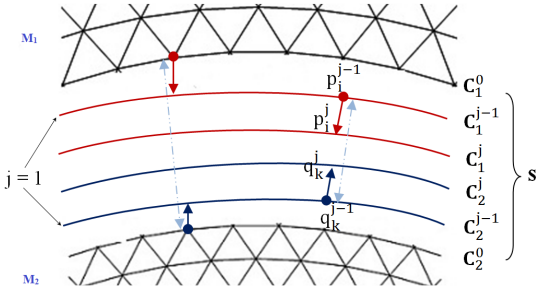

Figure 4: Boundary curves created in two directions.

We assume $N(C_1^0) \leq N(C_2^0)$ and let the density of vertices of the two boundary curves C_1^j and C_2^j be two functions $N(C_1^j)$ and $N(C_2^j)$ defined by:

$$N(C_1^j) = N(C_1^0) + \frac{j}{s+1}[N(C_2^0) - N(C_1^0)]$$
$$N(C_2^j) = N(C_2^0) - \frac{j}{s+1}[N(C_2^0) - N(C_1^0)] \quad (7)$$

The boundary curves are created in three phases.

5.2.1 *Phase 1: Create vertices of two new boundary curves by a linear interpolation.*

- Create vertices of the discrete boundary curve C_1^j in a direction from C_1^{j-1} to C_2^{j-1} (see Fig. 4): for each boundary vertex pair $(p_i^{j-1}, q_k^{j-1}) \in L_1^j$, we apply the linear interpolation equation (8) to create new boundary vertices $p_i^j \in C_1^j$.

$$p_i^j = p_i^{j-1} + \frac{j}{s+1}(q_k^{j-1} - p_i^{j-1}) \quad (8)$$

  Where i are the subscripts of boundary vertices of C_1^j, $1 \leq i \leq N(C_1^{j-1})$, and k are the subscripts of boundary vertices of C_2^{j-1}, $1 \leq k \leq N(C_2^{j-1})$.

- In the same way, we create the new boundary vertices $q_k^j \in C_2^j$ by (9).

$$q_k^j = q_k^{j-1} + \frac{j}{s+1}(p_i^{j-1} - q_k^{j-1}) \quad (9)$$

  Where k are the subscripts of boundary vertices of C_2^j, $1 \leq k \leq N(C_2^{j-1})$, and i are the subscripts of boundary vertices of C_1^{j-1}, $1 \leq i \leq N(C_1^{j-1})$.

Equations (8) and (9) have been chosen with a local linear expansion classically used in marching methods. We recursively compute (8) and (9) based on vertices of curves C_1^{j-1} and C_2^{j-1} but not C_1^0 and C_2^0. In addition, since C_1^{j-1} and C_2^{j-1} are then refined or coarsened by wavelet transforms, their resolutions are increased or reduced respectively.

5.2.2 *Phase 2: Project created boundary vertices onto surface CM using a tangent plane based local approximation.*

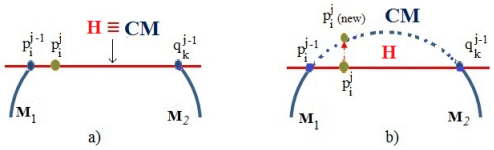

Figure 5: The connecting mesh CM created with and without using a local tangent plane approximation.

The goal of phase 2 is to improve the resulting surface CM after applying phase 1. Since new boundary vertices p_i^j and q_k^j of curves C_1^j and C_2^j are created by a linear interpolation in phase 1, they can lie on a flat surface H producing a flat surface CM as shown in Fig. 5a.

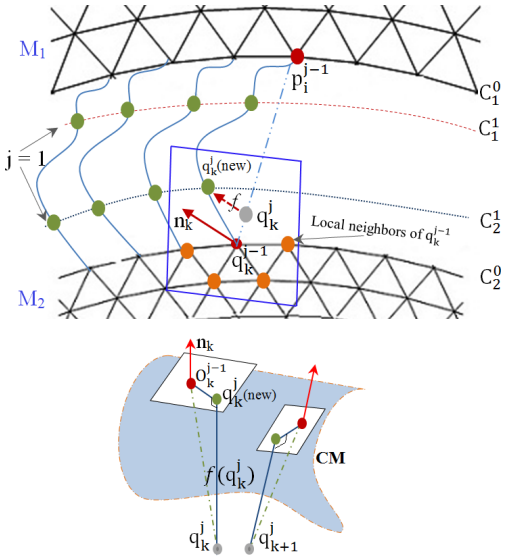

Figure 6: The vertices projected onto the surface CM.

When CM is a complex curved surface, these newly created boundary vertices may not be on the expected surface CM because we did not consider the curvature information in phase 1. As a result, the boundary curves are produced without respect of local curvatures. Therefore, the generated connecting mesh will not respect the expected continuity between the meshes. To solve this problem, we construct the connecting surface CM by a tangent plane local approximation. We first apply phase 1 (linear interpolation) to create new boundary vertices. We then project these vertices onto tangent planes as shown in Fig. 6. Projecting the created vertices $q_k^j \in C_2^j$ onto surface CM is performed as follows: First, for each vertex q_k^j, we find the closest vertex $q_k^{j-1} \in C_2^{j-1}$ and its local neighbors $Neighbors(q_k^{j-1})$ which have edges connected to q_k^{j-1} to determine the local control vertices of q_k^j (see Fig. 7). Next, we estimate the local tangent plane $Tp(q_k^{j-1})$ of surface CM. The plane $\mathrm{Tp}(q_k^{j-1})$ passes through the centroid vertex o_k^{j-1} (using (1)) with the unit normal vector n_k^{j-1} (using (2)). We construct the local signed

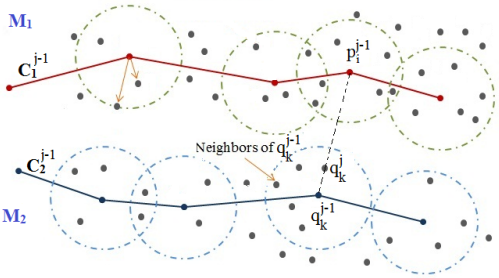

Figure 7: Selection of the local neighbors to construct a local tangent plane and a signed distance function.

distance function $f(q_k^j)$ using (3) whose value is referred to as the signed projection distance between q_k^j and $Tp(q_k^{j-1})$. Then, we use (4) to project them onto surface CM with the projection distances $f(q_k^j)$ along surface normals (see in Fig. 6). Finally, we update vertices q_k^j by their projections. We perform the same operation for vertices $p_i^j \in C_1^j$.

When the two boundary curves C_1^{j-1} and C_2^{j-1} are close together, we take the neighboring vertices from both curves to define the set of local neighboring vertices (or control vertices). For each vertex q_k^j, we keep the two closest vertices $p_i^{j-1} \in C_1^{j-1}$ and $q_k^{j-1} \in C_2^{j-1}$ with their neighbors. It permits us to take into account the local curvatures on both sides.

5.2.3 *Phase 3: Refine or coarsen the new boundary curves with wavelet transforms.*

Since the densities of vertices of C_1^j and C_2^j are now $N(C_1^{j-1})$ and $N(C_2^{j-1})$, we need to increase and reduce their densities to be $N(C_1^j)$ and $N(C_2^j)$. Taking advantage of the Lifted B-spline wavelet transform presented in section 3.1, we apply this transform for the multiresolution analysis of the boundary curves C_1^j and C_2^j to refine the curve C_1^j, coarsen the curve C_2^j. Then, we perform the vertex insertion or deletion operations to control the densities of vertices of C_1^j and C_2^j. Thus, the created boundary curves C_1^j, C_2^j, and the associated connecting mesh CM are changed gradually in resolution between both mesh areas.

5.3 Boundary curve connection

After creating two boundary curves C_1^j and C_2^j, we connect each new boundary curve to each previously created boundary curve, C_1^{j-1} to C_1^j and C_2^{j-1} to C_2^j, based on the method of stitching the matching borders proposed by G. Barequet et al. [Bar95]. The basic idea is the implementation of the boundary triangulation based on the distance between boundary vertices. We consider the distance between three adjacent vertices of two boundaries before connecting them together to create a triangular face (see Fig. 8). This process terminates when we reach the last vertices of both boundaries.

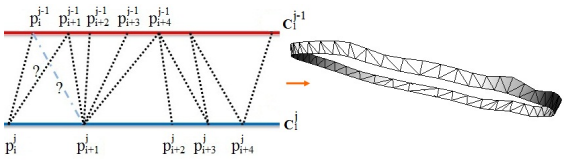

Figure 8: Figure shows a boundary curve connection.

6 RESULTS AND COMPARISONS

In this section we give some examples with experimental results to illustrate our algorithm. We also compare CM2D-TPW method with CM2D-RBFW method which is a mesh connection method based on a RBF local interpolation and a wavelet transform. CM2D-RBFW method is built on the work by Anh-Cang Phan et al. [Phan12] in which the connecting mesh is constructed by adding triangle strips to each boundary up to the time they are close enough to be linked. This method needs to solve a linear system that additionally requires off-surface constraints to specify implicit function values. It creates off-surface points by projecting on-surface points along the surface normals with a signed projection distance d. These points are used to construct RBF support and are mandatory to obtain valid solutions. Both methods have been implemented in Matlab to make possible their comparisons. All results were obtained on a PC 2.27GHz CPU Corei5 with 3GB Ram.

In Fig. 9, CM2D-TPW algorithm produces a connecting mesh of the Tiger model which consists of two meshes defined by subdivision surfaces (Loop and Butterfly), each mesh being at a different level of subdivision. From two original coarse meshes of this model, we first apply a Loop subdivision at level 2 and a Butterfly subdivision at level 1 to obtain two meshes M_1 and M_2 of different resolutions. We then implement our algorithm to connect them together. To understand the quality of the result, we plot the image of the connecting mesh and its zoom. Based on a set of tests, $s = 4$ is an empirical good value to apply CM2D-TPW algorithm for two subdivided meshes of the Tiger model as shown in Fig. 9. From the resulting mesh, we can see that our new method can generate a smooth connecting mesh with the progressive change in resolution between meshes because it is possible to constrain the surface to have specified tangent planes at subsets of control vertices to be interpolated.

To draw comparisons, we have chosen examples of a sphere to have accurate evaluations of the error and runtime. We have developed a test on four density-based discretizations of the sphere, since analytical description permits to compute the exact surface and relative errors. The numbers of vertices are 240, 3840, 61440, 983040 and the numbers of vertices of the removed strips are 66, 720, 5982, 70743, respectively. In this way, both meshes M_1 and M_2 have the same density of vertices for a given discretization level, and the process to obtain the compatible number of vertices of CM is the same for both methods. Hence, we define the errors E_{dist} and E_{max} as follows: $E_{dist} = \sqrt{\frac{\sum_{p_i \in CM}(R - dist(c,p_i))^2}{N}}$; $E_{max} = sup(R - d_i)$, $1 \leq i \leq N$; where: $d_i = dist(c, p_i)$ is the Euclidean distance between c and vertices p_i of CM; R, c are the radius and center of the sphere, respectively (in our tests, $c = (0,0,0)$ and $R = 10$). N is the number of vertices of CM.

Figs. 10-11 and Table 1 summarize the results. First, we apply CM2D-TPW algorithm for the discretiza-

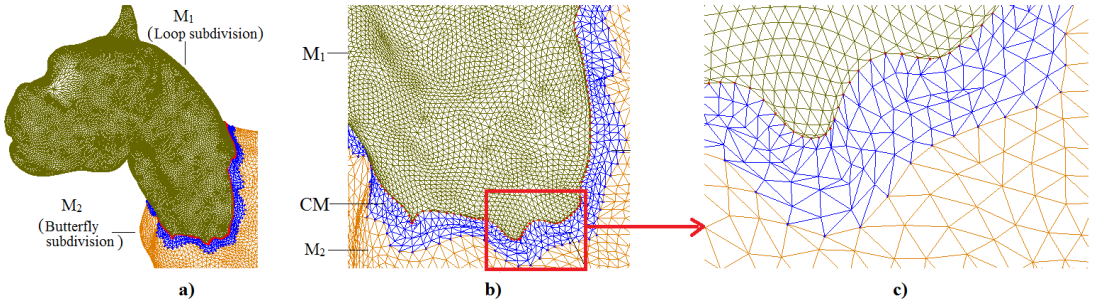

Figure 9: The Tiger model with CM2D-TPW algorithm: a) The connecting mesh CM produced with $s = 4$; b) Zoom of CM; c) Zoom of one of the interesting parts of CM.

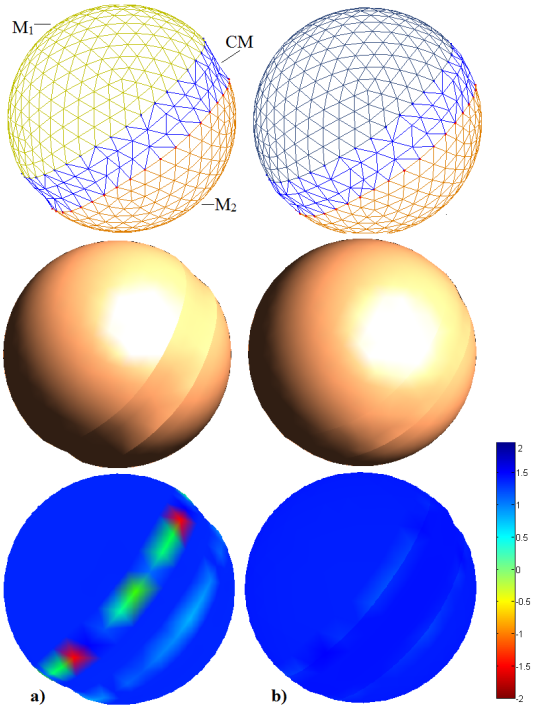

Figure 10: Model of Sphere 2: a) CM is produced by CM2D-RBFW method with $s = 2$ and $d = 0.004$; b) CM is produced by CM2D-TPW method with $s = 2$.

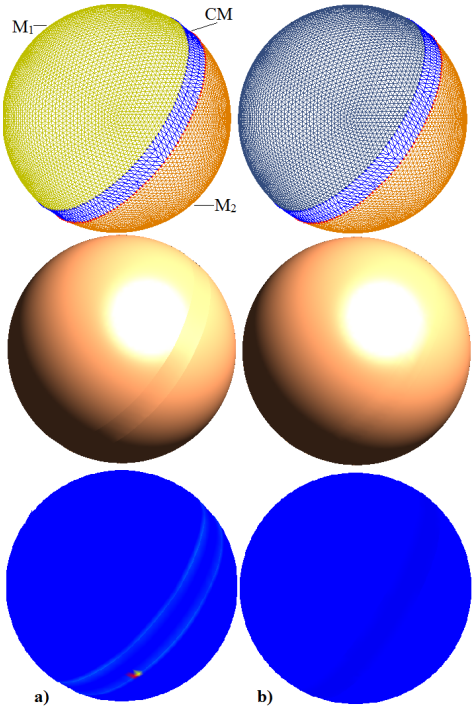

Figure 11: Model of Sphere 3: a) CM is produced by CM2D-RBFW method with $s = 2$ and $d = 0.004$; b) CM is produced by CM2D-TPW method with $s = 2$.

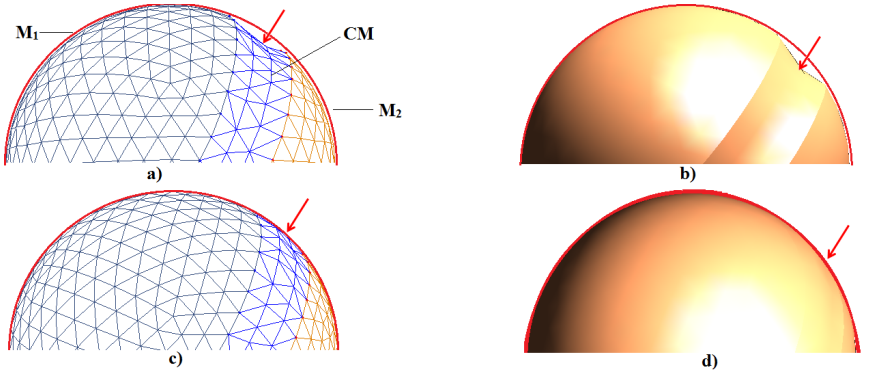

Figure 12: The surface continuity of Sphere preserved after applying CM2D-TPW method: a)-b) CM produced by linear interpolation; c)-d) CM produced by CM2D-TPW method.

tion models of the sphere with $s = 2$ as illustrated in Figs. 10b and 11b. Obviously, our method keeps the continuity between the meshes of the sphere model without destroying the Gaussian curvature and altering the original meshes because the newly inserted vertices are on the expected surface by a local approximation with tangent plane fitting (phase 2). As a result, it gives the high quality connecting meshes and smooth

Model	CM		E_{dist}		E_{max}		Runtime (secs)	
	V	F	RBFW	TPW	RBFW	TPW	RBFW	TPW
Sphere 1	38	40	0.9265	0.7581	2.3663	2.0479	0.4061	0.3159
Sphere 2	159	240	0.2022	0.0646	0.4861	0.2216	0.4592	0.3762
Sphere 3	639	960	0.034	0.0153	0.0578	0.0293	1.3859	0.917
Sphere 4	2641	3963	0.0634	0.0329	0.1453	0.0764	14.4955	9.0221

Table 1: Comparison of errors and runtimes of CM2D-RBFW and CM2D-TPW algorithms for spheres with center $c = (0,0,0)$, and radius $R = 10$; the numbers of vertices and faces of CM are in columns V and F.

surfaces. Then, we also use CM2D-RBFW method on these models (see Fig. 10a and Fig. 11a).

Figs. 12a-b show the connecting mesh and surface CM produced with linear interpolation by applying phase 1 and 3 of CM2D-TPW algorithm without phase 2. As a result, CM is hyperbolic and the surface continuity is not guaranteed. While Figs.12c-d present CM after applying all phases of the algorithm. Obviously, CM2D-TPW method generates a smooth surface with natural shape where continuity between meshes is preserved.

According to these experimental results, we can see that CM2D-TPW method gives better results compared to CM2D-RBFW method since errors to the real surface are smaller and Gaussian curvatures are much better respected. In addition, a well-known drawback of RBF based reconstruction methods is the difficulty to provide abrupt changes in a small distance (see [Luo08]). It requires much more estimation which includes estimating the linear constraints on the control vertices as well as the off-surface constraints to construct and solve a linear system for each interpolated vertex. Therefore, the time of computation will be inevitably longer or the memory requirements may exceed the capacity of the computer. As a consequence, the runtime of this algorithm is rapidly increasing when the vertex numbers of the models increase as illustrated in Table 1. We have applied the algorithm to various 3D objects with complex shapes. The runtime increases quadratically. Moreover, the most critical disadvantage is that it is very important for the user to make a decision on the choice of the basis functions and the user parameter values, i.e d-the signed distance and h-the shape parameter. This leads to the fact that the user chooses them by a rather costly trial and performs their numerical experiments over and over again until they end up with a satisfactory result consisting of the well-chosen values and an interpolated surface with a natural shape. In order to overcome these disadvantages, we have proposed a more reliable method to join two meshes. It produces surfaces of good approximation, computationally more efficient and occupied less memory compared to the C2MD-RBFW method. The memory storage will not become a problem when the numbers of vertices of the given meshes are large in practical applications. The computing time of this algorithm is smaller than CM2D-RBFW algorithm as shown in Table 1 while we have not taken into account the execution time of experiments for values d, h in CM2D-RBFW method.

7 CONCLUSION

We have introduced a new simple and efficient mesh connection method which produces a high quality connecting mesh and finally a smooth surface. The mesh is changed gradually in resolution from one area to the other one. CM2D-TPW method joins two meshes with different resolutions while maintaining the surface continuity and not destroying local curvatures. It keeps the original boundaries of the meshes and the closest faces around these boundaries while connecting them. The advantages of this method are: 1) It is simple, efficient, and local; 2) It generates smooth connecting surfaces; 3) There is no need to solve a system of linear equations. As a consequence, our algorithm is then numerically stable. These features make CM2D-TPW method become feasible and suitable for designing, joining and modeling 3D objects with complex shapes. Thus, it can be extended to applications related to pasting meshes, and filling holes.

ACKNOWLEDGMENTS

We would like to thank all reviewers for their valuable comments which help us to improve the paper.

8 REFERENCES

[Ale03] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. IEEE Trans. on Visualization and Comp. Graph.,9(1):3-15,2003.

[Ale04] M. Alexa, S. Rusinkiewicz, M. Alexa, and A. Adamson. On normals and projection operators for surfaces defined by point sets. In Eurograph. Symp. on Point-Based Graph., p. 149-155, 2004.

[Ber02] M. Bertram. Biorthogonal wavelets for subdivision volumes. In Proc. of the SMA'02, p. 72-82, New York, USA, 2002. ACM.

[Ber04a] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Generalized B-spline subdivision-surface wavelets for geometry compression. IEEE, 10:326-338, 2004.

[Bra06] John Branch, Flavio Prieto, and Pierre Boulanger. Automatic hole-filling of triangular

meshes using local Radial Basis Function. In 3DPVT, pages 727-734, 2006.

[Bar95] Gill Barequet and Micha Sharir. Filling gaps in the boundary of a polyhedron. Comp. Aided Geometric Design, 12(2):207-229, 1995.

[Car01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In Proc. of the SIGGRAPH'01, p.67-76, ACM, NY, USA, 2001.

[Cat98] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes, p.183-188. ACM, NY, USA, 1998.

[Cas05] G. Casciola, D. Lazzaro, L. B. Monte-fusco, and S. Morigi. Fast surface reconstruction and hole filling using radial basis functions, numerical algorithms, 2005.

[Dyn90] N. Dyn, D. Levin, and J. A. Gregory. A Butterfly subdivision scheme for surface interpolation with tension control. ACM Transactions on Graphics, 9:160-169, 1990.

[Doo78] D. Doo and M. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. CAD, 10(6):356-360, 1978.

[Dor97] Chitra Dorai, John Weng, and Anil K. Jain. Optimal registration of object views using range data. IEEE Trans. Pattern Anal. Mach. Intell., 19(10):1131-1138, 1997.

[Fu04] Hongbo Fu, Chiew-Lan Tai, and Hongxin Zhang. Topology free cut and paste editing over meshes. In GMP, pages 173-184, 2004.

[Hus10] 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 VRCAI'10, p. 207-210, USA, 2010. ACM.

[Hop92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. SIGGRAPH Comput. Graph., 26:71-78, 1992.

[Jia07] D. Jiang and N. F. Stewart. Reliable joining of surfaces for combined mesh-surface models. In Pro. of 21st ECMS, pages 297-303, 2007.

[Kho00] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In Proc. of the Comp. Graph. Conf., SIGGRAPH, pages 271-278, New York, 2000. ACM.

[Lev03] D. Levin. Mesh-independent surface interpolation. In Geometric Modeling for Scientific Visualization, volume 2, pages 37-49, 2003.

[Loo87] Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, 1987.

[Lou97] M. Lounsbery, T. DeRose, and J. D. Warren. Multiresolution analysis for surfaces of arbitrary topological type. ACM, p.34-73, 1997.

[Luo08] W. Luo, M. C. Taylor, and S. R. Parker. A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from England and Wales. Inter. Journal of Climatology, 28(7):947-959, 2008.

[Mal98] S. G. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 1998.

[Hus11] N. A. Husain, M. S. M. Rahim, and A. Bade. Iterative process to improve simple adaptive subdivision surfaces method with Butterfly scheme. In World Academy of Science, Engineering and Tech., pages 622-626, 2011.

[Ols08] L. J. Olsen and F. F. Samavati. A discrete approach to multiresolution curves and surfaces. In Proc. of the 2008 International Conf. on Computational Sciences and Its App., p. 468-477, Washington, DC, USA, 2008. IEEE.

[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, New York, NY, USA, 2012. ACM.

[Pak07] H. Pakdel and F. F. Samavati. Incremental subdivision for triangle meshes. In Journal of Computational Science Engineering, vol 3, No. 1, pages 80-92, 2007.

[Sam04] F. F. Samavati. Local filters of B-spline wavelets. In Proc. of International Workshop on Biometric Tech. 2004, p.105-110, 2004.

[Sar11] Scott A. Sarra. Radial basis function approximation methods with extended precision floating point arithmetic. Engineering Analysis with Boundary Elements, 35(1):68-76, 2011.

[Sto96] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann Pub., 1996.

[Suc09] N. Suciati and K. Harada. Wavelets-based multiresolution surface as framework for editing the global and local shapes. Inter. Journal of Comp. Science and Net. Security, 5:77-83, 2009.

[Swe96b] W. Sweldens and P. Schröder. Building your own wavelets at home. In Waveletes in Computer Graphics, p. 15-87, 1996.

[Swe98] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. SIAM J. Math. Anal., 29:511-546, 1998.

[Zor00] Denis Zorin and P. Schröder. Subdivision for Modeling and Animation. Technical report, SIGGRAPH 2000, Course Notes, 2000.