# Subdivision Surface Fitting to a Dense Mesh Using Ridges and Umbilics

**Xinhui Ma**[1], **Simeon Keates**[1], **Yong Jiang**[2], **Jiří Kosinka**[3]
[1]School of Engineering, Computing and Applied Mathematics, Abertay University, Dundee, UK
[2]Nanjing University of Information Science and Technology, China
[3]Computer Laboratory, University of Cambridge, UK

**Abstract:** Fitting a sparse surface to approximate vast dense data is of interest for many applications: reverse engineering, recognition and compression, etc. The present work provides an approach to fit a Loop subdivision surface to a dense triangular mesh of arbitrary topology, whilst preserving and aligning the original features. The natural ridge-joined connectivity of umbilics and ridge-crossings is used as the connectivity of the control mesh for subdivision, so that the edges follow salient features on the surface. Further more, the chosen features and connectivity characterise the overall shape of the original mesh, since ridges capture extreme principal curvatures and ridges start and end at umbilics. A metric of Hausdorff distance including curvature vectors is proposed and implemented in a distance transform algorithm to construct the connectivity. Ridge-colour matching is introduced as a criterion for edge flipping to improve feature alignment. Several examples are provided to demonstrate the feature-preserving capability of the proposed approach.

**Keywords:** Subdivision Surface Fitting, Dense Mesh, Feature Alignment, Ridges, Umbilics, Hausdorff Distance, Principal Curvature Vector.

## 1 Introduction

Fitting a sparse and smooth surface that approximates dense data is a desirable goal in applications such as compression, recognition, and reverse engineering. The simplified surface models or data are obtained for convenient downstream processing, e.g. surface design, animation and manufacturing. Commonly, there are two types of surfaces which are used to fit dense data: NURBS (Non-Uniform Rational B-Spline) and subdivision surfaces. Ma et al. (2004); Lavoué et al. (2007); Panozzo et al. (2011) fit these surfaces to dense meshes. Sometimes one is interested in fitting pure geometric surfaces. For example, the first author's previous work Ma and Cripps (2011) fits generalised Cornu spirals to dense surface points to preserve the original shape. Most existing methods fit B-spline surfaces to 3D points with simple topological type. In the case of complex topology, it is an extremely difficult task to handle continuity conditions among neighbouring surfaces. Subdivision surfaces can easily represent arbitrary topology in a compact form and achieve the same accuracy as NURBS.

Subdivision surfaces have recently become of great interest due to their high potential in rendering and shape design. A principal achievement in shape design is that subdivision surfaces are now compatible with the industry standard NURBS thanks to the work of Cashman et al. (2009) generalising the schemes of Catmull and Clark (1978) and Doo and Sabin (1978) to general degrees. The scheme of Loop (1987) is a popular subdivision scheme for triangulations. NieBner et al. (2012) achieve unprecedented quality and speed in rendering by fast evaluation of subdivision surfaces on GPUs (Graphical Processing Units). Subdivision surfaces ideally suit the paradigm of hardware tessellation on GPUs. Only the vertices of a coarse model need to be animated; the GPU can amplify the coarse geometry on-the-fly with very little memory bandwidth to produce a dense tessellation of the surface. Panozzo et al. (2011) predict that subdivision-based modelling is expected to replace polygonal modelling even for real time applications.

This paper proposes a new approach to fitting subdivision surfaces to dense meshes. Most existing algorithms aim at obtaining a mesh with a good quality of accuracy and regularity. However, we focus on preserving the features, since the convergence is improved by the alignment of the connectivity with salient features. The coarse control mesh is constructed from ridges, umbilics, and ridge-crossings of the original dense mesh. Ridges are geometrically and perceptually salient surface features and are used for shape recognition as said in Ohtake et al. (2004). Thirion (1996) states that these features can characterise the overall shape of the original data and are successfully used in image processing for registration. They also constitute a natural connectivity, since ridges start and end at umbilics and cross at ridge-crossings as stated in Porteous (1994). The natural connectivity matches the requirement of connectivity for a control mesh. Hence, we use these features to construct a coarse control mesh of subdivision surfaces to fit the overall shape of the original dense mesh.

The flow of the proposed approach is: First, ridges, umbilics, and ridge-crossings are extracted from the input dense mesh. Secondly, the significant features are filtered. Second, a connectivity of the feature points is constructed and its alignment with ridges is improved. Finally, a coarse control mesh is constructed and a subdivision surface is generated to fit the original dense mesh.

The main contributions of the approach are:

- The natural ridge-joined connectivity of umbilics and ridge-crossings is used as the connectivity of control mesh for subdivision.
- A metric of Hausdorff distance including curvature vectors is proposed and implemented in a distance transform algorithm for connectivity construction and feature alignment. An accurate arc-length estimation formula is derived to define the metric.
- A criterion of ridge-colour matching is introduced for edge flipping to improve feature alignment.

The proposed algorithm will work on noisy data as well thanks to the noise filtering capability of the adopted feature filtering approaches Ohtake et al. (2004); Cazals and Pouget (2005), as discussed in Subsection 3.4. Furthermore, since the proposed method aims to preserve the salient features using as few points as possible, detailed features and noisy data will not have much effect on the fitting quality.

The paper is structured as follows. In Section 2, previous related work is reviewed. In Section 3, we discuss why we choose ridges and umbilics, and how to extract and filter such features. Construction and optimisation of connectivity are introduced in Section 4. Section 5 derives the metric of Hausdorff distance including curvature vectors. The calculation of control mesh is presented in Section 6. In Section 7, we demonstrate and evaluate the proposed method on several examples. Finally, we give concluding remarks.

## 2 Previous Work

Subdivision surface fitting has been investigated by many researchers. Most existing methods can be classified as follows: mesh simplification, parametrisation, face clustering, feature extraction and others. Usually, a two-step process is adopted. Firstly, a coarse connectivity is constructed. The second step involves optimising the geometry and regularity of the connectivity. In this review, we will focus on methods that preserve features and alignment.

Hoppe et al. (1994) simplify a dense triangle mesh and construct a control mesh for Loop subdivision by optimising vertex positions. Lee et al. (2000) extend this approach using displacement mapping to approximate projectability during simplification. Suzuki et al. (1999) inversely refine an interactively defined initial control mesh. The control mesh is constructed using an iterative local approximation. Kanai (2001) simplifies the dense mesh using a modified QEM method. Ma et al. (2004) distinguish different types of vertices and edges during the simplification process to preserve sharp features. Panozzo et al. (2011) use fitmaps to achieve adaptive mesh simplification. Usually, following simplification, optimisation steps are used to optimise the geometry or connectivity using distance minimisation, collapsing, splitting or swapping edges of the control polyhedron. Approaches based on mesh simplification are easy to implement by using an established mesh simplification algorithm. However, they may need extensive computing time due to the large amount of data and geometric optimisation. It is also not clear how these methods align to salient features.

Parametrisation methods fit parameter equations for a selected domain and construct the coarse control mesh from these domains. Ma and Zhao (2002) interactively define a topological model for parametrisation and fit Catmull-Clark surfaces using linear least squares. Boier-Martin et al. (2004) introduce a method for computing parametrisations of triangulated manifolds over quadrilateral domains. The creation of the base domain is performed through a combination of clustering methods which control the shape and flatness of clusters. Lai et al. (2006) use feature sensitive parametrisation to allocate more parameter space to highly curved feature regions, so that more control points will be provided in those regions. Li et al. (2006) consider global parametrisation and quad domain remeshing to fit T-splines. Parametrisation methods can produce quality meshes but may need interactive definition of the topology model. They are suitable for simple topology cases, but can find complex topology objects difficult to handle.

Clustering algorithms are proposed by Cohen-Steiner et al. (2004), and driven by alignment of face normals, resulting in coarse but irregular meshes. Marinov and Kobbelt (2005) use a similar face-merge method to compute a coarse polygonal mesh, taking into account normal projectability. They compute a normal displacement map from the resulting control mesh. Such methods consider the geometric properties of surface normals, but without considering curvature which characterises the geometric features. Reverse subdivision is a multiresolution method to generate a sparse surface to fit the original dense mesh. Sadeghi and Samavati (2011) construct smooth reverse of the Loop and Catmull-Clark subdivision surfaces, balancing the two optimisation goals of finding a good approximation of the fine points and producing coarse points with minimum energy.

Feature-based methods construct a coarse control mesh by aligning the extracted salient features. Marinov and Kobbelt (2005) integrate curvature lines for meshing of isotropic regions. The QuadCover method proposed by Käberer et al. (2007) uses principal curvature frames to compute global parametrisations. Ling et al. (2008) fit sharp features by iteratively solving a nonlinear least squares problem based on the squared distances from the input mesh vertices to the fitted subdivision surface. Lavoué et al. (2007); Lavoué and Dupont (2009) extract features by segmentation and using semi-sharp subdivision surfaces to fit sharp features. Bommes et al. (2009) introduce cross field symmetry to formulate a mixed-integer problem for quadrangulation. Such methods achieve very good results in terms of regularity of the mesh, but they need interactive definition of feature and topological constraints. These methods have the advantage of feature preservation and potential for complex topology cases. However, how to choose appropriate features and align to the features are important issues.

In the above literature, most previous works aim at obtaining meshes with good surface shape and regularity. Few address feature preservation and alignment when constructing a control mesh. The typically used features have limitations in characterising salient edges and constituting topology. Basically, these features are based on surface normals, curvature lines and special subdivision rules. Surface normal based features lack curvature and topological information. Lines of curvature may not align to salient edges, e.g. the curvature lines of a twisted rounded cube do not align with the twelve salient edges of the cube.

We consider umbilics and ridges as the salient feature points and lines to characterise the shape of the input mesh. Ridge lines have maximum principal curvature along the corresponding principal directions. Umbilics are points with the same surface normal curvatures in all tangent directions. It is important for the connectivity of a reconstructed mesh to follow the salient features of an object. The main reason is given by D'Azevedo (2000): the convergence is improved by such alignment, for both remeshing and fitting. We discuss ridges and umbilics in detail in the following section.

# 3 Choosing Features for Control Mesh

We choose to place the images of control points at umbilics and ridge-crossings, because we want the edges to follow features in the surface. It is highly plausible that ridges of a surface give a good characterisation of the features, and ridges join umbilics. This section discusses why ridges and umbilics are chosen as the features to construct a control mesh. We briefly review the properties of these features; see Hallinan et al. (1999) and Cazals and Pouget (005a) for details. These properties illustrate that ridges and umbilics can characterise the overall shape and constitute a natural connectivity.

## 3.1 The Monge Form of a Surface

Given the principal directions $x$ and $y$ and the normal vector at a non-umbilic point, a surface can locally be represented using the Monge form:

$$z = \tfrac{1}{2}(k_1x^2 + k_2y^2) + \tfrac{1}{6}C_M(x, y) + \tfrac{1}{24}\textstyle\sum_{i=0}^{4} c_i x^{4-i} y^i + h.o.t,$$

where $k_1$ and $k_2$ are the maximum and minimum principal curvatures, respectively, and

$$C_M(x, y) = b_0x^3 + 3b_1x^2y + 3b_2xy^2 + b_3y^3$$

is referred to as the Monge cubic. The Taylor expansions of principal curvatures along their principal directions are

$$k_1(x) = k_1 + b_0x + \frac{P_1}{2(k_1 - k_2)}x^2 + h.o.t,$$

$$k_2(y) = k_2 + b_3y + \frac{P_2}{2(k_2 - k_1)}y^2 + h.o.t,$$

where

$$P_1 = 3b_1^2 + (k_1 - k_2)(c_0 - 3k_1^3),$$

$$P_2 = 3b_2^2 + (k_2 - k_1)(c_4 - 3k_2^3).$$

The Monge form provides a concise mathematical expression for the definition, properties, and extraction of ridges and umbilics.

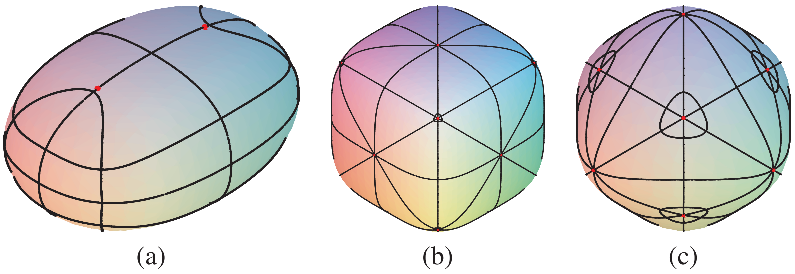


Figure 1: Ridges (black) and umbilics (red) on implicit polynomial surfaces: (a) a soap shape; (b) a rounded cube; (c) a rounded octahedron.

### 3.2 Ridges

One reason for choosing ridges to construct the control mesh is that they are geometrically and perceptually salient surface features and are used for shape recognition Ohtake et al. (2004). Thirion (1996) states that ridges and umbilicus are a decomposition of the curved surfaces as natural as the decomposition of the polyhedral surfaces into faces, edges and vertices. These features have been successfully used in 3D image processing registration. This is due to the fact that ridges are points of extrema of principal curvatures along their curvature lines. A ridge point is defined by:

**Definition 1** *A non-umbilical point ($k_1 \neq k_2$) is called a ridge point if the extremality coefficient $b_0 = dk_1/dx$ vanishes or $b_3 = dk_2/dy$ vanishes.*

Ridges can be classified into several types. To simplify notation, we assign a colour to each type. A ridge is called

- maximum elliptic (red) ridge if $b_0 = 0$ and $P_1 < 0$;
- maximum hyperbolic (green) ridge if $b_0 = 0$ and $P_1 > 0$;
- minimum elliptic (blue) ridge if $b_3 = 0$ and $P_2 < 0$;
- minimum hyperbolic (yellow) ridge if $b_3 = 0$ and $P_2 < 0$;
- maximum crest ridge if $b_0 = 0$ and $P_1 < 0$ and $|k_1| > |k_2|$;
- minimum crest ridge if $b_3 = 0$ and $P_2 < 0$ and $|k_2| > |k_1|$.

Crest ridges can be seen as the visually most salient curves on a surface. For example, Bogaevski et al. (2003) extract the ridges on some implicit surfaces. The overall shapes of these surfaces are characterised by their salient ridges, as shown in Fig. 1.

We remark that the maximum/minimum and colour assignments depend on the orientation of the normal of the surface. However, this ambiguity plays no role in our algorithms.

Ridges on a triangular mesh can be extracted using Cazals and Pouget (2005)'s marching method due to its robustness. Firstly, the Monge forms of the original vertices are estimated using jet fitting introduced by Cazals (2008), getting the estimated value of $b_0$ and $b_3$ for all vertices. For boundary points, their curvatures can be estimated by our circle fitting method presented in Ma and Cripps (2007). Then, a marching method is used to extract the ridges, as shown in Fig. 2. For the two vertices of an edge, if the signs of the two $b_0$ are different, the edge is marked as a ridge-passing edge. Ridges can be extracted by connecting the proportional internal points of all the adjacent ridge-passing edges.

Note that there are alternative methods to extract ridges on a mesh: Hildebrandt et al. (2005) use discrete differential geometry and externality smoothing method, Kim and Kim (2006) employ modified moving least square method, Clémençon et al. (2008) propose integrating a first-order ordinary differential equation, and Yoshizawa et al. (2008) use focal surface based finite difference scheme. These methods can also be used to extract ridges from meshes. However, the adopted method is accurate due to the high order of jet fitting and robustness of the marching method. It can extract both feature lines and feature points, i.e. ridges and umbilics, and has been implemented into Computational Geometry Algorithms Library (CGAL (2013)) to give convenient application.

The definition and examples of ridges demonstrate that ridges are geometrically salient features and natural decomposition of the curved surfaces. These features have been successfully used in shape recognition and registration. Hence, a control mesh based on ridges can align with the salient features and preserve the overall shape of the original dense mesh.

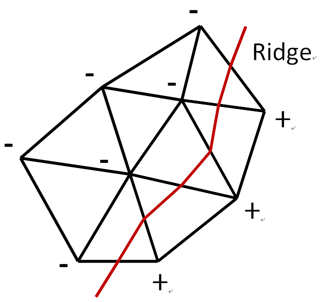


Figure 2: A marching method for ridge extraction.

## 3.3 Umbilics

Another reason for choosing umbilics and ridges is that their topological relationship provides a natural connectivity for control meshes. Ridges start and end at umbilics, points where the two principal curvatures are equal in value, i.e. $k_1 = k_2$. Umbilics can be classified in terms of their configuration of ridges, determined by the Monge cubic.

**Definition 2** *Generic umbilics are of two types:*

- *Elliptic or 3-ridge umbilic. The Monge cubic has three different real factor lines and three ridge lines meet at the umbilic.*
- *Hyperbolic or 1-ridge umbilic. The Monge cubic has only one real factor line and one ridge passes through the umbilic.*

Additionally, ridges cross transversally at generic umbilics and change from a minimum of $k_1$ to a maximum of $k_2$.

To summarise the topological relationship of ridges and umbilics, they have the following generic properties:

- Ridges start and end at umbilics, or form closed loops;
- A generic umbilic is either a 1-ridge umbilic (hyperbolic) or a 3-ridge umbilic (elliptic);
- Ridges of the same type (colour) do not cross, they only meet at 3-ridge umbilics. Two ridges of different types (colours) may cross at so-called ridge-crossings;
- A ridge contains an even number of turning points at which the ridge changes from elliptic to hyperbolic.

Umbilics on a triangular mesh can be extracted using the minimum value of $|k_1 - k_2|$ as given in Cazals and Pouget (2005). An example of extracted ridges and umbilics on a Catmull-Clark subdivision surface of a car model is shown in Fig. 3. It illustrates that elliptic ridges are salient features. However, hyperbolic ridges also play an important role. Note that in this paper the features and their connectivity are not obtained by tracing, i.e. the technique studied in Ray and Sokolov (2013). Since we aim to get a regular and feature aligned connectivity for subdivision, a tracing technique is not convenient to achieve that goal. The feature points and lines in Fig. 3 are extracted using a jet fitting and marching method, Cazals and Pouget (2005); Cazals (2008), which has already been implemented in CGAL (2013). The feature points will be connected using a modified distance transform algorithm to construct a triangulation connectivity. Then the alignment with salient feature lines will be improved using edge flipping and ridge colour matching. The detailed process will be given in Section 4.

We illustrated that ridges, umbilics, and ridges-crossings constitute a natural connectivity which can be used to construct a coarse control mesh for subdivision. Hence, ridges and umbilics are chosen as the features for subdivision surface fitting.

## 3.4 Feature Filtering

For real scanned objects, the configuration of umbilics and ridges may not match that of a smooth generic surface, especially for meshes with sharp features or featuring degenerate regions. For example, for a plane or a cylinder whose principal curvatures are constant, all points are ridge points. In such cases, sharp ridges or prominent ridges should be filtered.

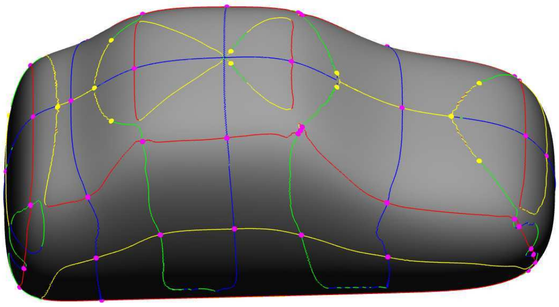


Figure 3: Extracted ridges (red and blue are elliptic, yellow and green are hyperbolic), umbilics (yellow), and ridge-crossings (purple) on a Catmull-Clark subdivision surface of a car model.

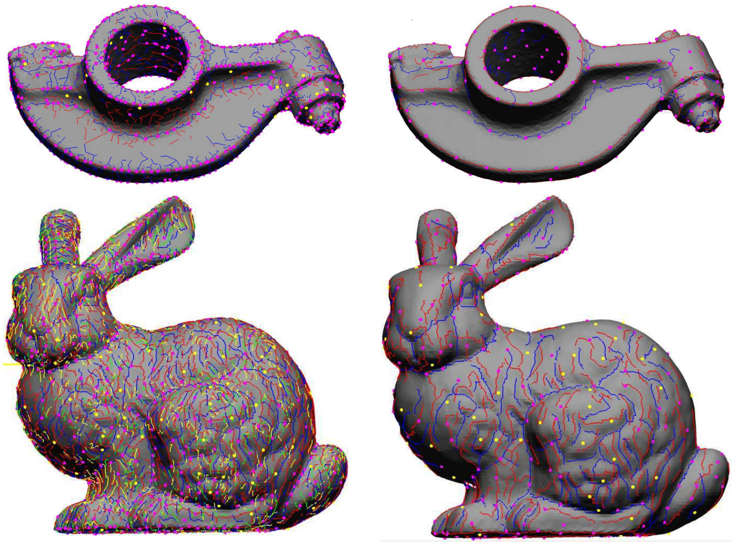


Figure 4: Filtering of features. Left: all extracted features which are too many to construct connectivity. Right: filtered features provide a sparse characterisation of the overall shape of the original meshes.

We use Ohtake et al. (2004)'s metric of strength to filter the salient ridges. The strength is defined by the integral of $k_1$ along a ridge line or a line connecting two feature points, and use the trapezoid approximation of the integral. Note that strength is a scale-independent threshold specified by the user. Similarly, strength is also used to filter ridge-crossings. Umbilics are filtered by threshold of curvature value.

Fig. 4 shows the feature filtering for the rocker-arm and Stanford bunny models. Note that the thresholds need to be interactively tuned by the users. Different models will have different thresholds. For the bunny model, the strength threshold is set to $1.0$ and the curvature threshold $0.1$. For the rock-arm model, the strength threshold is set to $10.0$ and curvature threshold to $1.0$. It is illustrated that the filtered features characterise the overall shape of the original meshes with noise being filtered out.

For moderate noisy input meshes, the noisy features will be removed by the filtering process. The extracted initial features may include small features caused by noise in the mesh. The feature filtering process uses the metric of strength defined by the integral of curvatures along a feature line. Noisy features are often short and have low strength measure, so they will be filtered out.

## 4 Constructing Connectivity

Because we are looking for a triangulation, there is a straightforward solution. We determine the Voronoi tessellation of the feature point set (umbilics and ridge crossings) and then take the dual. A distance transform algorithm starting from the feature points can give the Voronoi regions on the original mesh corresponding to the feature points as stated in Guan and Ma (1998). The dual of this is a candidate connectivity. A distance transform algorithm is driven by a suitable metric. In our approach, we introduce a metric based on Hausdorff distance that also takes into account curvature vectors. Details of how to derive the metric and its properties are given in Section 5.

**Algorithm 1:** 3D surface triangulation of feature points using distance transform

```
input  : A dense mesh with vertices $V$ and a set of feature points $P \subset V$
output: A coarse triangulation of the feature points $P$
// initialization
foreach $v \in V$ do
    $n[v] = NULL$;                    // set $v$'s nearest feature point to NULL
    $d[v] \leftarrow \infty$; // set the distance $d$ from $v$ to it's nearest feature point to
    infinity
foreach $p \in P$ do
    $n[p] \leftarrow p$;                    // set $p$'s nearest feature point to itself
    $d[p] \leftarrow 0$;  // set the distance $d$ from $p$ to it's nearest feature point to
    zero
    push $p$ to the priority queue $Q$;
// find the nearest feature point for each vertex
while $Q$ is not empty do
    $p \leftarrow Q$ pops out the top vertex with minimum $d$ value;
    foreach $v \in Adj[p]$ do                 // loop each adjacent vertex of $p$
        if $d[p] + length(p, v) < d[v]$ then
            $d[v] = d[p] + length(p, v)$ ;
            $n[v] = n[p]$;      // update $v$'s nearest feature point to $p$'s nearest
            feature point
            push $v$ to the priority queue $Q$;
// make coarse triangulation of the feature points
foreach triangle $t$ in the dense mesh do
    get $t$'s three vertices $v1$, $v2$ and $v3$;
    if $n[v1] \neq n[v2]$ and $n[v2] \neq n[v3]$ and $n[v3] \neq n[v1]$ then
        make a coarse triangle using the feature points $n[v1], n[v2]$ and $n[v3]$;
```

Since the generated connectivity may not always align with the filtered ridges, a colour matching and edge flipping method is introduced to improve ridge alignment.

## 4.1 Voronoi Tessellation and Triangulation

The distance transform maps each point into its shortest distance to points of interest as discussed in Fabbri et al. (2008). In our case, the interested points are the extracted and filtered umbilics and ridge-crossings. It is a useful tool in computational geometry and has many application in generating Voronoi diagrams. We use the distance transform algorithm, also known as the grass-fire algorithm, to find the nearest feature points for each vertex on the original 3D dense mesh. A priority queue is implemented in the algorithm to achieve linear complexity. It is a very efficient way to find the Voronoi tessellation of the feature points.

One way to think about the algorithm is to first imagine that foreground regions on the 3D dense mesh are made of some uniform slow burning inflammable material. Then consider simultaneously starting a fire at all feature points and letting the fire burn its way uniformly around the mesh. Then the source of the fire first reaching a vertex is the nearest feature point of the vertex. If we set fires simultaneously at the feature points on a sphere, these fires will meet each other at some edges which constitute a Voronoi diagram on a 3D surface. All the vertices in a Voronoi cell have nearest distances to the feature point in the cell. Algorithm 1 gives the implementation of distance transform to construct the triangulation connectivity of feature points on a dense mesh.

The dual of a Voronoi tessellation is normally a triangulation. The only case where it does not occur is when four or more regions meet in a single place. However, this is neatly resolved by the fact that each triangle can only have three source seeds. Thus a place where four (or more) regions meet is represented by two (or more) triangles. Thanks to geodesic distance metric implemented in the distance transform algorithm, the 3D Voronoi tessellation can be mapped to a 2D Voronoi tessellation. There will be no nested case, e.g. region A is enclosed by two regions B and C.

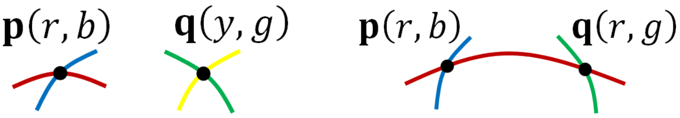


Figure 5: Left: No matching of colours, $\mathbf{p}$ and $\mathbf{q}$ do not connect. Right: Colours match, $\mathbf{p}$ and $\mathbf{q}$ are connected.

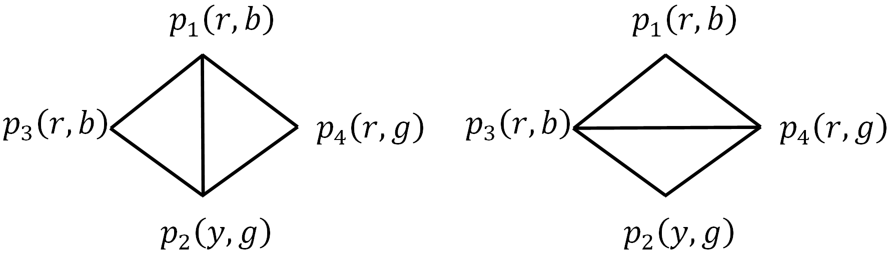


Figure 6: Edge flipping. Left: vertices $p_1, p_2$ have no colour matching. Right: vertices $p_3, p_4$ have colour matching.

What we have to do is to create an output triangle wherever there is an original facet whose three vertices are labelled with three different feature points as sources. This is created by looping over all the facets of the input mesh. The output triangles make up an initial connectivity of all the feature points. This connectivity is further improved to align better with features.

### 4.2 Feature Alignment by Colour Matching and Edge Flipping

An initial connectivity has been constructed using a modified distance transform algorithm to connect the feature points. In order to improve the alignment of the connectivity with significant ridges, edge flipping based on ridge colour matching is introduced.

Each feature point has one or more ridges passing through. These ridges may have different types represented by different colours. Each feature point is associated with colours of the ridges passing through it. For example, if two neighbouring feature points have one colour matching there may be a ridge connecting them; if they don't have any matching colours there should be no ridge connecting them; see Fig. 5.

In the initial connectivity, we consider one edge $p_1p_2$ shared by two triangles, as show in Fig. 6. If the two vertices $p_1, p_2$ of the edge have no colour matching and the other two vertices $p_3, p_4$ have matching colours, then we swap the edge $p_1p_2$ to $p_3p_4$.

There is also a priority of the colours $red > blue > yellow > green$, which can be used to choose the priority when two connections intersect each other as it respects the expected saliency of a ridge. This situation can occur in estimated features from noisy data.

A colour coding method is introduced to represent the priority of colour. A four-bit binary number is used to represent the four ridge colours. Red, blue, yellow and green are represented by $1000, 0100, 0010$ and $0001$ respectively. The code of a feature point is represented by the sum of codes of the ridges passing through the point. For example, the ridge colours at point $\mathbf{p}$ of Fig. 5 left are red and blue, so the colour code of $\mathbf{p}$ is $1000 + 0100 = 1100$.

The colour matching of a pair of feature points is measured by the value of 'logic and' of the colour codes of the two points. The value is named 'The matching value'. Zero means no colour matching. Positive means some colours match. The higher the matching value, the higher probability the two points should be connected. For example, in the left of Fig. 5, the matching value of points $\mathbf{p}$ and $\mathbf{q}$ is '1100 logic and $0011 = 0000$'. There is no colour match, so the two points should not be connected. In the right of Fig. 5, the matching value of points $\mathbf{p}$ and $\mathbf{q}$ is '1100 logic and $1001 = 1000$'. The matching colour is red, so the two point should be connected.

Matching value is used as the criterion in edge flipping to improve the alignment with salient ridges. Algorithm 1 gives the implementation of edge flipping and ridge colour matching for the initial connectivity to improve feature alignment.

We remark that there are alternative curvature-driven criteria for edge flipping. Li et al. (2006); Lai et al. (2010) align with another criterion, curvature directions, for connectivity optimisation. Here, we are interested in alignment with ridges as discussed above.

**Algorithm 2:** Improve feature alignment by edge flipping and ridge colour matching

```
input  : An initial triangulation connectivity and associated ridge colour codes $C[p]$ for each vertex $p$
output: An improved connectivity with better ridge alignment
foreach edge $e$ in the initial triangulation connectivity do
    $p_1, p_2 \leftarrow$ the two vertices of edge $e$;
    $t_1, t_2 \leftarrow$ the two triangles associated with edge $e$;
    $p_3, p_4 \leftarrow$ the two vertices opposite to edge $e$ in the two triangles $t_1, t_2$ respectively;
    if $(C[p_1]$ logic and $C[p_2]) < (C[p_3]$ logic and $C[p_4])$ then
        delete two old triangles $p_1, p_2, p_3$ and $p_1, p_2, p_4$;
        add two new triangles $p_1, p_3, p_4$ and $p_2, p_3, p_4$;
```

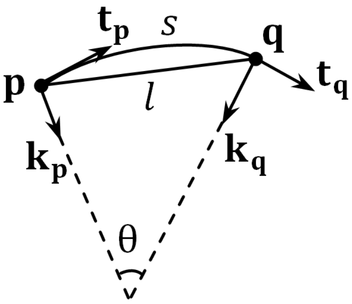


Figure 7: Arc-length estimation

# 5 Hausdorff Distance including Curvature Vectors

Several metrics can be used in the grass-fire algorithm to construct the connectivity, e.g. mesh distance, unwrapped Euclidean distance, and geodesic distance on the mesh. Since the vertices of the mesh lie on some smooth surface, these distances are not accurate to approximate the surface geodesic distance and may bring self-intersections to the connectivity. An accurate estimation of geodesic distance is needed for the grass-fire algorithm.

We introduce a modified Hausdorff distance including curvature vectors to estimate the geodesic distance between two vertices. The introduced metric works not only on meshes but also on point clouds, since it is independent of connectivity. The geodesic distance can be directly calculated from the estimated principal curvature vectors at the vertices. In the present work, the differential geometric properties have already been estimated during the stage of feature extraction.

We note that there are alternative approaches to geodesic distance estimation on meshes, Surazhsky et al. (2005); Bose et al. (2011); Aleksandrov et al. (2010). They can give bounded approximation of geodesic distance in some order of efficiency, but they are graph-based and only apply on polyhedral surfaces. The introduced geodesic distance estimation is suitable for both polyhedral surfaces and point clouds.

In our estimation, it is assumed that the two vertices in question lie on an unknown smooth surface and are close enough so that they can be expressed by each other using a single Taylor series. For example, two adjacent feature points match this assumption. We start by estimating the arc-length between two points on an unknown smooth planar curve.

## 5.1 Arc-length Estimation

As shown in Fig. 7, given two close points $\mathbf{p}$ and $\mathbf{q}$ and their estimated curvature vectors $\mathbf{k_p}$ and $\mathbf{k_q}$ on a smooth planar curve, we derive a formula to estimate the arc-length between the two points. It is assumed that $\mathbf{p}$ and $\mathbf{q}$ are close enough so that they can be expressed using Taylor series

$$\mathbf{q} = \mathbf{p} + s\mathbf{t_p} + \frac{s^2}{2}\mathbf{k_p} + O(s^3), \tag{1}$$

$$\mathbf{p} = \mathbf{q} - s\mathbf{t_q} + \frac{s^2}{2}\mathbf{k_q} + O(s^3), \tag{2}$$

from which it follows that

$$2(\mathbf{q} - \mathbf{p}) = s(\mathbf{t_p} + \mathbf{t_q}) + \frac{s^2}{2}(\mathbf{k_p} - \mathbf{k_q}) + O(s^3). \tag{3}$$

Approximating the squared arc-length $s^2$ by the squared chord length $l^2$ gives

$$2(\mathbf{q}-\mathbf{p}) \approx s(\mathbf{t_p}+\mathbf{t_q}) + \frac{l^2}{2}(\mathbf{k_p}-\mathbf{k_q}) + O(s^3)$$

and thus

$$s(\mathbf{t_p}+\mathbf{t_q}) \approx 2(\mathbf{q}-\mathbf{p}) - \frac{l^2}{2}(\mathbf{k_p}-\mathbf{k_q}) + O(s^3).$$

Taking norms on both sides yields

$$\|s(\mathbf{t_p}+\mathbf{t_q})\| \approx \|2(\mathbf{q}-\mathbf{p}) - \frac{l^2}{2}(\mathbf{k_p}-\mathbf{k_q}) + O(s^3)\|. \tag{4}$$

Let $\theta$ be the angle between unit tangent vectors $\mathbf{t_p}$ and $\mathbf{t_q}$. Then

$$\|\mathbf{t_p}+\mathbf{t_q}\| = 2\cos\frac{\theta}{2}$$

and substituting into (4) gives

$$s \approx \frac{1}{\cos\frac{\theta}{2}}\|(\mathbf{q}+\frac{l^2}{4}\mathbf{k_q}) - (\mathbf{p}+\frac{l^2}{4}\mathbf{k_p})\|. \tag{5}$$

This formula gives second geometric order estimation of arc-length between two adjacent points, given their estimated curvature vectors. It implies that the arc-length can be approximated as: firstly, the two points are transferred along their curvature vector with a displacement of $\frac{l^2}{4}\|\mathbf{k}\|$, which is related to their second order properties (curvature vector); then the distance between their new positions is scaled by $1/\cos\frac{\theta}{2}$, which is related to their first order properties (tangent vector). The resulting distance is the approximated arc-length between the two points.

## 5.2 Geodesic Distance Estimation

In a similar way, we can estimate the shortest arc-length between two adjacent points on a surface mesh. Theoretically, normal curvature vectors corresponding to the tangent direction along the two points are used to estimate geodesic distance. However, estimation of normal curvature vectors requires estimation of principal directions which are unstable near umbilics. We derive a simple formula to estimate geodesic distance directly using principal curvature vectors, following (5).

Note that principal curvature vectors are different from principal directions. Principal directions lie in tangent plane while principal curvature vectors are along the surface normal, although both of them can use the magnitude of principal curvatures. Following the formula 5, we use principal curvature vectors to estimate the geodesic distance between two adjacent points on a surface mesh.

At a point on a surface, there are two principal curvature vectors which can be treated as a set of two elements. Hausdorff distance is the appropriate tool to measure the distance between two sets. The symmetric Hausdorff distance, $D_H(A,B)$, between two sets $A$ and $B$ is defined by:

$$D_H(A,B) = \max(d_H(A,B), d_H(B,A)),$$

where

$$d_H(A,B) = \max_{a\in A}\min_{b\in B} d(a,b)$$

and $d(,)$ represents a more familiar metric, e.g. the rectilinear distance $L_1$, the Euclidean distance $L_2$, or the maximum $L_\infty$ metric. The Hausdorff distance measures the discrepancy between any element in one set and all elements in the other set. This is used to measure the distance between pairs of principal curvature vectors.

Inspired by formula (5), setting $\lambda = \frac{l^2}{4\cos\frac{\theta}{2}}$ and using the notation of Fig. 8, we introduce the modified Hausdorff distance

$$\begin{aligned} D_H = \frac{1}{\cos\frac{\theta}{2}} \max\{ &\max(\min(d_1,d_2), \min(d_3,d_4)), \\ &\max(\min(d_1,d_3), \min(d_2,d_4))\}. \end{aligned} \tag{6}$$

Substituting $d_i$ by formula (5), the geodesic distance between two points on a surface mesh can be estimated by

$$\begin{aligned} s \approx \frac{1}{\cos\frac{\theta}{2}} \max\{ &\max_{i=1,2}\min_{j=1,2} \|(\mathbf{q}+\frac{l^2}{4}\mathbf{k}_{\mathbf{q}i}) - (\mathbf{p}+\frac{l^2}{4}\mathbf{k}_{\mathbf{p}j})\|, \\ &\max_{j=1,2}\min_{i=1,2} \|(\mathbf{q}+\frac{l^2}{4}\mathbf{k}_{\mathbf{q}i}) - (\mathbf{p}+\frac{l^2}{4}\mathbf{k}_{\mathbf{p}j})\|\}. \end{aligned} \tag{7}$$

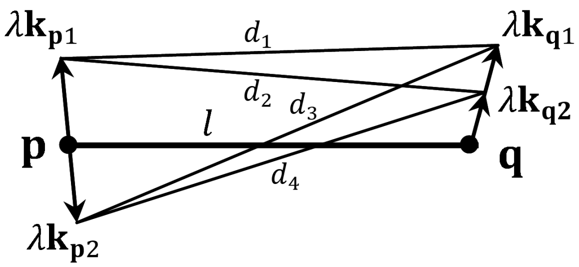


Figure 8: Hausdorff distance including principal curvature vectors.

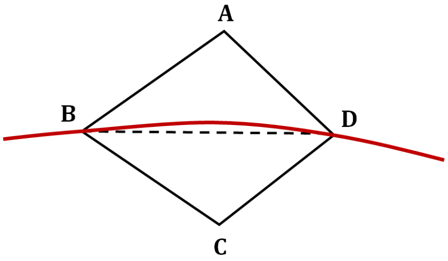


Figure 9: Ridge alignment using the proposed metric.

This estimation can achieve second order geometric accuracy. It directly uses principal curvature vectors to weave a complex estimation of normal curvature vectors near umbilics. The implementation of Hausdorff distance can solve the problem of which curvature vector at the second point corresponds to which at the first. This metric is used in the grass-fire algorithm.

## 5.3 Properties of the Modified Hausdorff Distance

The introduced metric has the potential to improve the alignment of the connectivity with ridge features, since it takes into account curvature information. Note that geodesic distance can be calculated by some practical anisotropic geodesy, e.g. Campen et al. (2013) using short-term vector Dijkstra algorithm. Similar results can be achieved using isotropic remeshing Lai et al. (2007), which re-samples the points by minimising an energy function of geodesic distance and connects these points using feature-sensitive local parametrisation. However, our approach directly adopts umbilics and ridge-crossings as the re-sampled feature points. We reconstruct the connectivity using distance transform based on the proposed Hausdorff distance metric to achieve feature alignment.

With $\triangle\mathbf{k_{pq}} = \mathbf{k_q} - \mathbf{k_p}$, equation (5) can be rewritten as

$$s \approx \frac{1}{\cos\frac{\theta}{2}}\|\mathbf{q} - \mathbf{p} + \frac{l^2}{4}\triangle\mathbf{k_{pq}}\|.$$

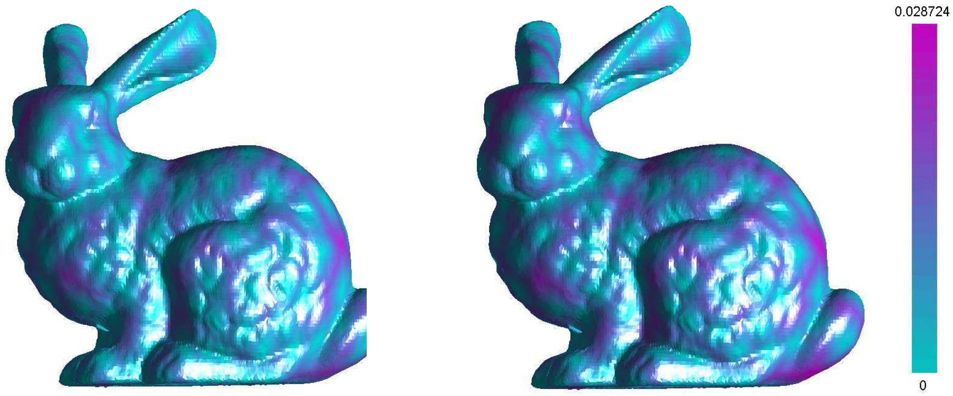
We can observe that the arc-length is determined by the term $\triangle\mathbf{k_{pq}}$ for a given chord length $l = \|\mathbf{q} - \mathbf{p}\|$, since $\theta$ can be calculated from curvature vectors. It means that, for two given points, the estimated arc-length is determined by the variation of curvature vectors not only by curvature values.

Consequently, two points on a ridge are more likely to be connected than two points on either side of the ridge if they have approximately the same chord length. The reason is that points on a ridge have lower variation of principal curvatures, hence smaller estimated geodesic distance.

As shown in Fig. 9, vertices $\mathbf{B}, \mathbf{D}$ lie on a ridge and $\mathbf{A}, \mathbf{C}$ lie on either side of the ridge with the same chord length $\|\mathbf{BD}\| = \|\mathbf{AC}\|$. Whether the connection should be $\mathbf{BD}$ or $\mathbf{AC}$ is determined by the variation of curvature vectors $\triangle\mathbf{k}_{i,\mathbf{BD}}$, $\triangle\mathbf{k}_{i,\mathbf{AC}}$, $i = 1, 2$. Since $\mathbf{B}, \mathbf{D}$ are ridge points with zero variation of principal curvature, it is more likely that $\triangle\mathbf{k}_{i,\mathbf{BD}} < \triangle\mathbf{k}_{i,\mathbf{AC}}$ and $s_{\mathbf{BD}} < s_{\mathbf{AC}}$. Thus, $\mathbf{B}, \mathbf{D}$ are more likely to be connected than $\mathbf{A}, \mathbf{C}$ and the connection is aligned with the ridge.

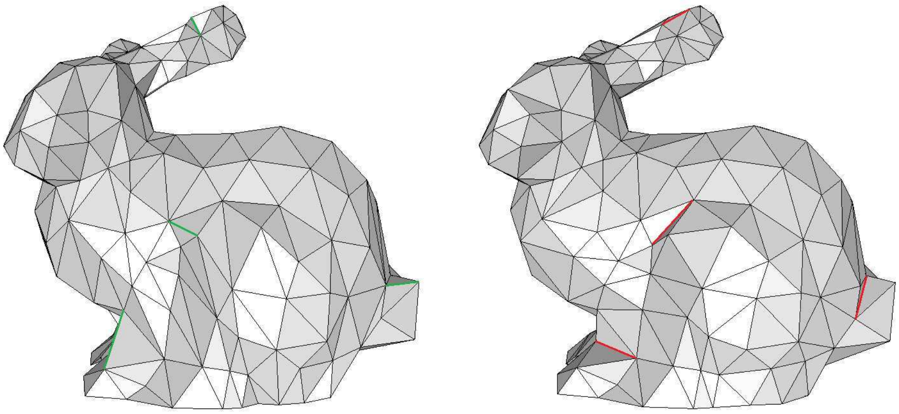
We note that the alignment with ridge happens when the geometric properties of $\mathbf{BD}$ and $\mathbf{AC}$ are similar. If there is a big difference $\|\mathbf{BD}\| \gg \|\mathbf{AC}\|$, the connection will cross the ridges.

In order to illustrate the proposed Hausdorff distance, we apply it as the metric for the grassfire algorithm introduced in Section 4.1 and produce the distance field. Then, we compare the distance field with that produced by the metric of unwrapped Euclidean distance.



(a) Unwrapped Euclidean distance (b) The proposed Hausdorff distance

Figure 10: Distance fields of grassfire algorithms using different metrics



(a) Unwrapped Euclidean distance (b) The proposed Hausdorff distance

Figure 11: Connectivity of grassfire algorithms using different metrics

As shown in Figure 10, (a) and (b) are the distance fields generated by the unwrapped Euclidean distance and the proposed Hausdorff distance respectively on the Stanford bunny model. The centers of green regions have zero distance values and they are fire sources. The fire sources are extracted feature points, i.e. umbilics and ridge-crossings. The purple regions construct the Voronoi diagram of the feature points. The green regions in Figure 10 (b) are 'fatter' than those in Figure 10 (a) along the valley between the top leg and the body. Similarly, on the face, the green regions in Figure 10 (b) are 'fatter' than those in Figure 10 (a) along the valley of neck. These illustrate that 10 (b) may have better feature alignment capability than 10 (a), since the proposed Hausdorff distance metric takes into account curvature information.

The proposed metric has the potential to improve alignment with salient features. Fig. 11 (a) and (b) illustrate the connectivities produced by the unwrapped Euclidean distance metric and the proposed Hausdorff distance metric respectively. Note the ridge areas at the top of the ear, along the connection between the leg and the main body, the connection between the leg and the feet, and the connection between the tail and the main body. The green edges in Fig. 11 (a) show that the connections produced by Euclidean distance cross the ridges in those areas. The red edges in Fig. 11 (b) show that the connections produced by the proposed distance align with the ridges in those areas.

# 6 Computing the Control Mesh

For the sake of completeness, we briefly review the Loop subdivision scheme developed in Loop (1987). In particular, we focus on limit point stencils as they are used in our algorithm to map feature points on the input surface to the vertices in the control mesh. Connectivity remains unchanged.

Given a triangular mesh, the subdivision process successively refines this mesh, resulting in a smooth limit surface. A vertex on the control mesh converges to a point $\mathbf{v}^{\infty}$ on the limit surface. Using eigenvalue analysis, one

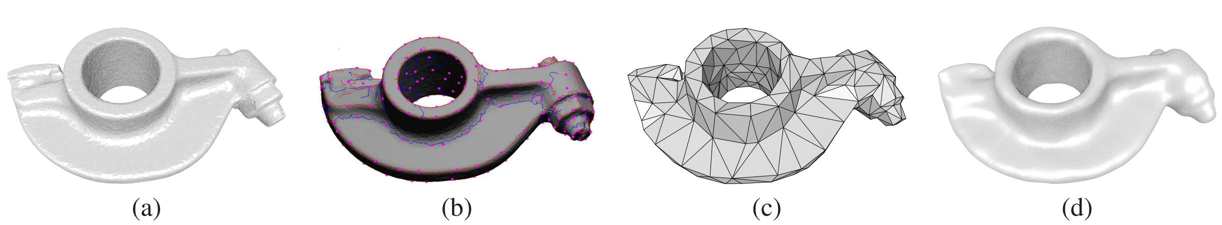


Figure 12: Subdivision surface fitting for the rocker arm model: (a) original mesh; (b) extracted ridges, umbilics, and ridge-crossings; (c) constructed connectivity; (d) fitted subdivision surface.

can show that this limit point is given by

$$\mathbf{v}^{\infty} = (1 - n\beta)\mathbf{v} + \beta \sum_{j=1}^{n} \mathbf{v}_j,$$

where the $\mathbf{v}_j$ are the neighbours of $\mathbf{v}$ in the control mesh, and $\beta = \frac{8\alpha}{3+8n\alpha}$ with $\alpha = \frac{1}{n}(\frac{5}{8} - (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n})^2)$ for $n > 3$ and $\alpha = \frac{3}{16}$ if $n = 3$.

Calculating control vertices from limit points is a reverse process of calculating limit points. Treating the filtered feature points, umbilics and ridge-crossings, as the limit points $\mathbf{v}^{\infty}$ of a subdivision surface, the control vertices $\mathbf{v}$ of the subdivision surface can be found by solving the linear system

$$\begin{array}{lcl} (1 - n_1\beta)\mathbf{v}_1 + \cdots + \beta\mathbf{v}_p + \cdots & = & \mathbf{v}_1^{\infty}, \\ \cdots + (1 - n_2\beta)\mathbf{v}_2 + \cdots + \beta\mathbf{v}_q + \cdots & = & \mathbf{v}_2^{\infty}, \\ \vdots & & \\ \cdots + \beta\mathbf{v}_r + \cdots + (1 - n_m\beta)\mathbf{v}_m & = & \mathbf{v}_m^{\infty}, \end{array}$$

where $m$ is the total number of feature points. On the left hand side of each equation, the points are $\mathbf{v}_i$ and their $n_i$ neighbours. All the control points $\mathbf{v}_i$ can be calculated by solving the linear system. The connectivity of the control mesh is kept exactly the same as the connectivity of feature points. Then the fitted subdivision surface can be generated using the standard Loop subdivision scheme.

The final Loop subdivision surface exactly goes through the filtered feature points. The final fitted surface only uses a subset of the original vertices in the fitting process, though they are all used in other stages for feature extraction and topological modelling.

To fit highly curved features, e.g. on the Rocker arm model, we use a refinement of the connectivity without changing the standard subdivision rules. A fine connectivity is produced by first subdividing the coarse connectivity using a mid-point interpolatory scheme, and further replacing each of the newly inserted vertices by the nearest vertex found in the original dense mesh. Then the refined connectivity is used to calculate the control mesh for the model with highly curved features.

To fit extreme explicit sharp features, some special subdivision rules would need to be used, e.g. semi-sharp subdivision surface fitting of Lavoué and Dupont (2009) or modified Loop subdivision rules to fit sharp features as in Ling et al. (2008). While these special rules could be incorporated, our implementation does not support them.

Note that some error control strategies could be adopted to improve the accuracy of the control mesh, such as using Panozzo et al. (2011)'s fitmaps to calculate adaptive control meshes, or using Ma et al. (2004)'s mid-point interpolatory and least squares to calculate the control mesh.

# 7 Examples

This section presents several examples to demonstrate the proposed approach. Different types of models are tested for evaluation: the rocker arm model, the Stanford bunny model, the Igea model and the Fertility model. The bunny model has ears, legs, and tail features. The Igea model has eyes, nose, and mouth features. The Fertility model has many geometrical features with complex topology and high resolution. These critical features test the performance of the proposed approach.

Fig. 12 shows the example of the rocker arm, a triangular mesh with 10044 vertices. The model is fitted by a single subdivision surface. The constructed connectivity has 239 vertices. Comparing Fig. 12 (a) to (d) illustrates that the overall shape is preserved during the fitting procedure. Comparing (b) to (c) shows that the connectivity is aligned with significant ridges along the inner and outer circles at the top and bottom of the model, and the intersection of the cylinder and the base.

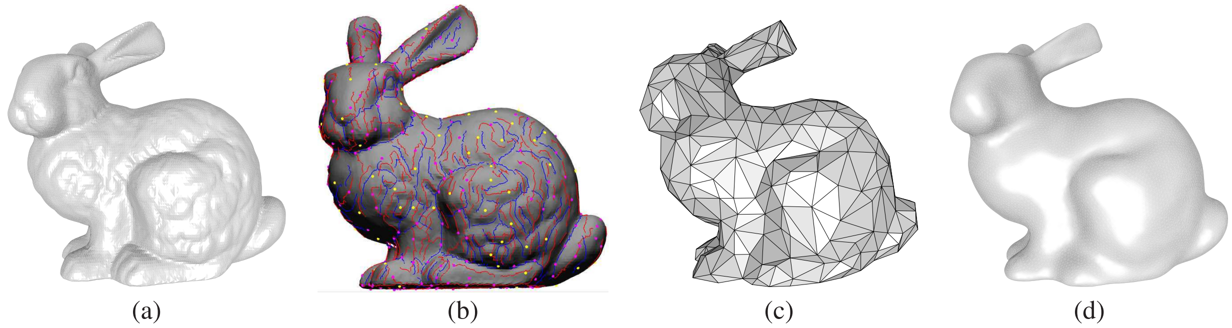


Figure 13: Subdivision surface fitting for the Stanford bunny model: (a) original mesh; (b) extracted ridges, umbilics, and ridge-crossings; (c) constructed connectivity; (d) fitted subdivision surface.

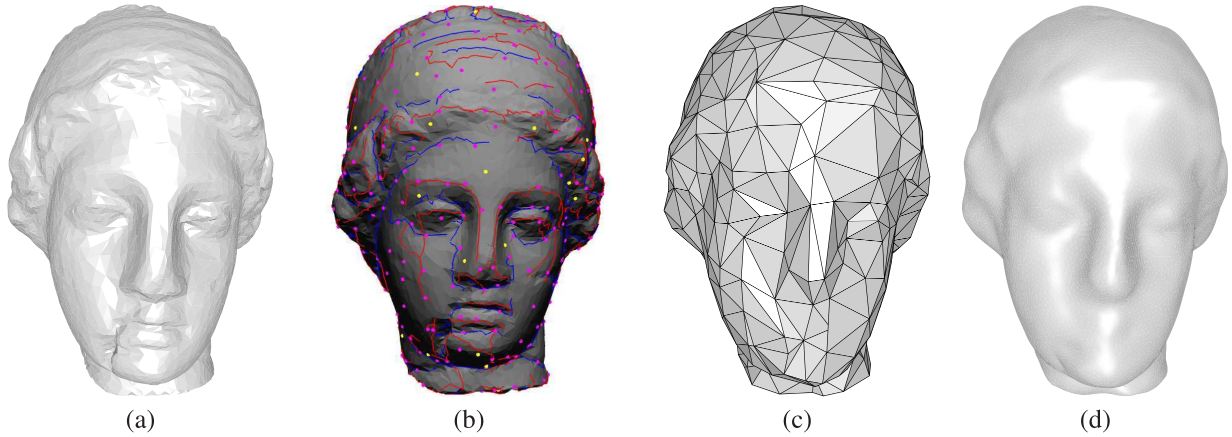


Figure 14: Subdivision surface fitting for the Igea model: (a) original mesh; (b) extracted ridges, umbilics, and ridge-crossings; (c) constructed connectivity; (d) fitted subdivision surface.

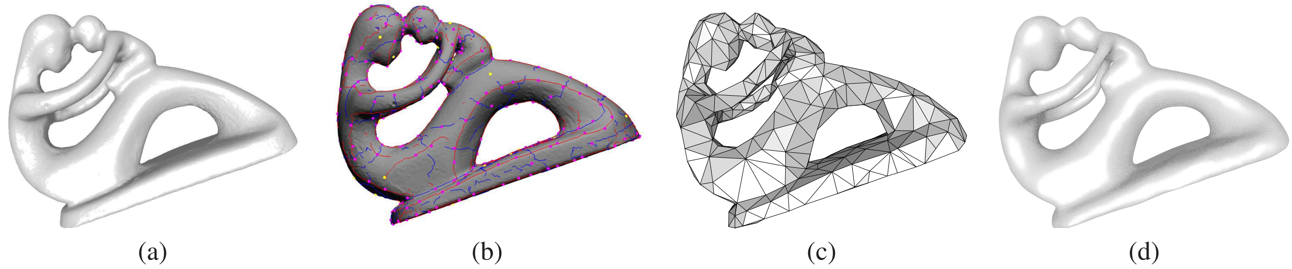


Figure 15: Subdivision surface fitting for the Fertility model: (a) original mesh; (b) extracted ridges, umbilics, and ridge-crossings; (c) constructed connectivity; (d) fitted subdivision surface.

Fig. 13 shows the Stanford bunny model. The original bunny mesh has $35947$ vertices whilst the constructed connectivity has only $381$ vertices. Comparing the fitted subdivision surface to the original mesh shows that the overall shape including ears, mouth, legs, and tail is preserved in the fitting procedure. From Fig. 13 (b) and (c), we can see that the connectivity is aligned with valleys (concave ridges) in the areas of the neck, the bottom of the ears, the intersection between legs and the main body, and the intersection between the tail and the main body of the bunny.

Fig. 14 shows the Igea model. It is reduced from $8268$ vertices to $383$ vertices. The overall shape of the Igea model is also preserved. Note the feature alignment in the areas of the neck, the top and the bottom of the nose, and the intersection of the hair and the face of the model.

Fig. 15 shows the results of the fitting processes for the Fertility model. The original dense Fertility mesh has $241607$ vertices. Since it is very expensive to estimate differential geometrical properties on such a dense mesh, the number of vertices is firstly reduced to $40000$ using Garland and Heckbert (1997)'s QEM-based mesh simplification. The number $40000$ is chosen as just over the maximum number of vertices of the other three popular models on which the proposed fitting approach works well. The salient features on the simplified Fertiltiy model

Table 1: Quantitative comparison of the proposed method and competing methods.

| | Rocker arm | Stanford bunny | Igea | Fertility |
|---|---|---|---|---|
| *No. of original vertices* | 10044 | 35947 | 8268 | 241607 |
| *No. of control points* | 239 | 381 | 383 | 354 |
| | *Hausdorff distance error* | | | |
| Lee | 0.02926 | 0.00974 | 0.02749 | 6.87943 |
| Kanai | 0.01866 | 0.00939 | 0.02251 | 3.32235 |
| Proposed | 0.01203 | 0.00879 | 0.01836 | 3.02488 |
| | *Average Euclidean distance error* | | | |
| Lee | 0.00381 | 0.00082 | 0.00266 | 0.63522 |
| Kanai | 0.00337 | 0.00071 | 0.00260 | 0.50667 |
| Proposed | 0.00195 | 0.00062 | 0.00164 | 0.38976 |

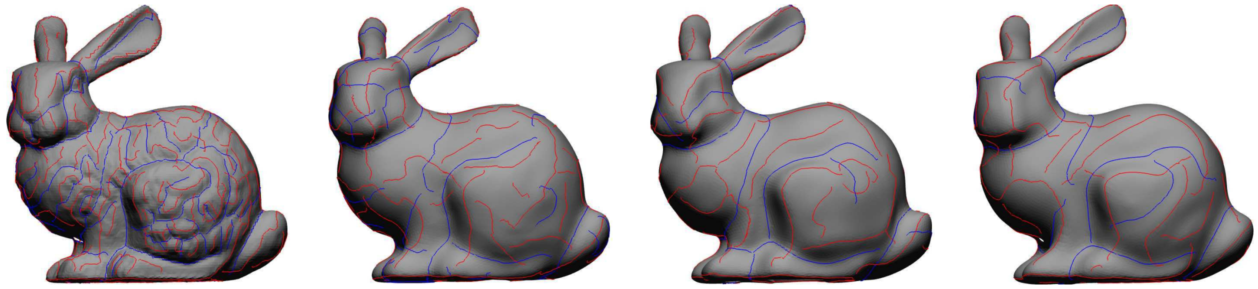
are shown in Fig. 15 (b). The constructed connectivity has only 254 vertices as shown Fig. 15 (c). The final fitted subdivision surface in Fig. 15 (d) shows that the complex topology and overall shape including heads, arms, bodies, legs and the base are preserved for both the mother and the baby. Comparing Fig. 15 (b) with (c), it is illustrated that the connectivity is aligned with valleys along the necks, the connection between the mother and the baby, and the connection between the mother and the base. The connectivity is also aligned with ridges along the profiles of the heads, arms, bodies, legs and the base.

The above examples illustrate the overall shape preserving quality and salient feature alignment of the proposed subdivision surface fitting approach. The feature alignment is improved using colour matching and a modified Hausdorff distance that includes curvature vectors. The results match the aim of the approach to fit the overall shape characterised by salient features using a relatively small number of control vertices.

Table 1 gives the quantitative comparison between the original surfaces and the surfaces fitted by the proposed method and the competing methods. Both error criteria of Hausdorff distance and average Euclidean distance are used for the comparison. The comparison focuses on the final errors instead of the processes. The processes are treated as black boxes. The comparison concerns the errors between the output (the fitted meshes) and the input (the original meshes) of the black boxes (the processes of the proposed method and the competing methods). These errors are computed between the final fitted subdivision surface and the original mesh. The distances are evaluated from vertices on one mesh to the faces of the other mesh. Table 1 illustrates that the proposed method has smaller geometric approximation errors than the competing methods of Lee et al. (2000) and Kanai (2001) in both error criteria of Hausdorff distance and average Euclidean distance.

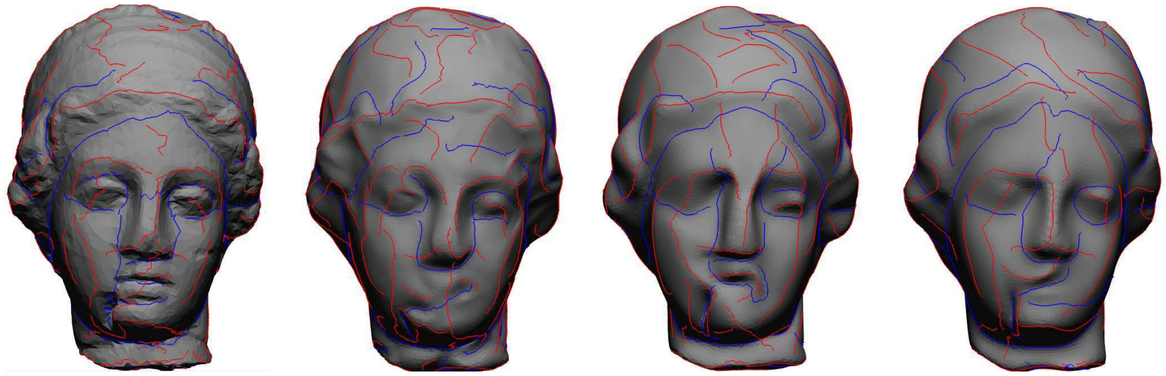
For feature preserving, the proposed method performs better than the other fitting approaches. Fig. 16 shows the features on the original bunny model and the fitted subdivision surfaces generated by the proposed method, Ma et al. (2004)'s simplification-based method and Kanai (2001)' method respectively, using the same number of control points, 381. On the bunny head, the proposed method preserves features better than the others. For example, the three valleys and the three ridges of the mouth are well preserved by the proposed method, while they are changed or removed by other two methods. The four valleys on the face are also well preserved by the proposed method, while they degenerate to one or disappear in other two methods. For the features on other parts of the bunny, there is no significance difference among all the methods. Fig. 17 illustrates the comparison on the Igea model. On the eyes, the proposed method preserves features better than others. For example, the creases at the top right eyes are missed in the simplification-based and Kanai's methods. The ridge on the left eyebrow is discarded in Kanai method. These features are preserved in the proposed method. For the other parts, there is no big difference between all the methods. Fig. 18 shows the comparison on the Fertility model. The valleys on the baby's face is removed by the simplification-based method. One valley on the mother's face is discarded by Kanai's method. Both competing methods missed the valley along the connection of the mother's head and her neck, and also missed some valleys on the mother's leg. The proposed method preserves all these features. The results are not surprising, since the connectivity of the proposed methods intends to follow the significant ridges and thus preserves features well.

The total computational time is few minutes running on an Intel i7 CPU 2.2G under a Windows 7 operating system. As shown in Table 2, the total running times are around $1.5$, $4$, $1$, and $4$ minute(s) for the models of Rocker arm, Stanford bunny, Igea, and Fertility, respectively. Note that the dense mesh of the Fertility model is pre-simplified to 40000 vertices, so that it has a similar number of vertices to the other models. It is illustrated that about $80\%$ of the running time is spent on feature extraction including estimation of differential geometry properties by jet fitting, extraction of ridges by a marching method, and extraction of umbilics by local minimisation. These



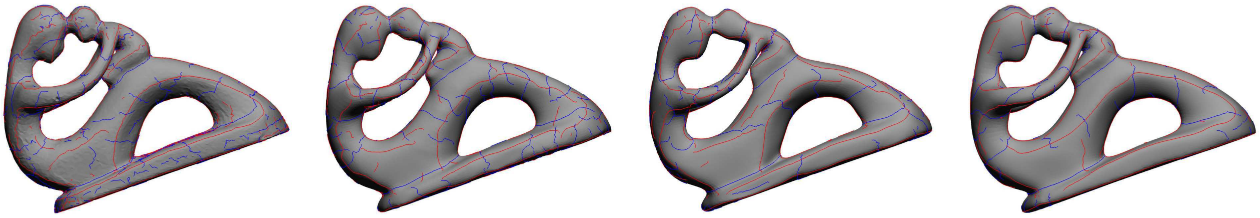
(Original) (Proposed) (Simplification) (Kanai)

Figure 16: Comparison of feature preserving of the Stanford bunny model for different subdivision surface fitting algorithms.



(Original) (Proposed) (Simplification) (Kanai)

Figure 17: Comparison of feature preserving of the Igea model for different subdivision surface fitting algorithms.



(Original) (Proposed) (Simplification) (Kanai)

Figure 18: Comparison of feature preserving of the Fertility model for different subdivision surface fitting algorithms.

processes are relatively efficient, since they are well established and have already been integrated into CGAL. In the fitting process, these functions are called from CGAL. The construction of connectivity spends about $20\%$ time, since the distance transform algorithm is linear, looping once over each original vertex by using a priority queue. The calculation of control points is within 1 second. Although the calculation by the linear solver is not linear, this will not be dominant because the number of filtered significant feature points is usually much smaller than that of the original vertices; the matrix of the system is very sparse. The final Loop subdivision is very efficient, spending less than 1 second. Table 3 compares the running times between different methods. The proposed method is slower than the competing methods, since it spends most of the time on feature extraction. Although the proposed method sacrifices some efficiency to preserve the accuracy and features, it is still reasonably efficient with total computation times of few minutes.

# 8 Conclusion

This paper proposed a method for constructing a smooth subdivision surface that fits a dense triangle mesh of arbitrary topology. This approach can preserve and align the salient features, and thus preserve the overall shape of the mesh. The natural ridge-joined connectivity of umbilics and ridge-crossings is used as the connectivity of

Table 2: Computational complexity of the proposed fitting processes

| | Rocker arm | Stanford bunny | Igea | Fertility |
|---|---|---|---|---|
| *No. of original vertices* | 10044 | 35947 | 8268 | 241607 |
| *No. of control points* | 239 | 381 | 383 | 354 |
| | *Computational times (seconds)* | | | |
| Mesh simplification | - | - | - | 9 |
| Estimation of curvatures | 18 | 57 | 15 | 62 |
| Extraction of ridges | 5 | 18 | 4 | 17 |
| Extraction of umbilics | 51 | 89 | 32 | 96 |
| Construction of connectivity | 11 | 68 | 9 | 73 |
| Calculation of control mesh | 1 | 1 | 1 | 1 |
| Subdivision | 1 | 1 | 1 | 1 |
| Total | 87 | 234 | 62 | 259 |

Table 3: Comparison of computational times of the proposed method and the competing methods

| | Rocker arm | Stanford bunny | Igea | Fertility |
|---|---|---|---|---|
| *No. of original vertices* | 10044 | 35947 | 8268 | 241607 |
| *No. of control points* | 239 | 381 | 383 | 354 |
| | *Computational times (seconds)* | | | |
| Simplification-based | 11 | 23 | 10 | 97 |
| Kanai | 5 | 14 | 4 | 76 |
| Proposed method | 87 | 234 | 62 | 259 |

control mesh for subdivision. These features can characterise the overall shape by capturing the extremal principal curvatures. A metric of Hausdorff distance including curvature vectors was derived to increase the speed and feature alignment of the grass-fire algorithm. The feature alignment is then improved by colour marching and edge flipping. Surfaces generated by this approach are suitable for GPU-based rendering with applications in animation or reverse engineering.

While this paper demonstrates the potential and flexibility of the described process, there is scope for future work. The accuracy of the approach is affected by the position and number of filtered umbilics and ridges. The fitting quality could be improved by a more robust feature extraction algorithm and optimal feature filtering method. This could be obtained by using a more detailed classification of ridges and umbilics. Panozzo et al. (2011)'s adaptive mesh can also be used to get a more accurate approximation. There exist some non-regular vertices, which are associated with different Gaussian curvatures. Cheng et al. (2004)'s post-processing step can be used to improve the regularisation of vertex valencies.

## Acknowledgment

The authors would like to thank Malcolm Sabin and Neil Dodgson for their valuable suggestions regarding the paper and the anonymous referees whose comments helped to improve the paper. The authors are pleased to acknowledge the financial support of EPSRC through Grant EP/H024816/1 and School of Computing, Engineering and Applied Mathematics in University of Abertay Dundee.

## References

Aleksandrov, L., Djidjev, H., Guo, H., Maheshwari, A., Nussbaum, D., and Sack, J.-R. (2010). Algorithms for approximate shortest path queries on weighted polyhedral surfaces. *Discrete & Computational Geometry*, 44:762–801.

Bogaevski, I., Lang, V., Belyaev, A., and Kunii, T. (2003). Color ridges on implicit polynomial surfaces. In *Proceedings GraphiCon 2003*, pages 161–164, Moscow, Russia. Moscow State University.

Boier-Martin, I., Rushmeier, H., and Jin, J. (2004). Parameterization of triangle meshes over quadrilateral domains.

In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 193–203, New York, NY, USA. ACM.

Bommes, D., Zimmer, H., and Kobbelt, L. (2009). Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28:77:1–77:10.

Bose, P., Maheshwari, A., Shu, C., and Wuhrer, S. (2011). A survey of geodesic paths on 3d surfaces. *Computational Geometry*, 44(9):486 – 498.

Campen, M., Heistermann, M., and Kobbelt, L. (2013). Practical anisotropic geodesy. *Computer Graphics Forum*, 32(5):63–71.

Cashman, T. J., Augsdörfer, U. H., Dodgson, N. A., and Sabin, M. A. (2009). NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Trans. Graph.*, 28(3):1–9.

Catmull, E. and Clark, J. (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350 – 355.

Cazals, F. (2008). Jet fitting 3: A generic C++ package for estimating the differential properties on sampled surfaces via polynomial fitting. *ACM Trans. Math. Softw.*, 35(3):24:1–24:20.

Cazals, F. and Pouget (2005). Topology driven algorithms for ridge extraction on meshes. Research Report RR-5526, INRIA.

Cazals, F. and Pouget, M. (2005a). Differential topology and geometry of smooth embedded surfaces: selected topics. *International Journal of Computational Geometry and Applications*, Vol. 15, No. 5:511–536.

CGAL (2013). Computational Geometry Algorithms Library. www.cgal.org.

Cheng, K.-S., Wang, W., Qin, H., Wong, K.-Y., Yang, H., and Liu, Y. (2004). Fitting subdivision surfaces to unorganized point data using SDM. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 16 – 24.

Clémençon, B., Borouchaki, H., and Laug, P. (2008). Ridge extraction and its application to surface meshing. *Engineering with Computers*, 24:287–304.

Cohen-Steiner, D., Alliez, P., and Desbrun, M. (2004). Variational shape approximation. *ACM Trans. Graph.*, 23(3):905–914.

D'Azevedo, E. F. (2000). Are bilinear quadrilaterals better than linear triangles? *SIAM Journal On Scientific Computing*, 22(1):198 – 217.

Doo, D. and Sabin, M. (1978). Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356 – 360.

Fabbri, R., Costa, L. D. F., Torelli, J. C., and Bruno, O. M. (2008). 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44.

Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Guan, W. and Ma, S. (1998). A list-processing approach to compute Voronoi diagrams and the euclidean distance transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(7):757 –761.

Hallinan, P. W., Gordon, G. G., Yuille, A. L., Giblin, P., and Mumford, D. (1999). *Two- and three-dimensional patterns of the face*. A. K. Peters, Ltd., Natick, MA, USA.

Hildebrandt, K., Polthier, K., and Wardetzky, M. (2005). Smooth feature lines on surface meshes. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. (1994). Piecewise smooth surface reconstruction. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 295–302, New York, NY, USA. ACM.

Käberer, F., Nieser, M., and Polthier, K. (2007). QuadCover - surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384.

Kanai, T. (2001). MeshtoSS: Converting subdivision surfaces from dense meshes. In *in Proc. of the Vision Modeling and Visualization Conference*, pages 325–332.

Kim, S.-K. and Kim, C.-H. (2006). Finding ridges and valleys in a discrete surface using a modified {MLS} approximation. *Computer-Aided Design*, 38(2):173 – 180.

Lai, Y.-K., Hu, S.-M., and Pottmann, H. (2006). Surface fitting based on a feature sensitive parametrization. *Computer-Aided Design*, 38(7):800 – 807.

Lai, Y.-K., Kobbelt, L., and Hu, S.-M. (2010). Feature aligned quad dominant remeshing using iterative local updates. *Computer-Aided Design*, 42(2):109 – 117.

Lai, Y.-K., Zhou, Q.-Y., Hu, S.-M., Wallner, J., and Pottmann, D. (2007). Robust feature classification and editing. *Visualization and Computer Graphics, IEEE Transactions on*, 13(1):34–45.

Lavoué, G. and Dupont, F. (2009). Semi-sharp subdivision surface fitting based on feature lines approximation. *Computers and Graphics*, 33(2):151 – 161.

Lavoué, G., Dupont, F., and Baskurt, A. (2007). A framework for quad/triangle subdivision surface fitting: Application to mechanical objects. *Computer Graphics Forum*, 26(1):1–14.

Lee, A., Moreton, H., and Hoppe, H. (2000). Displaced subdivision surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 85–94, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Li, W.-C., Ray, N., and Lévy, B. (2006). Automatic and interactive mesh to T-spline conversion. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 191–200, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Ling, R., Wang, W., and Yan, D. (2008). Fitting sharp features with Loop subdivision surfaces. *Computer Graphics Forum*, 27(5):1383–1391.

Loop, C. (1987). Smooth subdivision surfaces based on triangles. Msc thesis, department of mathematics, University of Utah, Utah, USA.

Ma, W., Ma, X., Tso, S.-K., and Pan, Z. (2004). A direct approach for subdivision surface fitting from a dense triangle mesh. *Computer-Aided Design*, 36(6):525 – 536.

Ma, W. and Zhao, N. (2002). Smooth multiple B-spline surface fitting with Catmull-Clark subdivision surfaces for extraordinary corner patches. *The Visual Computer*, 18:415–436.

Ma, X. and Cripps, R. J. (2007). Estimation of end curvatures from planar point data. In *Proceedings of the 12th IMA international conference on Mathematics of surfaces XII*, pages 307–319, Berlin, Heidelberg. Springer-Verlag.

Ma, X. and Cripps, R. J. (2011). Shape preserving data reduction for 3D surface points. *Computer-Aided Design*, 43(8):902 – 909.

Marinov, M. and Kobbelt, L. (2005). Optimization methods for scattered data approximation with subdivision surfaces. *Graphical Models*, 67(5):452 – 473.

NieBner, M., Loop, C., Meyer, M., and Derose, T. (2012). Feature-adaptive GPU rendering of Catmull-Clark subdivision surfaces. *ACM Trans. Graph.*, 31(1):6:1–6:11.

Ohtake, Y., Belyaev, A., and Seidel, H.-P. (2004). Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3):609–612.

Panozzo, D., Puppo, E., Tarini, M., Pietroni, N., and Cignoni, P. (2011). Automatic construction of quad-based subdivision surfaces using fitmaps. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1510 –1520.

Porteous, I. (1994). *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge Univ. Press.

Ray, N. and Sokolov, D. (2013). Tracing cross-free polylines oriented by a n-symmetry direction field on triangulated surfaces. *CoRR*, abs/1306.0706.

Sadeghi, J. and Samavati, F. F. (2011). Smooth reverse loop and catmull-clark subdivision. *Graphical Models*, 73(5):202 – 217.

Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S. J., and Hoppe, H. (2005). Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3):553–560.

Suzuki, H., Takeuchi, S., and Kanai, T. (1999). Subdivision surface fitting to a range of points. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 158 –167, 322.

Thirion, J.-P. (1996). The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision*, 19:115–128.

Yoshizawa, S., Belyaev, A., Yokota, H., and Seidel, H.-P. (2008). Fast, robust, and faithful methods for detecting crest lines on meshes. *Computer Aided Geometric Design*, 25(8):545 – 560.