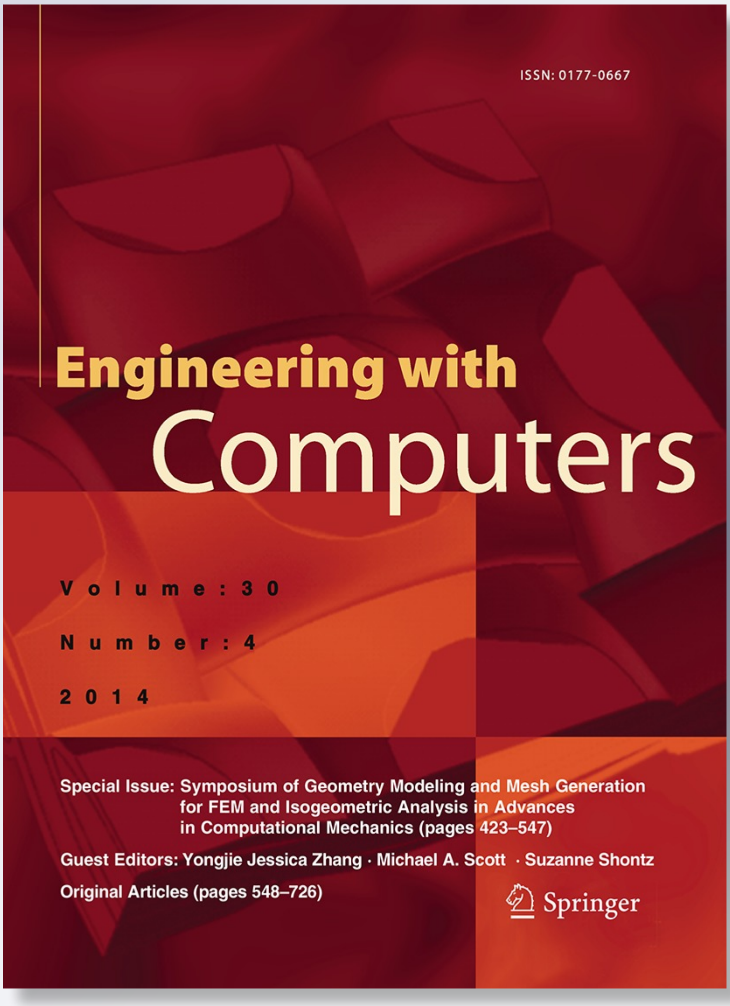
A computational approach to joint line detection on triangular meshes

Xunnian Yang, Jianmin Zheng & Desheng Wang

Engineering with Computers
An International Journal for Simulation-Based Engineering

ISSN 0177-0667
Volume 30
Number 4

Engineering with Computers (2014)
30:583-597
DOI 10.1007/s00366-012-0303-9


ISSN: 0177-0667
Engineering with
Computers
Volume: 30
Number: 4
2014
Special Issue: Symposium of Geometry Modeling and Mesh Generation for FEM and Isogeometric Analysis in Advances in Computational Mechanics (pages 423–547)
Guest Editors: Yongjie Jessica Zhang · Michael A. Scott · Suzanne Shontz
Original Articles (pages 548–726)
Springer

Engineering with Computers (2014) 30:583–597
DOI 10.1007/s00366-012-0303-9

ORIGINAL ARTICLE

A computational approach to joint line detection on triangular meshes

Xunnian Yang · Jianmin Zheng · Desheng Wang

Received: 15 July 2012 / Accepted: 15 November 2012 / Published online: 1 December 2012

Abstract In this paper, we present formulae for evaluating differential quantities at vertices of triangular meshes that may approximate potential piecewise smooth surfaces with discontinuous normals or discontinuous curvatures at the joint lines. We also define the C^1 and C^2 discontinuity measures for surface meshes using changing rates of one-sided curvatures or changing rates of curvatures across mesh edges. The curvatures are computed discretely as of local interpolating surfaces that lie within a tolerance to the mesh. Together with proper estimation of local shape parameters, the obtained discontinuity measures own properties like sensitivity to salient joint lines and being scale invariant. A simple algorithm is finally developed for detection of C^1 or C^2 discontinuity joint lines on triangular meshes with even highly non-uniform triangulations. Several examples are provided to demonstrate the effectiveness of the proposed method.

Keywords Discrete differential geometry · Triangle meshes · Discontinuity measures · Joint line detection

X. Yang (✉)
Department of Mathematics, Zhejiang University,
Hangzhou, China
e-mail: yxn@zju.edu.cn

J. Zheng
School of Computer Engineering, Nanyang Technological
University, Singapore, Singapore

D. Wang
Division of Mathematics, SPMS, Nanyang Technological
University, Singapore, Singapore

1 Introduction

Triangular meshes can be used to represent surfaces with complex shapes, they can be rendered efficiently and also they can be easily transferred among various CAD systems. A triangular mesh can be regarded as the tessellation or the approximation of an unknown piecewise smooth surface that may have C^1 or C^2 discontinuity joint lines [6]. A mesh edge is said to be C^1 or C^2 discontinuous if it approximates a C^1 or C^2 discontinuity joint line of the surface. The detection of normal or curvature discontinuity joint lines is useful for further shape processing of triangular meshes like patch segmentation [1, 12, 18], remeshing [25, 28], feature sensitive hole filling [27], etc.

In contrast to edge detection in computer vision [4], identification of C^1 or C^2 discontinuity joint lines for meshes is more challenging. Detection of C^1 or C^2 discontinuity joint edges depends heavily on proper estimation of normal vectors and curvatures on triangular meshes that may have highly irregular or non-uniform triangulations. Even discrete normals and curvatures have been properly estimated, naive comparison of discrete normals or curvatures does not guarantee the correct detection of all C^1 or C^2 discontinuities. It may miss true discontinuities in low curvature regions but introduce false ones in high curvature regions.

For a mesh with unknown C^1 or C^2 discontinuities, we assume that there is a unique normal vector at every edge and normal curvatures at mesh edges or at their opposite vertices are estimated as of local interpolating surfaces in a small tolerance to the mesh. The C^1 or C^2 discontinuity of a mesh edge can be measured by the changing rates of normal curvatures across the edge with respect to the normal curvatures at two opposite vertices or the changing rate of normal curvatures across the edge. Besides the fitting

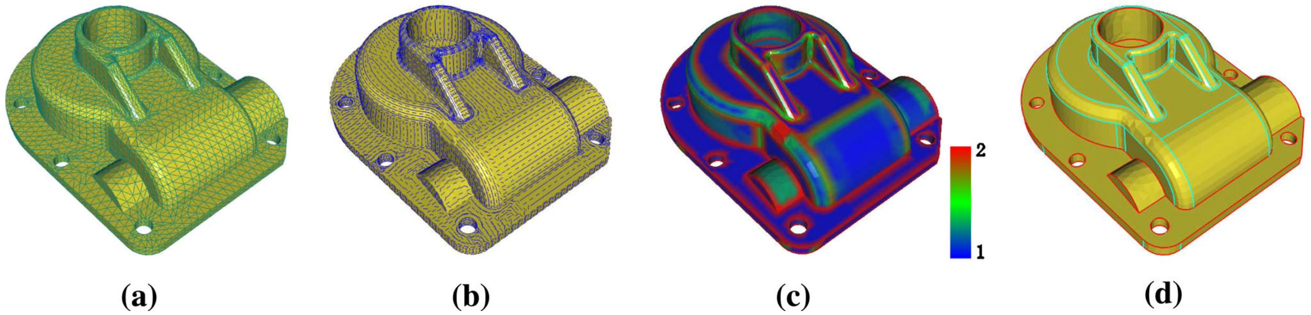


Fig. 1 Joint line detection on a CAD mesh: **a** the input triangular mesh; **b** the principal directions along edges; **c** the plot of C^2 discontinuity measure (the discontinuity measure of a vertex is defined as the maximum value of discontinuity measures of all abutting edges); **d** the detected C^1 (*red*) and C^2 (*cyan*) discontinuity joint edges (color figure online)

tolerance, we also estimate local shape parameters like crease directions at mesh edges for robust computation of discontinuity measures at edges. The obtained discontinuity measures have the properties like sensitivity to true joint lines and being scale invariant, etc. Then, the C^1 or C^2 discontinuity joint edges can be detected easily based on a simple threshold. Moreover, the initially detected joint edges can be smoothed further based on a few collective properties or assumptions of the joint lines.

Figure 1 illustrates the main steps of joint line detection on a triangular mesh. For a given surface mesh we first compute differential quantities and shape parameters. Figure 1b shows the crease directions across the mesh edges. Afterwards, we compute C^1 or C^2 discontinuity measures for every edge; see Fig. 1c for the plot of C^2 discontinuity measures. Based on the discontinuity measures we can detect joint lines; see Fig. 1d for the result by the proposed approach.

The organization of the paper is as follows. Section 2 introduces some related work. Notations and formulae for some basic differential quantities will be given in Sect. 3. The C^1 or C^2 discontinuity measures for mesh edges are evaluated in Sect. 4. An algorithm for C^1 or C^2 discontinuity joint line detection on triangular meshes is presented in Sect. 5. Section 6 provides our experimental results. Section 7 concludes the paper.

2 Related work

Usually, an edge with a dihedral angle greater than a given threshold is considered a sharp edge. Hubeli et al. [7] proposed to compute angles between normals of surface patches that are fitted to selected neighborhoods of mesh edges. Vidal et al. [23] proposed to compute angles between one-sided tangent planes. Jiao and Heath [9] proposed to detect joint edges based on the dihedral angles across edges together with the smoothness assumption of potential feature lines. Baker [3] and Jiao and Bayyana [8] improved the technique further by using more collective properties of feature lines and local geometric quantities of the mesh.

Inspired by the technique of tensor voting [19, 20], Page et al. [16] proposed to detect creases on surface meshes by a normal voting method. Shimizu et al. [17] defined edge strengths for meshes based on eigen analysis of covariance matrices of voted normals. Recently, Angelo and Stefano [2] proposed to detect C^1 discontinuity edges using a sharpness indicator defined by local fitting of paraboloid to mesh data. Similar to the edge strength measure, this method may miss salient feature edges in low curvature regions or introduce false detected edges in high curvature regions. Kim et al. [10] proposed to classify vertices on a triangular mesh into corners, edges or smooth surface regions by the tensor voting theory, too, and they in fact segment a surface into various feature parts.

In contrast to C^1 discontinuity detection, detection of joint lines across which the surface curvatures are not continuous is more challenging. Yamakawa and Shimada [28] proposed a polygon crawling technique to detect C^2 discontinuity edges on cylinder like surfaces. Recently, Jiao and Bayyana [8] proposed a method to detect C^2 discontinuity edges based on a set of observed rules and a heuristic algorithm. By this method, only edges lying on boundaries of flat regions may be detected.

Joint lines can also be obtained when a surface has been segmented into individual patches. Demarsin et al. [5] proposed to detect closed sharp features on point set surfaces using normal estimation and graph theory. Sunil and Pande [18] proposed to segment surface meshes into feature patches using region growing together with identification of salient edges. Várady et al. [22] proposed to recover primary surface patches in scan reconstructed surfaces and then extract feature lines as the surface boundaries. Though surface segmentation is itself an important research area in CAD and graphics [1, 26], but it is too costly to segment surfaces when only joint lines should be detected.

Our proposed algorithm depends on proper estimation of normal vectors or curvatures at mesh vertices. Though these two topics have been studied extensively in the literature [13, 14, 15, 20, 21], but all these methods assume that a triangular mesh is approximating a smooth surface and there is a unique normal vector or curvature tensor at a vertex. If a vertex lies on a joint line, these methods may not work. We will propose methods to compute normal vectors and normal curvatures in selected directions even when a vertex is a joint point of several surface patches.

3 Notations and basic geometric quantities

This section presents formulae for computing several basic geometric quantities for surface meshes. Before that, notations for variables used in the paper are given in Table 1.

Table 1 Notations for the variables used in the paper

$\mathbf{e} = \mathbf{p}_0\mathbf{p}_1$	A mesh edge and the end vertices of the edge
$\mathbf{u}_e$	The unit direction of the edge $\mathbf{e}$
$\mathbf{n}_e(\mathbf{n}_q)$	Unit normal vector at an edge (vertex)
$\mathbf{n}_q^e$	Unit normal at vertex $\mathbf{q}$ in a plane perpendicular to edge $\mathbf{e}$
θ_e	The signed dihedral angle at an edge
k_{qb}	Normal curvature at $\mathbf{q}$ in direction $\mathbf{b} - \mathbf{q}$
H_{qb}	The height of the arc that interpolates points $\mathbf{q}$, $\mathbf{b}$ and normal $\mathbf{n}_q$
T_q	The tensor matrix at a vertex $\mathbf{q}$
$\mathbf{t}_q$	Unit crease direction at a vertex
$\mathbf{t}_e$	Unit crease direction at an edge
$k_{\mathrm{l}}^{\mathrm{b}}(k_{\mathrm{l}}^{\mathrm{e}})$	The beginning (end) curvature of a spiral to the left side of edge $\mathbf{e}$
$k_{\mathrm{r}}^{\mathrm{b}}(k_{\mathrm{r}}^{\mathrm{e}})$	The beginning (end) curvature of a spiral to the right side of edge $\mathbf{e}$
$k_{\mathrm{l}}^{\mathrm{m}}(k_{\mathrm{r}}^{\mathrm{m}})$	The middle curvature of a spiral to the left (right) side of edge $\mathbf{e}$
$\mu_e^-(\mu_e^+)$	The curvature changing rate at the left (right) side of edge $\mathbf{e}$
μ_e^c	The curvature changing rate across an edge
$\mu_e^1(\mu_e^2)$	The C^1 (C^2) discontinuity measure of edge $\mathbf{e}$
ϕ_e	The angle between an edge $\mathbf{e}$ and the crease direction at the edge
ϕ_n	The angle between two planes spanned by vectors $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{n}_0$ or by vectors $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{n}_1$
ρ_e	Local roughness of edge $\mathbf{e}$
l_{m}	Mean edge length of the mesh
k_{ave}	The average of curvatures across a set of selected edges of the mesh
τ_{m}	The tolerance bound for a mesh which approximates a piecewise smooth surface

3.1 Normals and angles at edges

Assume $\mathbf{p}_0\mathbf{p}_1$ is an edge shared by two triangles f_0 and f_1 on the left or the right side, respectively. Let $\mathbf{n}_0$ and $\mathbf{n}_1$ be the unit normal vectors of the two triangles, the unit normal vector at the edge is given by

$$\mathbf{n}_e = \frac{\mathbf{n}_0 + \mathbf{n}_1}{\|\mathbf{n}_0 + \mathbf{n}_1\|}. \tag{1}$$

The signed dihedral angle across edge $\mathbf{p}_0\mathbf{p}_1$ can be computed as the angle between vectors $\mathbf{n}_0$ and $\mathbf{n}_1$. The angle is positive when the surface is local convex and the angle is negative otherwise. We have

$$\theta_e = \begin{cases} \arccos(\mathbf{n}_0 \cdot \mathbf{n}_1) & , \quad \text{if } (\mathbf{n}_0 \times \mathbf{n}_1) \cdot (\mathbf{p}_1 - \mathbf{p}_0) > 0 \\ -\arccos(\mathbf{n}_0 \cdot \mathbf{n}_1) & , \quad \text{otherwise} \end{cases} \tag{2}$$

We note that all angles and angle thresholds in this paper are in radian.

3.2 Normal curvatures at vertices or across edges

Assume $\mathbf{q}$ is a vertex neighboring to edge $\mathbf{p}_0\mathbf{p}_1$ on a triangle mesh; see Fig. 2a. To estimate normal curvature of the mesh at $\mathbf{q}$ in a direction perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$ we have to estimate normal vector at the vertex in a plane perpendicular to the edge. Instead of computing the intersection line between the plane and the mesh, we compute the normal vector at vertex $\mathbf{q}$ as a weighted sum of normal vectors of neighboring triangles. To define the normal vector properly when the vertex $\mathbf{q}$ is lying on a joint line between two faces, the coefficient for a facet normal $\mathbf{n}_i$ should be proportional to the vertex angle α_i of the ith neighbor triangle. The influence of $\mathbf{n}_i$ should be decreased further when $\mathbf{n}_i$ and edge $\mathbf{p}_0\mathbf{p}_1$ are far from being orthogonal. Let φ_i be the angle between line $\mathbf{p}_0\mathbf{p}_1$ and the plane through the ith neighbor triangle, we compute the normal vector that is approximately perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$ as

$$\bar{\mathbf{n}}_q^e = \frac{\sum_{i \in N(q)} \alpha_i \cos^{2k} \varphi_i \mathbf{n}_i}{\| \sum_{i \in N(q)} \alpha_i \cos^{2k} \varphi_i \mathbf{n}_i \|}, \tag{3}$$

where $N(q)$ means the index set of neighboring triangles at vertex $\mathbf{q}$ and k is an integer that controls the coefficients. Let $\mathbf{u}_e = \frac{\mathbf{p}_1 - \mathbf{p}_0}{\|\mathbf{p}_1 - \mathbf{p}_0\|}$, the angle φ_i can be computed by $\cos^2 \varphi_i = 1 - (\mathbf{u}_e \cdot \mathbf{n}_i)^2$; see Fig. 2b.

If k equals zero, $\bar{\mathbf{n}}_q^e$ reduces to an angle-weighted normal vector $\mathbf{n}_q$, which can be used as the normal vector of a potential smooth surface at vertex $\mathbf{q}$. With increased value of k, the weights of those neighboring triangles which are approximately perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$ will decrease drastically. In our experiments we choose $k = 10$ that can give reasonable estimation of normal curvatures.

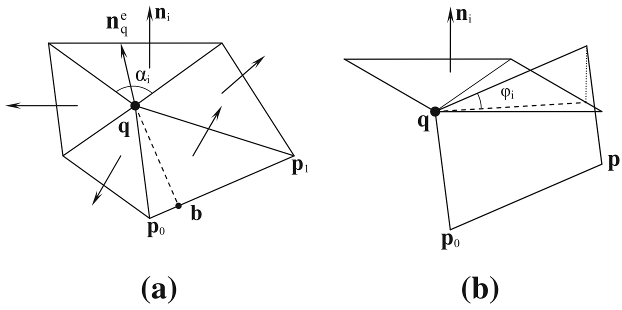


Fig. 2 Computing vertex normals for normal curvature estimation

The unit normal vector at vertex $\mathbf{q}$ in a plane perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$ is obtained as

$$\mathbf{n}^e_q = \frac{\bar{\mathbf{n}}^e_q - (\bar{\mathbf{n}}^e_q \cdot \mathbf{u}_e)\mathbf{u}_e}{\|\bar{\mathbf{n}}^e_q - (\bar{\mathbf{n}}^e_q \cdot \mathbf{u}_e)\mathbf{u}_e\|}. \tag{4}$$

Assume $\mathbf{b}$ is the perpendicular foot of vertex $\mathbf{q}$ onto the edge $\mathbf{p}_0\mathbf{p}_1$. Following the method in [15, 20], the normal curvature at vertex $\mathbf{q}$ in direction $\mathbf{qb}$ can be estimated by an interpolating circular arc. It yields

$$k_{qb} = \frac{2(\mathbf{q} - \mathbf{b}) \cdot \mathbf{n}^e_q}{\|\mathbf{q} - \mathbf{b}\|^2}. \tag{5}$$

Similarly, the normal curvature across edge $\mathbf{p}_0\mathbf{p}_1$ in a direction $\mathbf{bq}$ can be computed by $k_{bq} = \frac{2(\mathbf{b}-\mathbf{q})\cdot\mathbf{n}_e}{\|\mathbf{b}-\mathbf{q}\|^2}$.

Besides estimation of normal curvatures, we can also estimate the fitting error from a potential interpolating surface to the mesh by computing the heights of interpolating arcs. Let ϕ be the unsigned acute angle between vector $\mathbf{n}_q$ and vector $\mathbf{q} - \mathbf{b}$, we have $\cos\phi = \frac{|\mathbf{n}^e_q\cdot(\mathbf{q}-\mathbf{b})|}{\|\mathbf{q}-\mathbf{b}\|}$ and $\sin\phi = \sqrt{1 - \cos^2\phi}$. The height H_{qb} from the arc to the chord $\mathbf{q} - \mathbf{b}$ can be computed by

$$H_{qb} = \frac{\|\mathbf{q} - \mathbf{b}\|}{2\cos\phi}(1 - \sin\phi). \tag{6}$$

Since H_{qb} is determined by points $\mathbf{q}$, $\mathbf{b}$ and normal vector $\mathbf{n}^e_q$, we denote further $H_{qb} = H(\mathbf{q}, \mathbf{n}^e_q, \mathbf{b})$.

3.3 Estimating the fitting tolerance

Based on the assumption that a triangular mesh is an approximation of a piecewise smooth surface, we should then estimate the bound of fitting error for the mesh. Because an edge with a low absolute dihedral angle may lie on a flat region and an edge with a large dihedral angle is probably a feature edge, we then estimate local heights for mesh edges of which the dihedral angles belong to a properly chosen interval $[\theta_l, \theta_L]$. Assume $\mathbf{p}_0\mathbf{p}_1$ is an edge with two opposite vertices $\mathbf{q}_0$ and $\mathbf{q}_1$ on two sharing triangles, the local height at the edge can be computed as the distance between lines $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{q}_0\mathbf{q}_1$. We have

$$h_e = \frac{\|(\mathbf{p}_0 - \mathbf{q}_0) \cdot ((\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{q}_1 - \mathbf{q}_0))\|}{\|(\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{q}_1 - \mathbf{q}_0)\|}. \tag{7}$$

The fitting tolerance for a triangular mesh is given by

$$\tau_{\mathrm{m}} = \frac{3}{n} \sum_{\theta_l \le |\theta_e| \le \theta_L} h_e, \tag{8}$$

where n is the total number of edges that satisfy $\theta_l \le |\theta_e| \le \theta_L$.

4 C^1 and C^2 discontinuity measures

In this section we propose to measure C^1 or C^2 discontinuities across mesh edges by the changing rates of normal curvatures on either side or across the edges. A robust algorithm for shape aware discontinuity measure computation will also be given.

4.1 Curvature based discontinuity measures

4.1.1 Choosing opposite vertices for an edge

Basically, the opposite vertices for an edge $\mathbf{p}_0\mathbf{p}_1$ can be chosen as vertices $\mathbf{q}_0$ and $\mathbf{q}_1$ on two sharing triangles $\Delta\mathbf{p}_0\mathbf{p}_1\mathbf{q}_0$ and $\Delta\mathbf{p}_0\mathbf{p}_1\mathbf{q}_1$, respectively. Though this choice works well for most examples, it may still introduce false detected discontinuity edges when some triangles or small local flat regions have been over tessellated into more triangles on the same planes. To choose proper neighbor vertices for an edge $\mathbf{p}_0\mathbf{p}_1$ which may have over tessellated neighbor triangles, we check local shape of the mesh near vertex $\mathbf{q}_0$ or $\mathbf{q}_1$ and extend the vertices along some edges if local over tessellation has been detected.

We check vertex $\mathbf{q}_0$ on triangle $\Delta\mathbf{p}_0\mathbf{p}_1\mathbf{q}_0$ as follows, vertex $\mathbf{q}_1$ on triangle $\Delta\mathbf{p}_0\mathbf{p}_1\mathbf{q}_1$ can be checked in the same way. Without loss of generality, we assume the lengths of two edges abutting vertex $\mathbf{q}_0$ satisfy $\|\mathbf{p}_0 - \mathbf{q}_0\| < \|\mathbf{p}_1 - \mathbf{q}_0\|$; see Fig. 3. Let $\theta_{p_0q_0}$ and $\theta_{p_1q_0}$ be the dihedral angles at the two edges, respectively. Assume $\mathbf{q}_a$ is a 1-ring neighbor vertex to $\mathbf{q}_0$ which is computed by $\mathbf{q}_a = \arg\min_{\mathbf{q}\in N_1(\mathbf{q}_0),\mathbf{q}\ne\mathbf{p}_0,\mathbf{p}_1}\{|\theta_{q_0q} - \theta_{p_0q_0}|\}$. Opposite vertex $\mathbf{q}_0$ will be extended to a new position $\mathbf{q}_a$ if the following three conditions hold simultaneously.

1. The dihedral angles satisfy inequalities $|\theta_{p_0q_0}| > \varepsilon_0$ and $|\theta_{p_1q_0}| < \varepsilon_0$;
2. The acute angle ϕ between vectors $\mathbf{p}_0\mathbf{q}_0$ and $\mathbf{p}_1\mathbf{q}_0$ is large enough such as $\phi > \frac{\pi}{3}$;
3. The lines $\mathbf{p}_0\mathbf{q}_0$ and $\mathbf{q}_0\mathbf{q}_a$ are nearly parallel, but the normal vectors $\mathbf{n}_{q_0}$ and $\mathbf{n}_{q_a}$ do not. Practically, the

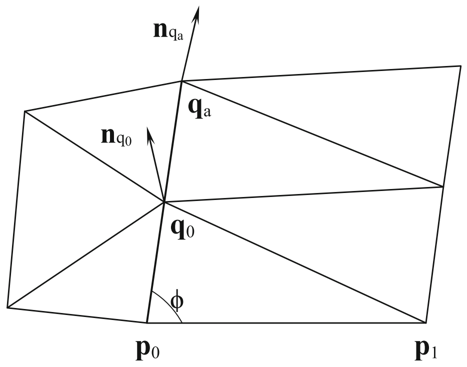


Fig. 3 Choosing opposite vertices for a mesh edge

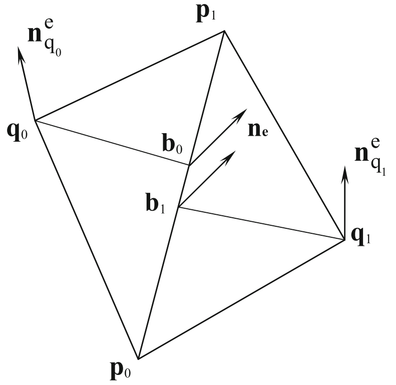


Fig. 4 Discontinuity measure computation for a mesh edge

vectors should satisfy $\frac{\mathbf{q}_a-\mathbf{q}_0}{\|\mathbf{q}_a-\mathbf{q}_0\|}\cdot\frac{\mathbf{q}_0-\mathbf{p}_0}{\|\mathbf{q}_0-\mathbf{p}_0\|} > 1-\varepsilon_0$ and $\mathbf{n}_{q_0}\cdot\mathbf{n}_{q_a} < 1-\varepsilon_0$.

In the following we will still denote the opposite vertices as $\mathbf{q}_0$ or $\mathbf{q}_1$ even they are the extended ones.

4.1.2 Robust and monotone curvatures

Assume that $\mathbf{p}_0\mathbf{p}_1$ is an arbitrary edge on a triangular mesh with two opposite vertices $\mathbf{q}_0$, $\mathbf{q}_1$ on two sides of the edge. Let $\mathbf{b}_0$ and $\mathbf{b}_1$ be the perpendicular feet of $\mathbf{q}_0$ and $\mathbf{q}_1$ onto the edge, respectively; see Fig. 4. We will define the discontinuity measures at edge $\mathbf{p}_0\mathbf{p}_1$ by computing normal curvatures at $\mathbf{q}_0$, $\mathbf{q}_1$, $\mathbf{b}_0$ and $\mathbf{b}_1$, all in a direction perpendicular to the edge.

Let $\mathbf{n}_e$, $\mathbf{n}^e_{q_0}$ and $\mathbf{n}^e_{q_1}$ be the normal vectors at the edge $\mathbf{p}_0\mathbf{p}_1$ or at two opposite vertices, respectively. The normal curvature across the edge or at their opposite vertices can be computed by Eq. (5). However, if the triangle mesh is highly irregular or have non-uniform triangulations, the computed curvatures may suffer even minor data noise when the chord $\|\mathbf{b}_0-\mathbf{q}_0\|$ is very short. We then modify Eq. (5) for robust curvature estimation using a filtered chord length. Let $l_\mathrm{b}=0.1l_\mathrm{m}$ be a lower bound of chord lengths, we replace $\|\mathbf{b}_0-\mathbf{q}_0\|$ in Eq. (5) by $d_0=\max\{\|\mathbf{q}_0-\mathbf{b}_0\|, l_\mathrm{b}\}$. By this refinement, too small chord length will be increased, and the normal curvatures at $\mathbf{b}_0$, $\mathbf{q}_0$ are then computed by

$$k_{b_0}=\frac{-2\bar{\mathbf{d}}_0\cdot\mathbf{n}_e}{d_0},\quad k_{q_0}=\frac{2\bar{\mathbf{d}}_0\cdot\mathbf{n}^e_{q_0}}{d_0}, \tag{9}$$

where $\bar{\mathbf{d}}_0=\frac{\mathbf{q}_0-\mathbf{b}_0}{\|\mathbf{q}_0-\mathbf{b}_0\|}$. Similarly, the normal curvatures at $\mathbf{b}_1$ and $\mathbf{q}_1$ can be obtained as

$$k_{b_1}=\frac{-2\bar{\mathbf{d}}_1\cdot\mathbf{n}_e}{d_1},\quad k_{q_1}=\frac{2\bar{\mathbf{d}}_1\cdot\mathbf{n}^e_{q_1}}{d_1}, \tag{10}$$

where $d_1=\max\{\|\mathbf{q}_1-\mathbf{b}_1\|, l_b\}$ and $\bar{\mathbf{d}}_1=\frac{\mathbf{q}_1-\mathbf{b}_1}{\|\mathbf{q}_1-\mathbf{b}_1\|}$.

Motivated by curve completion using spiral curves [11], we measure curvature changing rate at one side of an edge by assuming that the discrete curvatures are sampled from a spiral curve which also lies in a tolerance to the mesh. When the points $\mathbf{q}_0$, $\mathbf{b}_0$ and normal vectors $\mathbf{n}^e_{q_0}$, $\mathbf{n}_e$ all lie in a plane perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$, there exists a spiral that interpolates the boundary data in the plane [24]. Moreover, the interpolating spiral curve also lies in a region bounded by two circular arcs that interpolates two end points and either of the end normals; see Fig. 5a.

If $H_\mathrm{l}\equiv H(\mathbf{b}_0, \mathbf{n}_e, \mathbf{q}_0) < \tau_\mathrm{m}$, we regard the height of the interpolating spiral to the left side of edge $\mathbf{p}_0\mathbf{p}_1$ lying in the tolerance and choose k_{b_0} and k_{q_0} as the sampled curvatures of a spiral curve. If $H_l > \tau_\mathrm{m}$, the tolerance constrained interpolating curve should have a local minimum curvature value and the spiral segment can be a part of the interpolating curve; see Fig. 5b. We re-estimate end curvatures of a spiral segment when the height H_l or the dihedral angle at the edge is a large value. The discrete curvatures of a spiral segment at the left side of edge $\mathbf{p}_0\mathbf{p}_1$ are chosen as follows

$$\begin{cases} k^\mathrm{b}_\mathrm{l}=k_{b_0}, k^\mathrm{e}_\mathrm{l}=k_{q_0} & \text{if } (H_\mathrm{l}\le\tau_\mathrm{m})\text{and}(|\theta_e|<\theta_L) \\ k^\mathrm{b}_\mathrm{l}=\frac{-2\bar{\mathbf{d}}_0\cdot\mathbf{n}_e}{\mathrm{l}_\mathrm{b}}, k^\mathrm{e}_\mathrm{l}=0 & \text{otherwise} \end{cases} \tag{11}$$

We modify the end curvatures as of a local spiral segment when the height of an osculating arc exceeds the tolerance by experiment. Similarly, the normal curvatures at the right side of edge $\mathbf{p}_0\mathbf{p}_1$ are obtained as

$$\begin{cases} k^\mathrm{b}_\mathrm{r}=k_{\mathrm{b}_1}, k^\mathrm{e}_\mathrm{r}=k_{q_1} & \text{if } (H_\mathrm{r}\le\tau_\mathrm{m})\text{and}(|\theta_e|<\theta_L) \\ k^\mathrm{b}_\mathrm{r}=\frac{-2\bar{\mathbf{d}}_1\cdot\mathbf{n}_e}{l_\mathrm{b}}, k^\mathrm{e}_\mathrm{r}=0 & \text{otherwise} \end{cases} \tag{12}$$

4.1.3 Defining the discontinuity measures

Now we have spiral based sampling curvatures k^b_l and k^e_l at the left side of edge $\mathbf{p}_0\mathbf{p}_1$ and spiral based sampling curvatures k^b_r and k^e_r at the right. Edge $\mathbf{p}_0\mathbf{p}_1$ is a C^1 discontinuity joint edge if inequalities $k^\mathrm{b}_\mathrm{l}\gg k^\mathrm{e}_\mathrm{l}$ and $k^\mathrm{b}_\mathrm{r}\gg k^\mathrm{e}_\mathrm{r}$ hold. If either of the mentioned inequalities holds or the two curvatures k^b_l and k^b_r differ greatly, the edge is probably a C^2 discontinuity joint edge.

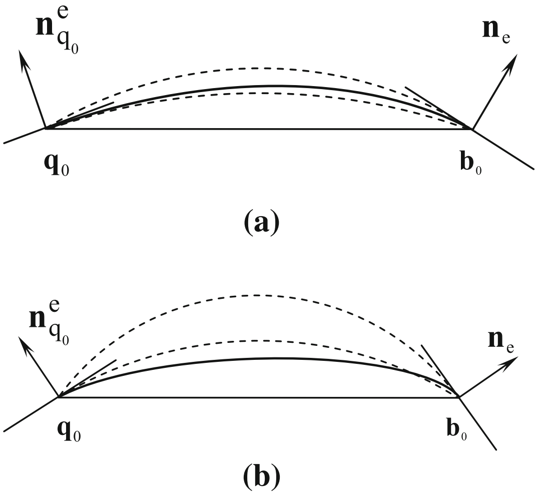

$\mathbf{n}^{\mathrm{e}}_{\mathrm{q}_0}$
$\mathbf{n}_{\mathrm{e}}$
$\mathbf{q}_0$
$\mathbf{b}_0$
(a)
$\mathbf{n}^{\mathrm{e}}_{\mathrm{q}_0}$
$\mathbf{n}_{\mathrm{e}}$
$\mathbf{q}_0$
$\mathbf{b}_0$
(b)

Fig. 5 Curvature computation using tolerance constrained curve interpolation

Instead of using curvature differences, we measure changing rates of curvatures using ratios of curvatures. The curvature ratios are scale invariant and can be used for detection of joint edges on high curvature regions and low curvature regions as well. The definition of curvature ratios requires that each pair of curvatures have a same sign. To achieve such a goal, we sample a midpoint curvature of the interpolating spiral at the left side of edge $\mathbf{p}_0\mathbf{p}_1$ as follows

$$k_{\mathrm{l}}^{\mathrm{m}} = \begin{cases} \frac{1}{2}(k_{\mathrm{l}}^{\mathrm{b}} + k_{\mathrm{l}}^{\mathrm{e}}) & \text{if } (k_{\mathrm{l}}^{\mathrm{b}} k_{\mathrm{l}}^{\mathrm{e}} > 0) \\ \frac{\mathrm{sgn}(k_{\mathrm{l}}^{\mathrm{b}})}{2} \max\{|k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{l}}^{\mathrm{e}}|\} & \text{otherwise} \end{cases} \tag{13}$$

where the function sgn() equals 1, −1 or 0 depending on the sign of a real number. The new sampled curvature of the right interpolating curve is given by

$$k_{\mathrm{r}}^{\mathrm{m}} = \begin{cases} \frac{1}{2}(k_{\mathrm{r}}^{\mathrm{b}} + k_{\mathrm{r}}^{\mathrm{e}}) & \text{if } (k_{\mathrm{r}}^{\mathrm{b}} k_{\mathrm{r}}^{\mathrm{e}} > 0) \\ \frac{\mathrm{sgn}(k_r^b)}{2} \max\left\{\frac{|k_{\mathrm{r}}^{\mathrm{b}}|}{2}, \frac{|k_{\mathrm{r}}^{\mathrm{e}}|}{2}\right\} & \text{otherwise} \end{cases} \tag{14}$$

Based on the sampled curvatures, we define the curvature changing rates at either side or across the edge $\mathbf{p}_0\mathbf{p}_1$ as follows

$$\mu_e^- = \frac{|k_{\mathrm{l}}^{\mathrm{b}}| + \varepsilon}{|k_{\mathrm{l}}^{\mathrm{m}}| + \varepsilon}, \tag{15}$$

$$\mu_e^+ = \frac{|k_{\mathrm{r}}^{\mathrm{b}}| + \varepsilon}{|k_{\mathrm{r}}^{\mathrm{m}}| + \varepsilon}, \tag{16}$$

$$\mu_e^c = \frac{\max\{|k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{b}}|\} + |k_{\mathrm{l}}^{\mathrm{b}} - k_{\mathrm{r}}^{\mathrm{b}}| + \varepsilon}{\max\{|k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{b}}|\} + \varepsilon}, \tag{17}$$

where $\varepsilon > 0$ is a number that can keep the denominator from being zero or reduce the influences of data noise.

From the definitions of the discontinuity measures in Eqs. (15–17), we can easily draw several interesting properties of the measures:

1. $\lim_{k_{\mathrm{l}}^{\mathrm{e}} \to k_{\mathrm{l}}^{\mathrm{b}}} \mu_e^- = \lim_{k_{\mathrm{r}}^{\mathrm{e}} \to k_{\mathrm{r}}^{\mathrm{b}}} \mu_e^+ = \lim_{k_{\mathrm{l}}^{\mathrm{b}} \to k_{\mathrm{r}}^{\mathrm{b}}} \mu_e^c = 1.$
2. If $\max\{|k_{\mathrm{l}}^{\mathrm{e}}|, |k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{e}}|\} \ll \varepsilon$, we have $\mu_e^- \approx 1$, $\mu_e^+ \approx 1$ and $\mu_e^c \approx 1$.
3. If $|k_{\mathrm{l}}^{\mathrm{b}}| \gg \varepsilon$ and $|k_{\mathrm{l}}^{\mathrm{b}}| \gg |k_{\mathrm{l}}^{\mathrm{e}}|$, we have $\mu_e^- \approx 2$.
4. If $|k_{\mathrm{r}}^{\mathrm{b}}| \gg \varepsilon$ and $|k_{\mathrm{r}}^{\mathrm{b}}| \gg |k_{\mathrm{r}}^{\mathrm{e}}|$, we have $\mu_e^+ \approx 2$.
5. If $|k_{\mathrm{l}}^{\mathrm{b}}| \gg |k_{\mathrm{r}}^{\mathrm{b}}|$ or $|k_{\mathrm{r}}^{\mathrm{b}}| \gg |k_{\mathrm{l}}^{\mathrm{b}}|$, we have $\mu_e^c \approx 2$.
6. If $k_{\mathrm{l}}^{\mathrm{b}} = -k_{\mathrm{r}}^{\mathrm{b}}$ and $|k_{\mathrm{r}}^{\mathrm{b}}| \gg \varepsilon$, we have $\mu_e^c \approx 3$.

Property 1 means that the one-sided discontinuity measures for edges sampled from curvature continuous surfaces are close to constant 1. From property 2 we know that the influences of data noise can be reduced greatly by choosing a proper value for the parameter ε. Properties 3 and 4 show that rapid change of curvature on one side of a joint edge can lead to a large discontinuity measure. Properties 5 and 6 imply that large jump of curvatures across an edge may lead to a large discontinuity measure.

For a given threshold μ_{T}, if $\mu_e^- > \mu_{\mathrm{T}} > 1$ and $\mu_e^+ > \mu_{\mathrm{T}} > 1$, the normal curvatures at the either side of edge $\mathbf{p}_0\mathbf{p}_1$ are greater than the normal curvatures at their opposite vertices, the edge is probably a C^1 discontinuity joint edge. We then measure the normal discontinuity across edges by

$$\mu_e^1 = \min\{\mu_e^+, \mu_e^-\}.$$

If one of measures μ_e^-, μ_e^+ or μ_e^c is much >1 (but still <3), the discrete curvatures either jump suddenly across the edge $\mathbf{p}_0\mathbf{p}_1$ or decrease rapidly at one side of the edge. In this case, the edge is a C^2 discontinuity joint edge. The curvature discontinuity at an edge can be measured by

$$\mu_e^2 = \max\{\mu_e^-, \mu_e^+, \mu_e^c\}.$$

4.2 Crease directions across edges and shape aware discontinuity measures

4.2.1 *Crease directions across mesh edges*

An efficient way to estimate crease directions at mesh vertices is by the normal voting technique [16]. We modify the technique a little for computing crease directions at vertices first, and then compute crease directions across mesh edges based on the piecewise smoothness assumption of the surfaces.

Assume $\mathbf{n}_i$ is the normal vector and $\mathbf{c}_i$ is the center of a triangle incident to vertex $\mathbf{q}$, $\mathbf{n}_q$ is the normal vector at vertex $\mathbf{q}$ (Fig. 6a), the voted normal by $\mathbf{n}_i$ is computed by

$$\mathbf{n}_i' = \frac{\mathbf{n}_i - 2(\mathbf{r}_i \cdot \mathbf{n}_i)\mathbf{r}_i + \mathbf{n}_q}{\|\mathbf{n}_i - 2(\mathbf{r}_i \cdot \mathbf{n}_i)\mathbf{r}_i + \mathbf{n}_q\|},$$

where $\mathbf{r}_i = \frac{\mathbf{q}-\mathbf{c}_i}{\|\mathbf{q}-\mathbf{c}_i\|}$. We have modified the previous approach by adding $\mathbf{n}_q$ in each voted normal. This modification can increase the robustness of crease direction estimation even when the vertex is lying on a joint line between two planes with opposite normals.

By the voted normal vectors, the tensor matrix at vertex $\mathbf{q}$ is obtained as $T_q = \sum_{i \in N_1(q)} \alpha_i \mathbf{n}'_i \mathbf{n}'^t_i$, where t means the transpose of a column vector. It is clear that the matrix T_q is symmetric and it can be decomposed into $T_q = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^t + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^t + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^t$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the eigenvalues and $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ are the corresponding eigenvectors of the matrix.

The crease direction at vertex $\mathbf{q}$ is then chosen as $\mathbf{t}_q = \mathbf{e}_3$. The crease direction across a mesh edge should be parallel to the edge when the edge is lying on the C^1 or C^2 discontinuity joint line between two surfaces. For a triangular mesh with no prior knowledge of joint lines, we set $\mathbf{t}_e = \mathbf{u}_e$ for an edge $\mathbf{p}_0\mathbf{p}_1$ if it is a potential C^1 or C^2 discontinuity joint edge judged by the following two simple rules:

1. If $H(\mathbf{b}_0, \mathbf{n}_e, \mathbf{q}_0) > \tau_m$ or $H(\mathbf{b}_1, \mathbf{n}_e, \mathbf{q}_1) > \tau_m$, edge $\mathbf{p}_0\mathbf{p}_1$ is a potential sharp edge and it will be dealt as a C^1 or C^2 discontinuity joint edge.
2. If edge $\mathbf{p}_0\mathbf{p}_1$ is a short edge of a thin triangle, especially, when $\max\{\|\mathbf{q}_0 - \mathbf{b}_0\|, \|\mathbf{q}_1 - \mathbf{b}_1\|\} > 10\|\mathbf{p}_1 - \mathbf{p}_0\|$, the edge is probably lying on a joint line between two different surfaces.

If none of above two conditions holds, we treat edge $\mathbf{p}_0\mathbf{p}_1$ as lying on a local smooth region (Fig. 6b). Assume that $\mathbf{t}_0$ and $\mathbf{t}_1$ are the two unit crease directions at two end vertices of the edge $\mathbf{p}_0\mathbf{p}_1$, the crease direction for the edge is chosen by default as $\mathbf{t}'_e = \frac{\mathbf{t}_0+\mathbf{t}_1}{\|\mathbf{t}_0+\mathbf{t}_1\|}$ when $\|\mathbf{t}_0 + \mathbf{t}_1\| > \|\mathbf{t}_0 - \mathbf{t}_1\|$ or $\mathbf{t}'_e = \frac{\mathbf{t}_0-\mathbf{t}_1}{\|\mathbf{t}_0-\mathbf{t}_1\|}$ otherwise. If an edge is connected to other discontinuity joint edges at one of its end vertices, the crease direction across the edge can be chosen as the crease direction at the other end vertex. Assume that the acute angles between vectors $\mathbf{t}_0$, $\mathbf{t}_1$ or $\mathbf{t}'_e$ with direction $\mathbf{p}_1 - \mathbf{p}_0$ are ϕ_0, ϕ_1 or ϕ'_e, respectively, the crease direction across the edge is chosen as follows

$$\mathbf{t}_e = \begin{cases} \mathbf{t}'_e, & \text{by default} \\ \mathbf{t}_0, & \text{if } \phi_0 < \min\{\phi'_e, \phi_1, \phi_E\} \\ \mathbf{t}_1, & \text{if } \phi_1 < \min\{\phi'_e, \phi_0, \phi_E\} \end{cases} \tag{18}$$

where ϕ_E is a threshold characterizing the smoothness of discrete crease directions.

4.2.2 Shape aware discontinuity measures for mesh edges

We compute shape aware discontinuity measures for mesh edges by choosing parameter ε for Eqs. (15–17) based on local shape analysis of the triangular mesh. In particular, the parameter should have low values for edges lying on boundaries of flat surfaces or joint lines between two curved surfaces but have larger values for edges lying on smooth surface regions.

The angle ϕ_e between edge direction $\mathbf{u}_e$ and crease direction $\mathbf{t}_e$ across edge $\mathbf{p}_0\mathbf{p}_1$ can be obtained by $\phi_e = \cos^{-1}(|\mathbf{t}_e \cdot \mathbf{u}_e|)$; see Fig. 7. From the assumption that the direction of a C^1 discontinuity joint edge is approximately parallel to the crease direction, the smaller the angle ϕ_e the smaller the value of ε should be for one-sided discontinuity measures μ_e^- and μ_e^+.

If a mesh is obtained by the tessellation of a piecewise smooth surface with a small error bound, a C^1 or C^2 discontinuity joint edge $\mathbf{p}_0\mathbf{p}_1$ and the normal vectors $\mathbf{n}_0$, $\mathbf{n}_1$ at the ends of the edge usually lie on a same plane exactly or approximately. To measure the coplanarity of vectors $\mathbf{n}_0$, $\mathbf{n}_1$ and $\mathbf{u}_e$, we compute the angle ϕ_n between two planes

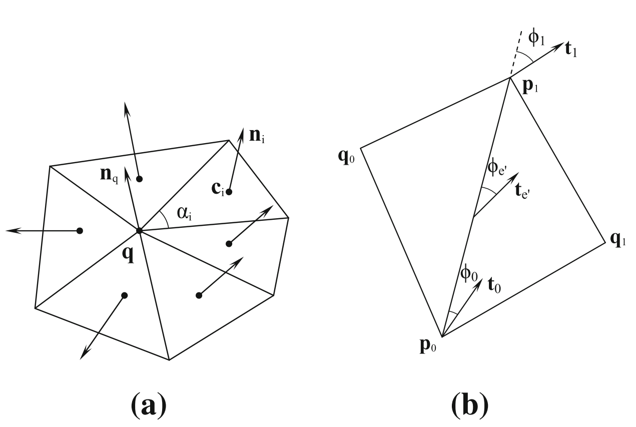


Fig. 6 Estimating the crease direction at **a** vertices or **b** across the edge

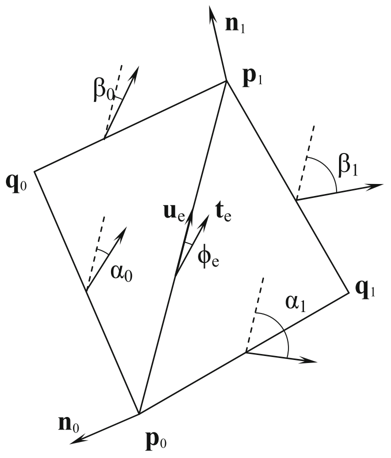


Fig. 7 Computing local shape parameters for the discontinuity measures of a mesh edge

spanned by vectors $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{n}_0$ or by vectors $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{n}_1$, respectively. With simple computation, we have $\phi_n = \cos^{-1}(|\frac{\mathbf{n}_0 \times \mathbf{u}_e}{\|\mathbf{n}_0 \times \mathbf{u}_e\|} \cdot \frac{\mathbf{n}_1 \times \mathbf{u}_e}{\|\mathbf{n}_1 \times \mathbf{u}_e\|}|)$. If ϕ_n is a small value, we should choose small value of ε for the discontinuity measures defined by Eqs. (15–17).

To distinguish boundary edges of flat surfaces or the joint edges between two curved surfaces from other roughly tessellated edges in smooth regions, we compute local roughness for mesh edges. Assume α_0, β_0, α_1 and β_1 are the angles between edge $\mathbf{p}_0\mathbf{p}_1$ and the principal directions at other four edges of two sharing triangles of edge $\mathbf{p}_0\mathbf{p}_1$ (Fig. 7). We compute the weighted angles for the two sharing triangles by

$$\bar{\theta}_0 = \cos^2\alpha_0|\theta_{p_0q_0}| + \cos^2\beta_0|\theta_{p_1q_0}|,$$

$$\bar{\theta}_1 = \cos^2\alpha_1|\theta_{p_0q_1}| + \cos^2\beta_1|\theta_{p_1q_1}|,$$

where $\theta_{p_0q_0}$, $\theta_{p_1q_0}$, $\theta_{p_0q_1}$ and $\theta_{p_1q_1}$ are the dihedral angles at four edges of the two sharing triangles. We define the local roughness of edge $\mathbf{p}_0\mathbf{p}_1$ as $\rho_e = \frac{\min\{\bar{\theta}_0, \bar{\theta}_1\}}{|\theta_{p_0p_1}|}$. If one neighbor triangle lies on a local flat region or the minor principal directions at two edges (other than $\mathbf{p}_0\mathbf{p}_1$) are perpendicular to edge $\mathbf{p}_0\mathbf{p}_1$, either $\bar{\theta}_0$ or $\bar{\theta}_1$ is a low value and the edge is probably a boundary edge of a flat surface or the joint edge between two curved surfaces. If ρ_e is a large number, both $\bar{\theta}_0$ and $\bar{\theta}_1$ are large enough as compared with $\theta_{p_0p_1}$, the edge $\mathbf{p}_0\mathbf{p}_1$ is probably a rough tessellated edge in a smooth surface region.

To overcome the influences of data noise on discontinuity measures further, we compute the average curvature across selected edges on low curved surface regions. For an arbitrary edge $\mathbf{p}_0\mathbf{p}_1$ with two opposite vertices $\mathbf{q}_0$ and $\mathbf{q}_1$ on two sharing triangles, $\mathbf{b}_0$ and $\mathbf{b}_1$ are the perpendicular feet of $\mathbf{q}_0$ and $\mathbf{q}_1$ on the edge. The curvature across the edge $\mathbf{p}_0\mathbf{p}_1$ is computed as $k_e = 2\sin(\frac{\theta_e}{2})/l_e$, where $l_e = \max\{l_a, 0.1 l_{\mathrm{m}}\}$ and $l_a = (\|\mathbf{q}_0 - \mathbf{b}_0\| + \|\mathbf{q}_1 - \mathbf{b}_1\|)/2$. The average curvature of selected edges is then computed as $k_{\mathrm{ave}} = \frac{1}{n}\sum_{\theta_l < |\theta_e| < \theta_T} |k_e|$, where θ_{T} is a threshold and n is the total number of edges of which the dihedral angles satisfy $\theta_l < |\theta_e| < \theta_{\mathrm{T}}$.

Based on the above analysis, we choose two shape parameters ε_1 and ε_2 heuristically as follows

$$\varepsilon_1 = (\sin^2\phi_e + \sin^2\phi_n)k_{max} + k_\varepsilon, \tag{19}$$

$$\varepsilon_2 = 2\sin\phi_n k_{max} + k_\varepsilon, \tag{20}$$

where $k_{\mathrm{max}} = \max\{|k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{b}}|\}$, $k_{\mathrm{min}} = \min\{|k_{\mathrm{l}}^{\mathrm{b}}|, |k_{\mathrm{r}}^{\mathrm{b}}|\}$ and $k_\varepsilon = \rho_e k_{\mathrm{min}} + 0.05k_{\mathrm{ave}} + 10^{-5}$. We choose the shape parameter $\varepsilon = \varepsilon_1$ for Eqs. (15) and (16) and choose $\varepsilon = \varepsilon_2$ for Eq. (17). The main steps for the computation of C^1 or C^2 discontinuity measures for mesh edges are summarized in Algorithm 1.

Algorithm 1. Discontinuity measure computation

```
input: A triangular mesh
output: discontinuity measures for all edges
1. Compute $\mathbf{n}_e$ and $\theta_e$ for each edge $\mathbf{p}_0\mathbf{p}_1$;
2. Compute $\tau_m$ and $k_{ave}$ for the mesh;
3. Compute crease directions for all vertices;
4. for each edge $\mathbf{p}_0\mathbf{p}_1$
     choose vertices $\mathbf{q}_0$, $\mathbf{q}_1$ on two sharing triangles;
     Compute perpendicular feet $\mathbf{b}_0$ and $\mathbf{b}_1$;
     Compute $H_0 = H(\mathbf{b}_0, \mathbf{n}_e, \mathbf{q}_0)$ and $H_1 = H(\mathbf{b}_1, \mathbf{n}_e, \mathbf{q}_1)$;
     if ($H_0 > \tau_m$ or $H_1 > \tau_m$) set $\mathbf{t}_e = \mathbf{u}_e$;
     else Compute $r_0 = \frac{\|\mathbf{q}_0 - \mathbf{b}_0\|}{\|\mathbf{p}_1 - \mathbf{p}_0\|}$ and $r_1 = \frac{\|\mathbf{q}_1 - \mathbf{b}_1\|}{\|\mathbf{p}_1 - \mathbf{p}_0\|}$;
         if ($r_0 > 10$ or $r_1 > 10$) set $\mathbf{t}_e = \mathbf{u}_e$;
         else compute $\mathbf{t}_e$ by Equation (18);
5. for each edge $\mathbf{p}_0\mathbf{p}_1$
     choose (extended) opposite vertices $\mathbf{q}_0$, $\mathbf{q}_1$;
     Compute normal vectors $\mathbf{n}_{q0}^e$ and $\mathbf{n}_{q1}^e$;
     Compute perpendicular feet $\mathbf{b}_0$ and $\mathbf{b}_1$;
     Compute $k_{b_0}$, $k_{q0}$, $k_{b_1}$, $k_{q1}$;
     Compute parameters $\varepsilon_1$ and $\varepsilon_2$ for the edge;
     Compute $\mu_e^-$, $\mu_e^+$, $\mu_e^c$ by Eqn.s (13-17).
```

5 The discontinuity detection algorithm

Based on the computed discontinuity measures for mesh edges, we present a practical algorithm for C^1 or C^2 discontinuity joint edge detection. An algorithm for smooth joint line generation from initially detected joint edges will also be presented.

5.1 Initial joint line detection

When the C^1 or C^2 discontinuity measures have been obtained for all mesh edges, the initial joint edges can just be selected using a given threshold μ_{T}. However, some obvious joint edges may still be missed in this way because of inaccurate discontinuity measures. We then refine the chosen edges a little to reduce false detected edges and pick missed joint edges as much as possible. The refinement is based as the following rules:

1. No three discontinuity joint edges lie on a same triangle;
2. Every C^1 or C^2 discontinuity joint edge lies on a smooth joint line;
3. The edges with large dihedral angles are possibly C^1 or C^2 discontinuity joint edges.

From rule 1 we can delete the edge with minimum discontinuity measure when the discontinuity measures of three edges of a triangle are all larger than the threshold μ_{T}. By rule 2 or rule 3 we can pick some missed joint edges by extending the current joint lines.

The extension of a joint line on a triangular mesh is based on the choice of smooth connected edge at an end vertex of the line. Assume $\mathbf{v}_0$ and $\mathbf{v}_1$ are the two vertices of an edge $\mathbf{e}$ on a triangle mesh, the smooth joint edge of edge $\mathbf{e}$ at vertex $\mathbf{v}_1$ is chosen from the 1-ring neighborhood of vertex $\mathbf{v}_1$. The other end vertex of the smooth connected edge $\mathbf{x}$ to edge $\mathbf{e}$ is picked by

$$\mathbf{v}_s = \arg\max_{\mathbf{v}_a \in N_1(\mathbf{v}_1)} \left\{ \frac{\mathbf{v}_1 - \mathbf{v}_0}{\|\mathbf{v}_1 - \mathbf{v}_0\|} \cdot \frac{\mathbf{v}_a - \mathbf{v}_1}{\|\mathbf{v}_a - \mathbf{v}_1\|} \bar{\theta}_a \right\},$$

where $\bar{\theta}_a = |\theta_{\mathbf{v}_1\mathbf{v}_a}| + \pi$. When we obtain a smooth joint edge $\mathbf{x}$ to a given edge $\mathbf{e}$, we compute the turning angle ϕ between the projections of edges $\mathbf{e}$ and $\mathbf{x}$ onto the tangent plane that passes through vertex $\mathbf{v}_1$ with normal vector $\mathbf{n}_{v_1}$.

```
Algorithm 2. Initial joint line detection
input: A triangular mesh, dihedral angles and discontinuity
measures at edges and a threshold $\mu_T$
output: a set of detected joint edges
1. for each edge $\mathbf{e}$
     if ($\mu_e > \mu_T$) index($\mathbf{e}$)=joint_edge;
     else index($\mathbf{e}$)=no_index;
2. for each triangle with edges $\mathbf{e}_0$, $\mathbf{e}_1$ and $\mathbf{e}_2$
     if ($\mathbf{e}_0$, $\mathbf{e}_1$ and $\mathbf{e}_2$ are all indexed joint_edges)
          delete the joint edge with min$\{\mu_{e_0}, \mu_{e_1}, \mu_{e_2}\}$;
3. for each vertex $\mathbf{v}$
     $M(\mathbf{v})$=the number of joint edges connecting $\mathbf{v}$;
4. for each joint edge $\mathbf{e} = \mathbf{v}_0\mathbf{v}_1$
     if ($M(\mathbf{v}_0) == 1 \,||\, M(\mathbf{v}_1) == 1$)
          find extended edge $\mathbf{x} = \mathbf{v}_0\mathbf{v}_2$ or $\mathbf{x} = \mathbf{v}_1\mathbf{v}_2$;
          compute angle $\phi$ between $\mathbf{e}$ and $\mathbf{x}$;
          if ($\mu_e > (\mu_T - 0.05)$ && $M(\mathbf{v}_2) \leq 1$)
               index($\mathbf{x}$)=joint_edge;
          if ($|\theta_x| > \theta_T$ && $|\phi| < \phi_T$ && $M(\mathbf{v}_2) \leq 1$)
               index($\mathbf{x}$)=joint_edge;
     end if
```

For every end C^1 or C^2 discontinuity joint edge $\mathbf{e}$ we find the smooth joint edge $\mathbf{x}$ and add the edge $\mathbf{x}$ as a discontinuity joint edge if it satisfies one of the following conditions:

1. The discontinuity measure $\mu_x > \mu'_{\mathrm{T}} = \mu_{\mathrm{T}} - 0.05$ and the edge $\mathbf{x}$ is connected to at most one other C^1 or C^2 discontinuity joint edge at the other end;
2. The dihedral angle of edge $\mathbf{x}$ satisfies $|\theta_x| > \theta_T$, the turning angle $\phi < \phi_{\mathrm{T}}$ and the edge $\mathbf{x}$ is connected to at most one other C^1 or C^2 discontinuity joint edge.

Condition 1 implies that some discontinuity joint edges can be detected by using adaptive thresholds while by condition 2 we may pick some discontinuity joint edges that have large dihedral angles and small turning angles with known discontinuity joint edges. We restrict the new extended edge connecting to at most one another C^1 or C^2 discontinuity joint edge to avoid the generation of new branch joint lines. See Algorithm 2 for the main steps for initial joint line detection.

5.2 Smooth joint line generation

Based on the assumption that C^1 or C^2 discontinuity joint lines of piecewise smooth surfaces are usually smooth and there is no isolated or short branches on the joint lines, we present an algorithm to detect smooth joint lines from the initially detected joint edges. The algorithm consists of three main steps: one-edge-length-branch pruning, one-edge-length-gap filling and short branch pruning.

We delete an end discontinuity joint edge if one of the following conditions holds:

1. The end joint edge $\mathbf{e}$ is an isolated joint edge;
2. Two end joint edges $\mathbf{e}$ and $\mathbf{x}$ are connected and the dihedral angles of the two edges are less than θ_{T};
3. The end joint edge $\mathbf{e}$ is connected to a joint line with turning angle $\phi > \phi_{\mathrm{T}}$ at the joint point;
4. The end joint edge $\mathbf{e}$ is connected to two or more other joint lines and the smallest turning angle $\phi > \phi_{\mathrm{T}}$ at the joint point.

```
Algorithm 3. Smooth joint line generation
input: A mesh and initially detected joint edges
output: refined discontinuity joint edges
1. for each vertex $\mathbf{v}$
     $M(\mathbf{v})$=the number of joint edges connecting $\mathbf{v}$;
2. for each end joint edge $\mathbf{e} = \mathbf{v}_0\mathbf{v}_1$
     if ($M(\mathbf{v}_0) == 1$ && $M(\mathbf{v}_1) == 1$) delete $\mathbf{e}$;
     if ($M(\mathbf{v}_0) == 1$ && $M(\mathbf{v}_1) == 2$)
          find joint edge $\mathbf{x} = \mathbf{v}_1\mathbf{v}_2$;
          if ($M(\mathbf{v}_2) == 1 \&\& |\theta_e| < \theta_T \&\& |\theta_x| < \theta_T$)
               delete edges $\mathbf{e}$ and $\mathbf{x}$;
          if ($M(\mathbf{v}_2) > 1$)
               compute angle $\phi$ between $\mathbf{e}$ and $\mathbf{x}$;
               if ($\phi > \phi_T$) delete joint edge $\mathbf{e}$;
          end if
     end if
     if ($M(\mathbf{v}_0) == 1 \&\& M(\mathbf{v}_1) > 2$)
          find closest attached joint edge $\mathbf{x}$ at $\mathbf{v}_1$;
          compute unsigned angle $\phi$ between $\mathbf{e}$ and $\mathbf{x}$;
          if ($\phi > \phi_T$) delete joint edge $\mathbf{e}$;
     end if
3. for each end joint edge $\mathbf{e}_a$ with end vertex $\mathbf{v}_a$
     find straight extended edge $\mathbf{e} = \mathbf{v}_a\mathbf{v}_b$;
     if ($M(\mathbf{v}_b) \geq 1$) find extended joint edge $\mathbf{e}_b$;
          compute angles $\phi_a$, $\phi_b$, $\phi_e$;
          if ($M(\mathbf{v}_b) == 1$ && $\max\{|\phi_a|, |\phi_b|\} < \phi_T$)
               add $\mathbf{e}$ as a joint edge;
          if ($\max\{|\phi_a|, \min\{|\phi_b|, |\phi_e|\}\} < \phi_T$
               $\&\& M(\mathbf{v}_b) > 1$) add $\mathbf{e}$ as a joint edge;
     end if
4. for every end joint edge $\mathbf{e} = \mathbf{v}_0\mathbf{v}_1$
     if ($M(\mathbf{v}_1) > 2$) delete joint edge $\mathbf{e}$;
     if ($M(\mathbf{v}_1) == 2$)
          find attached joint edge $\mathbf{x} = \mathbf{v}_1\mathbf{v}_x$;
          if ($M(\mathbf{v}_x) > 2$) delete joint edges $\mathbf{e}$ and $\mathbf{x}$;
     end if
```

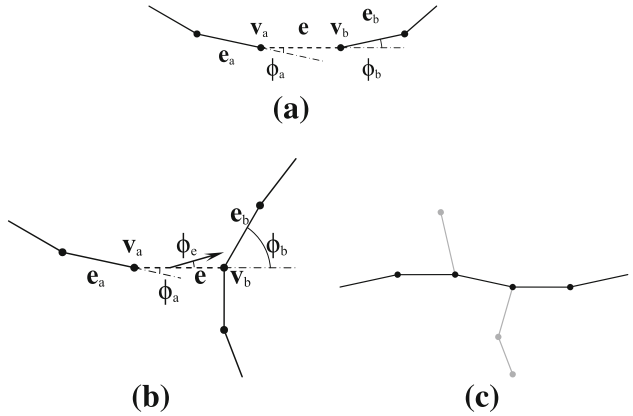


Fig. 8 One edge length gap filling and short branch pruning

We note that in case 2, both edge **e** and edge **x** can be deleted. Case 3 and case 4 imply that edge **e** turns rapidly from other joint lines and the edge will be deleted as a one-edge-length-branch.

The procedure for one-edge-length-gap filling is similar to joint edge extension in Sect. 5.1. If the smooth extended edge of a joint line is also connected to one or more other joint edges, the extended edge can be a missed gap edge of a smooth joint line or a missed joint edge from one joint line to other joint lines. Let $\mathbf{e} = \mathbf{v}_a\mathbf{v}_b$ be the smooth extended edge of $\mathbf{e}_a$ at $\mathbf{v}_a$, and $\mathbf{e}_b$ be the smooth extended joint edge of **e** at vertex $\mathbf{v}_b$; see Fig. 8a, b. We compute the turning angle ϕ_a between $\mathbf{e}_a$ and **e**, the turning angle ϕ_b between **e** and $\mathbf{e}_b$, or the angle ϕ_e between **e** and the crease direction $\mathbf{t}_e$ at **e** when $\mathbf{v}_b$ is the joint vertex of two or more joint edges. We add edge **e** as a new joint edge if one of the following conditions holds:

1. Both edges $\mathbf{e}_a$ and $\mathbf{e}_b$ are end joint edges and angles ϕ_a and ϕ_b are less than ϕ_T;
2. Edge $\mathbf{e}_b$ is not an end joint edge but angles ϕ_a and $\min\{\phi_b, \phi_e\}$ are less than ϕ_T.

We prune the short branches that have at most two edges as a third step for smooth joint line generation. Figure 8c shows two short branches which should be pruned by the smoothing procedure. The main steps for joint line smoothing are given in Algorithm 3.

5.3 Selection of parameters

As a triangular mesh can be the approximation or the tessellation of various (piecewise) smooth surfaces, the detection of joint lines on a surface mesh is not determined without proper assumptions. We detect the C^1 and C^2 discontinuity joint edges on a triangular mesh based on selection or computation of several parameters. The parameters can be grouped into three categories: fixed parameters, user chosen parameters and adaptively computed parameters for meshes or mesh edges.

The fixed parameters include:

ε_0: the threshold for characterizing whether two lines or vectors are parallel. In our experiments we choose $\varepsilon_0 = 0.001$.

ϕ_E: the threshold used for characterizing the smoothness of crease directions at mesh edges or at mesh vertices. We choose $\phi_E = \frac{\pi}{6}$ in our experiments.

θ_l, θ_L, θ_T: If the dihedral angle θ_e of an edge satisfies $|\theta_e| < \theta_l$, the edge is probably lying on a local flat region. If $|\theta_e| > \theta_L$, the edge is regarded as a C^1 discontinuity joint edge. If $|\theta_e| < \theta_\mathrm{T}$, it implies that the edge is lying on a local smooth surface region. We choose $\theta_l = 0.01, \theta_L = \frac{\pi}{4}$ and $\theta_\mathrm{T} = \frac{\pi}{10}$.

ϕ_T: this parameter is used as a threshold for characterizing the smoothness of a polygon on a surface mesh. In our experiments we choose $\phi_\mathrm{T} = \frac{\pi}{10}$.

The only parameter user can choose interactively is the threshold μ_T for C^1 or C^2 discontinuity measures. Based on the properties of the C^1 or C^2 discontinuity measures, the threshold μ_T can be selected in the interval (1,2]. Many more C^1 or C^2 discontinuity joint edges will be detected using a lower threshold and only salient discontinuity joint edges can be detected using a larger one. A default threshold $\mu_\mathrm{T} = 1.5$ can be used for automatic joint line detection which gives satisfying results for most examples.

The adaptively computed parameters include:

τ_m: the tolerance for construction of fitting curves and surfaces.

ε_1: the local shape parameter characterizing whether an edge is lying on the boundary curve of a flat surface or the C^1 discontinuity joint line between two curved surfaces.

ε_2: the local shape parameter characterizing whether an edge is lying on the C^2 discontinuity joint line between two surfaces.

Exact evaluation of these three parameters depends on the analytical representation of a piecewise smooth surface. By Eqs. (8, 19) and (20), we compute values for these parameters heuristically for triangle meshes.

6 Examples and comparisons

In this section we present several interesting examples to illustrate the results of C^1 or C^2 discontinuity joint edge detection. We have tested CAD-like meshes, surface meshes with re-triangulations as well as triangle meshes reconstructed from real data. To distinguish C^2 discontinuity edges

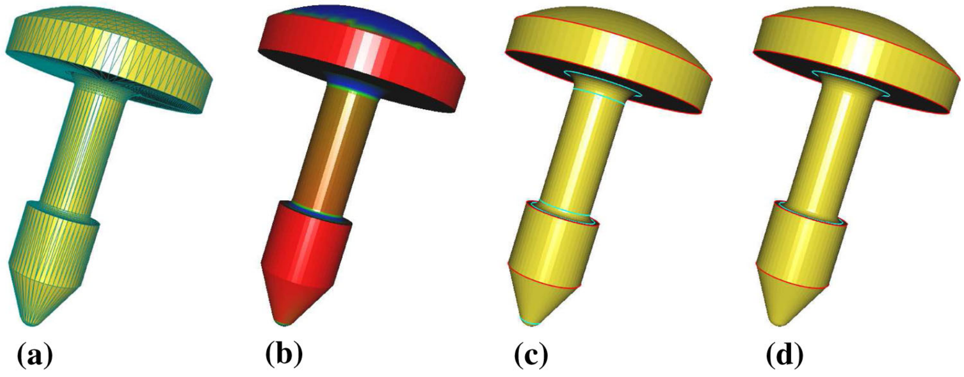


Fig. 9 C^1 and C^2 discontinuity joint line detection on a top shape model: **a** the triangular mesh; **b** the C^2 discontinuity measure; **c** the C^1 and C^2 discontinuity joint lines detected by the proposed algorithm (we choose ($\mu_T = 1.5$) in this and following examples); **d** the C^1 and C^2 discontinuity joint edges detected by Jiao's algorithm [8]

from C^1 discontinuity ones, we detect C^1 discontinuity edges using C^1 discontinuity measures first and detect C^2 discontinuity edges by computing C^2 discontinuity measures for the rest edges.

We have compared the results by our proposed C^1 discontinuity measure with the results by edge strength measure s given in [17] or by the sharpness indicator presented in [2]. The methods of edge strength measure and the sharpness indicator are either popular or among the latest ones for C^1 discontinuities detection. The comparisons of C^2 discontinuity edge detection by our proposed algorithm and by a recent method stated in [8] are also given.

First, we detect C^1 or C^2 discontinuity joint edges on a top shape model. The mesh illustrated in Fig. 9a is obtained by triangulating a piecewise smooth surface consisting of two spheres parts, three cylinders, one cone part and two blending surface patches, and there are totally 1,735 vertices and 3,466 triangles within the mesh. Except the joint lines between the cone and a small sphere patch or the boundary lines of two blending surfaces which are C^2 discontinuous joint lines, all other joint lines are C^1 discontinuous. Though the sizes and shapes of triangles on the mesh differ greatly, all C^1 and C^2 discontinuity joint lines have been detected correctly by our proposed procedure. Figure 9b, c illustrate the plot of discontinuity measures and the detected joint lines. According to Jiao and Bayyana's algorithm [8], only those C^2 discontinuity joint edges lying on planes may be detected. Several C^2 discontinuity joint lines have been missed by this method; see Fig. 9d.

Second, we compare the results of joint line detection on a casting model originally shown in Fig. 1a. Figure 1c, d show that the discontinuities across mesh edges can be measured properly by our proposed measures and almost all C^1 or C^2 discontinuity joint edges have been detected using the default threshold by our proposed algorithm. However, the joint lines cannot be detected satisfactorily by several existing methods. Figure 10a illustrates the detection result by Jiao and Bayyana's algorithm [8]. From the figure we can see that C^1 discontinuity joint edges have been detected correctly but C^2 discontinuity joint lines lying on non flat regions have been missed. When only C^1 discontinuity joint lines should be detected, both the sharpness indicator and the edge strength measure have introduced false detected joint edges in high curvature regions. See the red lines on the thin blending surfaces on top part of the model in Fig. 10b, c.

Third, we detect joint edges on a tessellated mesh that have noise. Figure 11a shows a tessellated mesh from a circulant equipment which is composed of several cylinders, cones and many parts of spheres. After the tessellation, a surface mesh consisting of 89,016 vertices and 178,040 triangles is obtained. Though the triangles have similar sizes, the vertices and edges in high curvature regions (like the thin cylinders) may still suffer noise. Since C^2 discontinuities are more sensitive to noise, we only detect C^1 discontinuities for this example. From Fig. 11b, c we can see that most C^1 discontinuity joint lines in either high curvature regions or low curvature regions have been detected successfully by our proposed C^1 discontinuity measure. As a comparison, the sharpness indicator and the edge strength measure have missed many salient C^1 discontinuity joint edges in low curvature regions and introduced false detected C^1 discontinuity joint edges in high curvature regions due to data noise. See the results in Fig. 11d, e.

Fourth, we detect C^1 discontinuity edges on a reconstructed cap model that has 7,513 vertices and 14,880 triangles; see Fig. 12a. The C^1 discontinuity measures are computed and plotted in Fig. 12b. From the figure we can see clearly that edges across which the surface normals change rapidly have higher values of C^1 discontinuity measures than other edges on smooth surface regions. The C^1 discontinuity joint lines on the surface mesh can be detected using the default discontinuity threshold by the proposed joint line detection algorithm; see the results in Fig. 12c. Figure 12d, e are the results detected by sharpness indicator or the edge strength measure, respectively. From these two figures we can see that both sharpness indicator and edge strength measure may introduce false

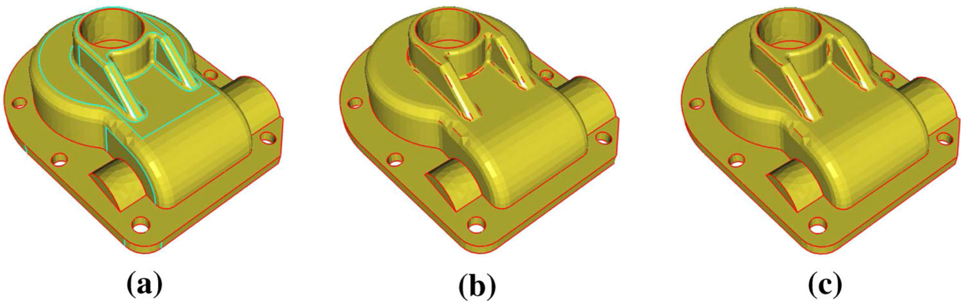


Fig. 10 Joint line detection on a casting model: **a** the C^1 and C^2 discontinuities detected by Jiao's algorithm; **b** the C^1 discontinuities detected using the sharpness indicator; **c** the C^1 discontinuities detected by the edge strength measure ($s > 0.1$)

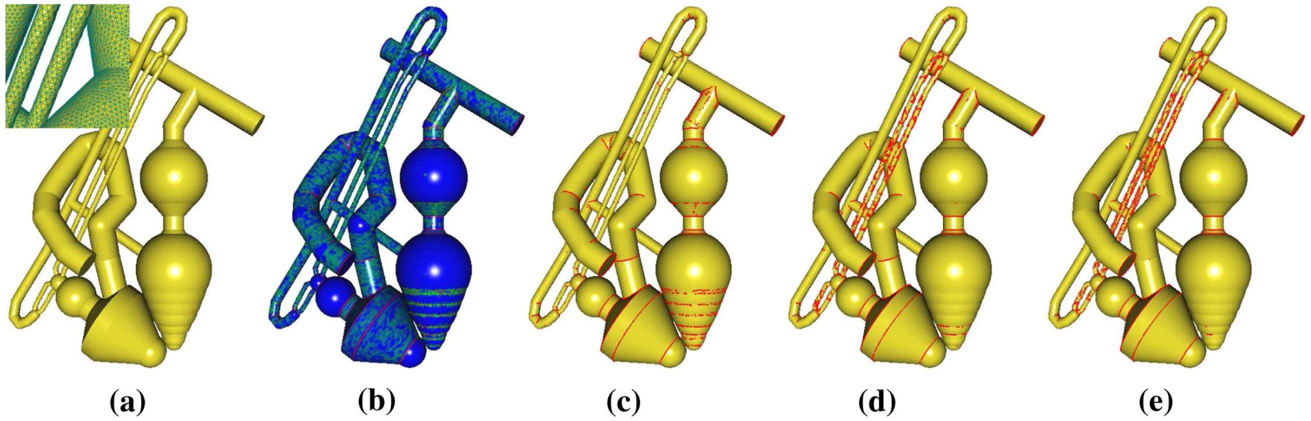


Fig. 11 C^1 discontinuity joint line detection on a circulant equipment model: **a** the triangular mesh; **b** the plot of C^1 discontinuity measure; **c** the C^1 discontinuity joint lines detected by the proposed algorithm; **d** the C^1 discontinuities detected using the sharpness indicator; **e** the C^1 discontinuities detected by the edge strength measure ($s > 0.05$)

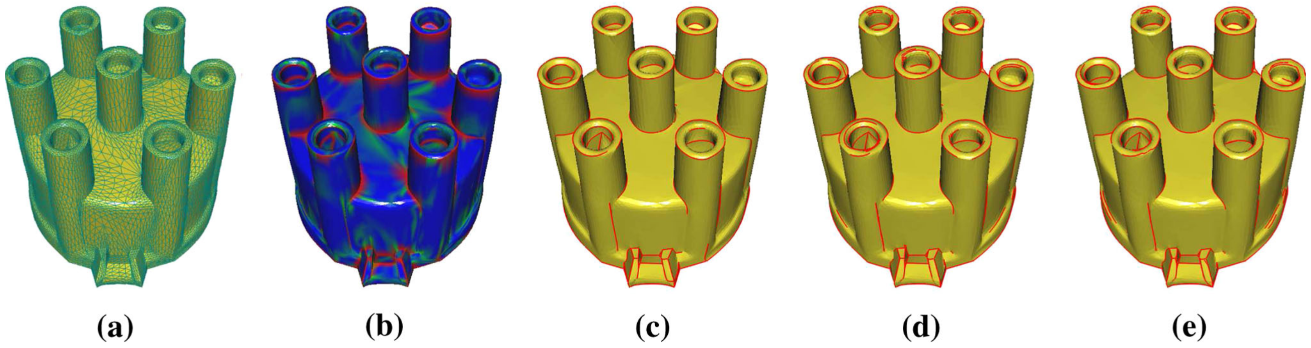


Fig. 12 C^1 discontinuity joint line detection on a cap model: **a** the triangular mesh; **b** the plot of C^1 discontinuity measure; **c** the C^1 discontinuity joint lines detected by the proposed algorithm; **d** the C^1 discontinuities detected using the sharpness indicator; **e** the C^1 discontinuities detected by the edge strength measure ($s > 0.05$)

C^1 discontinuity joint edges in smooth but high curvature regions. On another hand, some C^1 discontinuity joint edges shared by triangles with large or different sizes may be missed by these two measures.

Fifth, we detect C^1 discontinuity joint edges and C^2 discontinuity joint edges on the Fandisk model. Figure 13a shows a surface mesh consisting of 6,475 vertices and 12,946 triangles. We detect C^1 discontinuity joint edges and C^2 discontinuity joint edges consequently. Because C^1 discontinuity joint lines are also C^2 discontinuous, the C^1 discontinuity joint lines are set a high value of C^2 discontinuity measure in Fig. 13b. Figure 13c illustrates the detected joint lines by the proposed method. As a comparison, the result by Jiao and Bayyana's algorithm [8] for C^2 discontinuity line detection was given in Fig. 13d. From the figure we can see that an evident C^2 discontinuity joint line on the surface has been missed by their method. We note that even the model contains low noise, the sharpness indicator and the edge strength measure still introduce false C^1 discontinuity joint edges in high curvature regions or

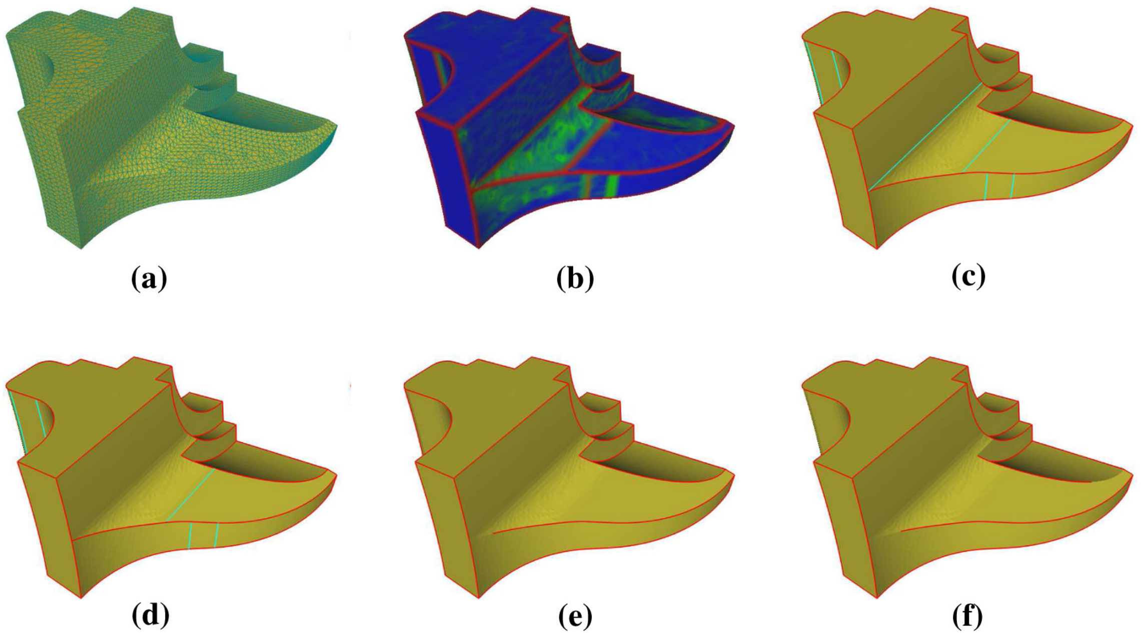


Fig. 13 C^1 and C^2 discontinuity joint line detection on the Fandisk model: **a** the triangular mesh; **b** the C^2 discontinuity measure; **c** the C^1 and C^2 discontinuity joint lines detected by the proposed algorithm; **d** the C^1 and C^2 discontinuity joint edges detected by Jiao's algorithm; **e** the C^1 discontinuities detected using the sharpness indicator; **f** the C^1 discontinuities detected by the edge strength measure ($s > 0.05$)

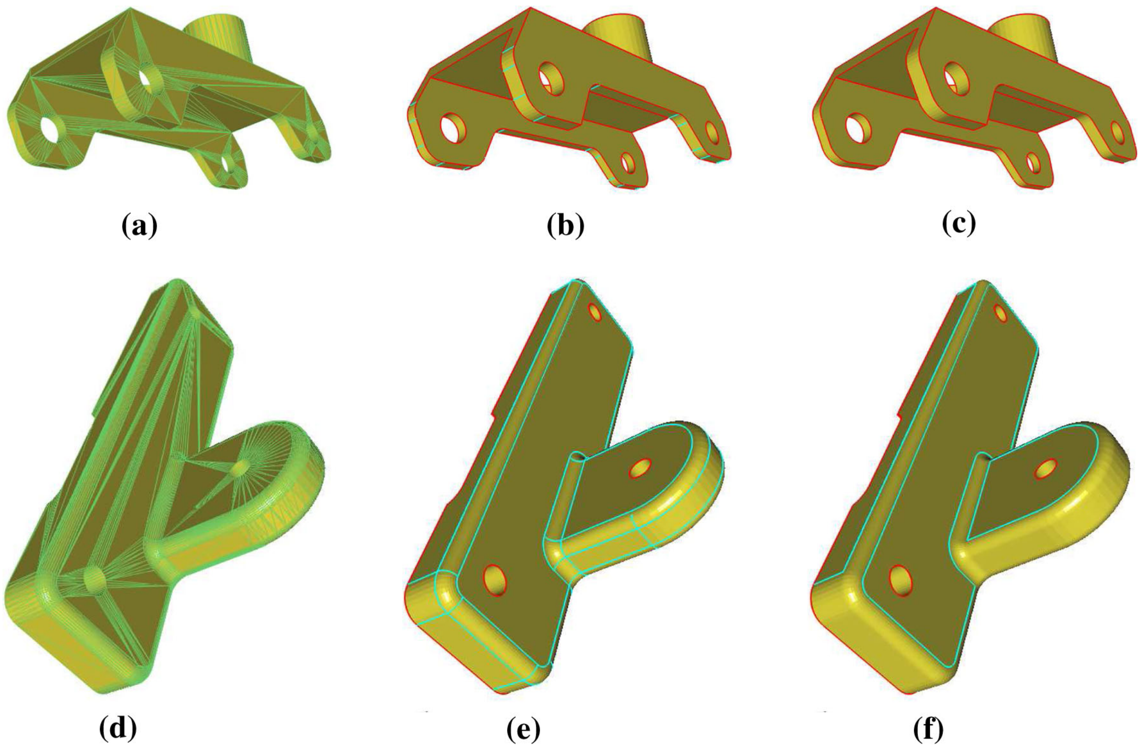


Fig. 14 C^1 and C^2 discontinuity joint line detection: **a**, **d** the triangular meshes; **b**, **e** the C^1 and C^2 discontinuity joint lines detected by the proposed algorithm; **c**, **f** the C^1 and C^2 discontinuity joint edges detected by Jiao's algorithm

Table 2 The numbers and the mean discontinuity measures of the detected joint edges by the proposed method

Model	C^1 discontinuity edges		C^2 discontinuity edges		Other edges	
	Number	Mean μ_e^1	Number	Mean μ_e^2	Number	Mean μ_e^2
Casting	848	1.992954	874	1.844504	13,614	1.046087
Top shape	278	1.960233	221	1.886938	4,700	1.060526
Fandisk	749	1.988232	201	1.896420	18,469	1.028607
Hub	663	1.984036	30	1.824270	1,305	1.003227
Part with holes	360	1.993069	554	1.832866	4,672	1.044054
Circulant equipment	3,099	1.797989	NA	NA	263,961	1.104138
Cap shape	1,204	1.890145	NA	NA	21,188	1.127248

Table 3 Bounds of absolute dihedral angles of the detected C^1 or C^2 discontinuity joint edges by the proposed method

Model	C^1 discontinuity edges	C^2 discontinuity edges	Other edges
Casting	(0.400215, 1.612509)	(0.015252, 0.812436)	(0, 0.979705)
Top shape	(0.499729, 1.570796)	(0, 0.177997)	(0, 0.308813)
Fandisk	(0.041001, 1.613320)	(0.014753, 0.200622)	(0, 0.395902)
Hub	(0.967451, 2.174142)	(0.075420, 0.087268)	(0, 0.174784)
Part with holes	(0.959965, 1.570796)	(0.006825, 0.087772)	(0, 0.176250)
Circulant equipment	(0.018235, 2.731584)	NA	(0, 2.508039)
Cap shape	(0.213996, 3.141592)	NA	(0, 3.069177)

miss true C^1 discontinuity joint edges in low curvature regions. See Fig. 13e, f for the results by the two measures, respectively.

Finally, we present two more examples to show C^1 and C^2 discontinuity joint line detection by the proposed method. The hub shape model in Fig. 14a has 658 vertices and 1,332 triangles and the mechanical part with three holes in Fig. 14d has 1,848 vertices and 3,704 triangles. From the two figures we can see that both meshes have many long and thin triangles. Though the C^1 discontinuity joint edges in these two models have large values of dihedral angles, but the C^2 discontinuity joint edges cannot be distinguished from other edges based on dihedral angles easily. By computing the discontinuity measures for all mesh edges, except for a few edges lying on low curvature regions, most C^2 discontinuity joint edges have been detected successfully by the proposed method; see Fig. 14b, e for the detection results. Figure 14c, f show that Jiao and Bayyana's algorithm [8] has missed many C^2 discontinuity joint lines.

The numbers and the mean discontinuity measures of C^1 or C^2 discontinuity joint edges for all the examples by the proposed method are given in Table 2. From the table we can also see that the joint edges are distinguished clearly from other edges lying on smooth surface regions by the discontinuity measures. Table 3 gives the lower and upper bounds of absolute dihedral angles of the C^1 discontinuity edges, the C^2 discontinuity edges or the other edges. From this table we learn that the C^1 or C^2 discontinuity joint edges cannot be detected correctly just by the dihedral angles at edges for general meshes.

7 Conclusions and discussions

This paper has presented new definitions of C^1 and C^2 discontinuity measures for edges on triangular meshes. Detailed formulae have been given for the computation of discontinuity measures using properly estimated discrete curvatures and shape parameters. A heuristic but practical algorithm has also been developed to detect smooth C^1 or C^2 discontinuity joint lines on surface meshes. Compared with other measures or algorithms for C^1 or C^2 discontinuity detection on surface meshes, the proposed technique has two main advantages. First, the proposed discontinuity measures are scale independent and the joint lines in low or high curvature regions can be detected in a same way. Second, the proposed measures do not depend on mesh triangulations much, C^1 or C^2 discontinuity joint lines on surface meshes with even highly irregular or non-uniform triangulations can be detected successfully.

As the joint line detection on a triangular mesh is an undetermined problem, we have chosen parameters heuristically for joint line detection on CAD-like meshes. It may be possible to set parameters in some other way such as statistics or optimization based methods for automatic detection of joint edges. At present, we detect joint lines that are consisting of mesh edges directly. In the future, we plan to compute discontinuity measures for mesh vertices and detect joint lines that pass a set of selected vertices approximately. The vertex based scheme is promising for approximate joint line detection for even wider types of meshes.

Acknowledgments We owe thanks to anonymous referees for their helpful comments on an earlier version of the paper. This work is supported by Natural Science Foundation of China Grants (60970077, 11290142), the ARC 9/09 Grant (MOE2008-T2-1-075), MOE RG 59/08 (M52110092), NRF 2007IDM-IDM002-010 of Singapore.

References

1. Agathos A, Pratikakis I, Perantonis S, Sapidis N, Azariadis P (2007) 3D mesh segmentation methodologies for cad applications. Comput Aided Des Appl 4(6):827–842
2. Angelo LD, Stefano PD (2010) C^1 continuities detection in triangular meshes. Comput Aided Des 42(9):828–839
3. Baker TJ (2004) Identification and preservation of surface features. In: Proceedings of the 13th international meshing roundtable, pp 299–310
4. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698
5. Demarsin K, Vanderstraeten D, Volodine T, Roose D (2007) Detection of closed sharp edges in point clouds using normal estimation and graph theory. Comput Aided Des 39(4):276–283
6. Elber G (1996) Error bounded piecewise linear approximation of freeform surfaces. Comput Aided Des 28(1):51–57
7. Hubeli A, Meyer K, Gross MH (2000) Mesh edge detection. In: Workshop Lake Tahoe, Lake Tahoe City, California, USA, October 15–17
8. Jiao X, Bayyana NR (2008) Identification of c^1 and c^2 discontinuities for surface meshes in CAD. Computer Aided Des 40(2):160–175
9. Jiao X, Heath M (2002) Feature detection for surface meshes. In: Proceedings of 8th international conference on numerical grid generation in computational field simulations, pp 705–714
10. Kim HS, Choi HK, Lee KH (2009) Feature detection of triangular meshes based on tensor voting theory. Comput Aided Des 41(1): 47–58
11. Kimia BB, Frankel I, Popescu AM (2003) Euler spiral for shape completion. Int J Comput Vis 54(1–3):159–182
12. Lavoué G, Dupont F, Baskurt A (2005) A new cad mesh segmentation method, based on curvature tensor analysis. Computer-Aided Design 37(10):975–987
13. Max N (1999) Weights for computing vertex normals from facet normals. J Graphics Tools 4(2):1–6
14. Meek DS, Walton DJ (2000) On surface normal and gaussian curvature approximations given data sampled from a smooth surface. Comput Aided Geom Des 17(6):521–543
15. Meyer M, Desbrun M, Schröder P, Barr AH (2003) Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and mathematics III, pp 35–57
16. Page DL, Koschan A, Sun Y, Paik JK, Abidi MA (2001) Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In: CVPR, pp 162–167
17. Shimizu T, Date H, Kanai S, Kishinami T (2005) A new bilateral mesh smoothing method by recognizing features. In: Proceedings of the ninth international conference on computer aided design and computer graphics, pp 281–286
18. Sunil VB, Pande SS (2008) Automatic recognition of features from freeform surface cad models. Comput Aided Des 40(4): 502–517
19. Tang CK, Medioni GG (1999) Robust estimation of curvature information from noisy 3D data for shape description. In: ICCV, pp 426–433
20. Taubin G (1995) Estimating the tensor of curvature of a surface from a polyhedral approximation. In: ICCV, pp 902–907
21. Thürmer G, Wüthrich C (1998) Computing vertex normals form polygonal facets. J Graphics Tools 3(1):43–46
22. Várady T, Facello MA, Terék Z (2007) Automatic extraction of surface structures in digital shape reconstruction. Comput Aided Des 39(5):379–388
23. Vidal V, Wolf C, Dupont F (2011) Robust feature line extraction on cad triangular meshes. In: Proceedings of the international conference on computer graphics theory and applications, GRAPP 2011, pp 106–112
24. Walton DJ, Meek DS (2009) G^1 interpolation with a single cornu spiral segment. J Comput Appl Math 223:86–96
25. Wang D, Hassan O, Morgan K, Weatherill N (2007) Enhanced remeshing from stl files with applications to surface grid generation. Commun Numer Methods Eng 23(3):227–239
26. Wang J, Yu Z (2011) Surface feature based mesh segmentation. Comput Graphics 35(3):661–667
27. Wang X, Liu X, Lu L, Li B, Cao J, Yin B, Shi X (2012) Automatic hole-filling of cad models with feature-preserving. Computers & Graphics 36(2):101–110
28. Yamakawa S, Shimada K (2010) Polygon crawling: feature edge extraction from a general polygonal surface for mesh generation. Eng Comput 26(3):249–264