Couple Points – A Local Approach to Global Surface Analysis

Christian Rössl and Holger Theisel

Visual Computing Group, University of Magdeburg, Germany

Abstract. We introduce the concept of couple points as a global feature of surfaces. Couple points are pairs of points $(\mathbf{x}_1, \mathbf{x}_2)$ on a surface with the property that the vector $\mathbf{x}_2 - \mathbf{x}_1$ is parallel to the surface normals both at $\mathbf{x}_1$ and $\mathbf{x}_2$. In order to detect and classify them, we use higher order local feature detection methods, namely a Morse theoretic approach on a 4D scalar field. We apply couple points to a number of problems in Computer Graphics: the detection of maximal and minimal distances of surfaces, a fast approximation of the shortest geodesic path between two surface points, and the creation of stabilizing connections of a surface.

Keywords: surface features, double normals, Morse theory, triangular meshes, geodesic paths

1 Introduction

Size, complexity and number of surfaces considered in Computer Graphics are continuously growing. One popular approach to deal with this is the extraction of characteristic features of a surface. Feature extraction has a variety of applications such as segmentation, shape matching, reverse engineering, compression and simplification of surfaces.

Generally, two kinds of surface features can be distinguished: local and global features. For local features it can be decided entirely by a local analysis whether or not a point on the surface belongs to the feature. For global features, this decision can be made only by a global analysis of the surface.

Examples for detecting local features on surfaces are the estimation of the curvature tensor (see, e.g., [1, 8, 19, 22] or the survey [11]), the estimation of surface normals, or the detection of sharp edges. Examples for global features are the detection of medial axes [3, 23], the shortest distance between a point and a surface [9], and the detection of the shortest geodesic path between two points on a surface. This is an interesting and well-studied problem which leads to solving the associated PDE by propagating wave fronts [15] or to unfolding a polyhedral surface [2, 24, 26] (see also [17] for a general survey). While such algorithms are efficient for single-source approximations of shortest paths, they may be rather expensive if paths between arbitrary point pairs have to be computed. Alternative methods apply energy (arc-length) minimization [10, 13, 20] for a curve in a manifold, where a reasonable initial guess is required, e.g., from searching a discrete shortest path. In contrast to the point-point problem which imposes a boundary value problem, the construction of a straightest geodesic path starting from a point and proceeding into a given direction is much simpler: such initial value problem depends entirely on a local analysis [21].

A combination of local and global features is the extraction of Morse complexes on a surface [4, 5, 18]. Given a smooth function m on a surface, Morse complexes divide the surface into areas of similar behavior of the gradient flow of m. To do so, critical points — i.e., points with a vanishing gradient of m — are extracted and classified into sources, sinks, and saddles. Then certain separation curves are integrated from the saddles in forward and backward gradient direction. Once the scalar field m is given, the extraction of the critical points is a local process. However, the integrated separation lines reveal a global feature: local changes of m can cause potential changes of the separation lines everywhere on the surface. Also note that the underlying Morse function may be obtained by a local or a global [12] analysis of the surface.

This paper introduces a new global feature of the surface called couple points. These are pairs of points $(\mathbf{x}_1, \mathbf{x}_2)$ on the surface with the property that the vector $\mathbf{x}_2 - \mathbf{x}_1$ is parallel to the surface normals both at $\mathbf{x}_1$ and $\mathbf{x}_2$. In order to extract and classify them, we do not apply a global analysis of the surface but a local feature analysis on pairs of surfaces instead. This way we are able to apply standard local analysis tools to achieve a global analysis of the surfaces. We show that in general there exists a finite number of isolated couple points. We believe that number, location, and classification of couple points reveals a relevant information about the global shape of surfaces.

We remark that the general concept of couple points is not new. In fact, the definition of double normals is equivalent to couple points, and there a number of interesting problems in geometry related to them, e.g., finding their minimum number for convex bodies in E^n [14] or for smooth curves [7] of certain topology. We are interested in the detection of couple points and applications in computer graphics.

The remainder of this article paper is organized as follows: Section 2 gives the definition of couple points and shows a first simple example. Section 3 collects properties of couple points on smooth surfaces. To do so, we apply Morse theoretic approaches to an appropriate 4D scalar field and show that the couple points correspond to the critical points of this 4D Morse complex. Section 4 shows how to extract and classify couple points on triangular meshes. In Section 5 we apply couple points to the following problems: first, we detect minimal (or maximal) distances between two surfaces (a problem which is far more complicated than the problem of finding the shortest distance between a point and a surface). Second, we use couple points to compute a fast approximation of the shortest geodesic path between two points on a surface. And third, we use couple points to find stabilizing connectors on surfaces in order to increase the surface stability. Finally, Section 6 draws conclusions and mentions future research.

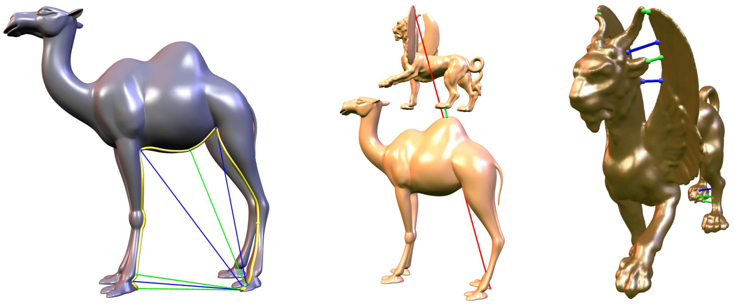

Fig. 1: Three applications of couple points. Left: a fast approximation of the shortest path between two surface points (yellow) passing trough five couple points. Middle: shortest (green) and largest (red) distance between two surfaces. Right: stabilizing connectors between parts of a surface.

2 Motivation and definition of couple points

To introduce the idea of couple points, we start with a simple 2D example. Given are two closed differentiable curves $\mathbf{x}_1, \mathbf{x}_2$ which do not intersect each other. We search for the minimal and maximal Euclidean distance of $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e., for a pair of points $(\mathbf{x}_1^{min}, \mathbf{x}_2^{min}) \in (\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_1^{max}, \mathbf{x}_2^{max}) \in (\mathbf{x}_1, \mathbf{x}_2)$ with

$$\forall (\mathbf{x}_1^i, \mathbf{x}_2^i) \in (\mathbf{x}_1, \mathbf{x}_2) \quad : \quad \|\mathbf{x}_2^{min} - \mathbf{x}_1^{min}\| \le \|\mathbf{x}_2^i - \mathbf{x}_1^i\| \le \|\mathbf{x}_2^{max} - \mathbf{x}_1^{max}\|.$$

Figure 2a gives an illustration. In this picture we can also observe a property of $(\mathbf{x}_1^{min}, \mathbf{x}_2^{min})$ (which will be proved later in Section 3): the vector $\mathbf{x}_2^{min} - \mathbf{x}_1^{min}$ is perpendicular to the tangent direction of $\mathbf{x}_1$ in $\mathbf{x}_1^{min}$ and of $\mathbf{x}_2$ in $\mathbf{x}_2^{min}$. A similar statement holds for $(\mathbf{x}_1^{max}, \mathbf{x}_2^{max})$. The transformation of this observation to surfaces gives reason for the following definition of couple points:.

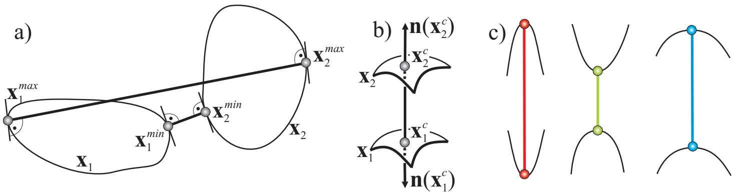

Fig. 2: a) Pair of points $(\mathbf{x}_1^{min}, \mathbf{x}_2^{min})$ and $(\mathbf{x}_1^{max}, \mathbf{x}_2^{max})$ with minimal and maximal distance. b) A couple point. c) Classification of couple points in sink (red), source (green) and saddle (blue).

Definition 1. *Given two differentiable surfaces* $\mathbf{x}_1, \mathbf{x}_2$ *together with their normal maps* $\mathbf{n}(\mathbf{x}_1), \mathbf{n}(\mathbf{x}_2)$*, a* couple point $\mathbf{x}^c = (\mathbf{x}_1^c, \mathbf{x}_2^c)$ *is a pair of points with*

$\mathbf{x}_1^c \in \mathbf{x}_1$, $\mathbf{x}_2^c \in \mathbf{x}_2$, *and* $\mathbf{n}(\mathbf{x}_1^c) \times (\mathbf{x}_2^c - \mathbf{x}_1^c) = \mathbf{n}(\mathbf{x}_2^c) \times (\mathbf{x}_2^c - \mathbf{x}_1^c) = (0,0,0)^T$. *Furthermore, let* $C(\mathbf{x}_1, \mathbf{x}_2)$ *be the set of all couple points between* $\mathbf{x}_1$ *and* $\mathbf{x}_2$.

Figure 2b illustrates a couple point. Given two surfaces $\mathbf{x}_1$ and $\mathbf{x}_2$, there is in general a finite number of isolated couple points between them. This paper is devoted to studying their properties and their applicability to a number of problems in Computer Graphics. Couple points can also be computed on a single surface, i.e., for instance $C(\mathbf{x}_1, \mathbf{x}_1)$ can be extracted.

3 Properties of couple points

In order to capture useful properties of couple points, we first assume $\mathbf{x}_1$ and $\mathbf{x}_2$ to be parametric surfaces. Then we apply a Morse theoretical approach to the 4D domain of $\mathbf{x}_1$ and $\mathbf{x}_2$ and show that our derived properties of couple points are independent of the particular parametrization of the surfaces.

Given two regularly parametrized surfaces $\mathbf{x}_1(u,v)$ over a 2D domain D_1 and $\mathbf{x}_2(s,t)$ over a 2D domain D_2, we define a *double surface* $\mathbf{x}^d$ as

$$\mathbf{x}^d(u,v,s,t) = (\mathbf{x}_1(u,v), \mathbf{x}_2(s,t)). \tag{1}$$

This means that $\mathbf{x}^d$ is a map from $D = D_1 \times D_2$ to $\mathrm{I\!R}^3 \times \mathrm{I\!R}^3$. A point on a double surface is called a *double point*. Furthermore, we define the *double normal*

$$\mathbf{n}^d(u,v,s,t) = (\mathbf{n}_1, \mathbf{n}_2) = \left(\frac{\mathbf{x}_{1u} \times \mathbf{x}_{1v}}{\|\mathbf{x}_{1u} \times \mathbf{x}_{1v}\|} \, , \, \frac{\mathbf{x}_{2s} \times \mathbf{x}_{2t}}{\|\mathbf{x}_{2s} \times \mathbf{x}_{2t}\|} \right). \tag{2}$$

where $\mathbf{x}_{1u}, \mathbf{x}_{1v}, \mathbf{x}_{2s}, \mathbf{x}_{2t}$ are the first order partials of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. We consider the 4D Morse function

$$m(u,v,s,t) = \|\mathbf{x}_2(u,v) - \mathbf{x}_1(s,t)\|^2 \tag{3}$$

which describes the (squared) Euclidean distance of $\mathbf{x}_1$ and $\mathbf{x}_2$. Its gradient $\mathrm{grad}(m) = (m_u, m_v, m_s, m_t)$ is a 4D vector field on D, its map onto $\mathbf{x}^d$ gives the *gradient double vector*

$$\begin{aligned} \mathbf{v}^d(u,v,s,t) &= (\mathbf{v}_1(u,v,s,t) \, , \, \mathbf{v}_2(u,v,s,t)) \qquad (4) \\ &= \left(m_u \frac{\mathbf{x}_{1v} \times \mathbf{n}_1}{\|\mathbf{x}_{1u} \times \mathbf{x}_{1v}\|} + m_v \frac{\mathbf{n}_1 \times \mathbf{x}_{1u}}{\|\mathbf{x}_{1u} \times \mathbf{x}_{1v}\|} \right. \, , \\ &\qquad \left. m_s \frac{\mathbf{x}_{2t} \times \mathbf{n}_2}{\|\mathbf{x}_{2s} \times \mathbf{x}_{2t}\|} + m_t \frac{\mathbf{n}_2 \times \mathbf{x}_{2t}}{\|\mathbf{x}_{2s} \times \mathbf{x}_{2t}\|} \right). \end{aligned}$$

$\mathbf{v}^d$ points into the direction of steepest ascent of the Euclidean distance of $\mathbf{x}_1$ and $\mathbf{x}_2$. Figure 3a illustrates $\mathbf{x}^d$, $\mathbf{n}^d$ and $\mathbf{v}^d$.

The following theorem will provide the foundation of our further treatment of couple points.

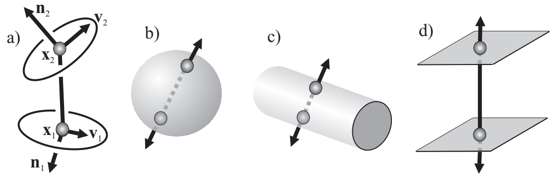

Fig. 3: a) A double point $\mathbf{x}^d = (\mathbf{x}_1, \mathbf{x}_2)$ with double normal $\mathbf{n}^d = (\mathbf{n}_1, \mathbf{n}_2)$ and gradient double vector $\mathbf{v}^d = (\mathbf{v}_1, \mathbf{v}_2)$. b)-e) Zero importance couple points: diametric on a sphere, opposite on a cylinder, opposite on parallel planes.

Theorem 1. *Given $\mathbf{x}^d$, $\mathbf{n}^d$, $\mathbf{v}^d$ as defined in (1) - (4), the following equation holds.*

$$\mathbf{v}^d = (\ 2((\mathbf{x}_2 - \mathbf{x}_1) \times \mathbf{n}_1) \times \mathbf{n}_1\ \ ,\ \ 2((\mathbf{x}_1 - \mathbf{x}_2) \times \mathbf{n}_2) \times \mathbf{n}_2\ )\ .$$

The proof is a straightforward exercise in algebra from (1)-(4). Theorem 1 shows that $\mathbf{v}$ is a geometric measure on $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e., it is independent of the particular parametrization of $\mathbf{x}_1$ and $\mathbf{x}_2$. In fact, $\mathbf{v}^d$ can directly be computed from $\mathbf{x}^d$ and $\mathbf{n}^d$. Also from Theorem 1 follows directly

Theorem 2. *Given $\mathbf{x}^d$, $\mathbf{n}^d$, $\mathbf{v}^d$ as defined from (1) - (4), the following three statements are equivalent:*

- $\mathbf{x}^d(u, v, s, t)$ *is a couple point concerning definition 1.*
- *grad*$(m) = (0, 0, 0, 0)^T$.
- $\mathbf{v}^d = ((0, 0, 0)^T, (0, 0, 0)^T) = \mathbf{0}^d$, *i.e.* $\mathbf{x}^d$ *is a critical (double) point w.r.t.* $\mathbf{v}^d$.

Let $\mathbf{x}^c = \mathbf{x}^d(u_c, v_c, s_c, t_c)$ be a couple point. We apply a local reparametrization of $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $\mathbf{x}_{1u}(u_c, v_c)$ and $\mathbf{x}_{1v}(u_c, v_c)$ are orthonormalized, and that $\mathbf{x}_{2s}(u_c, v_c)$ and $\mathbf{x}_{2t}(u_c, v_c)$ are orthonormalized as well. This can easily be done by locally computing normal and principal directions and using these vectors as the bases of a local coordinate system. Then we can further classify $\mathbf{x}^c$ by an eigen-analysis of the Hessian matrix

$$H_m(u, v, s, t) = \begin{pmatrix} m_{uu} & m_{uv} & m_{us} & m_{ut} \\ m_{vu} & m_{vv} & m_{vs} & m_{vt} \\ m_{su} & m_{sv} & m_{ss} & m_{st} \\ m_{tu} & m_{tv} & m_{ts} & m_{tt} \end{pmatrix}$$

at (u_c, v_c, s_c, t_c). Let λ_i $(i = 1, .., 4)$ be the eigenvalues of $H_m(u_c, v_c, s_c, t_c)$. We call $\mathbf{x}^c$ a *source* iff all eigenvalues of $H_m(u_c, v_c, s_c, t_c)$ are positive. In this case, m has a local minimum at (u_c, v_c, s_c, t_c): all double points on $\mathbf{x}^d$ in a small neighborhood of $\mathbf{x}^c$ have Euclidean distance smaller than $\|\mathbf{x}_2(s_c, t_c) - \mathbf{x}_1(u_c, v_c)\|$. $\mathbf{x}_c$ is a *sink* iff all eigenvalues of H_m are negative, i.e., m has a local maximum at (u_c, v_c, s_c, t_c). $\mathbf{x}^c$ is a *saddle* iff it has both positive and negative eigenvalues. Figure 2c illustrates source, sink, and saddle couple points for 2D curves. There, as well as in the remaining figures, we represent a source couple point with a green line, a sink with a red line, and a saddle with a blue line.

If we consider the couple points of only one surface $\mathbf{x}$ (i.e. we consider $C(\mathbf{x},\mathbf{x})$), and if $\mathbf{x}$ is a closed manifold, then a couple point $\mathbf{x}^c = (\mathbf{x}_1^c, \mathbf{x}_2^c) \in (\mathbf{x},\mathbf{x})$ can be further classified as

- *direct inside* if the line segment $\mathbf{x}_1^c, \mathbf{x}_2^c$ is completely inside the surface.
- *direct outside* if the line segment $\mathbf{x}_1^c, \mathbf{x}_2^c$ is completely outside the surface.
- *indirect* else.

This distinction is useful for applications which connect couple points by a straight line which should not have intersections with the surface, e.g., stabilizing connectors (see Section 5.3). Figure 4 illustrates this for a closed 2D curve.

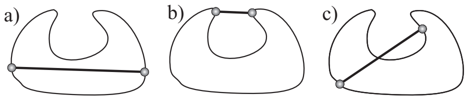

Fig. 4: Classification of couple points: a) direct inside, b) direct outside, c) indirect.

As we will see later, there may be a rather large or even infinite number of couple points in a surface. For instance in 2D curves of constant width, e.g., circle or Reuleaux triangle, have infinitely many couple points. To deal with this, i.e., to discard the unimportant ones, we also need to equip a couple point with an importance. Couple points which tend to disappear after slight changes of the surface should have a low importance. Also, couple points which are not isolated should have a zero importance. Here we assume a sufficiently smooth surface and use $\mathrm{imp}(\mathbf{x}^d) = \det(H_m)$ which fulfills the requirements mentioned above. Figure 3b-e shows some examples of non-isolated couple points with a zero importance, i.e., they are degenerate and are not considered here. We remark that this simple definition of importance is a local property. In particular, this local importance is not necessarily stable under perturbation of the shape or its parametrization by adding noise.

Let $\mathbf{e}_i$ $(i = 1,..,4)$ be the eigenvectors of $H_m(u_c, v_c, s_c, t_c)$. Then $\mathbf{e}_i$ can be transformed to double eigenvectors on $\mathbf{x}^d$ by

$$\mathbf{e}_i^c = (\ a_i\ \mathbf{x}_{1u}(u_c,v_c)\ +\ b_i\ \mathbf{x}_{1v}(u_c,v_c)\ ,\ c_i\ \mathbf{x}_{2s}(s_c,t_c)\ +\ d_i\ \mathbf{x}_{2t}(s_c,t_c)\ )$$

for $i = 1,..,4$. Note that $\mathbf{e}_i^c$ are uniquely defined by $\mathbf{x}^c$ and the curvature tensors at $\mathbf{x}_1(u_c,v_c)$ and $\mathbf{x}_2(s_c,t_c)$. Then we can compute the separation double line from $\mathbf{x}_c$ by applying a double stream line integration of $\mathbf{v}^d$ starting in $\mathbf{x}_c$ in the directions $\pm\mathbf{e}_i^c$. The integration direction (either forward or backward) is given by the signs of λ_i. This way, 8 separation double stream lines are created by a couple point $\mathbf{x}^c$. Figure 5 gives an illustration.

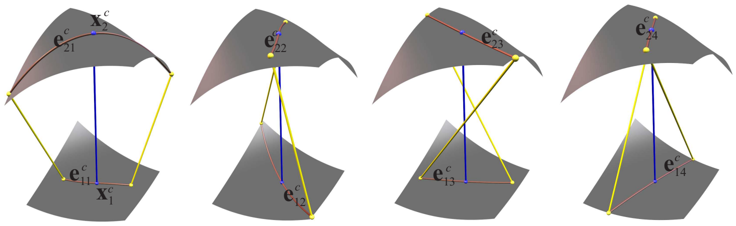

Fig. 5: A couple point $\mathbf{x}^c = (\mathbf{x}^c_1, \mathbf{x}^c_2)$ and its 4 double eigenvectors $\mathbf{e}^c_i = (\mathbf{e}^c_{1i}, \mathbf{e}^c_{2i})$. Each double eigenvector creates two double stream lines by integrating $\mathbf{v}^d$. Here, they end in double points (yellow) when one of the components reaches the boundary of the surface.

The set of all couple points together with all integrated double separation curves gives the topological skeleton of the Morse function. Figure 6 shows an example of a test surface containing 370 couple points. The close-up shows that the separation double curves cover the surface in a rather dense way.

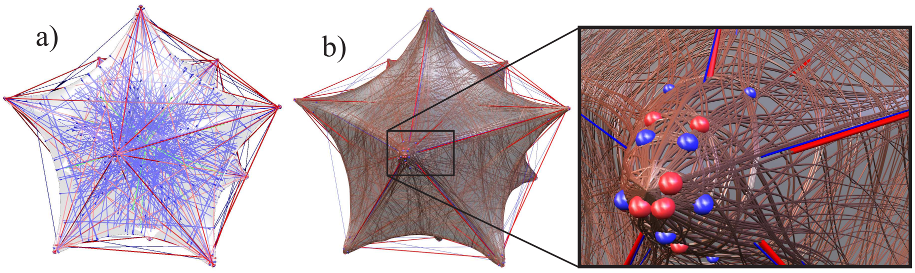

Fig. 6: a) Test surface with 370 couple points. b) All separation double curves.

4 Couple points for triangular meshes

Up to now we treated couple points for smooth surfaces. In this section we show how to apply the concept to piecewise linear approximations of smooth surfaces, i.e. to triangular meshes, as this is the standard surface representation in Computer Graphics. Here, we assume that each vertex is equipped with an either exact or estimated normal. Then the basic approach is to test each pair of triangles for couple points: given a triangle $\mathbf{t}$ with the vertices $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and their assigned normals $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$, and given a triangle $\widetilde{\mathbf{t}}$ with the vertices $\widetilde{\mathbf{p}_1}$, $\widetilde{\mathbf{p}_2}$, $\widetilde{\mathbf{p}_3}$ and the normals $\widetilde{\mathbf{n}_1}$, $\widetilde{\mathbf{n}_2}$, $\widetilde{\mathbf{n}_3}$, we search for couple points as barycentric combinations (α, β, γ) and $(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$ by solving the system

$$(\widetilde{\mathbf{x}} - \mathbf{x}) \times \mathbf{n} \;=\; (\widetilde{\mathbf{x}} - \mathbf{x}) \times \widetilde{\mathbf{n}} \;=\; (0,0,0)^T$$

with $\mathbf{x} = (\alpha\,\mathbf{p}_1 + \beta\,\mathbf{p}_2 + \gamma\,\mathbf{p}_3)$, $\widetilde{\mathbf{x}} = (\widetilde{\alpha}\,\widetilde{\mathbf{p}_1} + \widetilde{\beta}\,\widetilde{\mathbf{p}_2} + \widetilde{\gamma}\,\widetilde{\mathbf{p}_3})$, $\mathbf{n} = (\alpha\,\mathbf{n}_1 + \beta\,\mathbf{n}_2 + \gamma\,\mathbf{n}_3)$, $\widetilde{\mathbf{n}} = (\widetilde{\alpha}\,\widetilde{\mathbf{n}_1} + \widetilde{\beta}\,\widetilde{\mathbf{n}_2} + \widetilde{\gamma}\,\widetilde{\mathbf{n}_3})$, $\alpha + \beta + \gamma = 1$, $\widetilde{\alpha} + \widetilde{\beta} + \widetilde{\gamma} = 1$ for the unknowns $(\alpha, \beta, \gamma, \widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$. This ends up in finding the roots of a 6th order polynomial if the intersection line of the two triangle planes is excluded as a solution. This in turn means that two triangles which do not intersect each other can have up to 6 isolated couple points. Figure 7a gives an illustration.

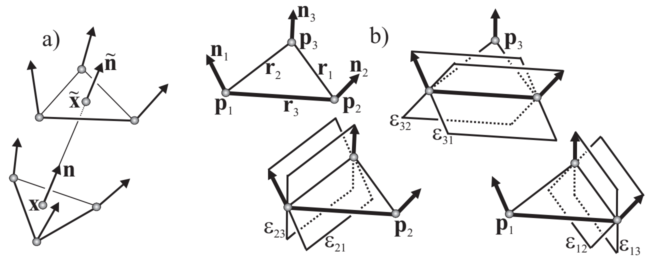

Fig. 7: a) A couple point between two triangles. b) The planes ϵ_{ij}.

In order to find the couple points between two triangles, we apply a subdivision approach. We test whether or not $\mathbf{t}$ can "see" $\widetilde{\mathbf{t}}$, i.e., whether there are barycentric coordinates (α, β, γ) such that the line $\mathbf{x} + \mu\mathbf{n}$ intersects $\widetilde{\mathbf{t}}$. Therefore, we introduce the 6 planes ϵ_{ij} $(i, j \in \{1, 2, 3\}, i \neq j)$ by demanding that ϵ_{ij} contains the line $\mathbf{r}_i$ and the vector $\mathbf{n}_i$. Here, $\mathbf{r}_1$ is the line through $\mathbf{p}_2$ and $\mathbf{p}_3$, $\mathbf{r}_2$ passes through $\mathbf{p}_3$ and $\mathbf{p}_1$, and $\mathbf{r}_3$ passes through $\mathbf{p}_1$ and $\mathbf{p}_2$. Figure 7b illustrates the planes ϵ_{ij}. Now the test is done by checking on which side of the planes the vertices $\widetilde{\mathbf{p}_1}$, $\widetilde{\mathbf{p}_2}$, $\widetilde{\mathbf{p}_3}$ are located. We use the notation opp$(\mathbf{p}, \epsilon, \mathbf{q})$ if the points $\mathbf{p}$ and $\mathbf{q}$ are located on opposite sides of the plane ϵ. Then we check the following conditions:

- If opp$(\widetilde{\mathbf{p}}_i, \epsilon_{12}, \mathbf{p}_1) \wedge$ opp$(\widetilde{\mathbf{p}}_i, \epsilon_{13}, \mathbf{p}_1)$ for all $i \in \{1, 2, 3\}$, then $\mathbf{t}$ cannot see $\widetilde{\mathbf{t}}$.
- If opp$(\widetilde{\mathbf{p}}_i, \epsilon_{21}, \mathbf{p}_2) \wedge$ opp$(\widetilde{\mathbf{p}}_i, \epsilon_{23}, \mathbf{p}_2)$ for all $i \in \{1, 2, 3\}$, then $\mathbf{t}$ cannot see $\widetilde{\mathbf{t}}$.
- If opp$(\widetilde{\mathbf{p}}_i, \epsilon_{31}, \mathbf{p}_3) \wedge$ opp$(\widetilde{\mathbf{p}}_i, \epsilon_{32}, \mathbf{p}_3)$ for all $i \in \{1, 2, 3\}$, then $\mathbf{t}$ cannot see $\widetilde{\mathbf{t}}$.

If none of these three conditions applies, we assume that $\mathbf{t}$ can "see" $\widetilde{\mathbf{t}}$. In a similar way we check whether $\widetilde{\mathbf{t}}$ can "see" $\mathbf{t}$. If either $\mathbf{t}$ cannot "see" $\widetilde{\mathbf{t}}$ or $\widetilde{\mathbf{t}}$ cannot "see" $\mathbf{t}$, no couple point exists between $\mathbf{t}$ and $\widetilde{\mathbf{t}}$. Otherwise we subdivide $\mathbf{t}$ and $\widetilde{\mathbf{t}}$ and apply our test recursively again until the size of $\mathbf{t}$ and $\widetilde{\mathbf{t}}$ is beyond a certain accuracy threshold.

As presented above, our algorithm for finding all couple points tests every pair of triangles for couple points and consequently has quadratic complexity in terms of the number of triangles. At this point, (hierarchical) space partition techniques can be applied to drastically improve on efficiency. Even simpler is a partition of normal directions which allows the fast enumeration of all candidates which eventually "see" a given triangle.

Once we have detected all couple points of a mesh, we are left with classifying them as sources, sinks, and saddles as well as to compute their eigenvalues and eigenvectors. All we need to have is an estimation of the curvature tensors of the surfaces at $\mathbf{x}_{1c}$ and $\mathbf{x}_{2c}$. Most existing algorithms to estimate the curvature tensor on a mesh do so per vertex. We follow [25] to estimate the curvature tensor at every inner point of a triangle as a smooth function over the triangle. This is done by considering both the linear interpolation of the vertices and the normals. This allows us to compute a classification of couple points as described in Section 3.

5 Results and Applications

In this section we demonstrate the extraction of couple points to a number of test data sets. Then we describe three areas of applications for couple points.

For the camel data set in Figure 8 (consisting of 78,144 triangles) we detected 2,721 couple points. Figure 8a shows all detected direct inside couple points. In Figure 8b, we picked two particular couple points: the longest one (i.e., the one with the largest Euclidean distance between its components) direct outside (located between ear and toe), and the longest direct inside.

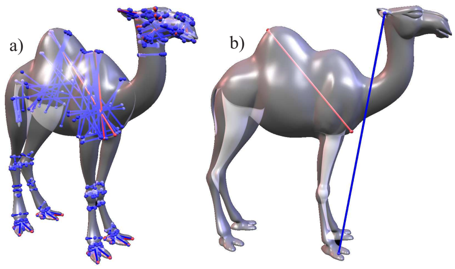

Fig. 8: Camel: a) all direct inside couple points, b) longest direct inside and direct outside couple points.

For the feline data set (99,732 triangles) shown in Figure 9, we detected 12,398 couple points. Figure 9a shows all detected direct inside couple points, Figure 9b does so for all direct outside couple points. Note that we filtered out couple points which are very close to each other for visualization. Figure 9c shows the three longest direct outside couple points, the largest direct inside couple point is shown in Figure 9d.

The ”Freezing Old Woman” data set shown in Figure 10 consists of 9,995 triangles. Figure 10b shows the most important direct inside couple points, while Figure 10c shows the longest direct inside couple point.

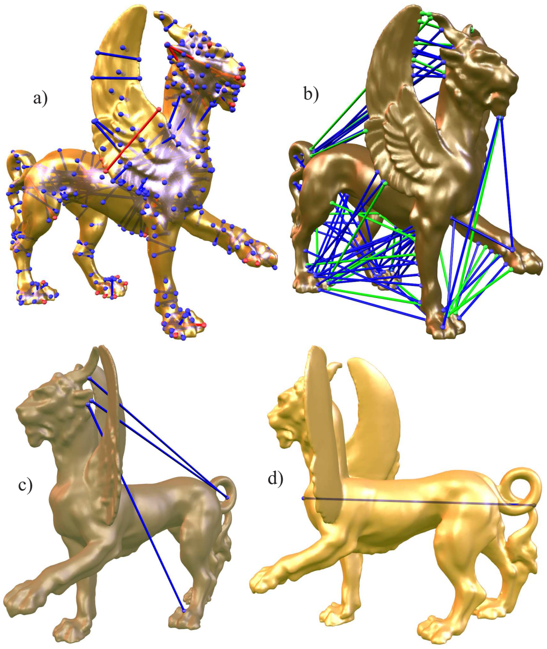

Fig. 9: Couple points of feline: a) all direct inside, b) direct outside, c) longest direct outside, d) longest direct inside.

5.1 Computing the maximal/minimal distance of surfaces

Given two surfaces, the computation of the minimal distance between them is a relevant problem in computer graphics and can be used for instance for collision detection or 3D path planning. This problem is far more complicated than the computation of the shortest distance between a point and a surface. If the surfaces are given as simple triangular meshes (i.e., without considering interpolated normals), the shortest distance may appear between two vertices, between a vertex and an inner point of a triangle, or between inner points of two edges. In the following, we consider piecewise linear surfaces with piecewise linear normals, a surface model often used in computer graphics as a compromise between simplicity and efficiency on the one side and (often visual) smoothness on the other side. For them, the solution generally appears at inner points of two triangles.

Theorem 2 states that the shortest distance between two surfaces is given by a couple point. Hence, to get the shortest distance between two surfaces $\mathbf{x}_1$ and $\mathbf{x}_2$, we can use $C(\mathbf{x}_1, \mathbf{x}_2)$ and select the couple point with the shortest Euclidean distance of its components. In a similar way we can compute the largest distance between two surfaces. Considering a single surface, we can also compute the shortest and largest distance as the distance of the components of couple points which are completely inside or outside the surface.

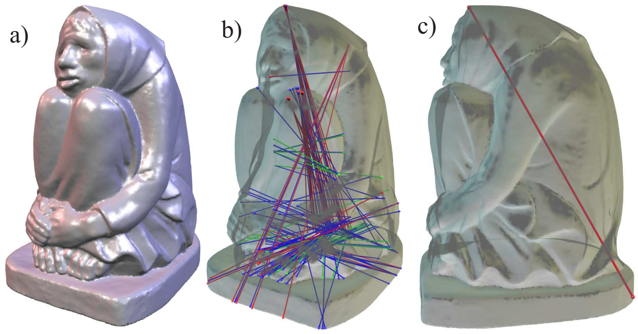

Fig. 10: a) Freezing Old Woman, b) most important direct inside couple points, c) longest direct inside couple point.

Figure 11a shows all detected direct outside couple points between the camel and the feline model for a certain relative position between them. (Note that in this picture we filtered out couple points which are very close to each other in both components.) Among them, the shortest couple point gives the shortest distance between the surfaces, as shown Figure 1 (middle). Figures 11b and 10c show the detected largest distance for the feline and the Freezing Old Woman data set, respectively.

Using our algorithm to compute the shortest distances between surfaces, the computing time is essentially the time necessary to extract all couple points (see Section 5). We are not aware of timings of any pre-existing solutions when the considered meshes contain normals at the vertices.

5.2 Approximations of the shortest geodesic paths between two points

Given two points $\mathbf{x}_1$, $\mathbf{x}_2$ on a surface, the computation of the shortest geodesic path connecting them is a rather expensive process which requests a global analysis of the surface. Here, couple points provide a way of computing a fast approximation of the shortest geodesic path. The basic idea is to apply a backward integration of $\mathbf{v}^d$ starting from the double point $(\mathbf{x}_1, \mathbf{x}_2)$. The motivation behind this approach is that $\mathbf{v}^d$ points into the direction of steepest ascent of the Euclidean distance of the components of a double point. Hence, a backward integration of $\mathbf{v}^d$ can be considered as a Greedy algorithm to obtain the shortest path between $\mathbf{x}_1$ and $\mathbf{x}_2$: at every integration step the integrator tries to reduce the Euclidean distance as much as possible.

Doing a backward integration of $\mathbf{v}^d$ starting from $(\mathbf{x}_1, \mathbf{x}_2)$, two cases are possible

- The components of the double points collapse to a single point $(\mathbf{x}_e, \mathbf{x}_e)$. Figure 12a gives an illustration.

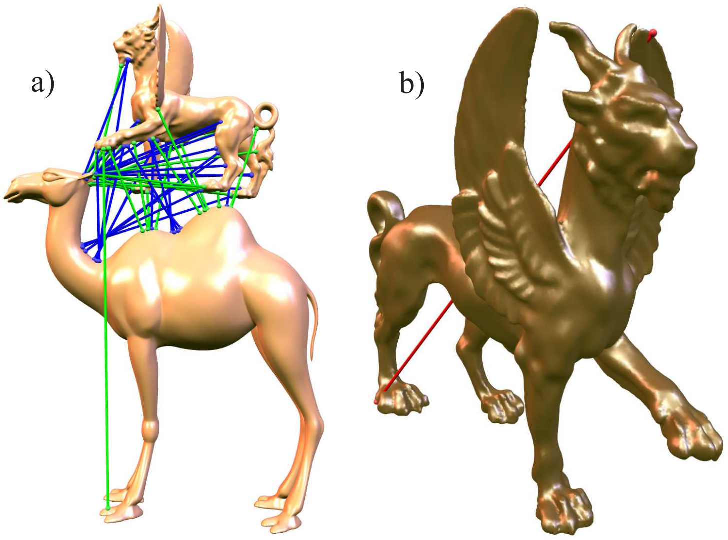

Fig. 11: a) all direct outside couple points between feline and camel, b) longest distance of feline.

- The integration gets stuck in a couple point $(\mathbf{x}_1{}^c, \mathbf{x}_2{}^c)$. Figure 12b illustrates this.

In the first case the algorithm stops, and the shortest path is the union of the two components of the integrated double curve. In the second case we need an algorithm to "get out" of the couple point, i.e., we need a shortest path between $\mathbf{x}_1{}^c$ and $\mathbf{x}_2{}^c$. Then the path between $\mathbf{x}_1$ and $\mathbf{x}_2$ is the union of the components of the backward integration from $(\mathbf{x}_1, \mathbf{x}_2)$ to $(\mathbf{x}_1{}^c, \mathbf{x}_2{}^c)$ and the shortest path between $\mathbf{x}_1{}^c$ and $\mathbf{x}_2{}^c$. Figure 12c illustrates this.

In order to get the shortest path between the components of a couple point $(\mathbf{x}_1{}^c, \mathbf{x}_2{}^c)$, we apply a pre-process to get the shortest paths between the components of all detected couple points. We integrate the separating double curves from each couple point as introduced in Section 3. This way, 8 double stream lines are emanating from a couple point. Among them, we consider all that collapse into a single point. The shortest path of them is the solution for $(\mathbf{x}_1{}^c, \mathbf{x}_2{}^c)$. Figure 13a illustrates an example where 6 of the 8 double stream lines collapse to a single point, while the remaining two get stuck in couple points. In case that all 8 separation double lines get stuck in couple points $(\mathbf{x}_{1i}{}^c, \mathbf{x}_{2i}{}^c)$, we recursively compute the shortest path as the union of the components of the integration double curve from $(\mathbf{x}_1{}^c, \mathbf{x}_2{}^c)$ to $(\mathbf{x}_{1i}{}^c, \mathbf{x}_{2i}{}^c)$ and the shortest path between $\mathbf{x}_{1i}{}^c$ and $\mathbf{x}_{2i}{}^c$. This algorithm is recursively carried out in a breadth first manner until a path for all couple points is found. Figure 13b gives an illustration.

The resulting data structure of our pre-processing is a distance-weighted sparse graph where each node represents a couple point. From each node, 8 edges

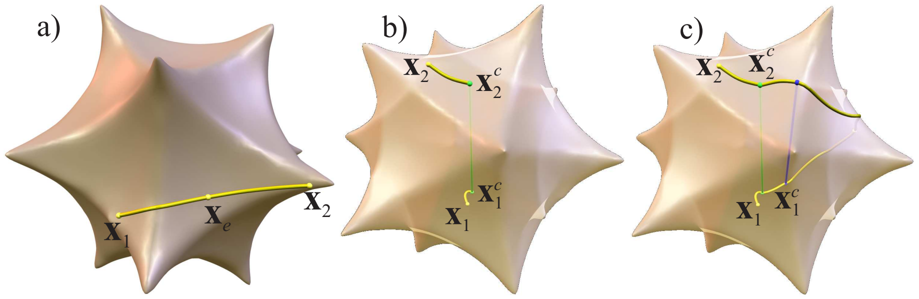

Fig. 12: a) Backward integration of $\mathbf{v}^d$ starting from $(\mathbf{x}_1, \mathbf{x}_2)$: the components of the double points collapse to a single point $\mathbf{x}_e$. b) Integration of $\mathbf{v}^d$ gets stuck in a couple point $({\mathbf{x}_1}^c, {\mathbf{x}_2}^c)$. c) The shortest path is completed by adding the shortest path between ${\mathbf{x}_1}^c$ and ${\mathbf{x}_2}^c$.

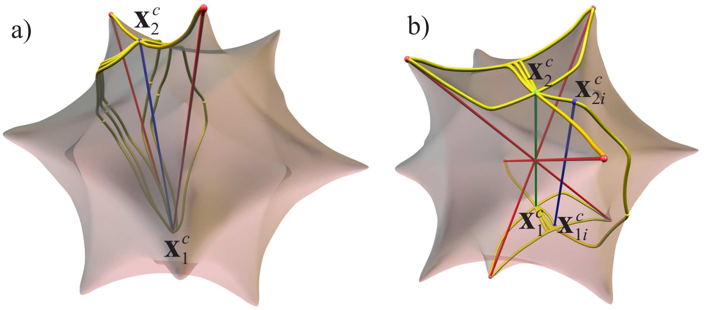

Fig. 13: a) The 8 separation double lines starting from a couple point $({\mathbf{x}_1}^c, {\mathbf{x}_2}^c)$ (blue): 6 collapse to single points. b) All separation lines from $({\mathbf{x}_1}^c, {\mathbf{x}_2}^c)$ (green) get stuck in a couple point: the shortest path is computed as the union of the components of the integrated double line from $({\mathbf{x}_1}^c, {\mathbf{x}_2}^c)$ to $({\mathbf{x}_{1i}}^c, {\mathbf{x}_{2i}}^c)$ and the shortest path between ${\mathbf{x}_{1i}}^c$ and ${\mathbf{x}_{2i}}^c$.

are leaving which represent the integrated double separation lines. These edges end either in the same node (in case that the integrated double line collapses into a point) or in another node (representing the couple point in which the integration gets stuck). If each edge is assigned with the path length of the corresponding double stream line, the problem of finding a shortest path between $({\mathbf{x}_1}^c$ and $({\mathbf{x}_2}^c$ is equivalent to finding a short loop in the graph. This way, we found solutions for all couple points for very few iteration steps.

Once the pre-process for a surface is done, the algorithm to detect a shortest path for a pair of surface points has just the cost of a numerical double stream line integration. Obviously, the cost of this depends on step size and accuracy of the chosen integration technique. Nevertheless it has a linear worst case complexity and is easily possible at interactive frame rates on standard personal computers.

Figure 14 shows two examples of computing the approximation of the shortest

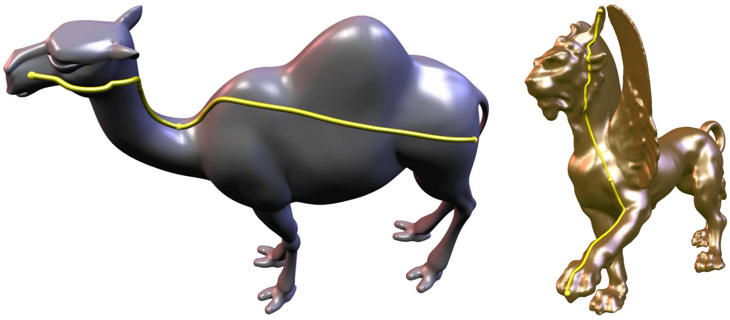

Fig. 14: Fast approximation of the shortest path between two surface points.

path between two surface points (yellow). In both examples, the backward integration of $\mathbf{v}_d$ collapsed into a single point without touching a couple point. Figure 1 (left) shows an example where the found shortest path is passing through 5 couple points.

Generally, our approach to get a shortest path between two surface points shares the advantages and disadvantages of all Greedy algorithms. It is fast, but is always possible to construct extreme examples in which the algorithm produces solutions far way from the globally optimal one. Figure 15 compares

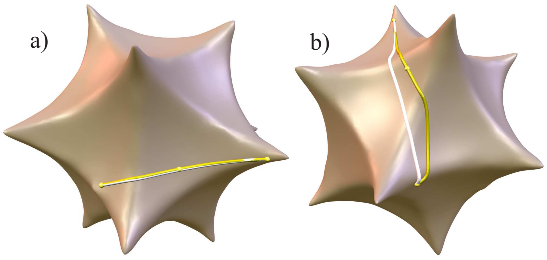

Fig. 15: Comparison between real geodesics (white lines) and our solution (yellow lines).

examples of our solutions (yellow lines) with the perfect geodesics (white lines). Figure 15a shows the coincidence between the lines while a certain difference can be observed in Figure 15b. However, the advantage of our approach is that it computes a path between two surface points only by a univariate numerical integration where estimators of real geodesics between two points have a higher complexity.

5.3 Computing stabilizing connectors between parts of a surface

Quite a number of classical statues and sculptures have lost parts because they got broken (see Figure 16a for an example). To prevent these accidents for instance during a transportation, stabilizing connectors can be included. These are static sticks which are connected to certain parts of the surface to prevent the breaking of parts. Figure 16b illustrates an example. Although the stability of a surface is a well-studied feature [6, 16], couple points provide a heuristic approach to find optimal stabilizing connectors. An "optimal" stabilizing connector should be a compromise between size, visual appearance and impact. It should be as small as possible in order to not disturb the visual impression of the sculpture, but it should stabilize the sculpture as much as possible. We believe that direct outside couple points are good candidates for stabilizing connectors because they touch the surface parallel to the normals and hence have a maximal effect of the stabilizing forces onto the surface.

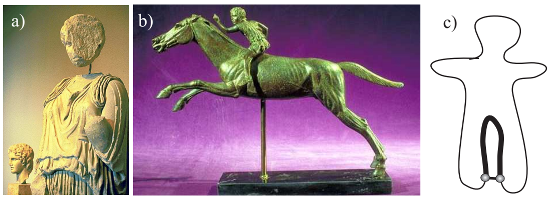

Fig. 16: a) A (real) statue with broken parts. b) A stabilizing connector for a real statue. c) A candidate couple point to be a stabilizing connector: the Euclidean distance between the points is rather short in comparison to the shortest path on the surface.

Our approach to find the optimal couple points to be used as stabilizing connectors is to consider the ratio between $\|\mathbf{x_2}^c - \mathbf{x_1}^c\|$ and the length of the shortest path between $\mathbf{x_1}^c$ and $\mathbf{x_2}^c$ for all outside direct couple points $(\mathbf{x_2}^c, \mathbf{x_1}^c)$. The smaller this ratio, the better the couple point is suited to be a stabilizing connector. A small value of $\|\mathbf{x_2}^c - \mathbf{x_1}^c\|$ ensures a small disturbance in the visual impression, while a relatively long path between $\mathbf{x_1}^c$ and $\mathbf{x_2}^c$ gives a good impact of the connector. As an example, consider the components of an outside direct couple point $(\mathbf{x_2}^c, \mathbf{x_1}^c)$ to be located on the two legs of a human statue where the position of the legs is rather parallel. Then the Euclidean distance between $\mathbf{x_1}^c$ and $\mathbf{x_2}^c$ is rather small in comparison to the shortest path between $\mathbf{x_1}^c$ and $\mathbf{x_2}^c$, hence the couple point is a good candidate for being a stabilizing connector. Figure 16c gives an illustration.

Figure 17 shows the 7 best suited couple points to be stabilizing connectors for the camel data set and the best 11 connectors for the feline data set, a front view is shown in Figure 1 (right).

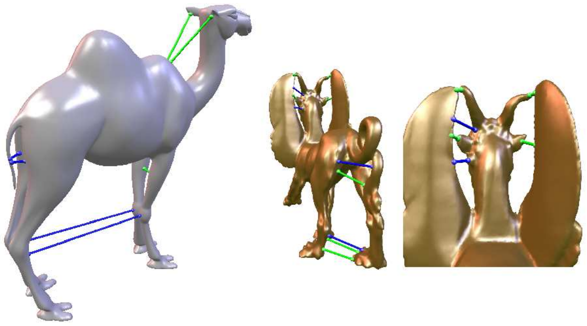

Fig. 17: Best couple points to serve as stabilizing connectors.

6 Conclusions

In this paper we have made the following contributions: We introduced the concept of couple points as a new global feature on surfaces. We applied local feature extracting techniques, namely a Morse theoretic approach on a 4D scalar field, to extract and classify couple points. We proposed a recursive algorithm to extract couple points for triangular meshes where the vertices are equipped with a normal. We applied couple points to compute the minimal and maximal distance between surfaces. We applied couple points to compute a fast approximation of the shortest geodesic path between two surface points. Finally, we proposed stabilizing connectors of a surface as appropriate couple points.

Nevertheless there is a number of open problems concerning couple points which are subject to future research. An interesting direction would be persistence of couple points under surface perturbation or surface deformation. For instance, we do not yet have control over the changes of couple points when the surfaces are continuously moved or deformed. In this case, couple points may drift on the surfaces, and they may appear and disappear at certain events. A careful study of these effects may make couple points applicable also to dynamic surfaces.

References

1. Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M.: Anisotropic polygonal remeshing. ACM Transactions on Graphics 22(3), 485–493 (Jul 2003)
2. Chen, J., Han, Y.: Shortest paths on a polyhedron. In: Symposium on Computational Geometry. pp. 360–369 (1990)
3. Culver, T., Keyser, J., Manocha, D.: Accurate computation of the medial axis of a polyhedron. In: Proc. ACM Symp. Solid Model. Appl. pp. 179–190 (1999)
4. Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Morse-smale complexes for piecewise linear 3-manifolds. In: Proc. 19th Sympos. Comput. Geom. 2003. pp. 361 – 370 (2003)
5. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical morse complexes for piecewise linear 2-manifolds. In: Proc. 17th Sympos. Comput. Geom. 2001 (2001)

6. Efimov, N.: Flächenverbiegungen im Grossen. Akademie-Verlag, Berlin (1957), (in German)
7. Ferrand, E.: On the Bennequin Invariant and the Geometry of Wave Fronts, Geometriae Dedicata 65, 219245 (1997)
8. Goldfeather, J., Interrante, V.: A novel cubic-order algorithm for approximating principal directions vectors. ACM Transactions on Graphics 23(1), 45–63 (2004)
9. Guéziec, A.: Meshsweeper: Dynamic point-to-polygonal-mesh and applications. IEEE Transactions on Visualization and Computer Graphics 7(1), 47–61 (2001)
10. Hahmann, S., Bonneau, G.P.: Smooth polylines on polygon meshes. In: Brunnett, G., Hamann, B., Müller, H. (eds.) Geometric Modeling for Scientific Visualization, pp. 69–84. Springer (2003)
11. Hahmann, S., Belyaev, A., Buse, L., Elber, G., Mourrain, B., Rössl, C.: Shape interrogation. In: de Floriani, L., Spagnuolo, M. (eds.) Shape Analysis and Structuring, chap. 1, pp. 1–52. Mathematics and Visualization, Springer, Berlin, Germany (2008)
12. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3D shapes. In: Proc. SIGGRAPH. pp. 203–212 (2001)
13. Hofer, M., Pottmann, H.: Energy-minimizing splines in manifolds. In: Proc. SIGGRAPH. pp. 284–293 (2004)
14. Kuiper, N. H.: Double normals of convex bodies, Israel J. Math. 2, 71-80 (1964)
15. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. USA 95(15), 8431–8435 (1998)
16. Minagawa, T., Rado, R.: On the infinitesimal rigidity of surfaces. Osaka Math. J. 4, 241–285 (1952)
17. Mitchell, J.: Geometric shortest paths and network optimization. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 633–701. Elsevier (1998)
18. Ni, X., Garland, M., Hart, J.: Fair morse functions for extracting the topological structure of a surface mesh. In: Proc. SIGGRAPH. pp. 613–622 (2004)
19. Petitjean, S.: A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys 34(2) (2001)
20. Pham-Trong, V., Biard, L., Szafran, N.: Pseudo-geodesics on three-dimensional surfaces and pseudo-geodesic meshes. Numerical Algorithms 26(4), 305–315 (2001)
21. Polthier, K., Schmies, M.: Straightest geodesics on polyhedral surfaces. In: Hege, H.C., Polthier, K. (eds.) Mathematical Visualization, pp. 135–150. Springer Verlag, Heidelberg (1998)
22. Rusinkiewicz, S.: Estimating curvatures and their derivatives on triangle meshes. In: 3DPVT. pp. 486–493 (2004)
23. Sheeny, D., Armstrong, C., Robinson, D.: Shape description by medial axis construction. IEEE Transactions on Visualization and Computer Graphics 2, 62–72 (1996)
24. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Transactions on Graphics 24(3), 553–560 (2005)
25. Theisel, H., Rössl, C., Zayer, R., Seidel, H.P.: Normal based estimation of the curvature tensor for triangular meshes. In: Proc. Pacific Graphics. pp. 288 – 297 (2004)
26. Xin, S.Q., Wang, G.J.: Improving Chen and Han's algorithm on the discrete geodesic problem. ACM Transactions on Graphics 28(4), 104:1–104:8 (Aug 2009)