Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Focal Surfaces of Discrete Geometry

Jingyi Yu[1], Xiaotian Yin[2], Xianfeng Gu[2], Leonard McMillan[3], and Steven Gortler[4]

[1]University of Delaware, Newark DE, USA
[2]State University of New York, Stony Brook NY, USA
[3]University of North Carolina, Chapel Hill NC, USA
[4]Harvard University, Cambridge MA, USA

Abstract

The differential geometry of smooth three-dimensional surfaces can be interpreted from one of two perspectives: in terms of oriented frames located on the surface, or in terms of a pair of associated focal surfaces. These focal surfaces are swept by the loci of the principal curvatures' radii. In this article, we develop a focal-surface-based differential geometry interpretation for discrete mesh surfaces. Focal surfaces have many useful properties. For instance, the normal of each focal surface indicates a principal direction of the corresponding point on the original surface. We provide algorithms to robustly approximate the focal surfaces of a triangle mesh with known or estimated normals. Our approach locally parameterizes the surface normals about a point by their intersections with a pair of parallel planes. We show neighboring normal triplets are constrained to pass simultaneously through two slits, which are parallel to the specified parametrization planes and rule the focal surfaces. We develop both CPU and GPU-based algorithms to efficiently approximate these two slits and, hence, the focal meshes. Our focal mesh estimation also provides a novel discrete shape operator that simultaneously estimates the principal curvatures and principal directions.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations; I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems;

1. Introduction

Surface representations are crucial to computer graphics, numerical simulation, and computational geometry. Sampled representations, such as triangle meshes, have long served as simple, but effective, smooth surface approximations. The approximation of a smooth surfaces from a sampled geometric model, whether explicit or not, requires consistent notions of first-order and second-order differential geometric attributes, such as principal curvatures and principal directions [Gri05]. Typically, differential geometric properties are derived from surface vertices, mesh connectivity, and, occasionally, by considering externally specified vertex normals. Polynomial patches are also frequently fitted to vertices and normals and then used to approximate differential surface properties. These methods can introduce ambiguities and inconsistencies, resulting in unexpected surfaces. Discrete operators have been also been introduced based on averaging Voronoi cells [MDSB03, CSM03]. However, no consensus has been reached on how to guarantee mesh smoothness and quality [GG06].

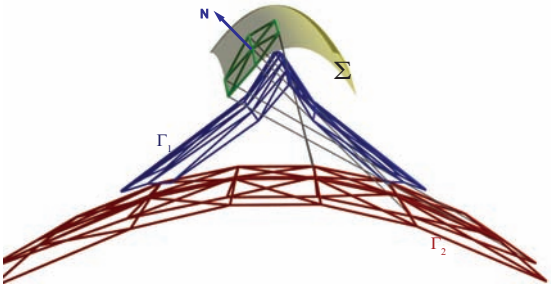

Figure 1: *A sampled mesh from a smooth surface Σ has two focal meshes Γ_1 (blue) and Γ_2 (red), each formed by the loci of the corresponding principal curvature's radii.*

Alternatively, the differential geometry of smooth three-

dimensional surfaces can be interpreted in terms of a pair of associated focal surfaces. For a smooth surface, its focal surfaces are formed by the loci of the principal radii of curvatures (Figure 1). The focal surfaces encapsulate many useful properties of the actual surface that they correspond to. Normals of the actual surface are tangent to both focal surfaces and the normal of a focal surface at the point of actual surface tangency corresponds to a principal direction on the original surface. Although the basic theory of the focal surfaces for smooth surfaces has been well explored [Koe94, Por94], very little work has been done to use focal surfaces for computing differential geometry attributes on discrete geometry.

In this paper, we present a new framework for approximating the focal surfaces of discrete meshes with known or estimated vertex normals. We employ a novel normal-ray surface representation, which locally parameterizes the surface normals about a point as rays. We show that neighboring rays form special ray congruencies due to the geometry of focal surfaces: these rays should simultaneously pass through two slits that rule the focal surfaces and correspond to the two principal directions. We then develop both CPU and GPU-based algorithms to efficiently approximate the ray congruency and, hence, the focal surfaces. Our focal surface estimation provides a new discrete shape operator that simultaneously estimates the principal curvatures and principal directions.

2. Previous Work

In many computational applications, it is common to approximate smooth surfaces with sampled representations composed of piecewise linear elements having connected boundaries. We refer to such models as surface meshes. It has long been recognized that higher-order geometric attributes are required to compute accurate shading, reflections, and other simulations. It is also commonplace for surface meshes to specify vertex normals separate from the modeling elements.

An objective of discrete differential geometry is to derive higher-order local surface properties that are simultaneously consistent with the given sampled surface mesh as well as some underlying smooth surface. Several efforts have been devoted to defining consistent differential geometric attributes for surface meshes. Many previous models are based on Cartan equations, which treat the principal directions as the orientations where the normal differentiation reaches extrema. In theory, the two principal directions can be computed by diagonalizing the second fundamental form. Intensive research has been carried out on estimating the second fundamental form of mesh surfaces via curvature tensors estimation using finite differences [Tau95, HP04], polynomial fitting [Ham93, CP03, GI04], and Voronoi cells [MDSB03, CSM03].

We present a new approach based on a different differential geometry entity– focal surfaces. In the literature, these loci of the reciprocal of the principal curvatures are interchangeably referred to as evolutes, caustics, centro-surfaces, and focal surfaces [Koe94]. Even before the invention of differential calculus, Huygens recognized that the loci of a curve's normal rays provide an enormous wealth of geometric insights. The focal surfaces alone are sufficient to characterize all of the signed-distance wavefronts emanating from the surface, from the surface itself to its Gauss map at infinity. In fact, the differential geometry of smooth surface can be completely characterized from the perspective of focal surfaces [Por94, PW01]. Thus, they subsume a wide range of higher-order properties commonly associated with surface points [PW06, TRZS04]. Recently, computer vision researchers have shown that focal surfaces are also closely related to the caustics of rays [SGN01]. For example, the focal surfaces of the reflection rays determine the unique distortions on curved mirrors [YM05]. We extend these insights of focal surfaces to discrete geometry.

Our key contributions include:

- A focal-surface-based framework for analyzing sampled geometric surfaces with associated vertex normals.
- A theory relating the normal rays, the focal surfaces, and the two-slit ray structure.
- A new class of CPU and GPU-based algorithms to estimate discrete differential geometry using focal surfaces.

Before proceeding, we explain our notation. Superscripts, such as S^x, S^y, and S^z represent the x and y and z component of S. Subscripts, such as f_x and f_y, represent the first-order partial derivatives of f with respect to x and y. Similarly, f_{xx} refers to the second-order partial derivative of f with respect to x.

3. Focal Surfaces and Focal Meshes

Traditionally, focal surfaces are modeled in terms of the principal curvature radii. Given a smooth locally parameterized surface

$$\Sigma = S(u,v) = [x(u,v), y(u,v), z(u,v)], \tag{1}$$

sample points on the two focal surfaces of S can be expressed as

$$\begin{aligned} \Gamma_1(u,v) &= S(u,v) - \frac{1}{K_1(u,v)} N(u,v) \\ \Gamma_2(u,v) &= S(u,v) - \frac{1}{K_2(u,v)} N(u,v) \end{aligned} \tag{2}$$

where N is the normal at S on Σ, and K_1 and K_2 are the two principal curvatures at S on Σ. We call Σ the *base surface* and Γ_1 and Γ_2 the *focal surfaces*. We assume $K_1 \geq K_2$ and we use $\vec{f}_1$ and $\vec{f}_2$ to represent the two principal directions at S. Except for parabolic points and planar points where one or both principal curvatures are zero, each point $S(u,v)$ on the base surface is associated with two focal points. Thus, generally, a smooth base surface has two focal surface sheets, $\Gamma_1(u,v)$ and $\Gamma_2(u,v)$.

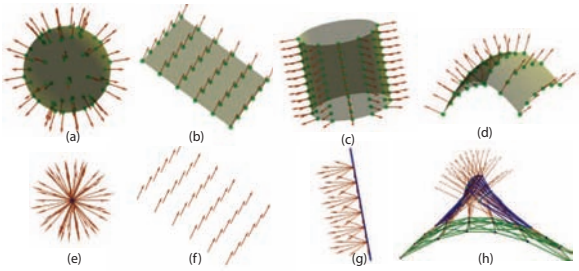

Figure 2: *Generally, smooth surfaces have two focal surfaces (h). Degeneracies occur at parabolic points (c) where only one of the focal surface exists, at umbilic points (a) where the focal surface converges into a common point, and at flat points (b), where the local surface is flat. The four types of focal surfaces can also be modeled with normal rays: normal rays of umbilic points pass through a single point (e); normal rays of flat points are parallel (f); normal rays of parabolic points intersect along a single line (g); and normal rays of generic points pass through exactly two parallel lines (h).*

3.1. Geometric Properties

Several focal-surface properties relate to the base surface's differential geometry. Here, we briefly summarize some of these properties relevant to this paper. A complete characterization can be found in [Por94] and [PW01].

Property 1: Normal Tangency The base surface normal N is tangent to both focal surfaces at $\Gamma_1(u,v)$ and $\Gamma_2(u,v)$.

Property 2: Principal Directions The normal of the two focal surfaces $\Gamma_1(u,v)$ and $\Gamma_2(u,v)$ correspond to the two principal directions of the base surface at $S(u,v)$. In Appendix A, we sketch a brief proof to Properties 1 and 2. See also [Do 76] (p. 210 ex. 9).

Property 3: Continuity The focal surfaces of polynomial base surfaces are two degrees less continuous. Thus, a smooth surface may have focal surfaces that are only C^0 continuous.

While the focal surfaces of smooth surfaces are well understood, little work has been done on estimating focal surfaces of surface meshes. This is probably because focal surfaces are directly related to the principal curvature directions. Thus, it is as difficult to compute the focal surfaces as it is to estimate curvatures along with their associated principal directions. A notable exception is the generalized focal surfaces introduced by Hahmann [HH92, Hah99], which are similar in spirit to our goals, but different in their approach. Hahmann's focal surface notion is developed as parameterized evolutes of uniform distance fields from the surface, whereas ours is determined by the structural constraints represented by a piecewise linear (PL) focal mesh.

3.2. Degeneracies

In general, a smooth base surface has two focal surfaces. However, degeneracies occur at special locations on the base surface. For example, at the points where exactly one principal curvature is zero, the corresponding focal-mesh point will lie at infinity (i.e. an assymptote of the focal sheet). These points lie on the parabolic curves of the surface and the second focal surface forms a cylindrical axis as shown in Figure 2(c) and 2(g). There may also exist points where both principal curvatures are zero. These surfaces are locally flat (Figure 2(b) and 2(f)) and are non-generic.

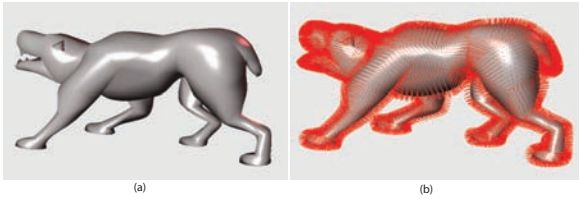

Figure 3: *A surface can be represented as a manifold of rays. (a) A smooth surface. (b) The normal ray manifold of the surface. Each normal ray is marked in red.*

Furthermore, points on the base surface can have the same curvature in all directions. These points correspond to the umbilics, around which local surface is sphere-like. The focal mesh formed by vertices around an umblic point can shrink into a point, as is shown in Figure 2(a) and 2(e). The presence of umbilic points does not affect our focal-mesh-based representation; we simply treat each focal point as different vertices on each focal mesh.

It is important to note that for polynomial surfaces their corresponding focal surfaces are 2 degrees less smooth. This suggests piecewise linear focal meshes are effective representations for focal surfaces. In addition, many higher order smooth surfaces have ridges or valleys, thus, their focal surfaces cusp in 3D and have only C^0 continuity as shown in Figure 2(h). This indicates that piecewise linear focal surfaces constructed from discrete meshes are reasonable choices and they are capable of modeling focal surface cusps, thus, preserving ridge-valley lines of the base mesh.

4. A Normal-Ray Focal Surface Model

Focal meshes have many appealing properties. However, using conventional differential geometry, it is equally difficult to compute focal meshes as it is to compute principal curvatures and directions. Many existing discrete shape operators use different routines to separately compute the curvature values and the curvature directions. In this section, we propose a unified tool that simultaneously estimates the principal curvatures and directions, as well as the focal meshes from a sampled surface.

Our method is based on the observation that the vertices and normals of the original surface can be represented as rays, where each ray has its origin at a vertex and direction given by its normal. A continuous surface can then be

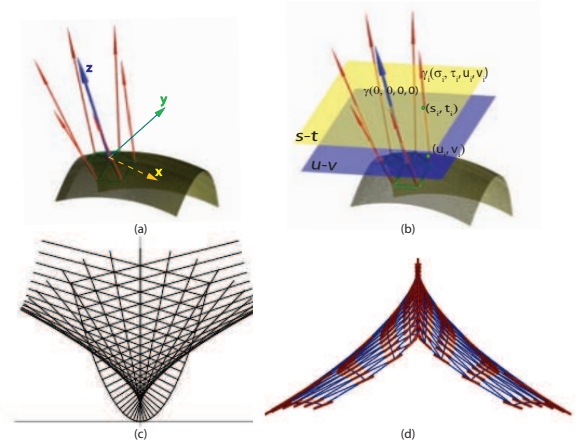

Figure 4: *Estimating focal meshes using the normal ray model. (a) We orient the local frame to align $z = 0$ plane (uv plane) with the surface tangent plane at P. (b) We choose the second plane to be $z = 1$ (st plane). Each neighboring normal ray can be parameterized by its intersections with the two planes as $[\sigma, \tau, u, v]$. (c) Focal surfaces (curve) formed by the foci of the normal rays of a parabolic surface. (d) Neighboring normal rays are constrained by the slits (red) that rule the focal surfaces (blue).*

mapped to a 2D manifold of rays, as is shown in Figure 3(b). The Normal Tangency Property reveals that focal surfaces are formed by the envelop of the normal rays as shown in Figure 4(c). Thus, locally, focal surfaces can alternatively be viewed as the foci of nearby normal rays.

In geometric optics, the ray foci sweep out the caustics of reflective or refractive surfaces [SGN01, YM05]. Our goal is to develop a new technique to robustly estimate the foci of discretely sampled normal rays, and hence, the focal surfaces. This requires computing the local ray structure in the ε neighborhood about a specific surface vertex. As follows, we first present a new normal-ray parametrization and then show how to compute the foci of normal rays under this parametrization.

4.1. Normal Ray Parametrization

At each surface vertex P, we orient the local frame to align $z = 0$ plane with the tangent plane at P. Thus, the normal at P corresponds to the z direction as shown in Figure 4(a). We assume P is the origin of $z = 0$ plane and we call this plane the *uv* plane. We position the second *st* plane at $z = 1$ parallel to the *uv* plane. At each vertex of the surface, we parameterize neighboring normals by their intersections with the *st* and *uv* plane at $[s,t,1]$ and $[u,v,0]$, as shown in Figure 4(b). The normal direction D can then be computed as $D = [\sigma, \tau, 1]$, where $\sigma = s - u$ and $\tau = t - v$. In this paper, we use $[\sigma, \tau, u, v]$ to parameterize all normal rays.

Under this ray parametrization, each normal ray maps to a point in a four-dimensional *ray* space. A patch of the surface normals maps to a 2D manifold in this ray space. In order to compute the focal surfaces at each vertex on the mesh, we examine how nearby normal rays behave on smooth surfaces. Notice the local surface is a height field (Monge patch) $z(x,y)$ under the new parametrization. Hence, the normal ray direction for every point $[x,y,z(x,y)]$ can be computed as $D = [-z_x, -z_y, 1]$. The $[u,v]$ coordinate of the ray can be computed by intersecting the normal ray with the *uv* plane as

$$\begin{aligned} [\sigma, \tau, 1] &= [-z_x, -z_y, 1] \\ [u, v, 0] &= P - z \cdot D = [x + z z_x, y + z z_y, 0] \end{aligned} \tag{3}$$

4.2. Computing Normal Ray Focus

Traditionally, the foci of rays can be computed using the Jacobian method [SGN01]. Assume each ray can be parameterized in (x,y) as $r(x,y) = S(x,y) + \lambda D(x,y)$, where S represents the origin of the ray and D represents the direction, the Jacobian method finds λ for each ray r that satisfies:

$$\begin{vmatrix} S_x^X + \lambda \cdot D_x^X & S_y^X + \lambda \cdot D_y^X & D^X \\ S_x^Y + \lambda \cdot D_x^Y & S_y^Y + \lambda \cdot D_y^Y & D^Y \\ S_x^Z + \lambda \cdot D_x^Z & S_y^Y + \lambda \cdot D_y^Y & D^Z \end{vmatrix} = 0 \tag{4}$$

We have shown [YM05] that under $[\sigma, \tau, u, v]$ ray parametrization, the Jacobian method can be alternatively derived by computing the triangular area formed by the intersections of the three rays r, $r + r_x$, and $r + r_y$ on $z = \lambda$ plane. When this area goes to zero, the three rays will focus at a *line slit*.

Therefore, we set out to find λ that satisfies:

$$Area(\lambda) = \begin{vmatrix} u + \lambda\sigma & v + \lambda\tau & 1 \\ (u + u_x) + \lambda(\sigma + \sigma_x) & (v + v_x) + \lambda(\tau + \tau_x) & 1 \\ (u + u_y) + \lambda(\sigma + \sigma_y) & (v + v_y) + \lambda(\tau + \tau_y) & 1 \end{vmatrix} = 0 \tag{5}$$

Notice that Equation (5) is quadratic in λ. Thus, it can be written as

$$A\lambda^2 + B\lambda + C = 0 \tag{6}$$

where

$$\begin{aligned} &A = \sigma_x \tau_y - \sigma_y \tau_x, \quad B = \sigma_x v_y - \sigma_y v_x - \tau_x u_y + \tau_y u_x \\ &C = u_x v_y - u_y v_x \end{aligned} \tag{7}$$

We call Equation (6) the *normal-ray characteristic equation*.

The two solutions to this quadratic equation correspond to the foci of the normal rays, and therefore, the focal surfaces. Furthermore, Equation (6) not only computes the position of the focal surfaces but also reveals a ruling on the focal surface, as is shown in Figure 4(d). This is because the three rays focus on two *slits* that have orthogonal directions. We show in Appendix B that the directions of the two slits correspond to the two principal direction on the base surface.

4.3. Characteristic Equation

The characteristic equation (6) computes the depth of the focal surfaces as λ_1 and λ_2, where $\lambda_1 = -\frac{1}{\kappa_1}$ and $\lambda_2 = -\frac{1}{\kappa_2}$. Since the coefficients of a quadratic equation must satisfy

$$\lambda_1 + \lambda_2 = -\frac{B}{A}, \quad \lambda_1 \cdot \lambda_2 = \frac{C}{A}, \tag{8}$$

we can directly derive the mean and the Gaussian curvature in terms of the coefficients of the characteristic equation without solving for λ_1 and λ_2:

$$\begin{aligned} K &= \kappa_1\kappa_2 = \frac{1}{\lambda_1\lambda_2} = \frac{A}{C} \\ 2H &= \kappa_1 + \kappa_2 = -\frac{1}{\lambda_1} - \frac{1}{\lambda_2} = \frac{B}{C} \end{aligned} \tag{9}$$

Notice at the points whose the Gaussian curvature K is zero, we must have $A = 0$ and the characteristic equation degenerates to a linear equation with at most one solution. These points correspond to the parabolic points of the base surface, therefore, only one focal surface exists. If the mean curvature H is also zero, then we must have $A = 0$ and $B = 0$, and therefore, Equation (6) has no solution. This happens when the surface is locally flat and has no focal surface. Finally, we can compute the discriminant $\Delta = B^2 - 4AC$. If $\Delta = 0$, then the quadratic characteristic equation has double roots. This indicates the surface has two identical principal curvatures and the surface point is umbilic.

5. Estimating Focal Surface on Discrete Meshes

The normal-ray characteristic equation (6) reveals that under two plane parametrization, nearby normal rays will pass through two slits that rule the two focal surfaces and are parallel to the principal directions. In this section, we show how to apply the local two-slit model to construct the focal meshes from discrete mesh surfaces. Our method provides a discrete shape operator that simultaneously estimates the principal curvatures and principal directions.

Given a sampled surface with vertex normals, we first map the local neighborhood about each vertex to normal rays. Our goal is to fit the optimal two-slit model at each vertex, where optimality is defined such that the neighboring rays lie close to the two slits.

Notice, the two-slit structure of normal rays is subject to the following three constraints. First, the vertex normal and the two slits should be perpendicular to each other. Second, the two slits should be perpendicular to each other (since they correspond to the two principal directions). Finally, the vertex normal ray should pass through both slits.

To satisfy the first constraint, at each vertex P, we use a local $[\sigma,\tau,u,v]$ normal ray parametrization. Thus, the normal ray γ at P must have ray-space coordinates $[0,0,0,0]$. We then compute all normal ray coordinates for all vertex normals within the 1-ring neighborhood of γ as $\gamma_i[\sigma_i,\tau_i,u_i,v_i]$.

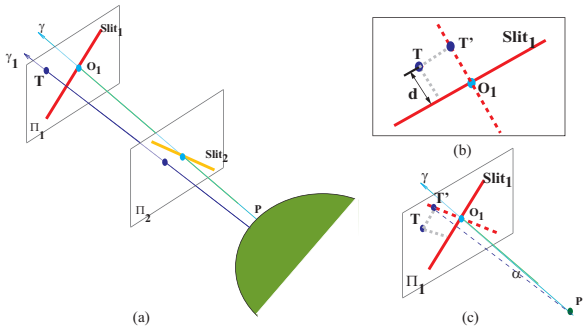

Figure 5: *Estimating focal surfaces via normal ray congruency. (a) Neighboring normal rays around γ should lie close to two slits. To find the optimal two slits, we can minimize the distance from the neighboring normal ray γ_1 to the slit (b). We can also minimize the angular difference between γ_1 and γ (c).*

Next, we parameterize each slit pair that is perpendicular to the normal ray and parallel to the uv plane as $[q_1^x, q_1^y, \lambda_1] + \mu[\cos(\theta_1), \sin(\theta_1), 0]$ and $[q_2^x, q_2^y, \lambda_2] + \mu[\cos(\theta_2), \sin(\theta_2), 0]$.

By the second constraint, the two slits must be perpendicular to each other. Thus, we have $\theta_2 = \theta_1 + \frac{\pi}{2}$. Finally, to guarantee that γ passes through both slits, each slit l_j should pass through the origin $O_j[0,0,\lambda_j]$ on plane $z = \lambda_j$, $j = 1,2$. Therefore, we can rewrite the two slits as

$$\begin{aligned} l_1 &= [0,0,\lambda_1] + \eta[\cos(\theta_1), \sin(\theta_1), 0] \\ l_2 &= [0,0,\lambda_2] + \eta[-\sin(\theta_1), \cos(\theta_1), 0] \end{aligned} \tag{10}$$

Our goal is to find the two slits l_1 and l_2 where neighboring normal rays γ_i form an envelop. Recall that l_1 and l_2 can be equivalently represented in terms of θ_1, λ_1, and λ_2 as shown in equation (10), we can formulate this optimization as a least square problem:

$$\min_{\lambda_1,\lambda_2,\theta_1} \sum_{\gamma_i \in 1 \text{ ring neighbor of } \gamma} (E^2(\gamma_i, l_1) + E^2(\gamma_i, l_2)) \tag{11}$$

where $E^2(\gamma_i, l_j)$ measures the closeness between the normal ray γ_i to slits l_j.

One way to measure E is to compute the intersection point T of ray γ_i with $z = \lambda_j$ plane and then calculate the distance d from T to l_j as shown in Figure 5(b). Notice that $T = [u_i + \lambda_j\sigma, v_i + \lambda_j\tau, \lambda_j]$, thus, $E(\gamma_i, l_j)$ can be computed as point-to-line distance:

$$\begin{aligned} E^2(\gamma_i, l_1) &= (\cos(\theta_1)(v_i + \lambda_1\tau) - \sin(\theta_1)(u_i + \lambda_1\sigma))^2 \\ E^2(\gamma_i, l_2) &= (-\sin(\theta_1)(v_i + \lambda_2\tau) - \cos(\theta_1)(u_i + \lambda_2\sigma))^2 \end{aligned} \tag{12}$$

However, λ_1 or λ_2 can be very large around the parabolic curves. This could lead to numeric problems when using classical non-linear optimizations. Therefore, we choose to measure the angular difference.

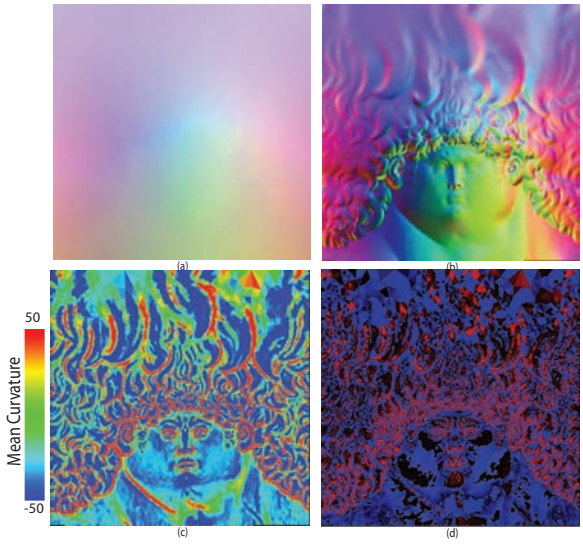

Figure 6: *GPU-based Focal Surface Estimation. (a) and (b) are the geometry and the normal image of the David Head model. (c) and (d) are the estimated mean and Gaussian curvatures. The red color on (d) represents the hyperbolic points ($K < 0$), the blue represents the elliptic points ($K > 0$), and the black represents the parabolic points ($K \approx 0$).*

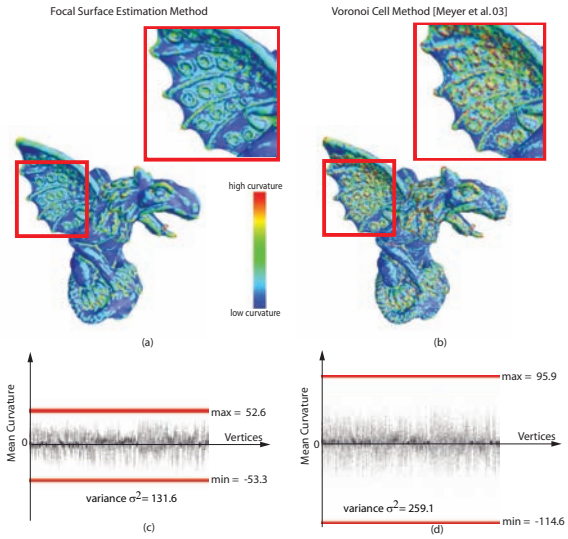

Figure 7: *The color-coded mean curvature image illustrates that the focal surface curvature estimation (a) is less sensitive to mesh connectivity than [MDSB03], especially shown on the wings of the gargoyle model. Plots (c) and (d) compare the distributions of mean-curvature estimates using the focal surface algorithm and the Voronoi edge method.*

Notice that l_1, l_2 and γ define the principal axes at P. For each intersection point T on plane $z = \lambda_1$, we project $\overline{O_1T}$ onto the second principal direction l_2 as $\overline{O_1T'}$ as shown in Figure 5(c). We then measure the angular difference between γ and $\overline{PT'}$ as

$$|\tan\alpha| = |\frac{|\overline{O_1T'}|}{\lambda_1}| = |\frac{\cos(\theta_1)(v_i+\lambda_1\tau) - \sin(\theta_1)(u_i+\lambda_1\sigma)}{\lambda_1}| \tag{13}$$

Substituting $\lambda_1 = -\frac{1}{\kappa_1}$ and $\lambda_2 = -\frac{1}{\kappa_2}$ into Equation (13), we have

$$\begin{aligned} E^2(\gamma_i, l_1) &= ((u_i\kappa_1 - \sigma_i)\sin(\theta_1) - (v_i\kappa_1 - \tau_i)\cos(\theta_1))^2 \\ E^2(\gamma_i, l_2) &= ((u_i\kappa_2 - \sigma_i)\cos(\theta_1) + (v_i\kappa_2 - \tau_i)\sin(\theta_1))^2 \end{aligned} \tag{14}$$

When κ_i goes to infinity, equation (14) may still cause similar numerical problems as (12). However, large curvatures indicate rapid normal changes across neighboring vertices. for most smooth mesh surfaces, it is rarely the case. However, nearly flat patches that map to infinite λ often occur. Thus, we choose to use metric (14) to optimization equation (11).

To find the optimal $\{\kappa_1, \kappa_2, \theta_1\}$, we use the Levenberg-Marquardt optimization to minimize equation (11). To avoid getting trapped in a local minimum, we choose a near optimal initial condition. Notice, given a fixed θ_1, equation (14) is quadratic in κ_1 and κ_2 and equation (11) has a global optimal solution. Thus, we sample several θ_1 and find the corresponding κ_1 and κ_2 as the initial conditions and choose the one with the minimum error.

5.1. GPU-based Approximation

The Levenberg-Marquardt optimization robustly estimates both principal curvatures and directions. However, it converges slowly and is not suitable for real-time applications. Here we present a GPU-based implementation that extends the two slit algorithm for real-time curvature estimation.

Recall that the normal ray characteristic equation (6) only requires three neighboring rays. Therefore, a triangular mesh, we can simply solve equation (6) for each triangle using the three corresponding normal rays at the vertices. This results in a per-triangle-based estimation of the focal mesh which maintains the same connectivity as the underlying mesh. Furthermore, if we only need to compute the mean or the Gaussian curvature, we can directly use the A, B, and C coefficients of the characteristic equation shown in equation (9), and the quadratic equation (6) need not be solved.

However, if the mesh is densely triangulated, computing per-triangle-based focal surfaces can still be very expensive. Therefore, we present an image-space approximation using a two-pass GPU algorithm. In the first pass, the geometry is rasterized into two textures, one storing the position of the geometry per pixel, the second storing the normal of the geometry per pixel. At the second pass, we use the fragment shader to fetch neighboring sets of three pixels from both textures. We then map them to normal rays under the local

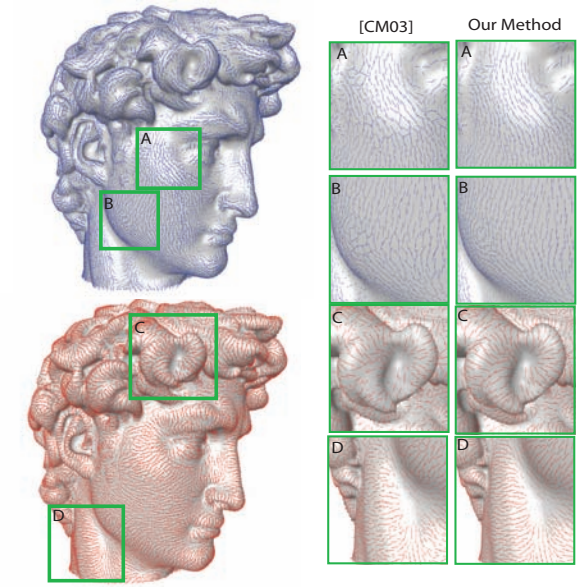

Figure 8: *Left: The estimated min (blue) and max principal curvature direction (red) using our focal surface algorithm. Right: We compare our method with [CSM03] on different parts of the model.*

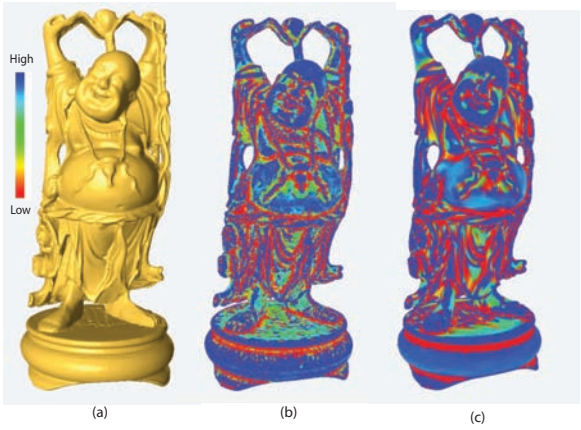

Figure 9: *Estimating curvatures on noisy meshes. We apply the focal surface algorithm on a dense Buddha model with over 1M triangles (a) and estimate the mean curvature field (b). The quality of the estimation can be further improved by applying the optimization to a two-ring neighborhood of rays (c).*

$[\sigma, \tau, u, v]$ parametrization and solve the corresponding characteristic equation.

The quality of the GPU based algorithm can further be improved by using the geometry and the normal images of a surface [GGH02] to achieve a more uniform spatial and angular sampling. In Figure 6, we show the estimated mean and Gaussian curvature using the geometry image of the David Head model. However, the GPU-based approach is less accurate than the CPU-based optimization. In particular, since the focal surfaces can lie very far away from the base surface, the estimated z values can be very large, and may challenge the precision and the dynamic range of the GPU's floating-point representation.

6. Results and Discussions

We have compared our focal surface approximation algorithm with two popular discrete differential shape operators. We implemented the Voronoi-edge algorithm [MDSB03] and a modified version of [CSM03] based on the code of [All].

Accuracy. We compare both simple analytical surfaces and complex 3D models. For simple analytical surfaces such as the torus, we find all three methods perform well, although [MDSB03] generates a slightly better estimates. On complex models such as the gargoyle, the focal surface method performs best in the sense that it gives the smoothest and most consistent estimations of the principal curvatures, as shown in Figure 7. Notice, at the rings on the wing of the gargoyle, [MDSB03] generates discontinuous and noisy mean curvatures. Our method produces much smoother mean-curvature estimates. In Figure 7(c), we plot the range of the estimated mean curvatures using both methods for comparison. Our estimates lie in a narrower band than Meyer's method. This is because the Voronoi edge approach relies heavily on the connectivity of the mesh and uses only vertex positions while our method uses both vertex position and normals and does not require connectivity constraints. We also compared our principal direction estimation with [CSM03] on the head model of Michelangelo's David, which consists of 25K vertices and 50K faces. In Figure 8, we plot the estimated principal directions. Both methods generate reasonably smooth principal direction fields, although ours is smoother in areas of high curvature and in hyperbolic regions, such as the eye, the hair, the chin, and the neck regions.

Speed. The performance of the ray-foci estimation algorithm depends mainly on the size of the data and the valence of the vertices. Compared with the Voronoi edge methods, our CPU-based algorithm is slower because it requires non-linear optimization to find the optimal two slits. On a Dell Precision PWS670 with 3.2 GHz CPU, we experienced computation times of 0.66 millisecond per vertex when applying the Levenberg-Marquart optimization to the gargoyle model (Figure 7) with 100K faces and an average valence of 6. The Voronoi cell method [MDSB03] takes 7 seconds to complete while ours takes 60 seconds (33 seconds in non-linear optimization and 27 seconds in mesh processing). In the example of computing principal directions on the David Head model, our algorithm took 16 seconds whereas the Voronoi-edge method [CSM03] takes about 2 seconds. Our GPU-based focal surface estimation, however, is highly efficient. On an NVidia GeForce7800, our algorithm achieves 105 fps

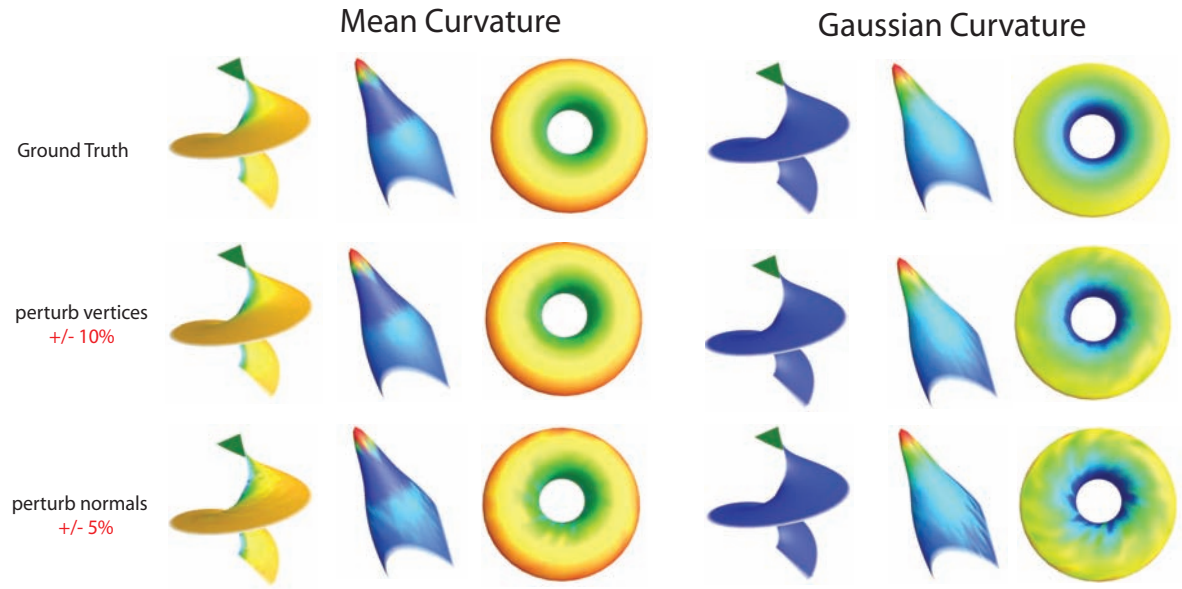

Figure 10: *Top row: the ground truth mean and Gaussian curvature of a helix, a pear-shaped surface, and a torus. Mid row: we perturb the vertices of each model by ±10% but keep the original normals. Bottom row: we perturb the normals of each model by ±5% but keep the original vertices. Our focal surface approximation algorithm robustly estimates the curvature fields despite the high noise level in both vertices and normals.*

for computing the Gaussian and mean curvature and 70 fps for the focal surfaces on the David Head model with an image resolution of 512x512.

Robustness. One limitation of our work is that our focal surface estimation algorithm requires the vertex normals. For analytical surfaces, we associate exact normals at each vertex. For scanned surfaces, we approximated the normal at each vertex by averaging the face normals of the triangles sharing the vertex. Although more sophisticated normal estimation algorithm such as [Max99] could be used, we found that simple schemes were sufficient in our experiments. We have also observed that we can improve the smoothness of the estimated curvature fields by optimizing all normal rays over a 2-ring neighborhood of each vertex as shown in Figure (9).

We have tested how noisy vertices and normals affect the quality of our estimation. In Figure 10, we perturb both the vertices and the normals with random noises. Our method is able to produce smooth and accurate Gaussian and mean curvature fields despite high noise levels. We have also found our method is more sensitive to noise in normals than in vertices.

Finally, the quality of our focal-surface-based estimation does not rely on mesh connectivity. This can be particularly important when approximating curvatures on complex models with noisy vertex connectivity, such as the Gargoyle model. In our experiment, Voronoi cell algorithms can be sensitive to mesh connectivity especially in regions where the concavity or the convexity of local surfaces flip due to triangulation. For example, around the rings of the gargoyle model, the estimated mean curvatures can flip signs across the neighboring triangles using [MDSB03] as shown in Figure 7 whereas our method maintains consistent curvature fields under the same triangulation.

7. Conclusions and Future Work

We have presented a new framework to interpret the differential geometry of smooth three-dimensional surfaces in terms of a pair of associated focal surfaces. Focal surfaces have many useful properties. For instance, the normal of each focal surface indicates a principal direction of the corresponding point on the original surface. We have employed a novel normal-ray surface representation, which locally parameterizes the surface normals about a point as rays. We have shown how to construct consistent piecewise linear focal surfaces by computing the congruency of normal rays from a discrete mesh. We have developed both CPU and GPU-based algorithms to efficiently approximate the ray congruency and hence, the focal meshes.

Our focal surface estimation provides a new discrete shape operator that simultaneously estimates the principal curvatures and principal directions. We have observed that, starting from the same mesh without vertex normals, our focal-surface-based algorithm computes smoother principal curvature and principal direction fields than the Voronoi cell approaches [MDSB03, CSM03]. This is because our method

imposes a stronger differential geometry constraint, i.e., the consistency of the focal surface, a higher-order differential attribute than the normals and the vertices.

In the future, we plan to explore how to use the estimated PL focal surfaces to model smooth surfaces. We also plan to compare our method with the recently proposed integral invariants algorithm [YLHP06] and the midedge normal shape operator [GGRZ06]. Finally, we would like to investigate how to use focal surfaces to fix vertex normals of scanned surfaces.

References

[All] ALLIEZ P.: Estimating curvature tensors on triangle meshes. http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/demos/curvature/

[CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), Eurographics Association, pp. 177–187.

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted delaunay triangulations and normal cycle. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry* (New York, NY, USA, 2003), ACM Press, pp. 312–321.

[Do 76] DO CARMO M.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[GG06] GATZKE T., GRIM C.: Estimating curvature on triangular meshes. In *International Journal of Shape Modeling* (2006), vol. 12.

[Gri05] GRINSPUN E.: Discrete differential geometry: an applied introduction. In *ACM SIGGRAPH 2005 Courses* (2005).

[GGRZ06] GRINSPUN E., GINGOLD Y., REISMAN J., ZORIN D.: Computing discrete shape operators on general meshes. In *Eurographics (Computer Graphics Forum)* (2006), vol. 25.

[GI04] GOLDFEATHER J., INTERRANTE V.: A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph. 23*, 1 (2004), 45–63.

[GGH02] GU X., GORTLER S., HOPPE H.: Geometry images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 355–361.

[Hah99] HAHMANN S.: Visualization techniques for surface analysis. In *C. Bajaj (ed.): Advanced Visualization Techniques, John Wiley,* (1999).

[Ham93] HAMANN B.: Curvature approximation for triangulated surfaces. *G. Farin et al., editor, Geometric Modelling* (1993), 139–153.

[HH92] HAGEN H., HAHMANN S.: Surface interrogation algorithms. In *IEEE Proceedings Visualization* (1992), pp. 70–76.

[HP04] HILDEBRANDT K., POLTHIER K.: Anisotropic filtering of non-linear surface features. *Computer Graphics Forum 23*, 3 (2004), 04–25.

[Koe94] KOENDERINK J. J.: *Solid Shape*. MIT Press, Cambridge, 1994.

[Max99] MAX N.: Weights for computing vertex normals from facet normalss. *Journal of Graphics Tools 4*, 2 (1999), 1–6.

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III* (2003), 35–58.

[Por94] PORTEOUS I. R.: *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge University Press, Cambridge, 1994.

[PW01] POTTMANN H., WALLNER J.: *Computational Line Geometry*. Springer, 2001.

[PW06] POTTMANN H., WALLNER J.: The focal geometry of circular and conical meshes. In *Geometry Preprint 163* (2006), TU Wien.

[SGN01] SWAMINATHAN R., GROSSBERG M. D., NAYAR S. K.: Caustics of catadioptric cameras. In *Proc. of IEEE International Conference on Computer Vision* (2001).

[Tau95] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of the Fifth International Conference on Computer Vision* (1995), IEEE Computer Society, p. 902.

[TRZS04] THEISEL H., ROSSL C., ZAYER R., SEIDEL H.-P.: Normal based estimation of the curvature tensor for triangular meshes. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 288–297.

[YLHP06] YANG Y.-L., LAI Y.-K., HU S.-M., POTTMANN H.: Robust principal curvatures on multiple scales. In *Proceedings of 4th Eurographics Symposium on Geometry Processing (2006)* (2006), pp. 223–226.

[YM05] YU J., MCMILLAN L.: Modelling reflections via multiperspective imaging. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 117–124.

Appendix A: Proof To Property 1 and 2

To prove Property 1 and 2, we compute the derivative of Γ_1 along the first principal direction $\vec{f}_1$:

$$\begin{aligned} D_{\vec{f}_1}\Gamma_1 &= D_{\vec{f}_1}S - \frac{1}{K_1}D_{\vec{f}_1}\vec{N} + \frac{1}{K_1^2}D_{\vec{f}_1}(K_1)\vec{N} \\ &= \vec{f}_1 - \frac{1}{K_1}K_1\vec{f}_1 + \frac{1}{K_1^2}D_{\vec{f}_1}(K_1)\vec{N} = \frac{1}{K_1^2}D_{\vec{f}_1}(K_1)\vec{N} \end{aligned} \quad (15)$$

Since $D_{\vec{f}_1}\Gamma_1$ is tangent to Γ_1, $\vec{N}$ must be tangent to Γ_1. Similarly, $\vec{N}$ is also tangent to Γ_2.

Next, we compute the derivative of Γ_1 along the second principal direction $\vec{f}_2$:

$$\begin{aligned} D_{\vec{f}_2}\Gamma_1 &= D_{\vec{f}_2}S - \frac{1}{K_1}D_{\vec{f}_2}\vec{N} + \frac{1}{K_1^2}D_{\vec{f}_2}(K_1)\vec{N} \\ &= \vec{f}_2 - \frac{1}{K_1}K_2\vec{f}_2 + \frac{1}{K_1^2}D_{\vec{f}_2}(K_1)\vec{N} \end{aligned} \quad (16)$$

Thus, $D_{\vec{f}_2}\Gamma_1$ is a linear combination of $\vec{N}$ and the second principal direction $\vec{f}_2$. Since $D_{\vec{f}_2}\Gamma_1$ and $\vec{N}$ are both tangent to Γ_1, $\vec{f}_2$ must also be tangent to Γ_1. Furthermore, since $\vec{N}$, $\vec{f}_1$, and $\vec{f}_2$ form a frame, $\vec{f}_1$ must be the normal of Γ_1. Similarly, $\vec{f}_2$ must be the normal of Γ_2.

Appendix B: The Slits and the Principal Directions

Lemma 1. *Each slit l derived from the ray characteristic equation (6) is tangent to its corresponding focal surfaces Γ.*

Proof. To compute the direction of l, we first intersect the three generator rays r, $r+r_x$, and $r+r_y$ with $z=\lambda$ plane, where λ is the solution to equation (6). The three intersection points can be computed as:

$$\begin{aligned} T_1 &= [u+\lambda\sigma, v+\lambda\tau, \lambda] \\ T_2 &= [(u+u_x)+\lambda(\sigma+\sigma_x), (v+v_x)+\lambda(\tau+\tau_x), \lambda] \\ T_3 &= [(u+u_y)+\lambda(\sigma+\sigma_y), (v+v_y)+\lambda(\tau+\tau_y), \lambda] \end{aligned} \quad (17)$$

Since λ corresponds to the depth where T_1, T_2, and T_3 lie on a line, we only need to use two of them to compute the direction of l. Without loss of generality, assume T_1 and T_2 do not coincide, the direction of l is

$$\overline{T_2T_1} = [u_x+\lambda\sigma_x, v_x+\lambda\tau_x, 0] \quad (18)$$

To prove direction $\overline{T_1T_2}$ is the tangent direction of Γ, we show the two tangent directions, $\vec{\Gamma}_x$ and $\vec{\Gamma}_y$, and $\overline{T_2T_1}$ are co-planar. We compute the determinant of $[\vec{\Gamma}_x, \vec{\Gamma}_y, \overline{T_2T_1}]^T$ as:

$$\begin{vmatrix} u_x+\lambda_x\sigma+\lambda\sigma_x & v_x+\lambda_x\tau+\lambda\tau_x & \lambda_x \\ u_y+\lambda_y\sigma+\lambda\sigma_y & v_y+\lambda_y\tau+\lambda\tau_y & \lambda_y \\ u_x+\lambda\sigma_x & v_x+\lambda\tau_x & 0 \end{vmatrix} \quad (19)$$

It is easy to verify that equation (19) is the same to the LHS of equation (5). Since the ray characteristic equation is zero for λ on the focal surface, equation (19) must also be zero. Therefore, $\overline{T_2T_1}$ must be tangent to Γ. □

Lemma 2. *The slits l_1 and l_2 derived from the ray characteristic equation (6) are perpendicular to each other.*

Proof. Assume λ_1 and λ_2 are the two roots to the ray characteristic equation (6). We intersect the three generator rays r, $r+r_x$, and $r+r_y$ with $z=\lambda_1$ plane at P_1, P_2 and P_3 and with $z=\lambda_2$ plane at Q_1, Q_2 and Q_3. The directions of l_1 and l_2 can be computed as:

$$\begin{aligned} \vec{d}_1 &= P_2 - P_1 = [u_x+\lambda_1\sigma_x, v_x+\lambda_1\tau_x, 0] \\ \vec{d}_2 &= Q_2 - Q_1 = [u_x+\lambda_2\sigma_x, v_x+\lambda_2\tau_x, 0] \end{aligned} \quad (20)$$

Next, we show $\vec{d}_1 \cdot \vec{d}_2 = 0$.

By modeling the local surface around S as a Monge patch $z(x,y)$, we can compute the $[\sigma,\tau,u,v]$ coordinate for each normal ray using equation (3) as

$$\begin{aligned} [\sigma,\tau,1] &= [-z_x, -z_y, 1] \\ [u,v,0] &= [x+zz_x, y+zz_y, 0] \end{aligned} \quad (21)$$

Since under our parametrization, S is the origin of the frame and the normal at S is the z axis, we must have

$$\begin{aligned} &x = y = z = z_x = z_y = 0 \\ &\sigma_x = -z_{xx}, \tau_x = -z_{xy}, u_x = 1, v_x = 0 \end{aligned} \quad (22)$$

Substituting equation (22) into $\vec{d}_1 \cdot \vec{d}_2$, we have

$$\vec{d}_1 \cdot \vec{d}_2 = (1-\lambda_1 z_{xx})(1-\lambda_2 z_{xx}) + \lambda_1\lambda_2 z_{xy}^2 \quad (23)$$

Furthermore, we can compute the Gaussian and the Mean curvature of the Monge patch as:

$$\begin{aligned} K &= \frac{z_{xx}z_{yy} - z_{xy}^2}{(1+z_x^2+z_y^2)^2} = z_{xx}z_{yy} - z_{xy}^2 \\ 2H &= \frac{(1+z_x^2)z_{yy} + (1+z_y^2)z_{xx} - 2z_xz_yz_{xy}}{(1+z_x^2+z_y^2)^{\frac{3}{2}}} = z_{xx}+z_{yy} \end{aligned} \quad (24)$$

Since $\lambda_1 = -\frac{1}{\kappa_1}$ and $\lambda_2 = -\frac{1}{\kappa_2}$, we have

$$\begin{aligned} \lambda_1+\lambda_2 &= \frac{1}{\kappa_1}+\frac{1}{\kappa_2} = \frac{2H}{K} = \frac{z_{xx}+z_{yy}}{z_{xx}z_{yy}-z_{xy}^2} \\ \lambda_1\lambda_2 &= \frac{1}{\kappa_1\kappa_2} = \frac{1}{K} = \frac{1}{z_{xx}z_{yy}-z_{xy}^2} \end{aligned} \quad (25)$$

Finally, substituting equation (25) into equation (23) gives $\vec{d}_1 \cdot \vec{d}_2 = 0$. Thus, the two slits must be perpendicular to each other. □

Lemma 2 reveals that the two slits and the normal are perpendicular to each other. Furthermore, by Lemma 1 and the Normal Tangency Property, l_1 and N are both tangent to Γ_1, therefore, l_2 must be the normal direction of Γ_1. Similarly, l_1 must be the normal direction of Γ_2. Thus, by the Principal Direction Property, l_1 and l_2 must correspond to the principal directions.