Decomposing The Tangent of Occluding Boundaries According to Curvatures and Torsions

Huizong Yang and Anthony Yezzi

Georgia Institute of Technology
huizong.yang@gatech.edu anthony.yezzi@ece.gatech.edu

Abstract. This paper develops new insight into the local structure of occluding boundaries on 3D surfaces. Prior literature has addressed the relationship between 3D occluding boundaries and their 2D image projections by radial curvature, planar curvature, and Gaussian curvature. Occluding boundaries have also been studied implicitly as intersections of level surfaces, avoiding their explicit description in terms of local surface geometry. In contrast, this work studies and characterizes the local structure of occluding curves explicitly in terms of the local geometry of the surface. We show how the first order structure of the occluding curve (its tangent) can be extracted from the second order structure of the surface purely along the viewing direction, without the need to consider curvatures or torsions in other directions. We derive a theorem to show that the tangent vector of the occluding boundary exhibits a strikingly elegant decomposition along the viewing direction and its orthogonal tangent, where the decomposition weights precisely match the geodesic torsion and the normal curvature of the surface respectively only along the line-of-sight! Though the focus of this paper is an enhanced theoretical understanding of the occluding curve in the continuum, we nevertheless demonstrate its potential numerical utility in a straight-forward marching method to explicitly trace out the occluding curve. We also present mathematical analysis to show the relevance of this theory to computer vision and how it might be leveraged in more accurate future algorithms for 2D/3D registration and/or multiview stereo reconstruction.

Keywords: Occluding boundaries, differential geometry, radial curves

1 Introduction

Occluding boundaries constitute one of the most important features of 3D surfaces with respect to a given viewpoint. They convey fruitful information such as silhouettes, topology, occlusions, etc., and have been intensively studied in both computer vision and computer graphics. One of the most beautiful mathematical properties, first published by Koenderink [15], is the elegant relationship between the radial 3D curvature of the occluding boundary, the 2D curvature of their perspective projection, and the 3D Gaussian curvature of the underlying

surface. Less attention, however, has been directed toward understanding the local geometry of the 3D occluding boundaries themselves (without reference to the geometry of their 2D projections), which would have special relevance in formulating variational methods involving perspective projections of 3D shapes.

As such, we study the local behavior of the 3D occluding boundary curve itself, motivated by the goal to capture properties that can be conveniently computed on the surface for future use in numerical algorithms. Understanding the geometrical properties of occluding boundaries is not only important for classical 3D computer vision problems such as camera pose estimation, but can also guide the design of data-driven methods [33]. We will develop a theorem which shows that the first order structure of the occluding boundary curve can be determined entirely by second order properties of the underlying surface just along the viewing direction, greatly simplifying its future exploitation in numerical algorithms by obviating the need to estimate principal curvatures and directions.

1.1 Relationship to prior work

Marr [24] first analyzed occluding contours and related them to projections from various types of 3D solid shapes. While numerous useful conclusions were derived, a major point of insight was developed by Koenderink [15], showing that inflection correspondences between occluding contours and boundary curves are related by a specific property that the Gaussian curvature of the occlusion point (3D) is the product of its apparent curvature (2D) and radial curvature (3D). Lazbenik etc. [20], Hebert and Ponce [25] later extended Koenderink's work into pure projective geometry and enriched the understanding of solid shapes. A point to note about Koenderink's results that it did not invoke properties along the occluding tangent direction but rather along the radial (viewing) direction.

Numerous methods utilize occluding boundary cues to recover 3D shape [31, 12, 32, 11]. Specifically, Ikeuchi and Horn [12] attempted to determine the self-shadow boundary of objects but presumed the boundary curve to be laid on a plane, which is generally not the case. In addition to 3D shape inference, occluding boundary curves can also serve to accomplish recognition tasks [14]. Still, computing space curves satisfying certain conditions is not an easy task.

Recently, Fabbri and Kimia [9] developed a comprehensive theory on multi-view differential occluding boundary curves, classified as non stationary curves in their frame work where these curves move as the view point changes. They derive differential equations to describe the evolution of points on the occluding curve with respect to a moving viewpoint, whereas our complementary focus will be to study the structure of a static occluding curve (fixed viewpoint).

More generic, yet closely related work is found in the computer graphics community [18, 17, 3, 1, 34, 7, 23, 8] for calculating the intersection between two surfaces. In [18], Kriegman and Ponce represented space curves as the intersection of two implicit surface functions (alternatively approximated by polynomial functions). Occluding boundary curves could fit into this framework as a special case. Given an initial point [2], Kriegman and Ponce directly solved these equations numerically through looking for the next closest projection on a 2D

subspace spanned by constant basis, then correcting it back to the intersection with Newton's method. Ye and Maekawa [34], furthermore, focused on computing the differential properties of the intersection curves by two implicit surfaces, including the tangent vectors of these curves. They also analyzed the singularities of different types of intersections, and gave the analytical expressions for these tangent vectors with respect to the normals of two surfaces. Düldül and Çalişkan also analyzed the curvatures and torsions of the intersection curves. Abdel-All [1] extended to characterizing the singularities of the intersection curves. They also gave necessary and sufficient conditions for those curves to be straight, planar, helical, and helix circular. While analytical solutions were studied and present in aforementioned works, calculations are restricted to pairs of implicit functions while the theorem we develop gives the solution in terms of the local geometric properties of the observed surface alone, independently of its representation.

1.2 Contributions of this paper

While algorithms already exist to compute occluding boundaries as the intersections between the object surfaces and visibility hulls via surface intersection methods [3, 29], the relationship between the boundaries and the local surface patch has not been unveiled. This is the primary goal of this paper, and specifically to do so invoking only properties along the radial (viewing) direction. The resulting occluding tangent decomposition theorem is therefore the only intended contribution of this work, allowing numerical schemes in future algorithms to directly compute the occluding tangent from the surface and without calculating principal curvatures nor the principal directions along the surface, which become ill-defined at umbilical points and unstable to compute at near-umbilical points.

To validate the theorem numerically and illustrate how it may be exploited practically, we will demonstrate a new explicit tracing algorithm which leverages the results of the theorem in a very direct and obvious manner. More sophisticated and mature strategies can clearly be developed, and although our simple illustrative approach turned out surprisingly well, it is not our intent to claim any premature contribution in actual curve extraction algorithms here. Rather, the goal of our numerical experiments is to reinforce the theory numerically and demonstrate how it might be used for practical benefit.

1.3 Outline of paper

The rest is structured as follows: we cover some preliminaries and derive the main theorem (Theorem 1) and contribution of this paper in Section 2. We follow with computational experiments in Section 3 to validate its numerical applicability by leveraging the theorem in an explicit rim tracing strategy. For both simple and more complicated geometries, we show how the decomposition in this Theorem can be used to directly trace out the occluding boundary (considering only local, not yet global, occlusion conditions thus far) by taking small steps in the resulting tangent direction from an arbitrarily detected starting point.

We follow this numerical validation of the theorem by further mathematical analysis, in section 4, relating the variation of the projected 2D curve to that of the 3D occluding boundary curve, revealing the explicit appearance of the 3D tangent vector and thereby illustrating the relevance of this theory to the future design of improved variational methods in computer vision problems such as silhouette matching, 2D/3D registration, and multiview stereo reconstruction.

2 Occluding tangent decomposition

Occluding tangents are the tangent vectors to a space curve that divides the exterior of orientable surfaces into visible and invisible parts from certain view points (shown in Fig. 1). Formally we define the *local* occluding curve as

$$\mathcal{S}^* : \{X : \mathbf{n} \cdot (X - P) = 0,\ X \in \mathcal{S}\}$$

where $\mathbf{n}$ denotes the surface normal at point X, P the view point. Note that $\mathcal{S}^*$ is a superset of the *global* occluding rim $\mathcal{S}^{**}$ that projects to the silhouette boundary as self occlusions can make portions $\mathcal{S}^*$ project inside the silhouette.

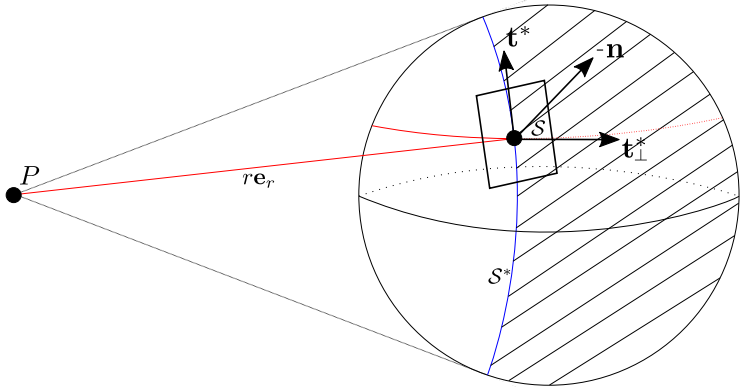

Fig. 1. Illustration of the occluding boundary curve $\mathcal{S}^*$ (blue) of a sphere. The red curve on the sphere is the *radial curve* and the grey straight dotted lines the lines of sight that touch the upper and the lower bound of the occluding boundary. P represents the view point, $\mathbf{e}_r$ the unit view direction and r the distance from view point to the surface point S.

2.1 Occluding boundary curve

If we consider a fixed viewpoint $P \in \mathbb{R}^3$ which is off the surface, we may construct a local parameterization of the surface by a radial distance $r = \|S - P\|$ and a radial unit vector $\mathbf{e}_r = (S - P)/r$ along the viewing direction

$$S(u, v) = P + r(u, v)\ \mathbf{e}_r(u, v)$$

where (u, v) denote the local surface parameters. We define the tangent vectors

$$\mathbf{t}_r = \mathbf{e}_r - (\mathbf{e}_r \cdot \mathbf{n})\mathbf{n}, \qquad \mathbf{t}_{r\perp} = \mathbf{n} \times \mathbf{t}_r \tag{1}$$

where $\mathbf{t}_r$ is the projection of $\mathbf{e}_r$ onto the tangent plane of S and where $\mathbf{t}_{r\perp}$ points in the orthogonal transverse direction. The resulting 3D frame $\mathbf{t}_r$, $\mathbf{t}_{r\perp}$, $\mathbf{n}$ becomes degenerate at any front-to-parallel point where $\mathbf{e}_r = \pm\mathbf{n}$ (such points in an angular sense, however, are maximally distant from occluding boundaries).

Accordingly, the occluding boundary curve $\mathcal{S}^*$ is pre-classified by the condition $\mathbf{t}_r = \mathbf{e}_r$ (recalling that $\mathcal{S}^*$ is a superset of the actual global occluding rim $\mathcal{S}^{**}$). We may equivalently express $\mathcal{S}^*$ by the the zero level set of the following local occlusion function $g : \mathcal{S} \to \mathbb{R}$ along the surface.

$$g(S) \doteq \underbrace{(S - P)}_{r\,\mathbf{e}_r} \cdot \mathbf{n} \tag{2}$$

We will use the * superscript notation to denote the restriction of any entity to/on the zero level set of g. Along the occluding boundary $\mathcal{S}^*$ the radial and transverse tangent vectors become unit vectors $\mathbf{t}_r^* = \mathbf{e}_r^*$ and $\mathbf{t}_{r\perp}^* = \mathbf{n}^* \times \mathbf{e}_r^*$.

Using the convention of an *inward-pointing normal* $\mathbf{n}$ (so that positive curvature indicates convexity and negative curvature concavity), a point on the surface is visible only if $g(S) > 0$ and is occluded if $g(S) < 0$. The occluding boundary curve therefore travels along the sign transition, and based on our sign convention, the intrinsic gradient $\nabla_{\mathcal{S}}\, g(\mathcal{S}^*)$ along the occluding boundary will point *inward toward* the potentially visible subset of S.

2.2 Decomposition

Given the function g defined in (2), the surface tangent vector $\mathbf{t}_\perp^*$ pointing to the visible side of the surface can be found by taking the intrinsic gradient of g at $\mathcal{S}^*$. Rotating this tangent vector by $\pi/2$ around the unit normal yields the tangent $\mathbf{t}^*$ of the occluding curve.

If we seek an occluding tangent vector $\mathbf{t}^*$ which follows the occluding boundary $\mathcal{S}^*$ in the counterclockwise direction when seen from the viewpoint P, then its counterclockwise rotation around the *inward unit normal* (the convention for $\mathbf{n}$ where positive curvature indicates convexity) will point *away from* the visible subset of $\mathcal{S}$ (where $g(S) < 0$) which is in the opposite direction of $\nabla_{\mathcal{S}} g$. By looking at both the covariant and contravariant components of the shape operator in this direction, we may prove the formal theorem below (see complete proof in the appendix).

Theorem 1. *Letting $\mathbf{e}_r$, κ_r, τ_r, denote the radial direction, radial curvature, and radial torsions respectively at a point S^* on the occluding boundary, we may construct the (unnormalized) occluding tangent vector $\mathbf{t}^*$ and its orthogonal complement $\mathbf{t}_\perp^*$ in terms of $\mathbf{e}_r$ and $\mathbf{n} \times \mathbf{e}_r$ as follows.*

$$\mathbf{t}_\perp^* = \kappa_r\, \mathbf{e}_r + \tau_r\, (\mathbf{n} \times \mathbf{e}_r) \tag{3}$$

$$\mathbf{t}^* = \begin{cases} \kappa_r\, (\mathbf{e}_r \times \mathbf{n}) + \tau_r\, \mathbf{e}_r, & \textit{for inward } \mathbf{n} \\ \kappa_r\, (\mathbf{n} \times \mathbf{e}_r) + \tau_r\, \mathbf{e}_r & \textit{for outward } \mathbf{n} \end{cases} \tag{4}$$

What is particularly appealing about this decomposition is that the second order properties of the surface need be computed only along the radial (line-of-sight) direction to determine the direction of the occluding boundary. Its two degrees of freedom within the tangent plane are captured by two second order pieces of information, the normal curvature and geodesic torsion in a single direction. Said in another way, the direction of the occluding boundary can therefore be determined purely from the geodesic along the radial direction (i.e. by examining its curvature and torsion).

3 Numerical demonstration

In this section we demonstrate experiments that contain two synthetic shapes (a sphere and a torus) and a bunny example [22], in order to validate the computational feasibility of our proposed theorem by both the qualitative and quantitative results. Note that the achievement on the best accuracy is not our main purpose for it depends on implementations and data representations. For validation purpose, we utilize the implicit representation defined on voxel grid and take a simple curves extraction strategy shown to be able to achieve decent accuracy later. Implementation details and calculations of the coefficients for this implicit representation are included in the supplementary material.

Now that the occluding tangent direction is known to each point on the occluding boundaries, straightforwardly following the tangent direction produces the curves. As such, an intuitive algorithm (the marching method) is: to move the point along the normalized tangent direction $\mathbf{t}^*$ for a small step followed by recomputing the unit tangent vector at the new point and moving it again to trace out the curve. The error accumulates without any surprises, resulting a drifting problem. However, the drift can be easily addressed by solving a system that makes sure the point is both on the object surface and the visibility hull.

In computer graphics, 3D objects like these are commonly represented as triangle meshes consisting of vertices and edges. This data representation enables one to evaluate the function g mentioned in section 2.1 at each vertex of a mesh, and to extract the zero level [30] set as a comparison. We evaluate the discrepancy in the metric of voxel length by averaging all the closest distance from the estimated occluding points to the references curves. Note that for synthetic examples, we compute the errors among the estimated curves to the analytical occluding boundaries with the same distance measurement.

3.1 Sphere case

A simple case starts with a sphere. If the view point is set in the exterior space of this object, neither self-occlusion nor multiple occluding boundary curves occur due to the convexity of the whole shape. Specifically, exactly one curve appears.

Fig. 2 shows the occluding boundary curve found by our tracing algorithm and the reference curve, displayed as the thin blue line and the red thick band, respectively. The curve generated by the simple marching algorithm is able to

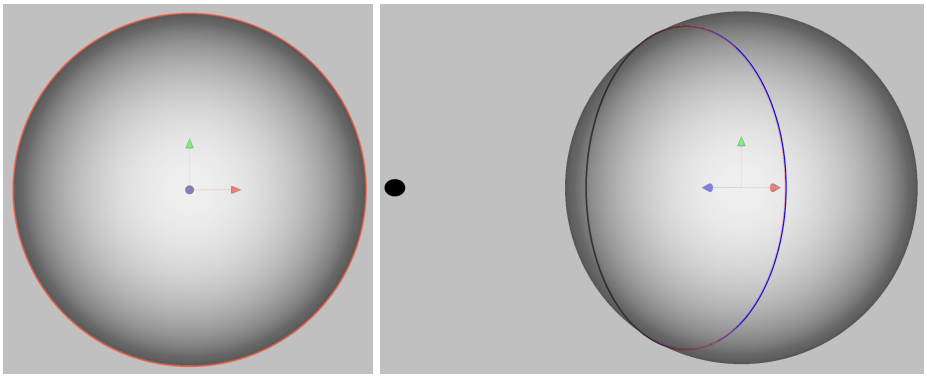

Fig. 2. Occluding boundary extraction of a sphere whose opacity is set such that one can observe invisible areas from other angles. The black dot represents the view point. The right image is viewed from a tilted angle, the left viewed from the black dot.

match the analytical ground truth with an average error 0.03% of the voxel length. Additionally, the estimated curve (blue) differs from the reference curve (red) by 0.6% of the voxel length, a discrepancy level hard to perceive (Fig. 2).

3.2 Torus case

In order to examine whether our proposed algorithm manages to extract the occluding boundaries under varying curvatures, we also demonstrate a more sophisticated 3D object, a torus, on which self-occlusion might occur from certain views and curvatures vary. Different from the sphere example, at least one curve exists and exactly two candidate curves compose the local occluding boundaries.

Three cases are tested at different view points. Among them are three categories of occlusion for the inner part–no occlusion, half occlusion and full occlusion. The easiest situation is illustrated from the top row of Fig. 3. Clearly, homogeneous curvatures along the occluding boundaries and zero torsions prevent the method from numerical issues. The third row of Fig. 3 indicates the results of full occlusion case, wherein varying curvatures and torsions makes the task harder but we can still benefit from the fronto-parallel voxel grid. Moreover, only one occluding curve (i.e. the outer rim) should occur, while two are displayed because invisible points rejection has not been taken. We may nevertheless observe from the zoomed in figures that the predicted occluding boundary curves align with the reference curves pretty well. We do not zoom in heavily as the relevant scale to the voxel would become imperceptible. The average distance difference is less than 0.6% of the voxel length while the error to the analytical ground truth remains around 0.04%.

Being the most complex case, the slanted view point not only brings up different curvatures of the occluding curves as well, but aslo makes the voxel grid non-fronto-parallel with potential numerical issues. In fact, it turns out that

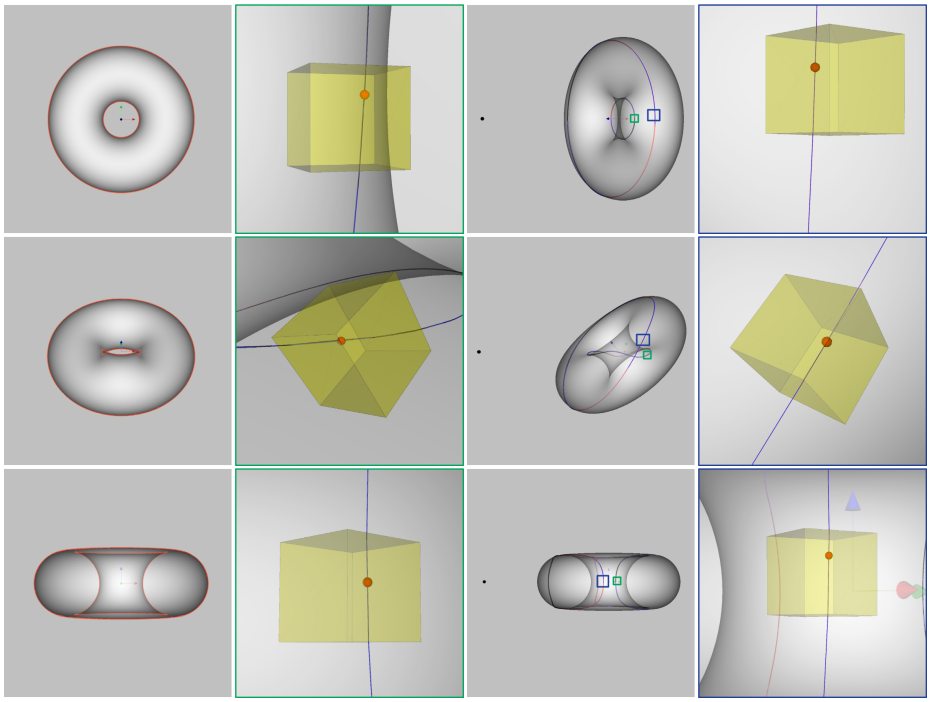

Fig. 3. Torus example. The torus is set to be relatively transparent in order to visualize the invisible areas. The columns from left to right are: view from the view point (black dot), zoomed-in details (marked by a green box) of occluded curve for the inner part, view from a tilted angle, zoomed-in details (marked by a dark blue box) of the visible curve (the outer part). Red curves are the reference superset of occluding boundaries and blue ones the predicted set by our theorem. A voxel is overlaid at the zoomed-in figures to qualitatively demonstrate the error.

our theorem with the simple method yields 0.06% mean error to the analytical ground truth. The estimated curves only differs from the reference curves by an average of 0.7% of the voxel length.

3.3 Bunny case

We also demonstrate the numerical feasibility on real data, Stanford bunny [22], instead of purely on synthetic data. Real data is more challenging as the surface curvatures have a wider range that could cause numerical issues. We show in Fig. 4 that our theorem manages to handle the real data. The average discrepancy to the reference curves for this example is 7% of the voxel length. As shown from the previous synthetic examples, reference curves can deviate from the real occluding boundaries for the normal calculation in the marching cube method has inaccuracy especially around highly bent regions. Hence, the reference curves

cannot guarantee to be the most accurate ones, and we can still conclude the numerical feasibility of our theorem given the acceptable discrepancy.

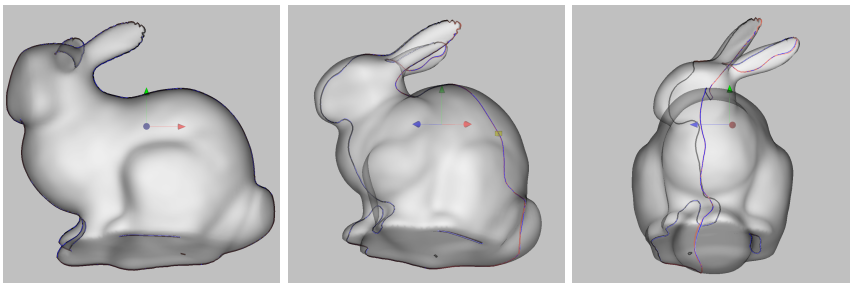

Fig. 4. Real data demonstration. We omitted the view point rendering to make the second image as the same size with the others. The first image is rendered from the viewpoint, the second from a rotated view and the third from the side view. Discrepancies around the tail region are perceptible.

From the experiments we show that a simple and naive strategy can also bring significant accuracy without suffering from numerical problems. We believe more sophisticated methods to perform even better.

4 Discussion for potential applications

4.1 Occluding curve and tangent in 3D pose estimation

To illustrate the relevance of the new theorem 1 in practical computer vision problems, we show how the occluding curve (and its tangent) play an important role in variational approaches to 2D/3D registration. While the contribution of this paper is theoretical, we feel it is nevertheless important to outline how its introduction to the computer vision community might drive upcoming research in this community, particularly mathematical approaches that require grounding in solid geometric theory. A few existing methods [5, 26] within the 2D/3D registration area, which we believe to be the most promising area for future benefits from the knowledge of the occluding curve structure relayed by the theorem.

Silhouette matching (the simplest case) Given a known 3D shape but with unknown pose with respect to a viewing camera, a simple region based method to estimate the camera pose parameters is to minimize the area-mismatch M between the projected silhouette of the shape (which depends upon each pose parameter λ) and a binary mask B representing a segmented/labeled region corresponding to the detected object within the domain Ω of the camera image. We may express this mismatch in integral form

$$M = \int_{\Omega} |\chi_S - B| \, d\mathbf{x}$$

where χ_S denotes the characteristic function of the 2D silhouette within the domain Ω of the camera image. Most minimization strategies require calculation of the derivative of the mismatch with respect to each pose parameter λ, which can be expressed in terms of the normal derivative of the projected silhouette 2D boundary contour $\mathbf{c}$ with respect to λ as follows.

$$\frac{\partial M}{\partial \lambda} = \int_{\mathbf{c}} \underbrace{(1-2B)}_{\substack{\text{signed} \\ \text{mismatch}}} \underbrace{\left\langle \frac{\partial \mathbf{c}}{\partial \lambda}, \hat{\mathbf{n}} \right\rangle}_{\substack{\text{normal} \\ \text{sensitivity}}} \underbrace{\| \frac{\partial \mathbf{c}}{\partial s} \|}_{\text{arclength}} ds \tag{5}$$

where s denotes an arbitrary contour parameter and $\hat{\mathbf{n}}$ the 2D unit normal.

However, direct computation of the derivative $\frac{\partial \mathbf{c}}{\partial \lambda}$ is not possible using only the structure of the silhouette boundary $\mathbf{c}$ itself. Rather, it depends on the differential structure of the back-projected 3D occluding curve $\mathbf{C}$ as follows[1]

$$\left\langle \frac{\partial \mathbf{c}}{\partial \lambda}, \hat{\mathbf{n}} \right\rangle \| \frac{\partial \mathbf{c}}{\partial s} \| = -\frac{L_h L_v}{Z^3} \left\langle \frac{\partial \mathbf{C}}{\partial s}, \mathbf{C} \times \frac{\partial \mathbf{C}}{\partial \lambda} \right\rangle \tag{6}$$

where $\mathbf{C} = (X, Y, Z)$ denotes the corresponding 3D occluding curve in camera frame coordinates (which depend on each camera pose parameter λ) and where $L_h$and L_v denote the horizontal and vertical focal scales of the camera (first two diagonal elements of the intrinsic matrix). Since s is an arbitrary contour parameter which parameterizes both the projected 2D silhouette boundary $\mathbf{c}$ as well as its originating 3D occluding curve $\mathbf{C}$, it may be conveniently chosen in reference to $\mathbf{C}$, by plugging (6) into (5) to obtain the following expression for the mismatch derivative as an integral along the occluding 3D curve

$$\frac{\partial M}{\partial \lambda} = -L_h L_v \int_{\mathbf{C}} \frac{1-2B}{Z^3} \left\langle \underbrace{\frac{\partial \mathbf{C}}{\partial s}}_{\substack{\text{occluding} \\ \text{3D tangent}}}, \mathbf{C} \times \frac{\partial \mathbf{C}}{\partial \lambda} \right\rangle ds \tag{7}$$

where the derivative $\frac{\partial \mathbf{C}}{\partial \lambda}$ of the camera coordinates $\mathbf{C} = (X, Y, Z)$ of each occluding 3D point with respect to the camera pose parameter λ is now straightforward. Detailed derivation can be found in [5].

Note that the integral (7) leverages not only the extracted occluding curve points, illustrating the usefulness for but also makes explicit use of the occluding

[1] This assumes no radial/tangential lens distortion, otherwise we must also multiply the right hand side of (6) by the determinant of the distortion model's Jacobian.

tangent vector $\frac{\partial \mathbf{C}}{\partial s}$ at each point as well. As such, not only does (7) illustrate an important application in modern computer vision for occluding curve extraction, but it also suggests that extraction algorithms based on the explicit representation of the occluding tangent developed in this paper are likely to yield numerically superior estimates pose derivatives compared to algorithms which construct the tangent afterwards more coarsely based on differences between nearby extracted points. Tangent estimates using the decomposition Theorem 1 would provide a more direct numerical link to the second order structure of the underlying 3D surface, thus further compensating the loss of information via discretization by the use of additional "off curve" sample points needed for the numerical second derivative calculations.

More complex 2D/3D registration with appearance models While the importance of the occluding curve, including the explicit use of its tangent estimate in (7), is immediately evident for the simplest case of silhouette matching, it is equally relevant for more complicated silhouette matching strategies which pair the geometry of a known 3D shape together with an appearance model. In such cases the derivative of the more sophisticating matching criterion with respect to any pose parameter λ will consist of a region based integral inside the silhouette, which will evaluate the residual error between the image (or image features) and the appearance model, together with a contour integral along the silhouette boundary which will once again involve the normal sensitivity (6) of the silhouette boundary with respect to the pose parameter λ. Due to the problem of computing this sensitivity, several modern schemes either ignore this integral component [16], or are formulated purely around point features [13], [4], [19], rather than region based properties, in order to avoid this all together [27]. For example, some works [26] embed curves or surfaces into a level set function and bypass computing the normal sensitivity by adding an approximated Heaviside function with accuracy however sacrificed. The theorem 1 presented here now reveals a practically exploitable structure about the occluding curve, around which tractable and accurate numerical schemes can be developed to better estimate this boundary-based contribution to the pose-gradients of blended geometric and appearance based models.

4.2 Variational approaches to 3D reconstruction

Finally, we note that in various generative model based approaches to 3D stereo reconstruction, ranging from shape-from-silhouettes to more sophisticated appearance matching methods [10], [28], [27], the normal sensitivity (6) of the silhouette boundary once again arises, whether with respect to a pose parameter λ, or a more general shape parameter in its place. Once again, new strategies which leverage the decomposition Theorem 1 to numerically compute the occluding tangent can, in turn, exploit expression (6) to more accurately compute such sensitivities. While we assume this gain in accuracy is probably less important for early iterates in these reconstruction techniques, the gain in accuracy

is likely to become more important and noticeable near convergence in the later iterates with these techniques.

5 Conclusion and discussion

We have derived in details the tangent of occluding boundary curves in terms of local geometric properties only along radial directions. In particular, we decompose the tangent into the radial direction and its orthogonal complement weighted by the radial curvatures and geodesic torsions along the line-of-sight. The theorem 1, as shown by itself, does not depend on the representations of the surface. In addition to the pioneer work from Koenderink [15], who formally gave an elegant formula between the surface curvatures, apparent curvature, we provide a new perspective of understanding the occluding tangent along the radial and its orthogonal tangent directions.

We also undertake three experiments to validate the numerical feasibility of this theorem from a simple strategy and demonstrate the trivial error. While the formula does not suffice to extract the global occluding boundary curves, additional visibility test [18], [21], [6] can be applied to reject invisible parts and segments whose projections fall within silhouettes for retrieving global occluding boundaries.

Although we mainly focus on the theoretical contribution in this work, we also provide the explicit connections between the Theorem 1 and three computer vision applications. We hope this new theorem to invoke improvements on more vision tasks.

A Appendices

A.1 Intrinsic gradient

If $f : \mathcal{S} \to \mathbb{R}$ is a differentiable function defined on $\mathcal{S}$, the "intrinsic gradient" would naturally correspond to the projection onto the tangent plane of the standard $\mathbb{R}^3$ gradient of any local differentiable extension $\hat{f} : \mathbb{R}^3 \to \mathbb{R}$, where $\hat{f}(S) = f(S)$.

$$\nabla_S f = \nabla \hat{f} - \left(\nabla \hat{f} \cdot \mathbf{n}\right) \mathbf{n}$$

However, we can intrinsically define the gradient of f, without reference to any extension function $\hat{f}$, as the unique tangent vector $\nabla_S f$ which satisfies the equality $\partial_{\mathbf{t}} f = \nabla_S f \cdot \mathbf{t}$ for any tangent vector $\mathbf{t}$ (where $\partial_{\mathbf{t}} f$ denotes the directional derivative of f along the vector $\mathbf{t}$). We can solve for this vector using the first fundamental form coefficients E, F, G together with the partial derivatives of f with respect to the surface parameters u, v as follows.

$$\nabla_S f = \begin{bmatrix} \frac{\partial S}{\partial u} & \frac{\partial S}{\partial v} \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial f}{\partial u} \\ \\ \frac{\partial f}{\partial v} \end{bmatrix}$$

A.2 Orthogonal decomposition of the shape operator

We first orthogonally decompose the shape operator (which actually generalizes Theorem 1) to see both its covariant action and its contravariant action which is ignored in classical differential geometry but highly relevant to our exploration.

Lemma 1. *The action of the shape operator* $\mathbb{S}$ *on any tangent vector* $\mathbf{t} \in \mathbb{R}^3$ *can be decomposed into a covariant component parallel to its argument* $\mathbf{t}$ *and a contravariant component along the perpendicular tangent direction* $\mathbf{t}_\perp = \mathbf{n} \times \mathbf{t}$

$$\mathbb{S}(\mathbf{t}) = \kappa(\mathbf{t})\ \mathbf{t} + \tau(\mathbf{t})\ \mathbf{t}_\perp$$

where $\kappa(\mathbf{t})$ *is the normal curvature (by definition) in the direction of* $\mathbf{t}$ *and where* $\tau(\mathbf{t})$ *matches the geodesic torsion in the direction of* $\mathbf{t}$*.*

Proof. As an operator from the tangent space back to the tangent space, we may decompose the action of the shape operator into two orthogonal directions within the tangent plane, one parallel to its argument $\mathbf{t}$ and the other along the perpendicular tangent direction $\mathbf{t}_\perp$ as follows,

$$\begin{aligned}
\mathbb{S}(\mathbf{t}) &= \left(\mathbb{S}(\mathbf{t}) \cdot \frac{\mathbf{t}}{\|\mathbf{t}\|}\right) \frac{\mathbf{t}}{\|\mathbf{t}\|} + \left(\mathbb{S}(\mathbf{t}) \cdot \frac{\mathbf{t}_\perp}{\|\mathbf{t}\|}\right) \frac{\mathbf{t}_\perp}{\|\mathbf{t}\|} \\
&= \underbrace{\left(\mathbb{S}\left(\frac{\mathbf{t}}{\|\mathbf{t}\|}\right) \cdot \frac{\mathbf{t}}{\|\mathbf{t}\|}\right)}_{\kappa(\mathbf{t})} \mathbf{t} + \underbrace{\left(\mathbb{S}\left(\frac{\mathbf{t}}{\|\mathbf{t}\|}\right) \cdot \frac{\mathbf{t}_\perp}{\|\mathbf{t}\|}\right)}_{\tau(\mathbf{t})} \mathbf{t}_\perp \\
&= \kappa(\mathbf{t})\mathbf{t} + \tau(\mathbf{t})\mathbf{t}_\perp
\end{aligned} \tag{8}$$

A.3 Proof for Theorem 1

Proof. We first compute the intrinsic gradient of the local occlusion function g.

$$\begin{aligned}
\nabla_s g &= \begin{bmatrix} \frac{\partial S}{\partial u} & \frac{\partial S}{\partial v} \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \nabla_{u,v} \left((S - P) \cdot \mathbf{n}\right) \\
&= \begin{bmatrix} \frac{\partial S}{\partial u} & \frac{\partial S}{\partial v} \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} \underbrace{\frac{\partial S}{\partial u} \cdot \mathbf{n}}_{0} + \underbrace{(S - P)}_{r\mathbf{e}_r} \cdot \frac{\partial N}{\partial u} \\ \underbrace{\frac{\partial S}{\partial v} \cdot \mathbf{n}}_{0} + \underbrace{(S - P)}_{r\mathbf{e}_r} \cdot \frac{\partial N}{\partial v} \end{bmatrix} \\
&= r \begin{bmatrix} \frac{\partial S}{\partial u} & \frac{\partial S}{\partial v} \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial N}{\partial u} \cdot \mathbf{e}_r \\ \frac{\partial N}{\partial v} \cdot \mathbf{e}_r \end{bmatrix}
\end{aligned}$$

Notice that

$$\frac{\partial N}{\partial u}\cdot\mathbf{t}_r = \frac{\partial N}{\partial u}\cdot(\mathbf{e}_r-(\mathbf{e}_r\cdot\mathbf{n})\mathbf{n}) = \frac{\partial N}{\partial u}\cdot\mathbf{e}_r-(\mathbf{e}_r\cdot\mathbf{n})\underbrace{\frac{\partial N}{\partial u}\cdot\mathbf{n}}_{0}$$

$$\frac{\partial N}{\partial v}\cdot\mathbf{t}_r = \frac{\partial N}{\partial v}\cdot(\mathbf{e}_r-(\mathbf{e}_r\cdot\mathbf{n})\mathbf{n}) = \frac{\partial N}{\partial v}\cdot\mathbf{e}_r-(\mathbf{e}_r\cdot\mathbf{n})\underbrace{\frac{\partial N}{\partial u}\cdot\mathbf{n}}_{0}$$

which allows us to substitute $\mathbf{e}_r$ with $\mathbf{t}_r = \left[\frac{\partial S}{\partial u}\ \frac{\partial S}{\partial v}\right] t_r$, where t_r denotes the 2D intrinsic representation of $\mathbf{t}_r$ in the basis $\frac{\partial S}{\partial u}$ and $\frac{\partial S}{\partial v}$

$$\begin{aligned}\nabla_{\!S} g &= r\ \left[\tfrac{\partial S}{\partial u}\ \tfrac{\partial S}{\partial v}\right]\begin{bmatrix}E & F\\ F & G\end{bmatrix}^{-1}\left[\tfrac{\partial N}{\partial u}\ \tfrac{\partial N}{\partial v}\right]^T\left[\tfrac{\partial S}{\partial u}\ \tfrac{\partial S}{\partial v}\right] t_r\\ &= -r\ \underbrace{\left[\tfrac{\partial S}{\partial u}\ \tfrac{\partial S}{\partial v}\right]\begin{bmatrix}E & F\\ F & G\end{bmatrix}^{-1}\begin{bmatrix}e & f\\ f & g\end{bmatrix}}_{\text{Weingarten}} t_r\end{aligned}$$

where e, f, g are the coefficients of second fundamental form $\mathbf{II}$. The appearance of the Weingarten formulas allows us to recognize this part of the expression as the shape operator $\mathbb{S}$ applied to the radial tangent vector $\mathbf{t}_r$ and therefore write

$$\frac{\nabla_{\!S} g}{r} = -\mathbb{S}(\mathbf{t}_r) \tag{9}$$

Since we define in Lemma 1 the direction of $\mathbf{t}^*_\perp$ opposite to direction of $\nabla_{\!S} g$, we may then write

$$\mathbf{t}^*_\perp = -\frac{\nabla_{\!S} g}{r} = \mathbb{S}(\mathbf{t}_r) = \kappa_r\,\mathbf{t}_r + \tau_r\,\mathbf{t}_{r\perp}$$

Subsequently, the occluding tangent is computed as follows,

$$\begin{aligned}\mathbf{t}^* &= -\mathbf{n}\times\mathbf{t}^*_\perp = -\kappa_r\,\mathbf{n}\times\mathbf{e}_r - \tau_r\,\mathbf{n}\times(\mathbf{n}\times\mathbf{e}_r)\\ &= -\kappa_r\,\mathbf{n}\times\mathbf{e}_r - \tau_r\left(\mathbf{n}\underbrace{(\mathbf{n}\cdot\mathbf{e}_r)}_{\frac{g(S)}{r}=0} - \mathbf{e}_r\underbrace{(\mathbf{n}\cdot\mathbf{n})}_{1}\right) = -\kappa_r\,\mathbf{n}\times\mathbf{e}_r + \tau_r\,\mathbf{e}_r\end{aligned}$$

For the *outward unit normal* $\mathbf{n}$, we negate $\mathbf{n}$ in the equations above. Therefore, we have

$$\mathbf{t}^* = \begin{cases}\kappa_r\,(\mathbf{e}_r\times\mathbf{n}) + \tau_r\,\mathbf{e}_r, & \text{for inward } \mathbf{n}\\ \kappa_r\,(\mathbf{n}\times\mathbf{e}_r) + \tau_r\,\mathbf{e}_r & \text{for outward } \mathbf{n}\end{cases}$$

References

1. Abdel-All, N.H., Badr, S.A.N., Soliman, M., Hassan, S.A.: Intersection curves of two implicit surfaces in r3. J. Math. Comput. Sci. **2**(2), 152–171 (2012)

2. Abdel-Malek, K., Yeh, H.J.: On the determination of starting points for parametric surface intersections. Computer-Aided Design **29**(1), 21–35 (1997)
3. Bajaj, C.L., Hoffmann, C.M., Lynch, R.E., Hopcroft, J.: Tracing surface intersections. Computer aided geometric design **5**(4), 285–307 (1988)
4. Chen, H.C., Jia, W., Yue, Y., Li, Z., Sun, Y.N., Fernstrom, J.D., Sun, M.: Model-based measurement of food portion size for image-based dietary assessment using 3d/2d registration. Measurement Science and Technology **24**(10), 105701 (2013)
5. Dambreville, S., Sandhu, R., Yezzi, A., Tannenbaum, A.: Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior. In: ECCV. pp. 169–182. Springer (2008)
6. De Vivo, F., Battipede, M., Gili, P.: Occlusion points identification algorithm. Computer-Aided Design **91**, 75–83 (2017)
7. DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A.: Suggestive contours for conveying shape. In: ACM SIGGRAPH 2003 Papers, pp. 848–855 (2003)
8. Düldül, B.U., Çalişkan, M.: The geodesic curvature and geodesic torsion of the intersection curve of two surfaces. Acta Universitatis Apulensis. Mathematics-Informatics **24**, 161–172 (2010)
9. Fabbri, R., Kimia, B.B.: Multiview differential geometry of curves. International Journal of Computer Vision **120**(3), 324–346 (2016)
10. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods and the stereo problem. IEEE (2002)
11. Giblin, P.: Reconstruction of surfaces from profiles. In: Proc. 1st International Conference on Computer Vision, London, 1987 (1987)
12. Ikeuchi, K., Horn, B.K.: Numerical shape from shading and occluding boundaries. Artificial intelligence **17**(1-3), 141–184 (1981)
13. Kaptein, B., Valstar, E., Stoel, B., Rozing, P., Reiber, J.: A new model-based rsa method validated using cad models and models from reversed engineering. Journal of biomechanics **36**(6), 873–882 (2003)
14. Kehtarnavaz, N., Defigueiredo, R.: Recognition of 3d curves based on curvature and torsion. In: Digital and Optical Shape Representation and Pattern Recognition. vol. 938, pp. 357–364. International Society for Optics and Photonics (1988)
15. Koenderink, J.J.: What does the occluding contour tell us about solid shape? Perception **13**(3), 321–330 (1984)
16. Kolev, K., Klodt, M., Brox, T., Cremers, D.: Continuous global optimization in multiview 3d reconstruction. International Journal of Computer Vision **84**(1), 80–96 (2009)
17. Kriegman, D.J., Ponce, J.: A new curve tracing algorithm and some applications. In: Curves and Surfaces, pp. 267–270. Elsevier (1991)
18. Kriegman, D.J., Ponce, J.: Geometric modeling for computer vision. In: Curves and Surfaces in Computer Vision and Graphics II. vol. 1610, pp. 250–260. International Society for Optics and Photonics (1992)
19. Lamecker, H., Wenckebach, T.H., Hege, H.C.: Atlas-based 3d-shape reconstruction from x-ray images. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 1, pp. 371–374. IEEE (2006)
20. Lazebnik, S., Ponce, J.: The local projective shape of smooth surfaces and their outlines. International Journal of Computer Vision **63**(1), 65–83 (2005)
21. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM Transactions on Graphics (TOG) **37**(6), 1–11 (2018)
22. Lindstrom, P., Turk, G.: Fast and memory efficient polygonal simplification. In: Proceedings Visualization'98 (Cat. No. 98CB36276). pp. 279–286. IEEE (1998)

23. Lone, M.S., Shahid, M.H., Sharma, S.: A new approach towards transversal intersection curves of two surfaces in $\mathbb{R}^3$. Geometry, Imaging and Computing **3**(3), 81–99 (2016)
24. Marr, D.: Analysis of occluding contour. Proceedings of the Royal Society of London. Series B. Biological Sciences **197**(1129), 441–475 (1977)
25. Ponce, J., Hebert, M.: On image contours of projective shapes. In: European Conference on Computer Vision. pp. 736–749. Springer (2014)
26. Prisacariu, V.A., Reid, I.D.: Pwp3d: Real-time segmentation and tracking of 3d objects. IJCV **98**(3), 335–354 (2012)
27. Runz, M., Li, K., Tang, M., Ma, L., Kong, C., Schmidt, T., Reid, I., Agapito, L., Straub, J., Lovegrove, S., et al.: Frodo: From detections to 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14720–14729 (2020)
28. Sandhu, R., Dambreville, S., Yezzi, A., Tannenbaum, A.: A nonrigid kernel-based framework for 2d-3d pose estimation and 2d image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(6), 1098–1115 (2010)
29. Soliman, M.A.L., Abdel-All, N.H., Hassan, S.A., Badr, S.A.N., et al.: Intersection curves of implicit and parametric surfaces in r 3. Applied Mathematics **2**(08), 1019 (2011)
30. Thirion, J.P., Gourdon, A.: The 3d marching lines algorithm. Graphical Models and Image Processing **58**(6), 503–509 (1996)
31. Vaillant, R.: Using occluding contours for 3d object modeling. In: European Conference on Computer Vision. pp. 454–464. Springer (1990)
32. Vaillant, R., Faugeras, O.D.: Using extremal boundaries for 3-d object modeling. IEEE Transactions on Pattern Analysis & Machine Intelligence **14**(02), 157–173 (1992)
33. Wang, C., Fu, H., Tao, D., Black, M.: Occlusion boundary: A formal definition & its detection via deep exploration of context. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
34. Ye, X., Maekawa, T.: Differential geometry of intersection curves of two surfaces. Computer Aided Geometric Design **16**(8), 767–788 (1999)