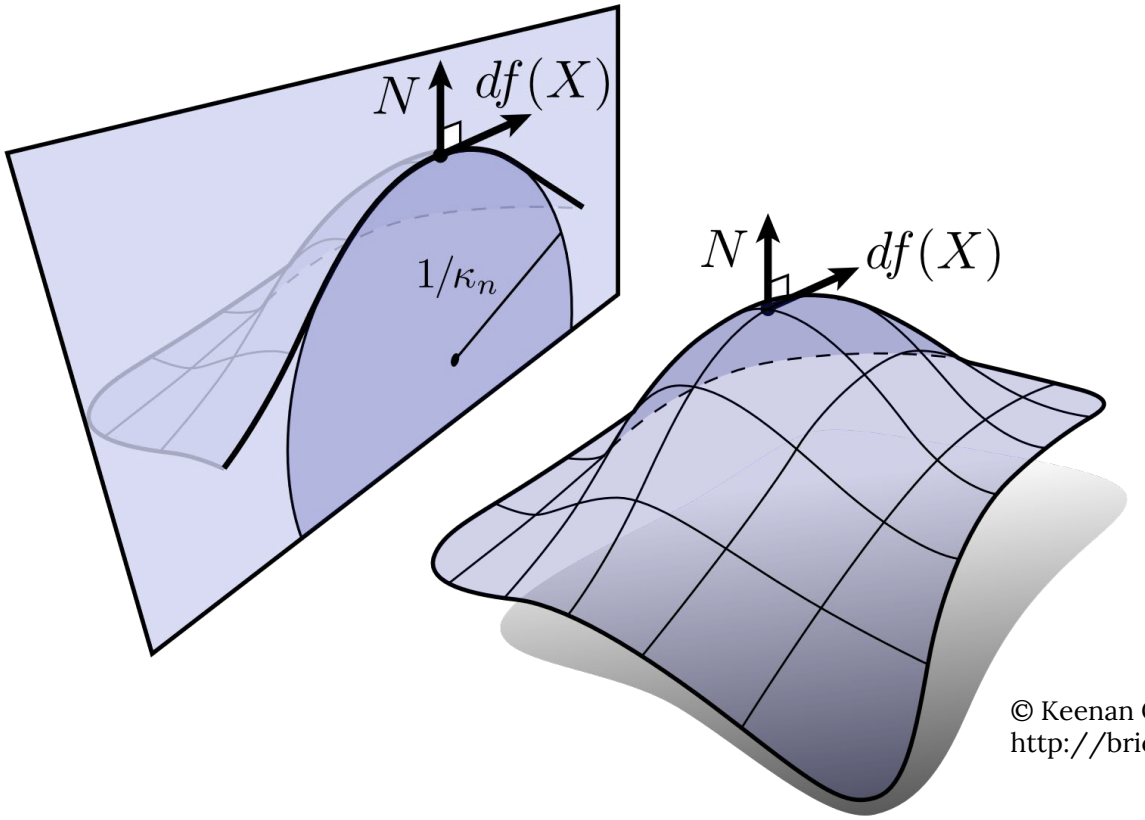
IFT 6113
SURFACES



Mikhail Bessmeltsev

http://tiny.cc/ift6113

SURFACE PARAMETERIZATION

$$f : U \to \mathbb{R}^n$$

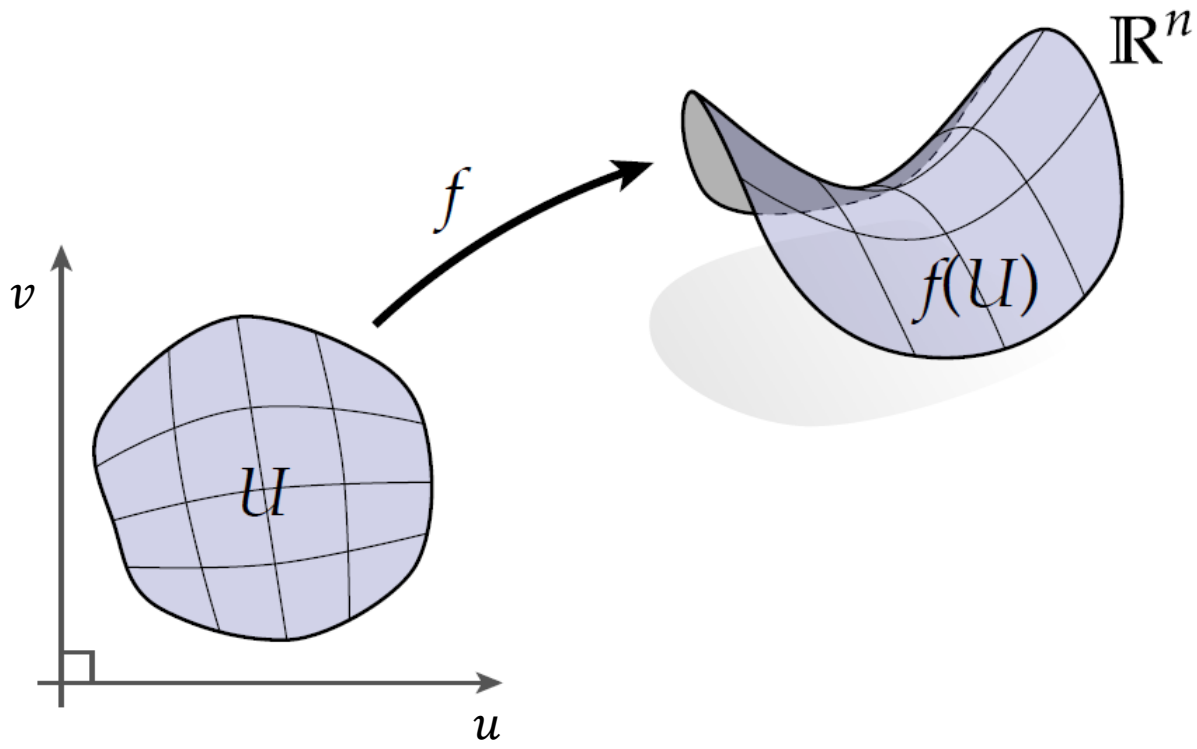
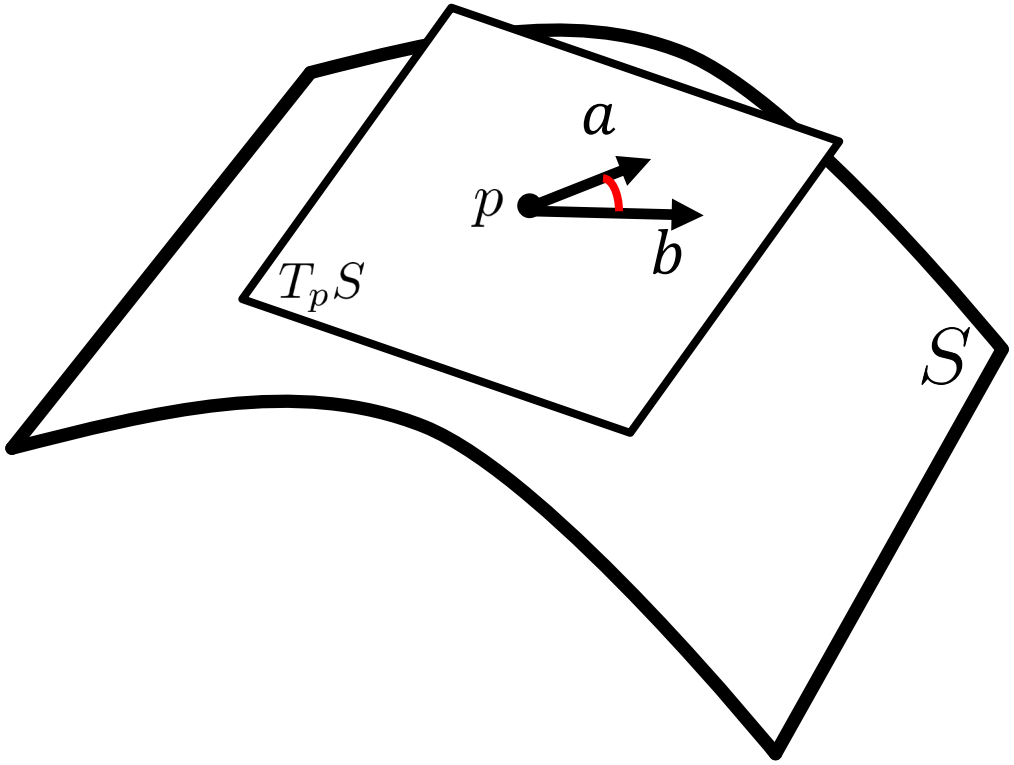


Image © Keenan Crane

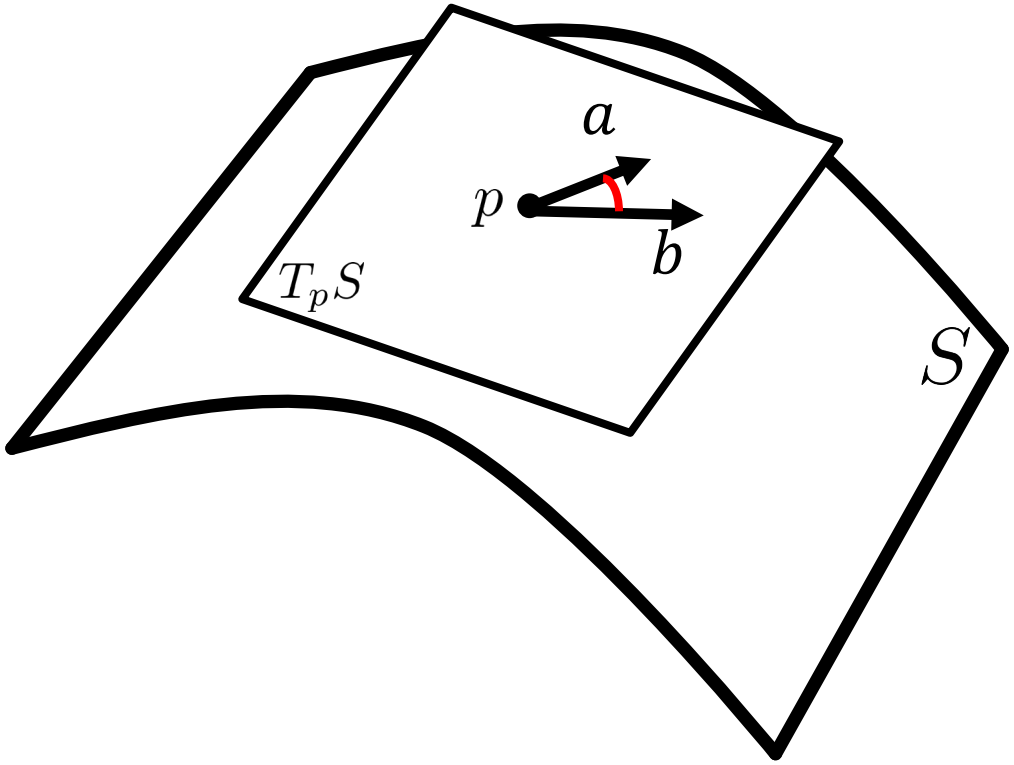
ANGLES AND LENGTHS

How to measure angle between two vectors?



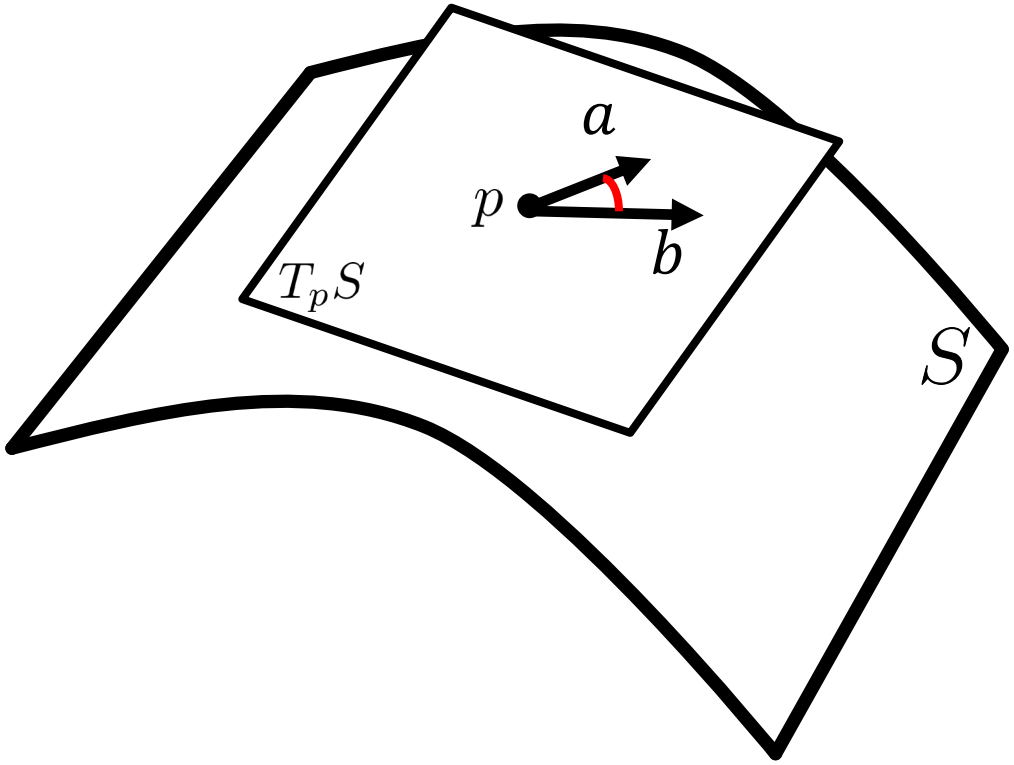
ANGLES AND LENGTHS

$$\cos\alpha = \frac{\langle a, b\rangle}{\|a\|\|b\|}$$



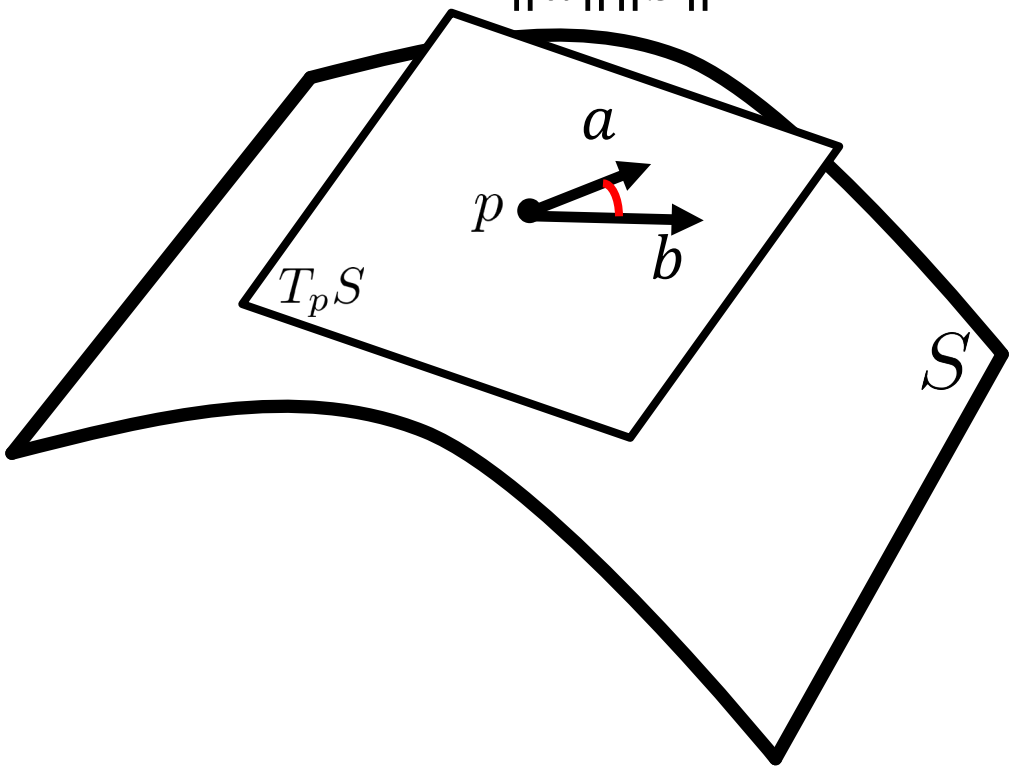
ANGLES AND LENGTHS

$$\cos\alpha = \frac{\langle a, b\rangle}{\sqrt{\langle a, a\rangle}\sqrt{\langle b, b\rangle}}$$



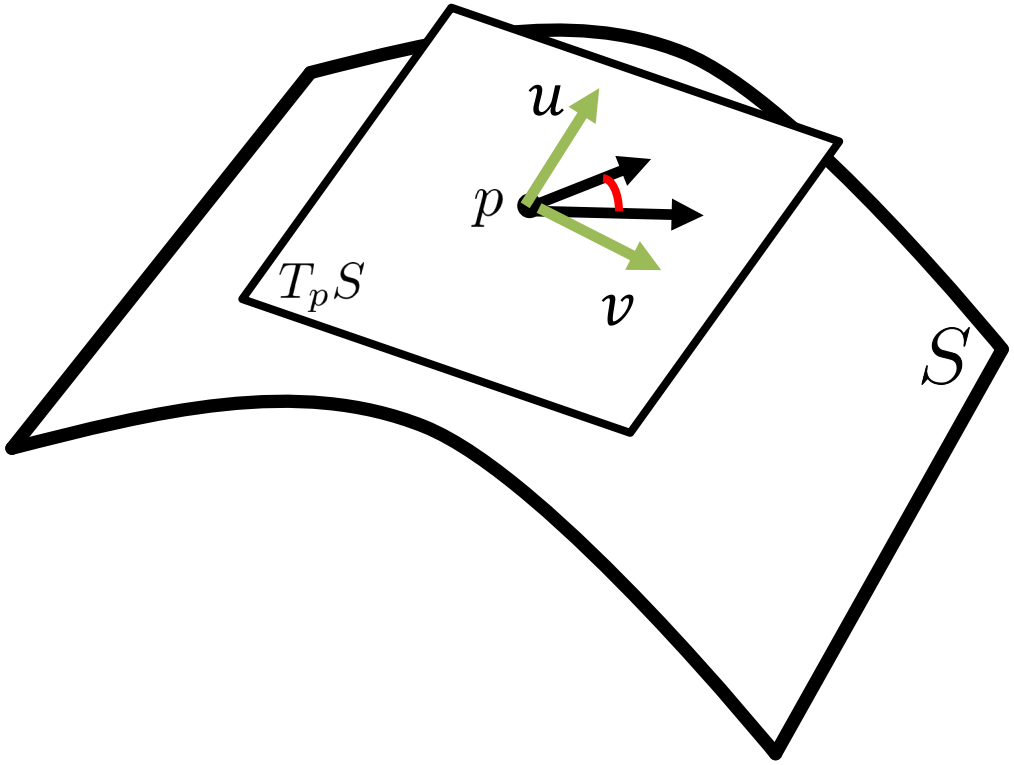
ANGLES AND LENGTHS

$$a = (x_a, y_a, z_a)^T$$
$$b = (x_b, y_b, z_b)^T$$
$$\cos\alpha = \frac{a^T b}{\|a\|\|b\|}$$

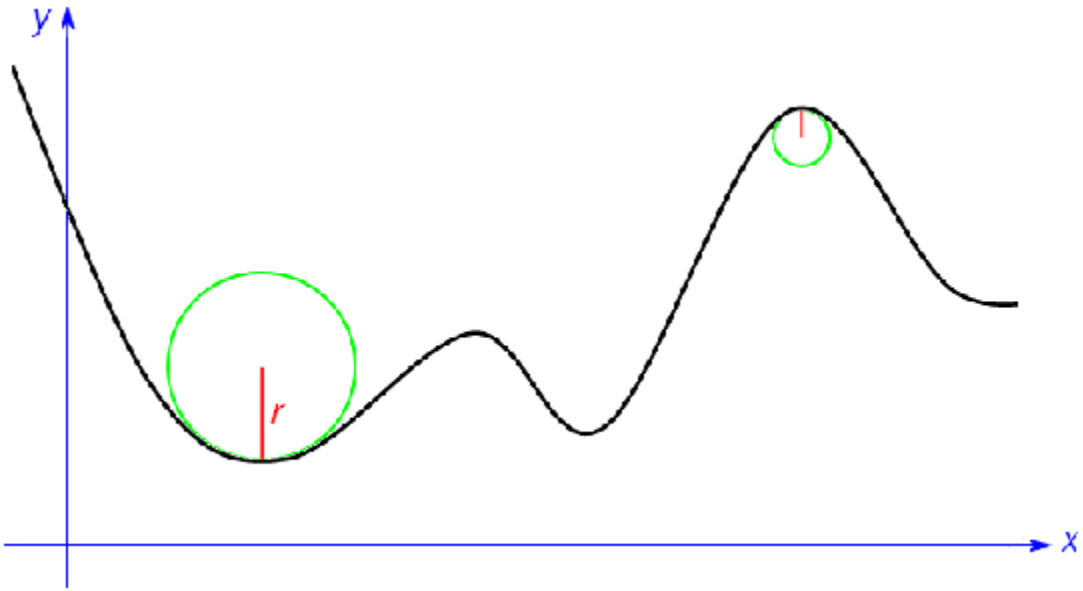


ANGLES AND LENGTHS

$$a = (u_a, v_a)$$
$$b = (u_b, v_b)$$
$$\alpha = ???$$



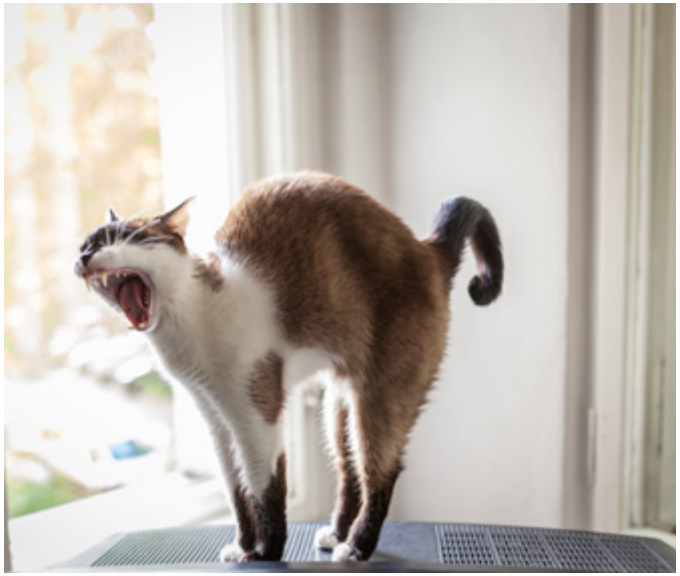
PREVIOUSLY



$$r(s) := \frac{1}{k(s)} \qquad \gamma''(s) = k(s) \cdot \vec{n}\,(s)$$

TODAY

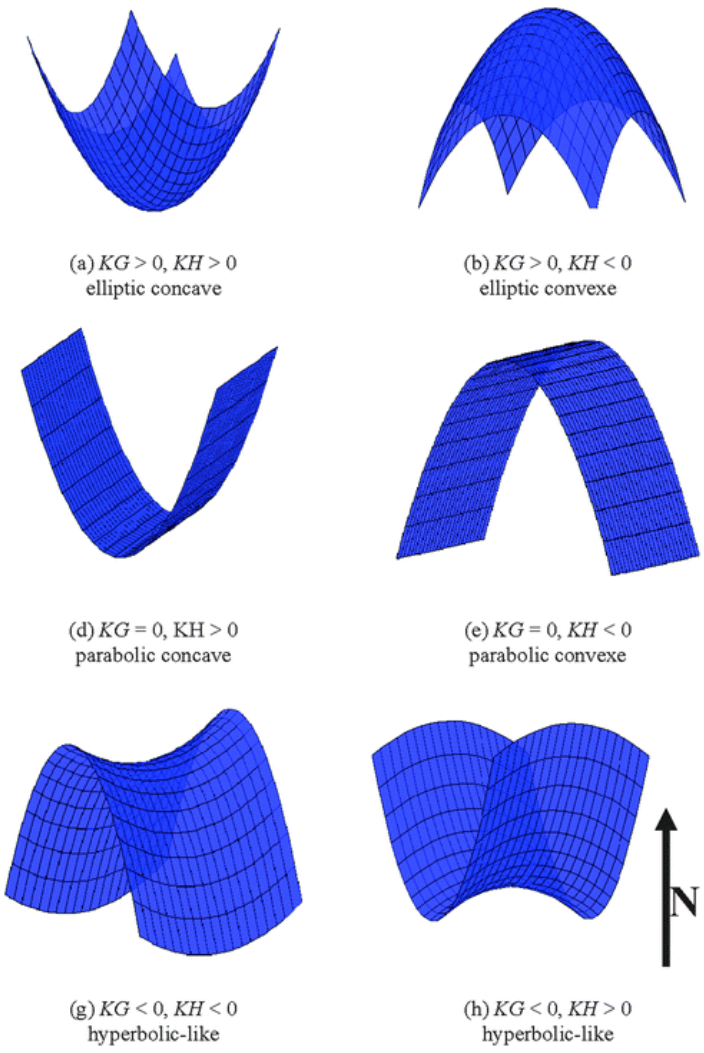
Quantify how a surface **bends.**



http://www.vetstreet.com/our-pet-experts/why-does-my-cat-arch-her-back

Curvature.

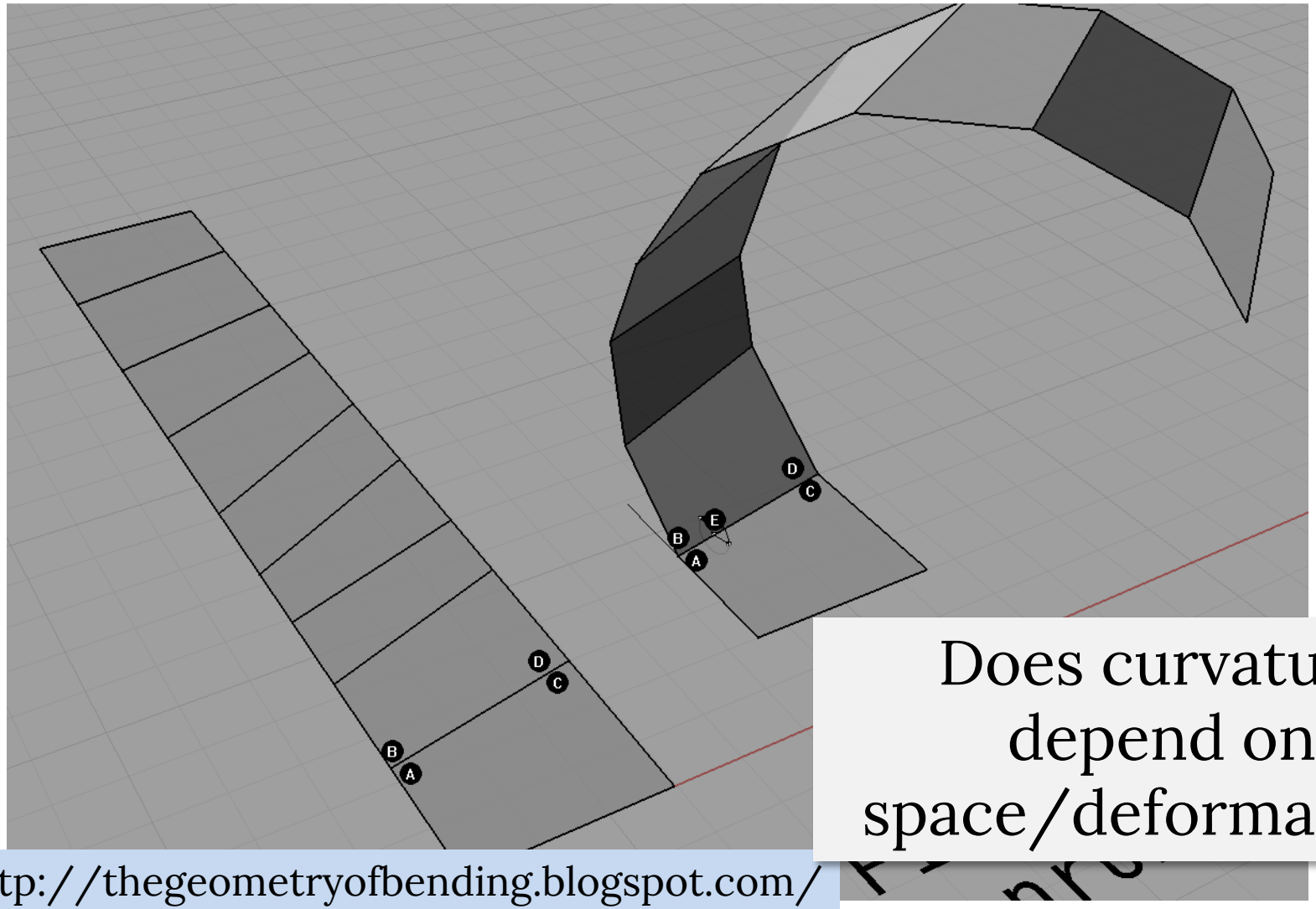
HIGH-LEVEL QUESTIONS



How to distinguish?

http://pubs.rsc.org/is/content/articlelanding/2013/cp/c3cp44375b

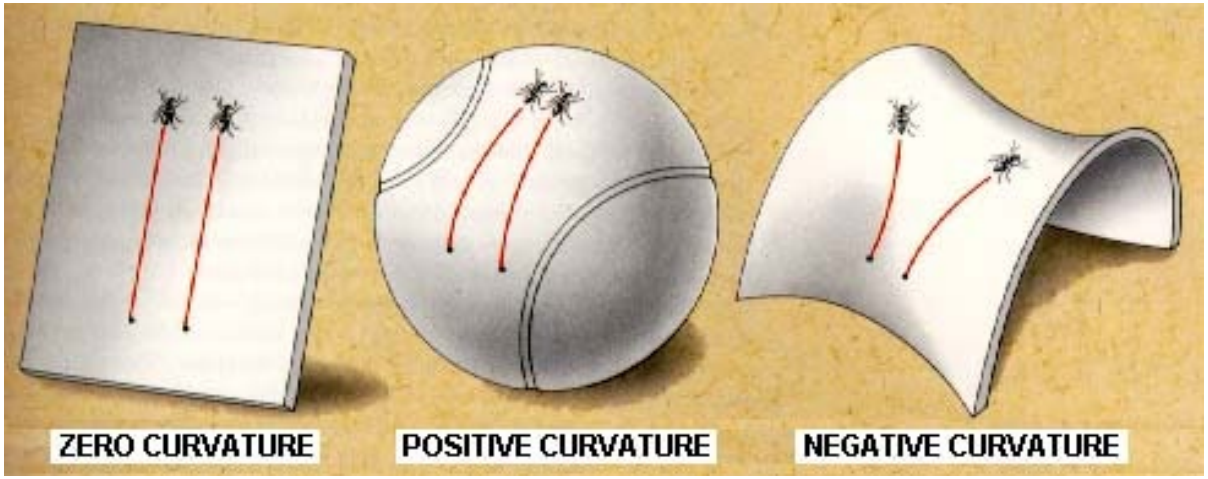
HIGH-LEVEL QUESTIONS



Does curvature depend on space/deformation?

http://thegeometryofbending.blogspot.com/

HIGH-LEVEL QUESTIONS



Does surrounding space matter?

http://starchild.gsfc.nasa.gov/docs/StarChild/questions/question35.html

PRACTICAL APPLICATION

The Best Way to Eat Pizza, According to Science, Means You Probably Have Been Doing It Wrong

Share this

By LUCIA PETERS Oct 10 2014

Congratulations, New Yorkers: Here's proof that you are apparently the only people in the United States who have mastered the best way to eat pizza. As part of their "It's Okay to Be Smart" Web series, PBS' latest video delves into all the scientific trivia you never thought you'd need to know about everyone's favorite late-night drunk food — including the most scientifically sound way to gobble down your slice.

Bend It Like Gauss:

https://www.bustle.com/articles/43697-the-best-way-to-eat-pizza-according-to-science-means-you-probably-have-been-doing-it

Recall:

FRENET FRAME: CURVES IN $\mathbb{R}^3$

$$\frac{d}{ds}\begin{pmatrix} T \\ N \\ B \end{pmatrix} = \begin{pmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{pmatrix}\begin{pmatrix} T \\ N \\ B \end{pmatrix}$$

- **Binormal:** $\boldsymbol{T \times N}$
- **Curvature:** In-plane motion
- **Torsion:** Out-of-plane motion

Theorem:

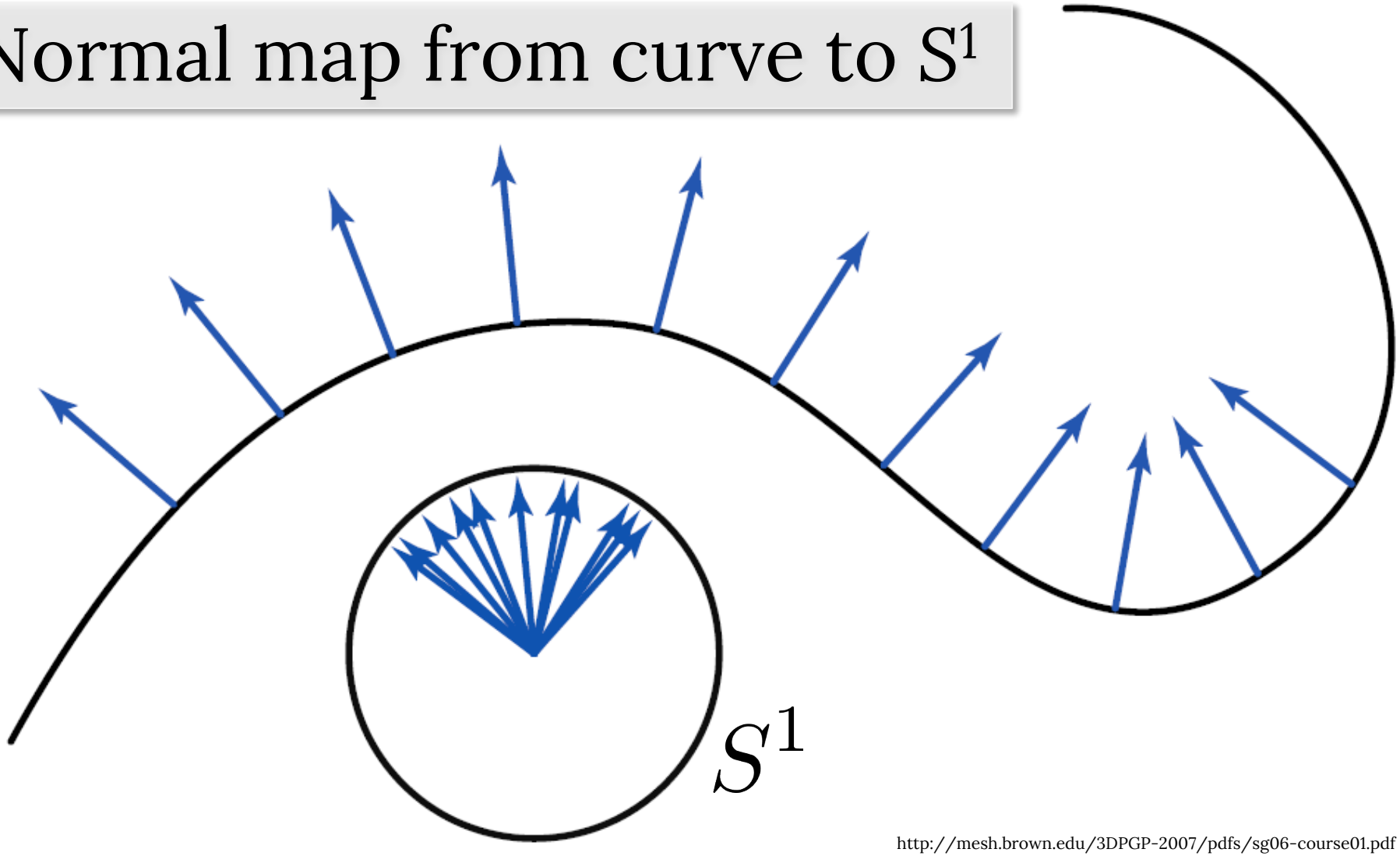
Curvature and torsion determine geometry of a curve up to rigid motion.

Can we say something about **surface curvature** using *curve* curvature/torsion?

Recall:

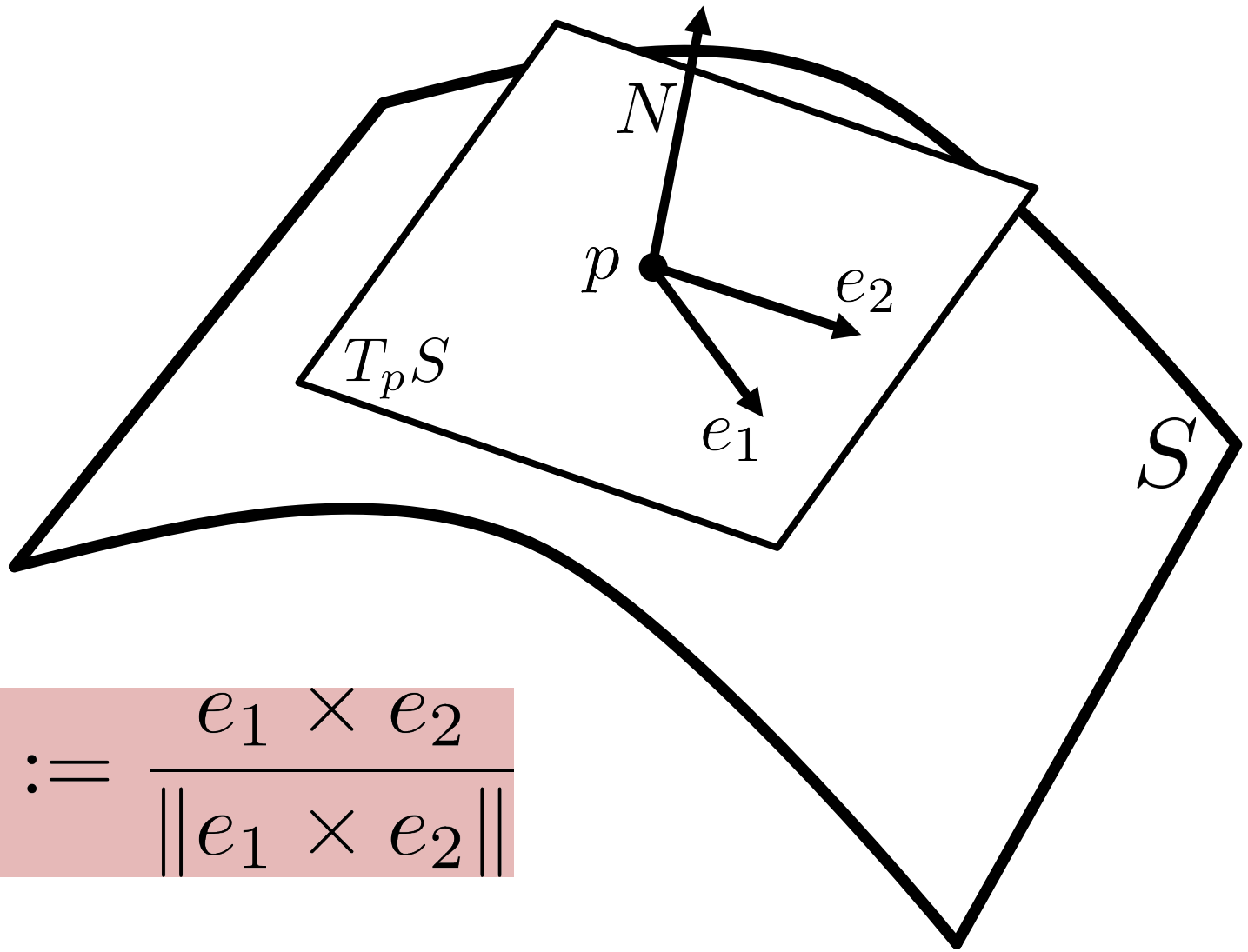
GAUSS MAP

Normal map from curve to S^1



http://mesh.brown.edu/3DPGP-2007/pdfs/sg06-course01.pdf

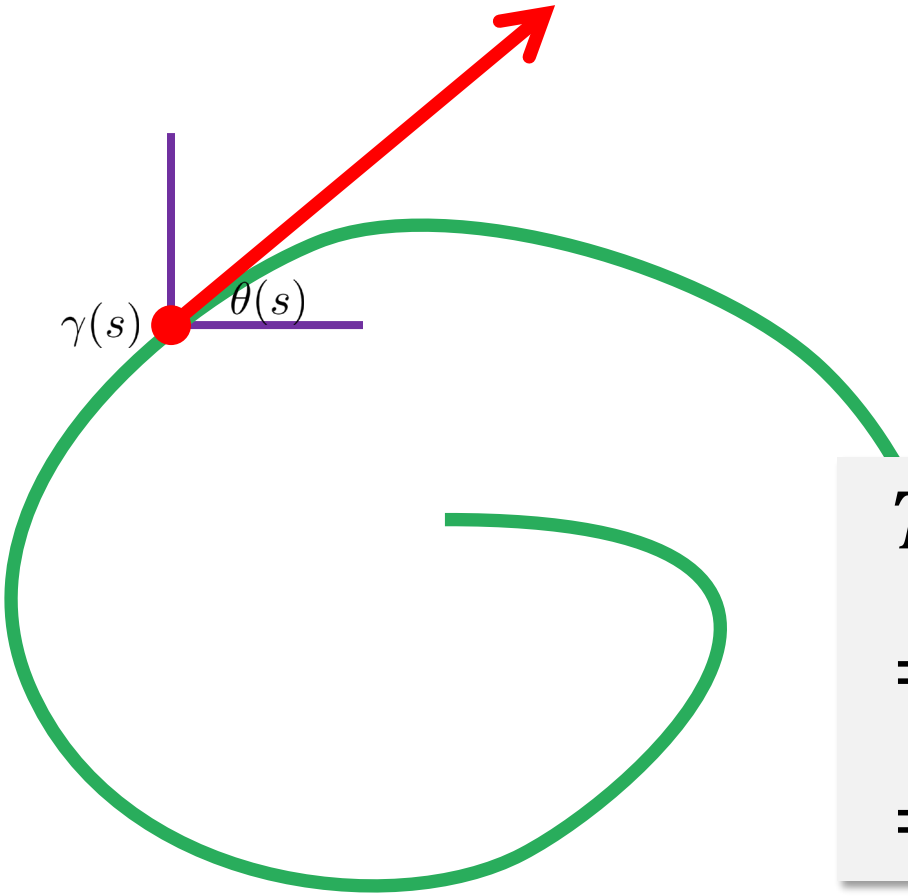
UNIT NORMAL



$$N := \frac{e_1 \times e_2}{\|e_1 \times e_2\|}$$

Recall: SIGNED CURVATURE ON PLANE CURVES

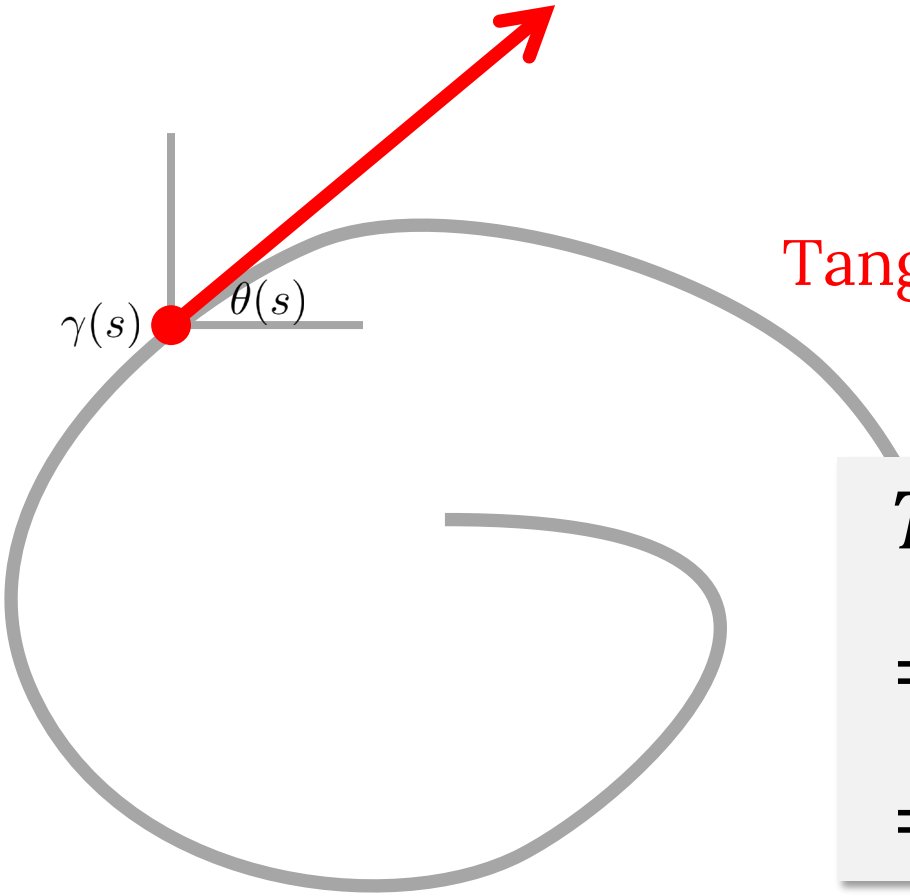
$$T(s) = (\cos\theta(s), \sin\theta(s))$$



$$\begin{aligned} & T'(s) \\ &= \theta'(s) \cdot \begin{pmatrix} -\sin\theta(s) \\ \cos\theta(s) \end{pmatrix} \\ &= k(s) \cdot \vec{n}(s) \end{aligned}$$

Recall: SIGNED CURVATURE ON PLANE CURVES

$$T(s) = (\cos\theta(s), \sin\theta(s))$$



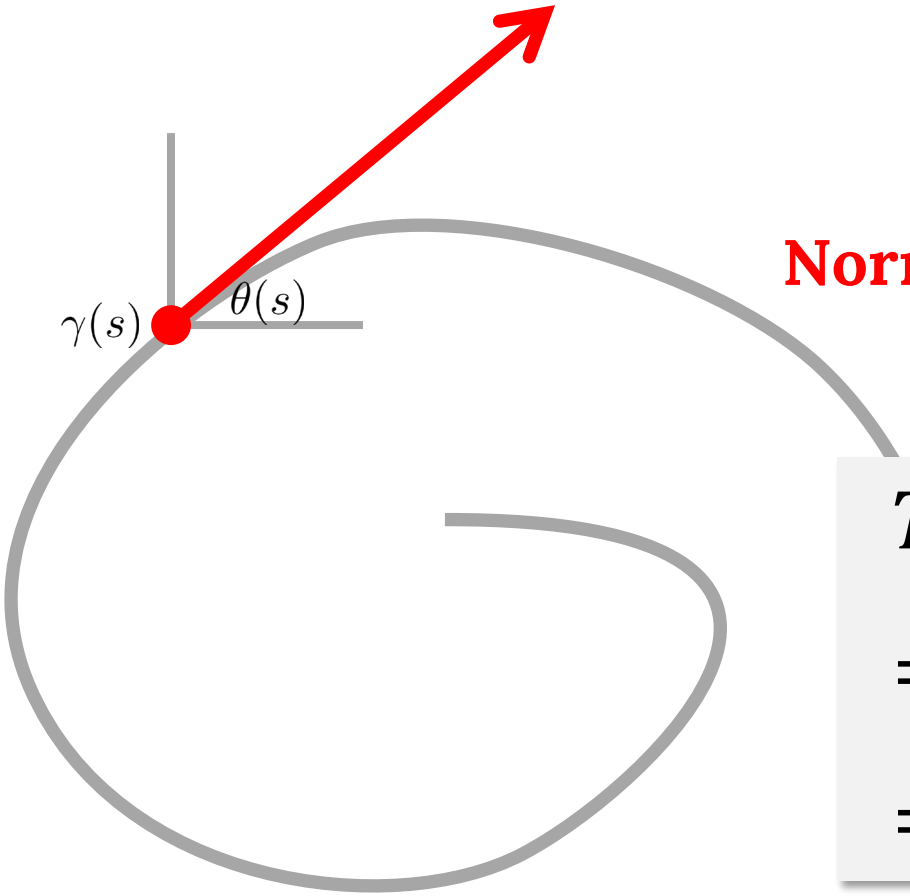
$$\theta'(s) = k(s)$$

Tangent rotates due to curvature

$$\begin{aligned} T'(s) &= \theta'(s) \cdot \begin{pmatrix} -\sin\theta(s) \\ \cos\theta(s) \end{pmatrix} \\ &= k(s) \cdot \vec{n}(s) \end{aligned}$$

Recall: SIGNED CURVATURE ON PLANE CURVES

$$T(s) = (\cos\theta(s), \sin\theta(s))$$

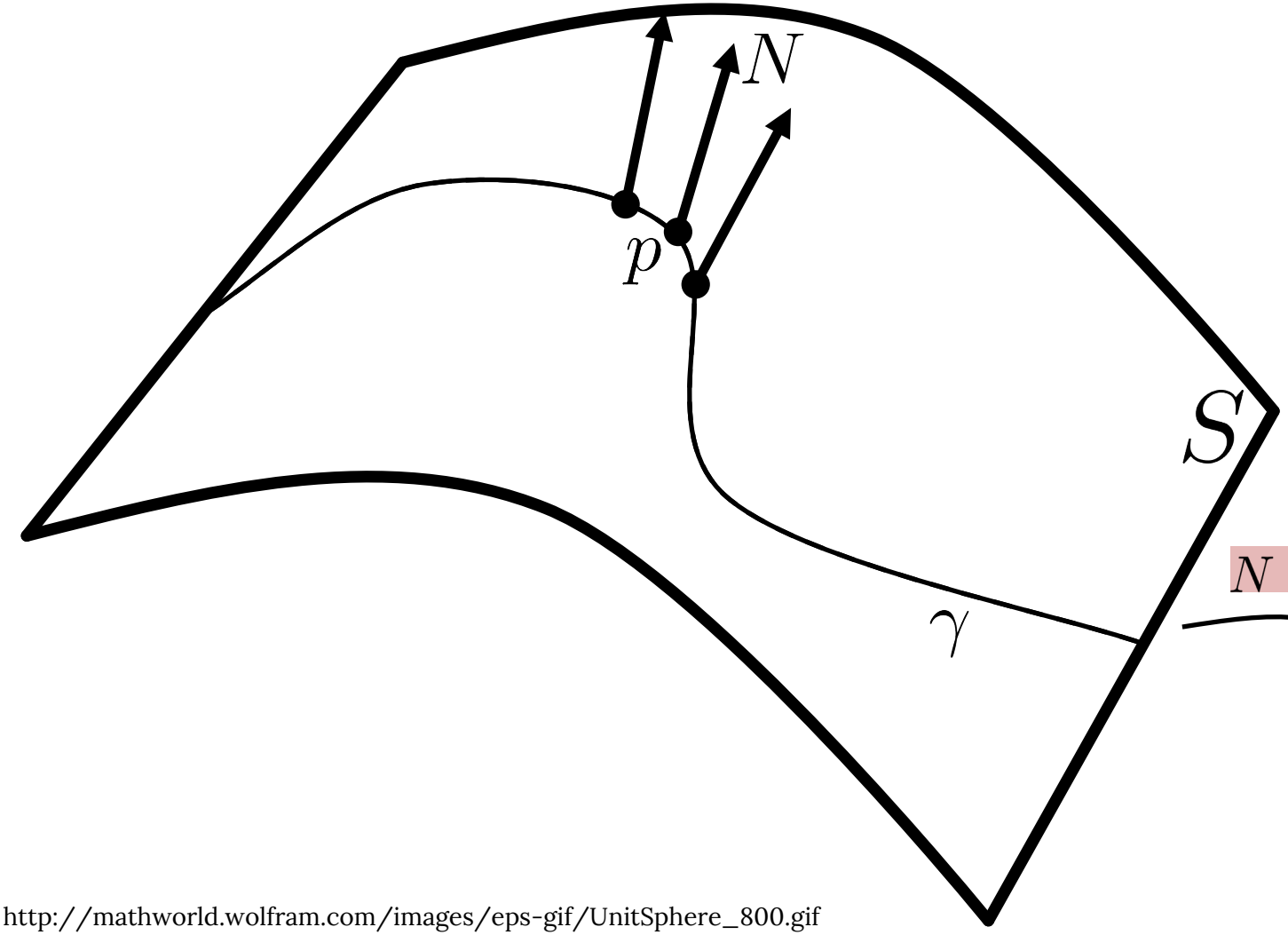


$$\theta'(s) = k(s)$$

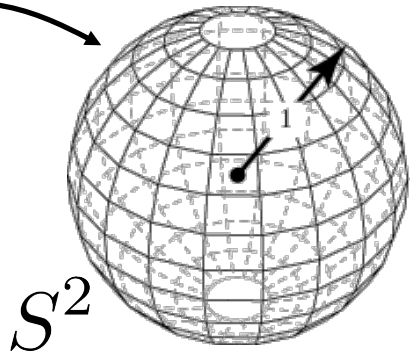
Normal rotates due to curvature

$$\begin{aligned} & T'(s) \\ &= \theta'(s) \cdot \begin{pmatrix} -\sin\theta(s) \\ \cos\theta(s) \end{pmatrix} \\ &= k(s) \cdot \vec{n}(s) \end{aligned}$$

GAUSS MAP FOR SURFACE

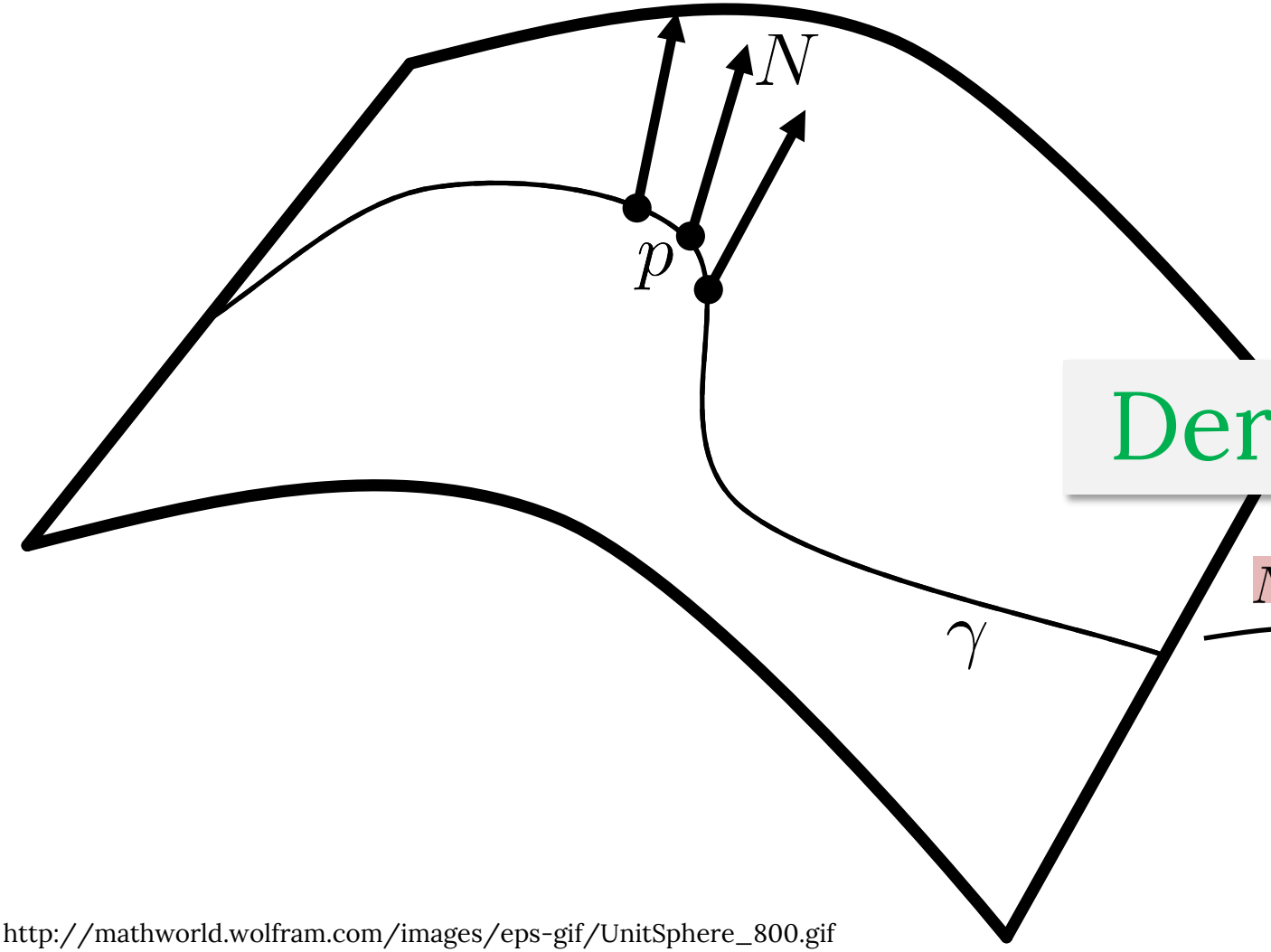


$N : S \to S^2$



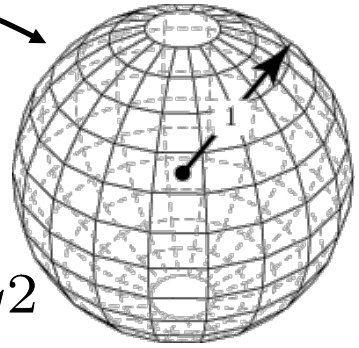
http://mathworld.wolfram.com/images/eps-gif/UnitSphere_800.gif

GAUSS MAP FOR SURFACE



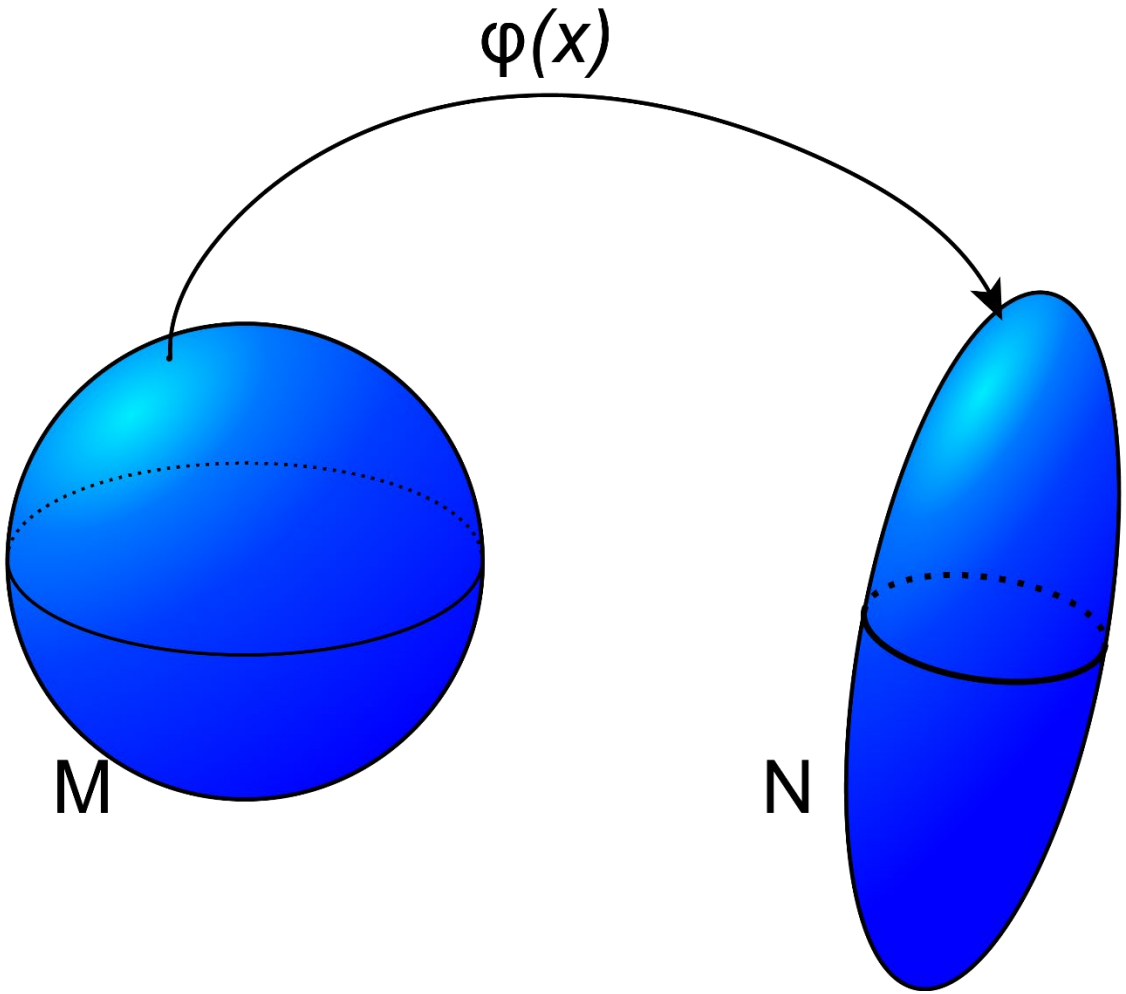
Derivative?

$N : S \to S^2$

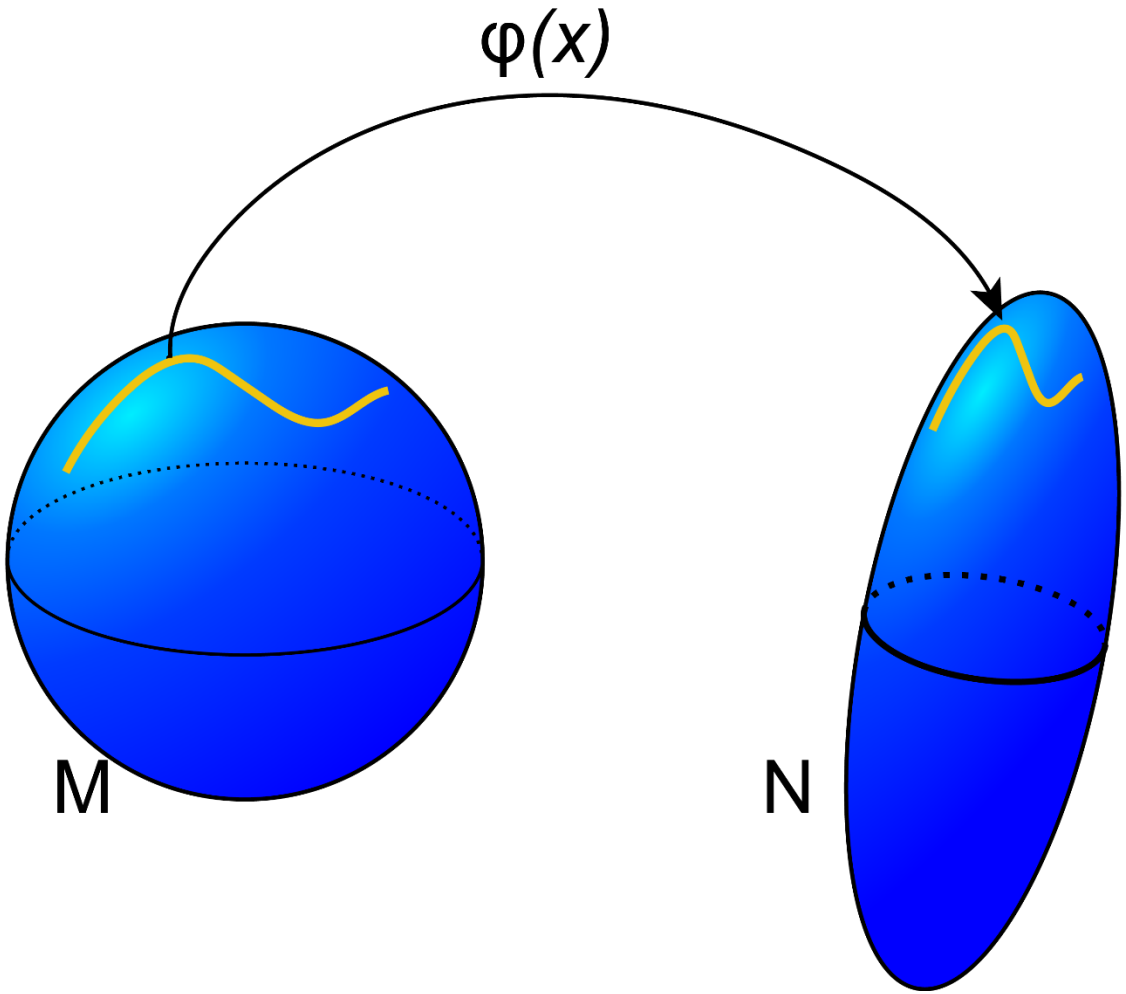


http://mathworld.wolfram.com/images/eps-gif/UnitSphere_800.gif

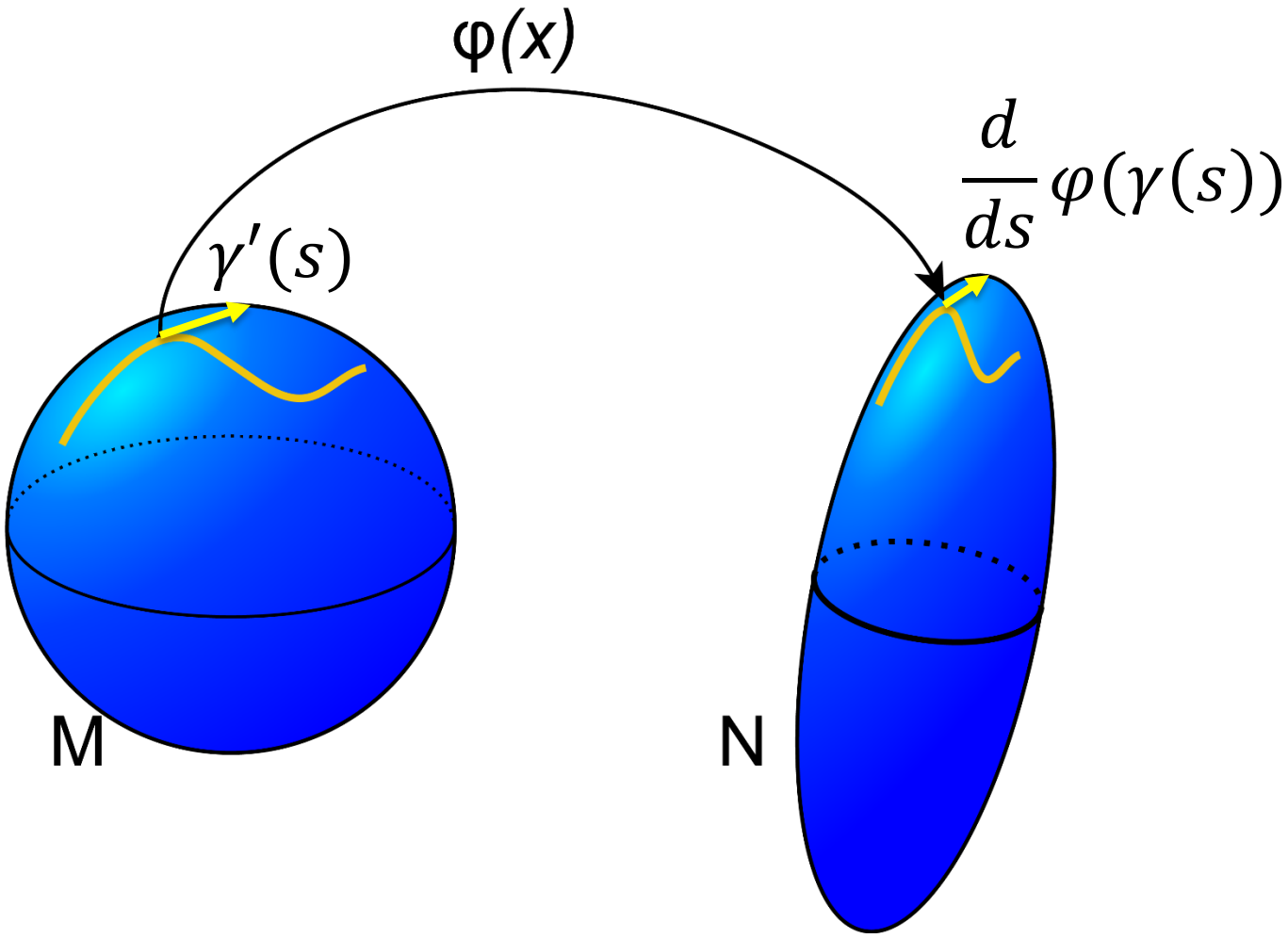
SMOOTH MAPS



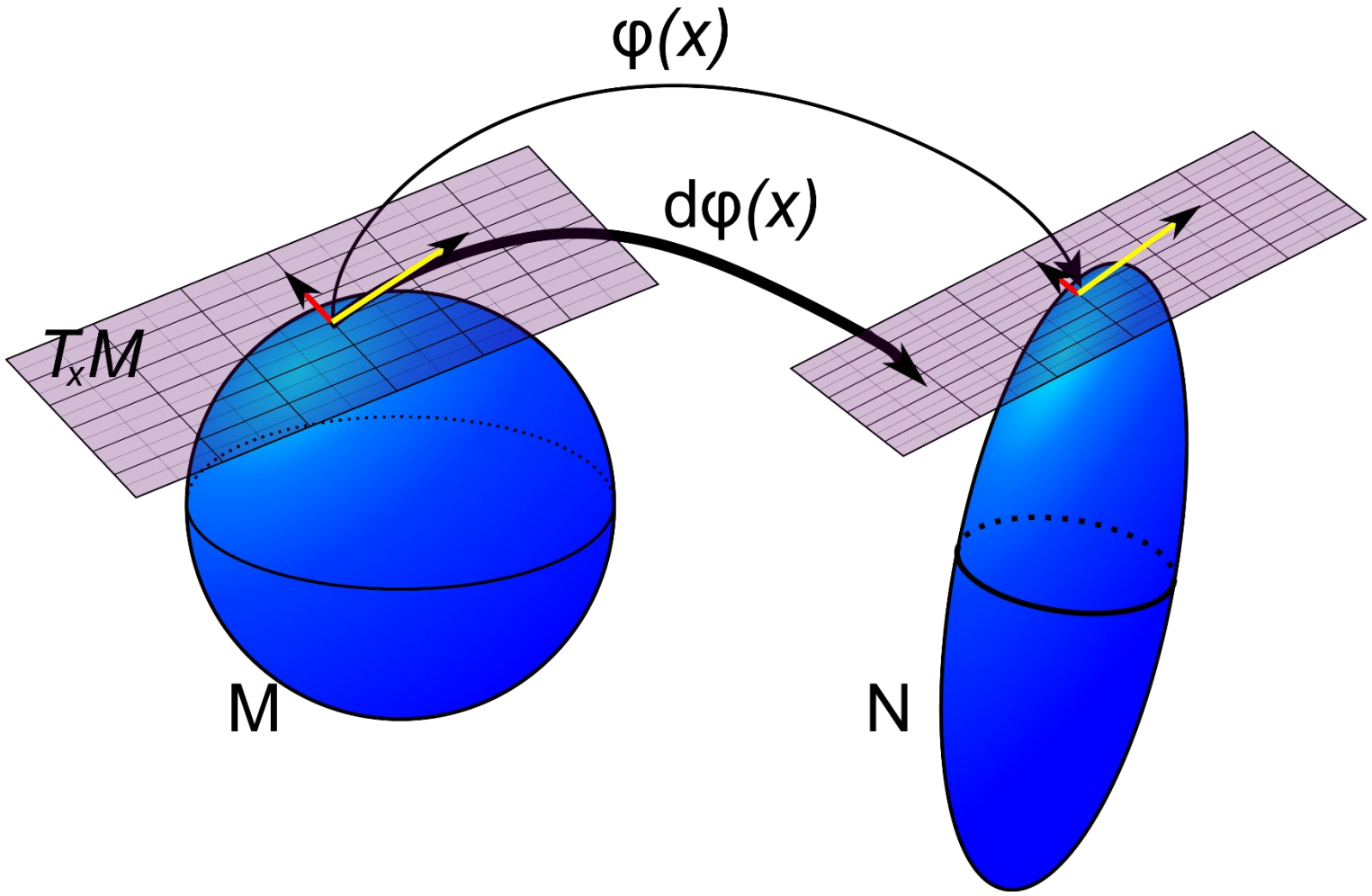
SMOOTH MAPS



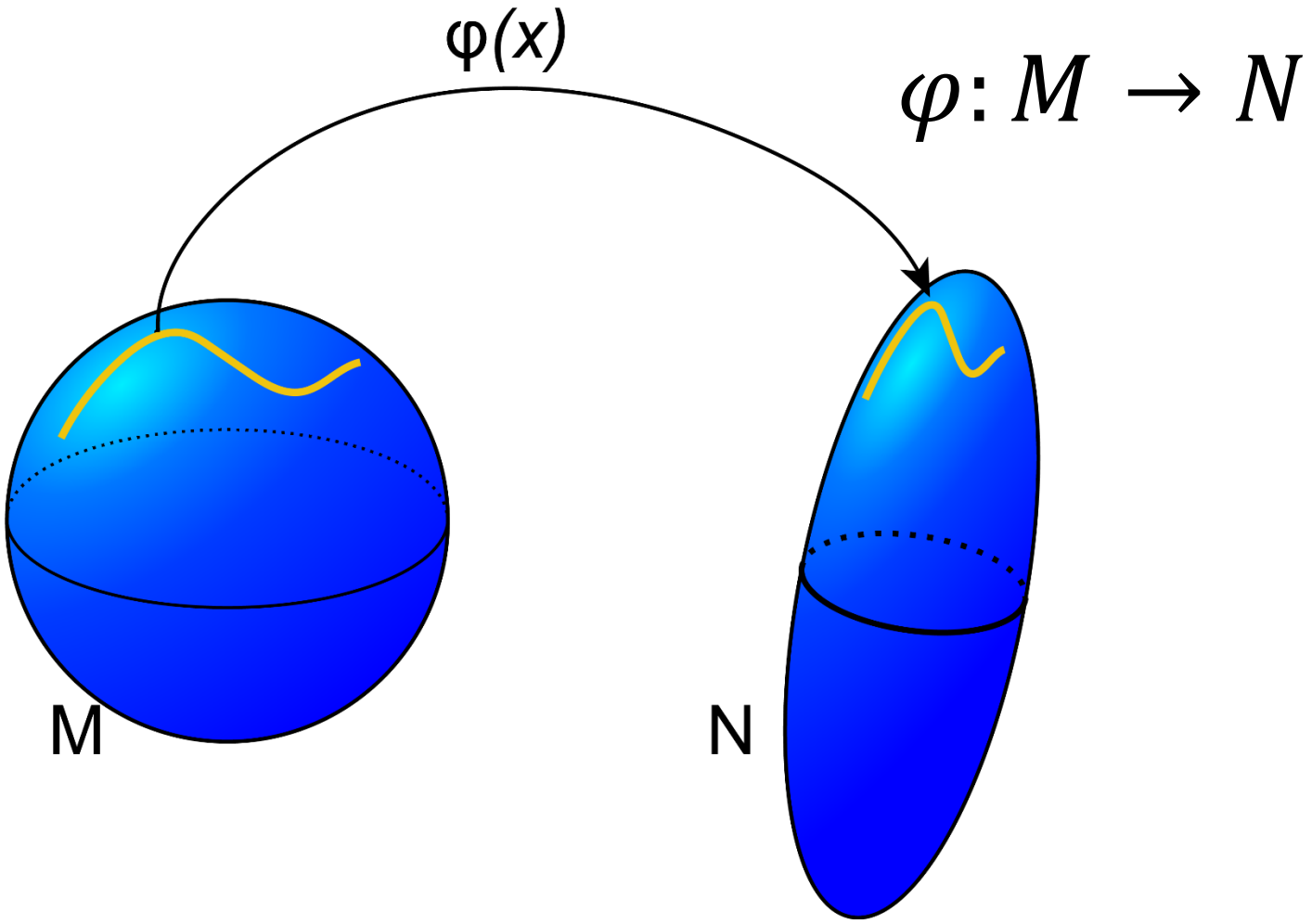
SMOOTH MAPS



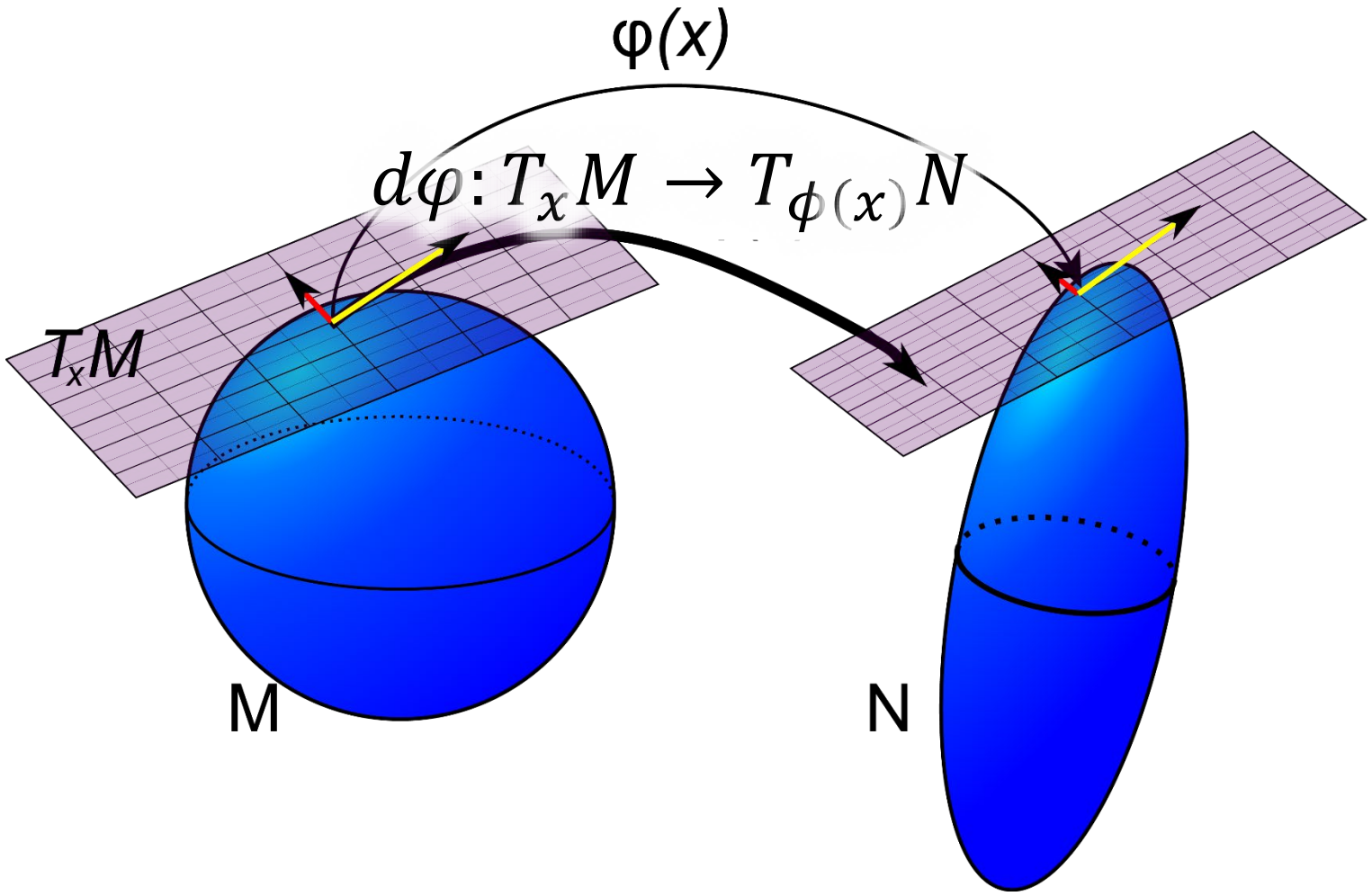
SMOOTH MAPS



SMOOTH MAPS



SMOOTH MAPS



DIFFERENTIAL OF A MAP

Definition

$$d\varphi : T_x M \to T_{\phi(x)} N$$

$$d\varphi \cdot \gamma'(s) \equiv \frac{d\varphi(\gamma(s))}{ds}$$

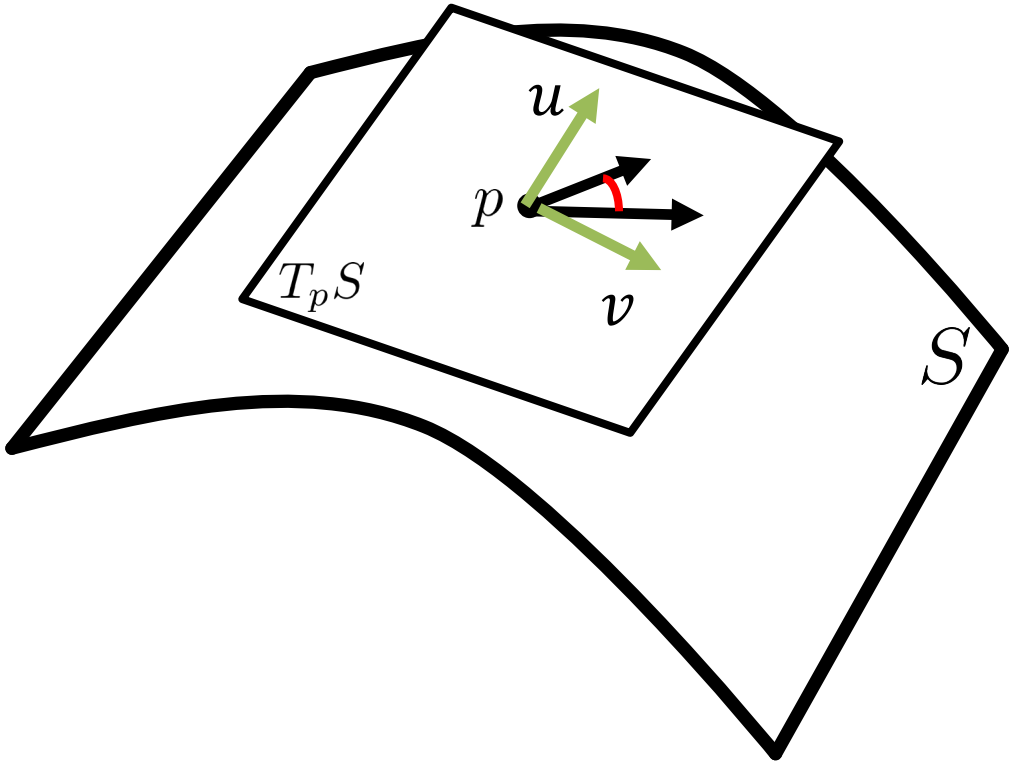
Linear map of tangent spaces

Image from Wikipedia

ANGLES AND LENGTHS

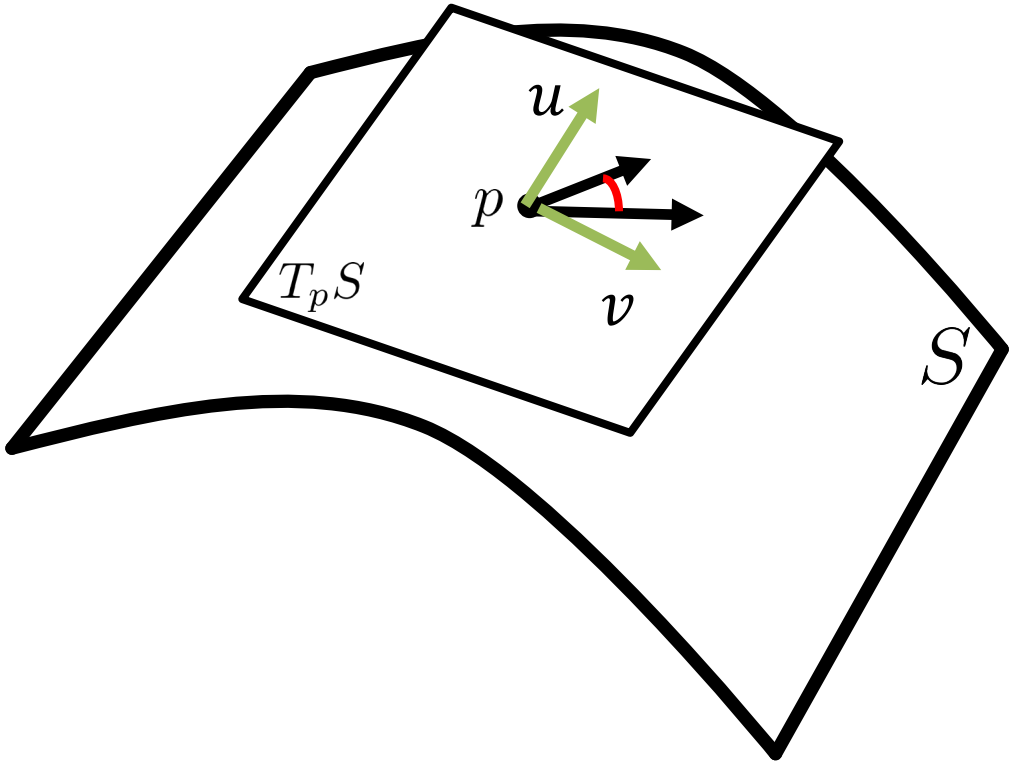
$$a = (u_a, v_a), \qquad b = (u_b, v_b)$$

$$\cos\alpha = \frac{\langle d\varphi(a), d\varphi(b)\rangle}{\sqrt{\langle d\varphi(a), d\varphi(a)\rangle}\sqrt{\langle d\varphi(b), d\varphi(b)\rangle}}$$



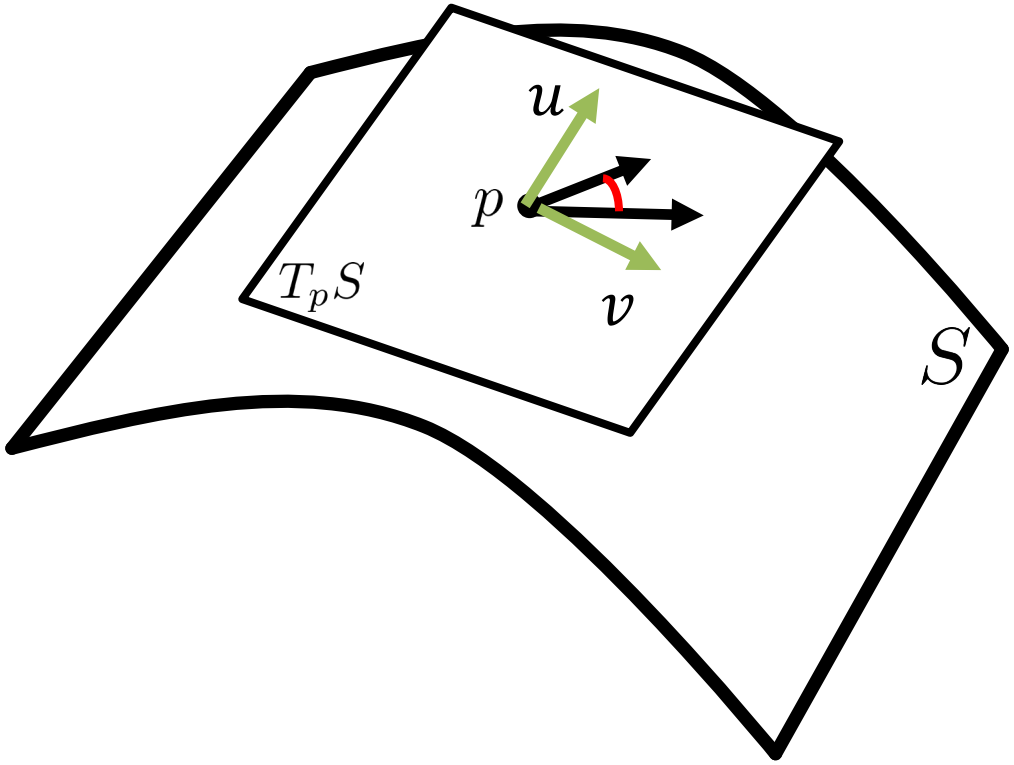
ANGLES AND LENGTHS

$$\cos\alpha = \frac{\langle Ja, Jb\rangle}{\sqrt{\langle Ja, Ja\rangle}\sqrt{\langle Jb, Jb\rangle}}$$



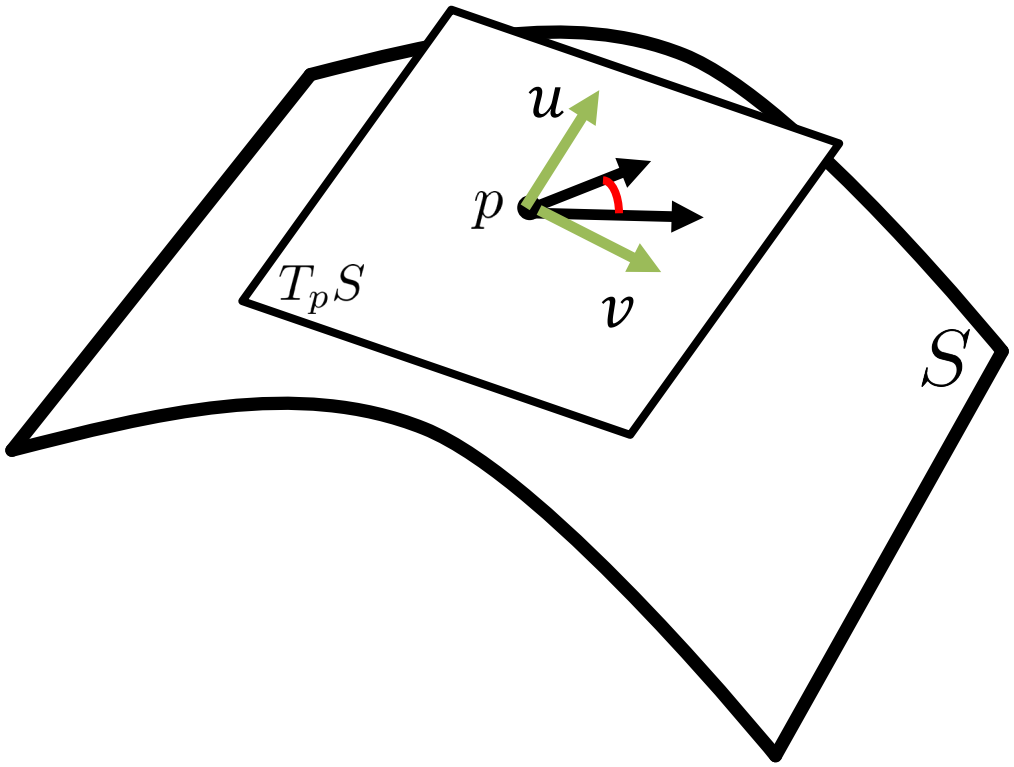
ANGLES AND LENGTHS

$$\cos\alpha = \frac{a^T J^T J b}{\sqrt{a^T J^T J b}\sqrt{a^T J^T J b}}$$



ANGLES AND LENGTHS

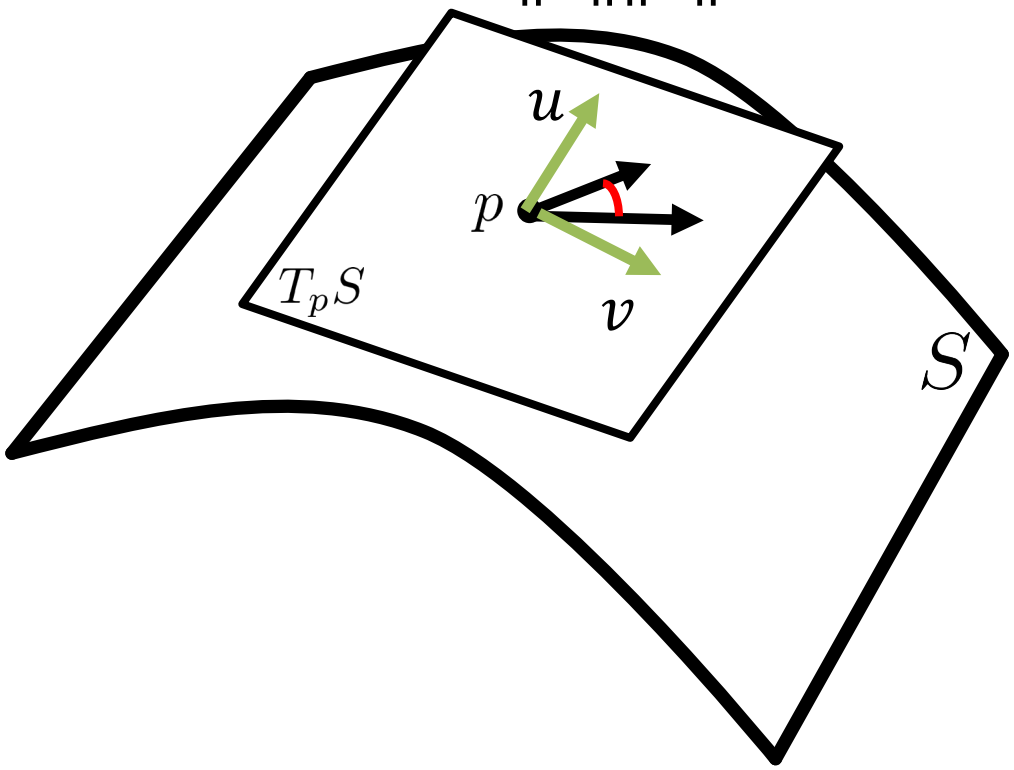
$$\cos\alpha = \frac{a^T \boldsymbol{J^T J} b}{\sqrt{a^T \boldsymbol{J^T J} b}\sqrt{a^T \boldsymbol{J^T J} b}}$$



METRIC TENSOR

AKA FIRST FUNDAMENTAL FORM

$$a = (u_a, v_a)$$
$$b = (u_b, v_b)$$
$$\cos\alpha = \frac{\boldsymbol{g}(a,b)}{\|a\|\|b\|}$$

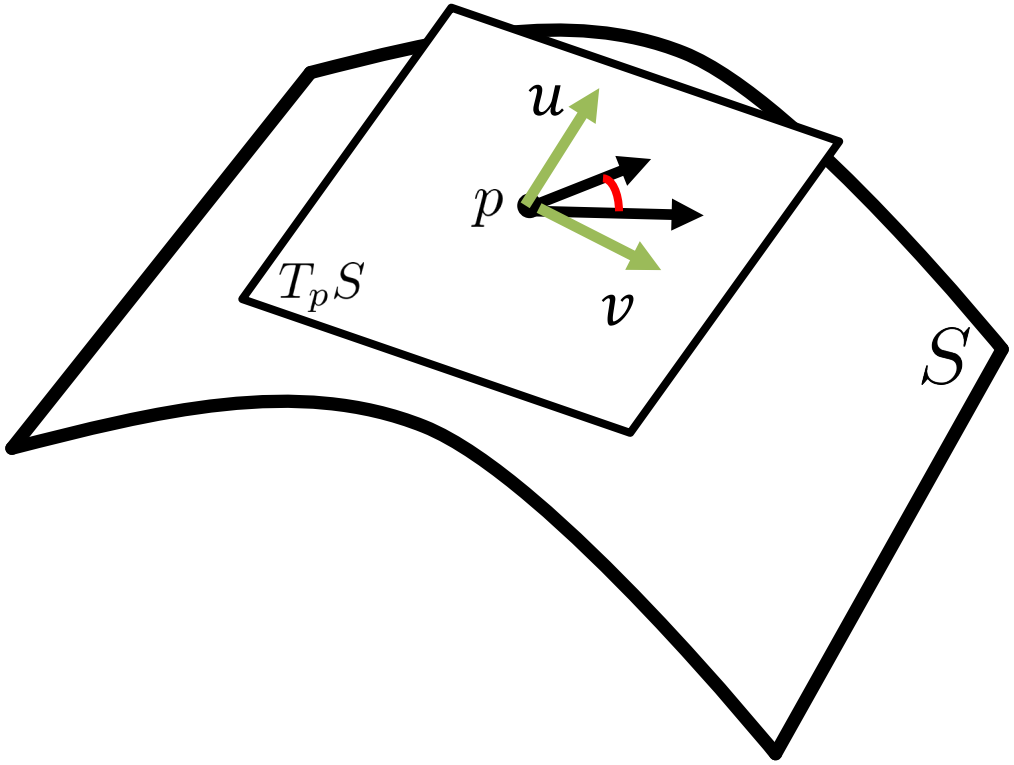


METRIC TENSOR

AKA FIRST FUNDAMENTAL FORM

$$g : T_pS \times T_pS \to \mathbb{R}$$

g is symmetric, positive definite



CALCULATION ON BOARD

Where is the

derivative of N?

Spoiler alert: T_pS

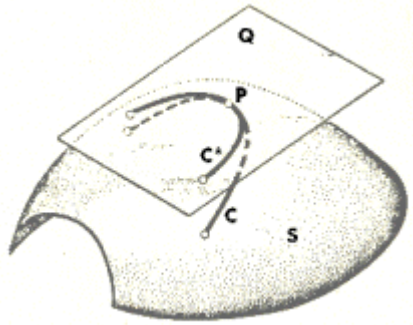
SECOND FUNDAMENTAL FORM

$$DN_p : T_pS \to T_pS$$

$$A_p(V, W) := -\langle DN_p(V), W \rangle$$

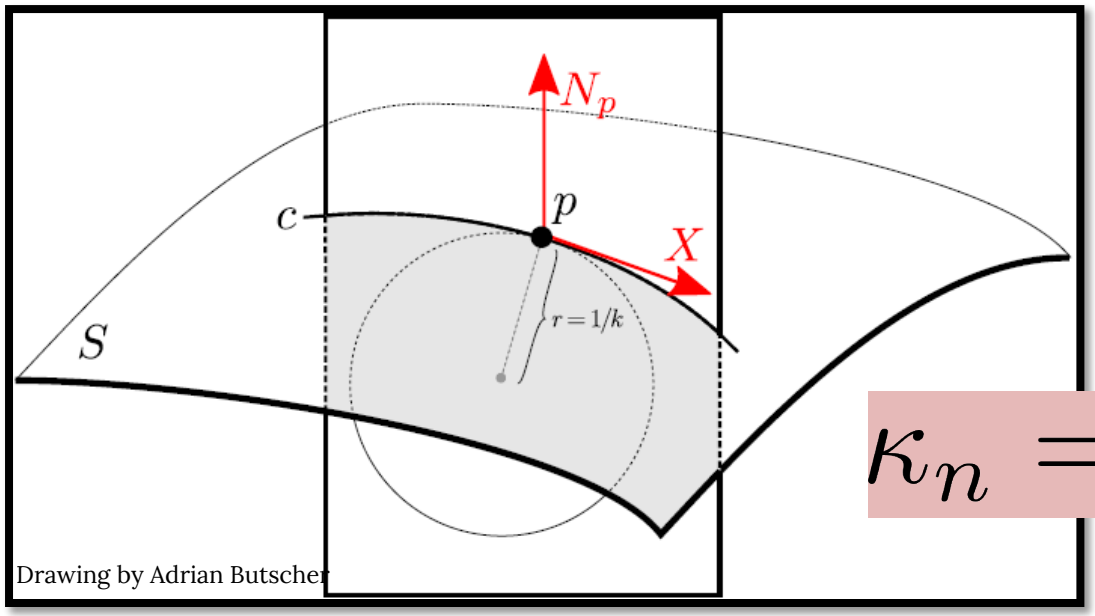
“Shape operator”

RELATIONSHIP TO CURVATURE OF CURVES



http://www.solitaryroad.com/c335.html

$$\kappa_g := \vec{\kappa} \cdot (\vec{N} \times \vec{T})$$



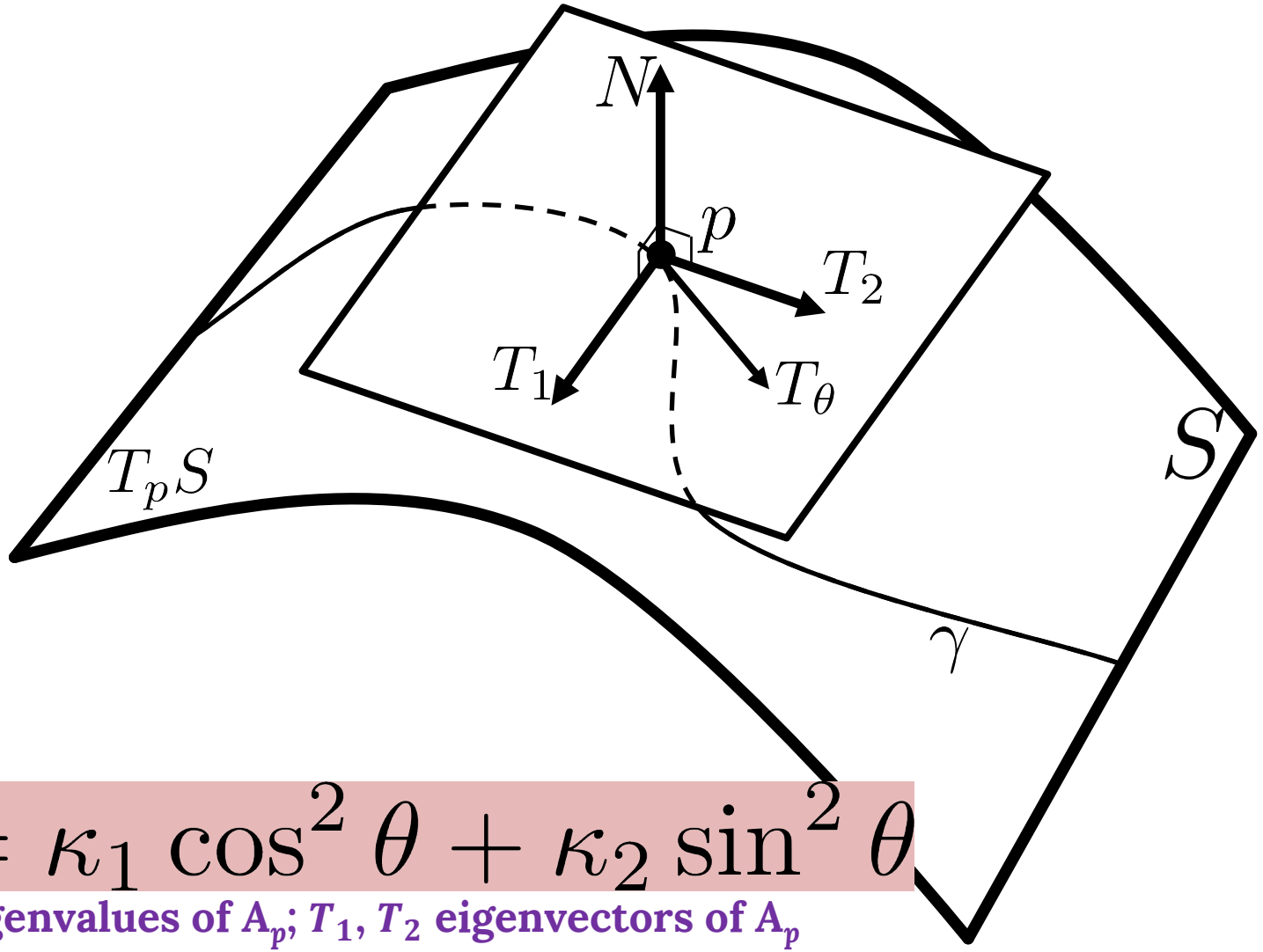
Drawing by Adrian Butscher

$$\kappa_n = A_p(X, X)$$

A_p IS SELF-ADJOINT

(on board)

PRINCIPAL DIRECTIONS AND CURVATURES



$$\kappa_\theta = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta$$

κ_1, κ_2 eigenvalues of A_p; T_1, T_2 eigenvectors of A_p

PRINCIPAL CURVATURES

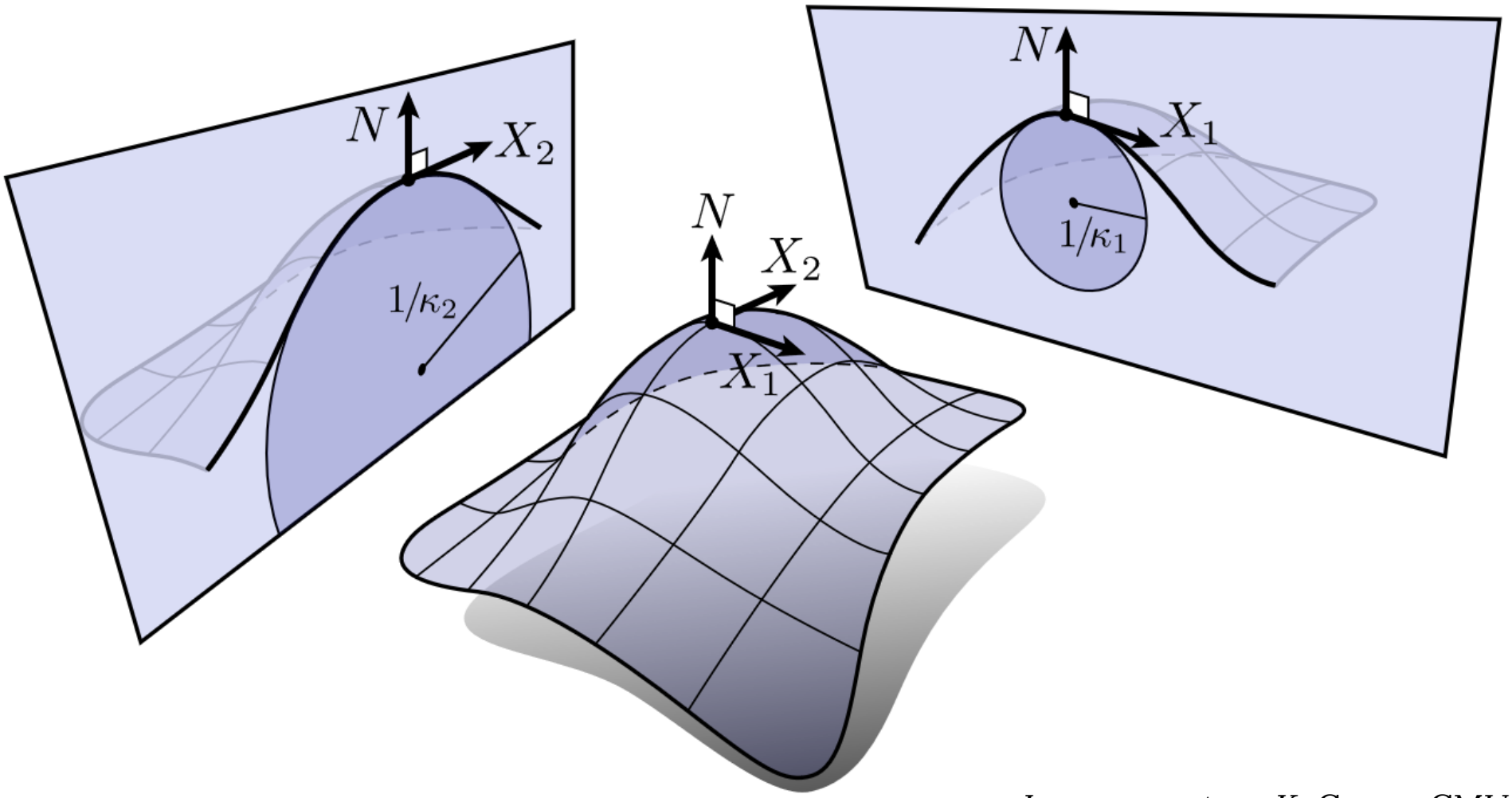
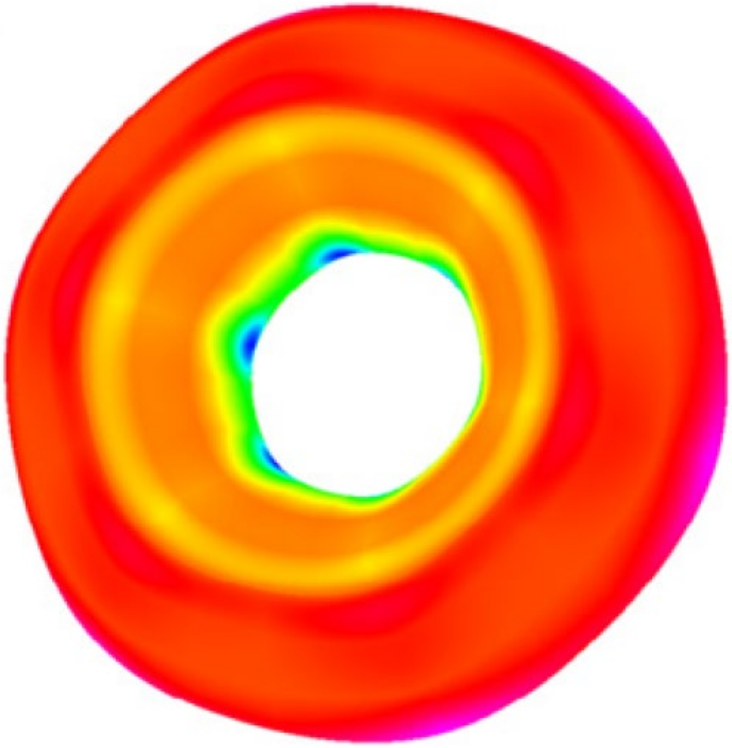
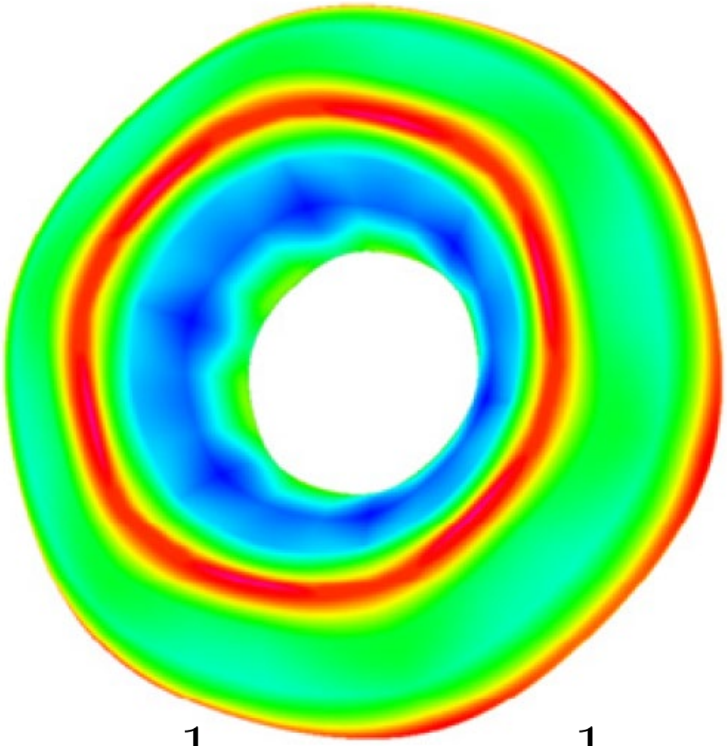


Image courtesy K. Crane, CMU

EXTRINSIC CURVATURE



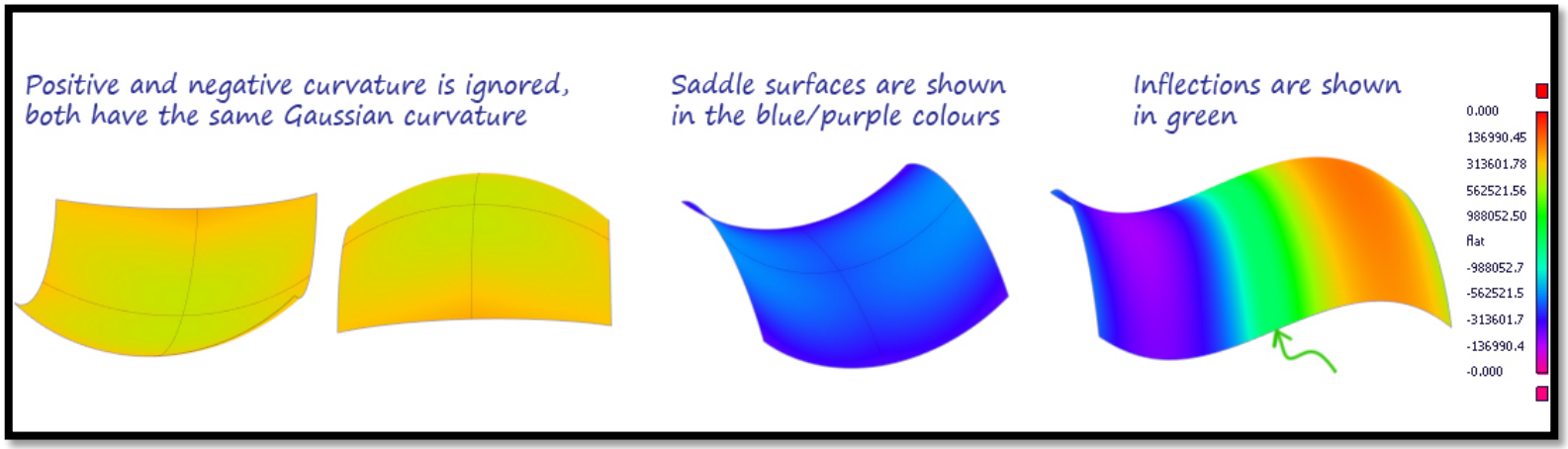
$$K := \kappa_1 \kappa_2 = \det \mathrm{I\!I}$$

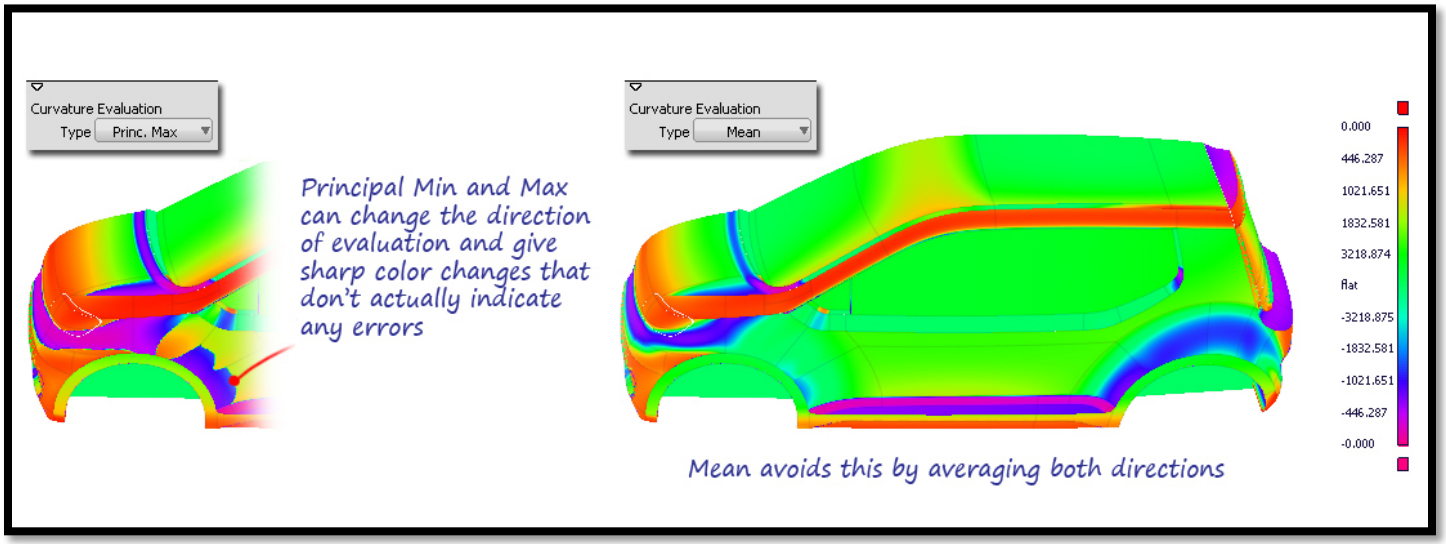


$$H := \frac{1}{2}(\kappa_1 + \kappa_2) = \frac{1}{2}\mathrm{tr}\,\mathrm{I\!I}$$

http://www.sciencedirect.com/science/article/pii/S0010448510001983

INTERPRETATION





http://www.aliasworkbench.com/theoryBuilders/TB7_evaluate3.htm

UNIQUENESS RESULT

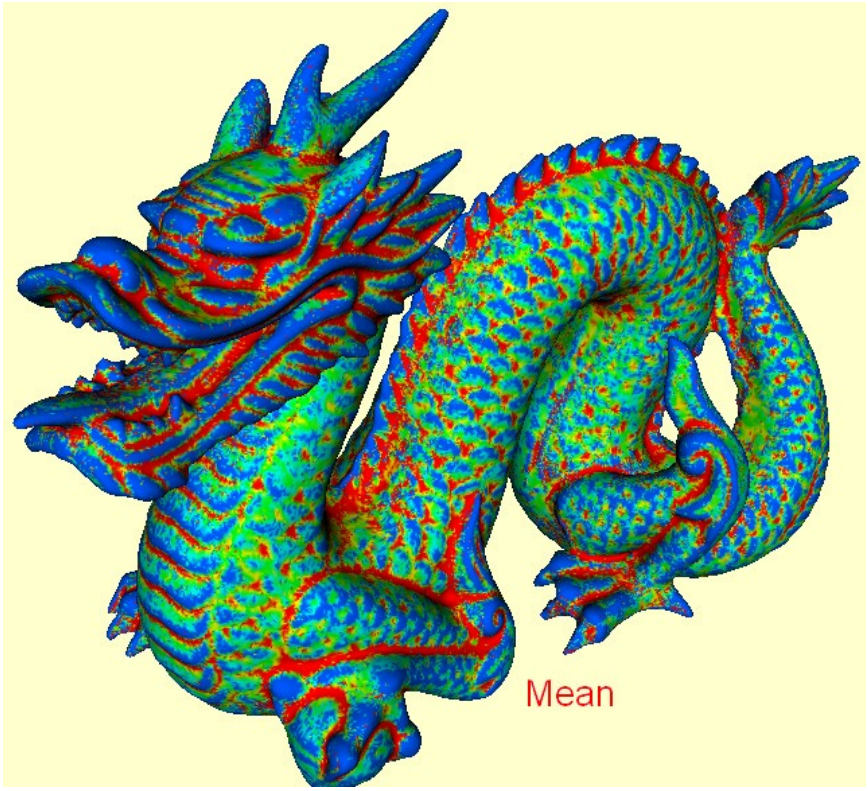
Theorem:

A smooth surface is determined up to rigid motion by its first and second fundamental forms.

WHO CARES?

Curvature completely determines local surface geometry.

USE AS A DESCRIPTOR



http://graphics.ucsd.edu/~iman/Curvature/

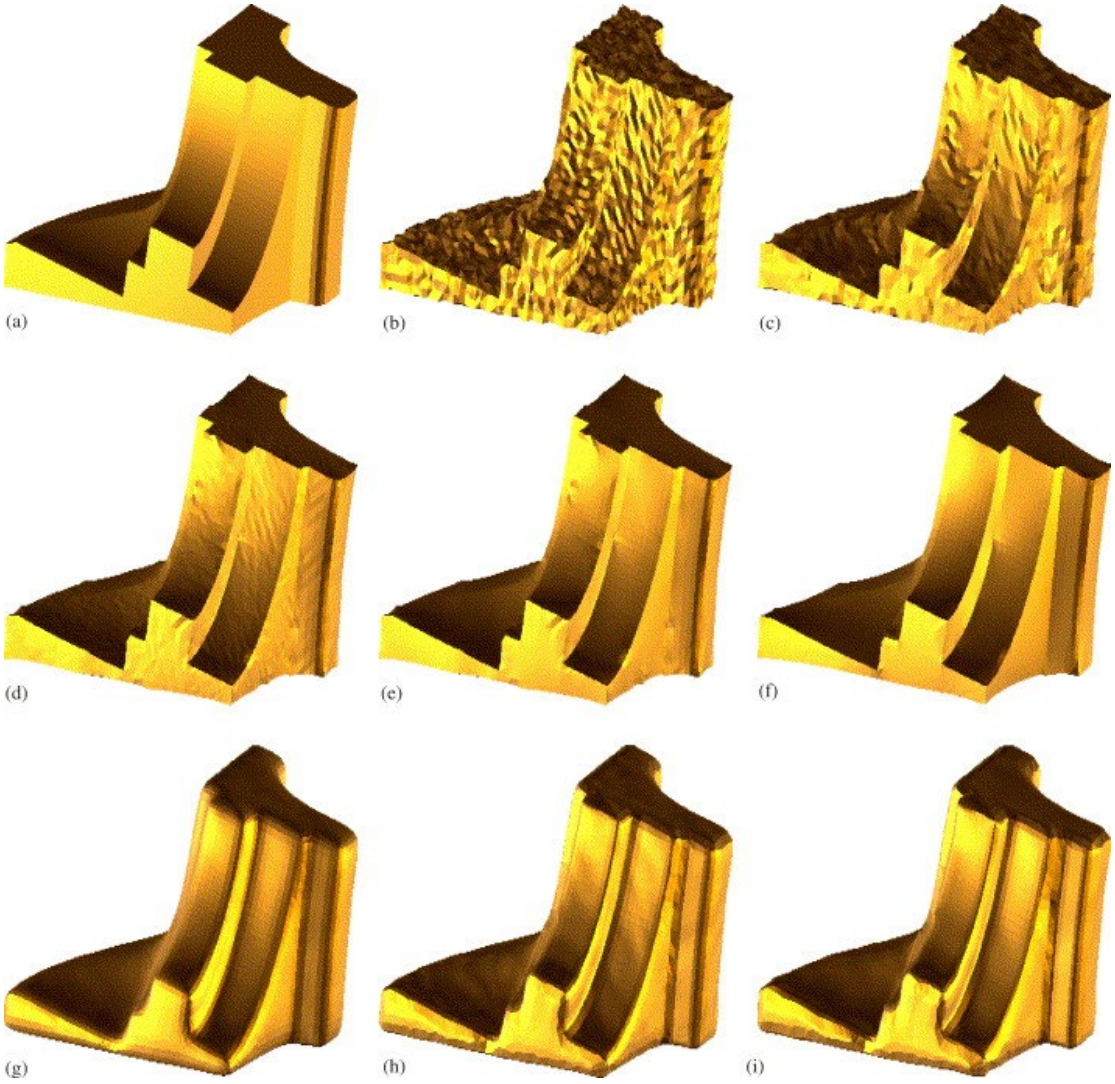
SMOOTHING AND RECONSTRUCTION

Linear Surface Reconstruction from Discrete Fundamental Forms on Triangle Meshes

Wang, Liu, and Tong

Computer Graphics Forum 31.8 (2012)

FAIRNESS MEASURE



Triangular Surface Mesh Fairing via Gaussian Curvature Flow
Zhao, Xu
Journal of Computational and Applied Mathematics 195.1-2 (2006)

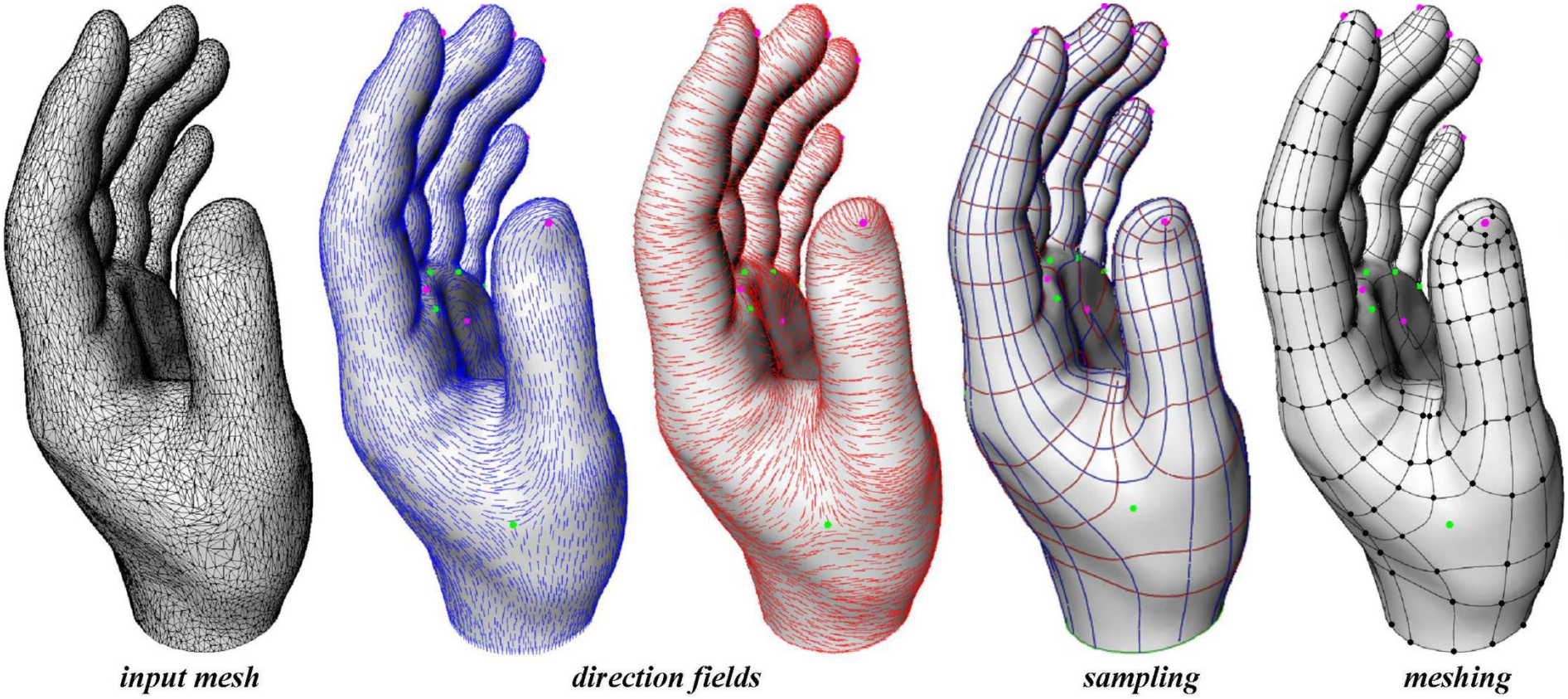
...and many more

GUIDING RENDERING

Highlight Lines for Conveying Shape
DeCarlo, Rusinkiewicz
NPAR (2007)

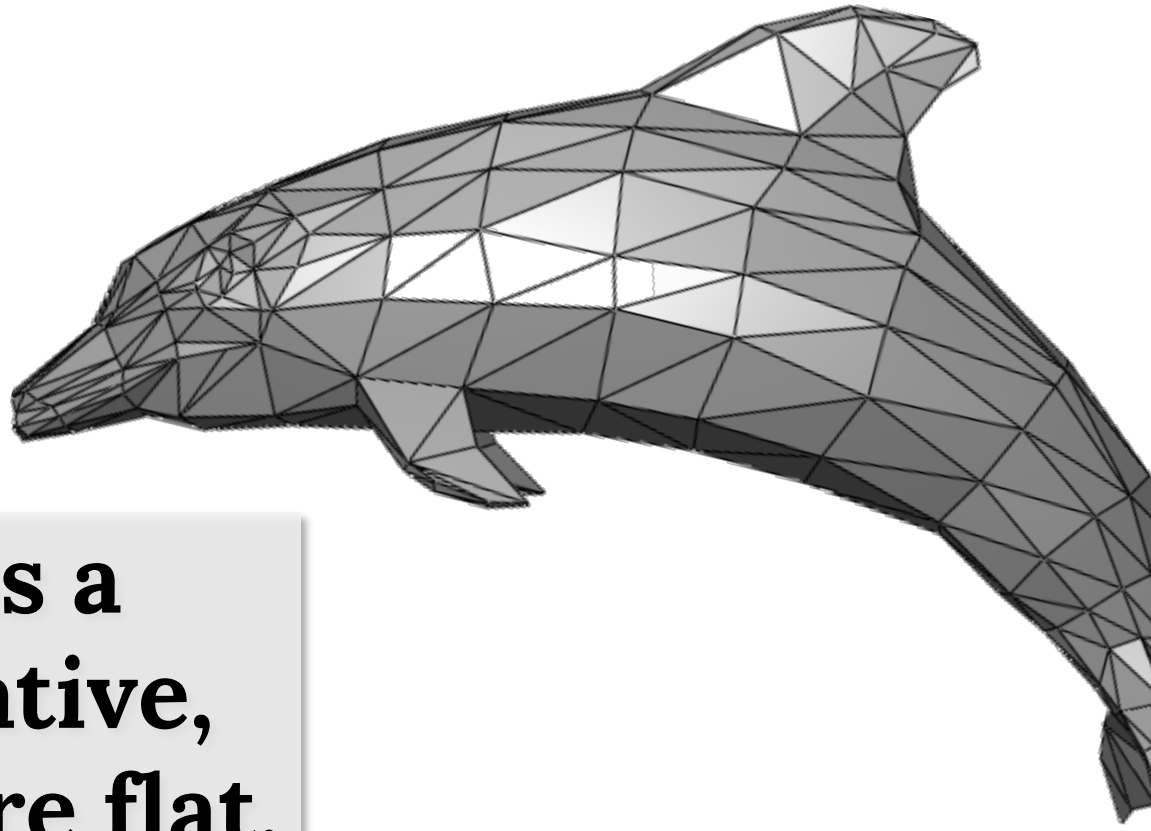
http://www.cs.rutgers.edu/~decarlo/pubs/npar07.pdf

GUIDING MESHING



Anisotropic Polygonal Remeshing
Alliez et al.
SIGGRAPH (2003)

CHALLENGE ON MESHES



Curvature is a second derivative, but triangles are flat.

http://upload.wikimedia.org/wikipedia/commons/f/fb/Dolphin_triangle_mesh.png

STANDARD CITATION

ESTIMATING THE TENSOR OF CURVATURE OF A SURFACE FROM A POLYHEDRAL APPROXIMATION

Gabriel Taubin

IBM T.J.Watson Research Center
P.O.Box 704, Yorktown Heights, NY 10598
taubin@watson.ibm.com

ICCV 1995

Abstract

Estimating principal curvatures and principal directions of a surface from a polyhedral approximation with a large number of small faces, such as those produced by iso-surface construction algorithms, has become a basic step in many computer vision algorithms. Particularly in those targeted at medical applications. In this paper we describe a method to estimate the tensor of curvature of a surface at the vertices of a polyhedral approximation. Principal curvatures and principal directions are obtained by computing in closed form the eigenvalues and eigenvectors of certain 3×3 symmetric matrices defined by integral formulas, and closely related to the matrix representation of the ten-

mate principal curvatures at the vertices of a triangulated surface. Both this algorithm and ours are based on constructing a quadratic form at each vertex of the polyhedral surface and then computing eigenvalues (and eigenvectors) of the resulting form, but the quadratic forms are different. In our algorithm the quadratic form associated with a vertex is expressed as an integral, and is constructed in time proportional to the number of neighboring vertices. In the algorithm of Chen and Schmitt, it is the least-squares solution of an overdetermined linear system, and the complexity of constructing it is quadratic in the number of neighbors.

2 The Tensor of Curvature

TAUBIN MATRIX

$$M := \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

$$\kappa_\theta := \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta$$

$$T_\theta := T_1 \cos \theta + T_2 \sin \theta$$

TAUBIN MATRIX

$$M := \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

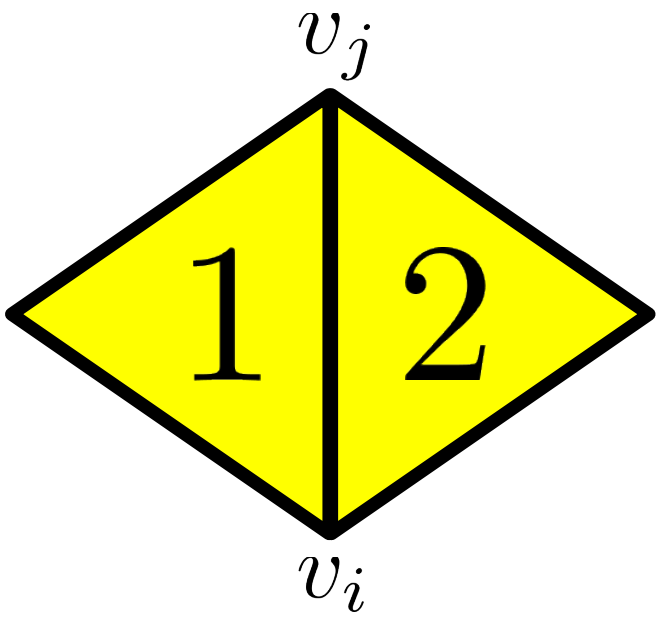
- Eigenvectors are N, T_1, and T_2
- Eigenvalues are $\frac{3}{8}\kappa_1 + \frac{1}{8}\kappa_2$ and $\frac{1}{8}\kappa_1 + \frac{3}{8}\kappa_2$

TAUBIN'S APPROXIMATION

$$M := \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa_\theta T_\theta T_\theta^\top \, d\theta$$

$$\tilde{M}_{v_i} := \sum_{v_j \sim v_i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^\top$$

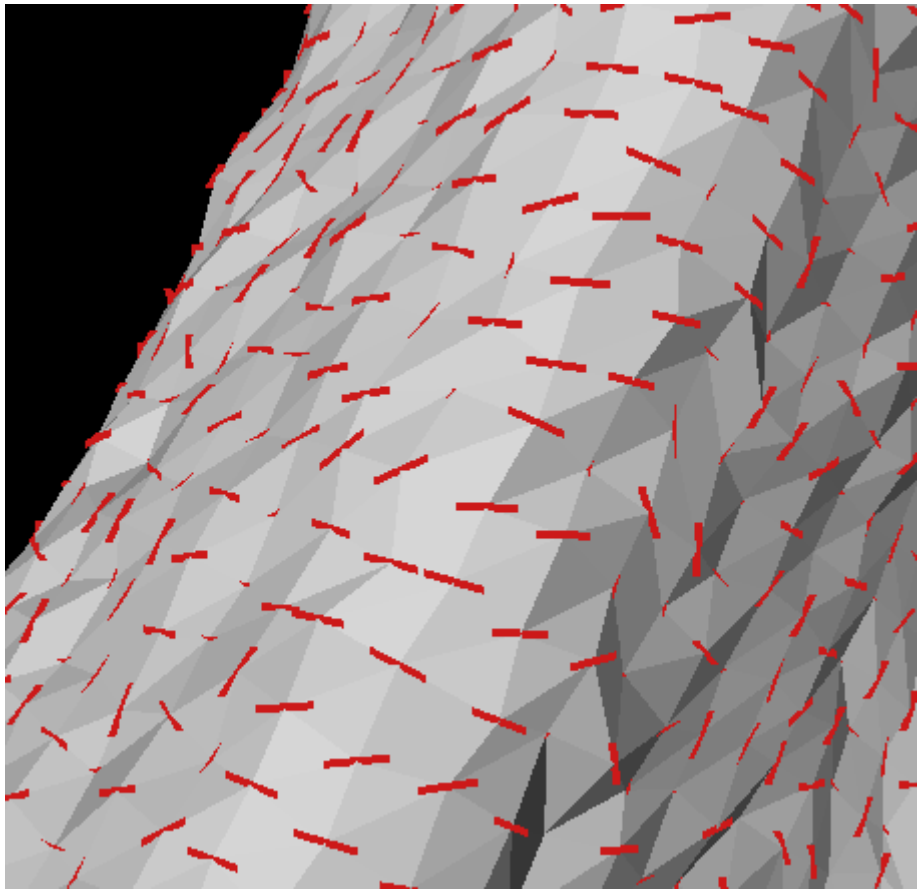
TAUBIN'S APPROXIMATION



Divided difference approximation

$$\tilde{M}_{v_i} := \sum_{v_j \sim v_i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^{\top}$$

PROBLEM



http://iristown.engr.utk.edu/~koschan/paper/CVPR01.pdf

Local estimates are noisy

MAIN TAKE-AWAY

Use application to motivate choice of curvature.

Simulation, smoothing, analysis, meshing, nonphotorealistic rendering, ...

ANOTHER EXAMPLE

Estimating Curvatures and Their Derivatives on Triangle Meshes

Szymon Rusinkiewicz
Princeton University

3DPVT '04

Abstract

The computation of curvature and other differential properties of surfaces is essential for many techniques in analysis and rendering. We present a finite-differences approach for estimating curvatures on irregular triangle meshes that may be thought of as an extension of a common method for estimating per-vertex normals. The technique is efficient in space and time, and results in significantly fewer outlier estimates while more broadly offering accuracy comparable to existing methods. It generalizes naturally to computing derivatives of curvature and higher-order surface differentials.

1 Introduction

As the acquisition and use of sampled 3D geometry become more widespread, 3D models are increasingly becoming the focus of analysis and signal processing techniques previously applied to data types such as audio, images, and video. A key component of algorithms such as feature detection, filtering, and indexing, when applied to both geometry and other data types, is the discrete estimation of differential quantities. In

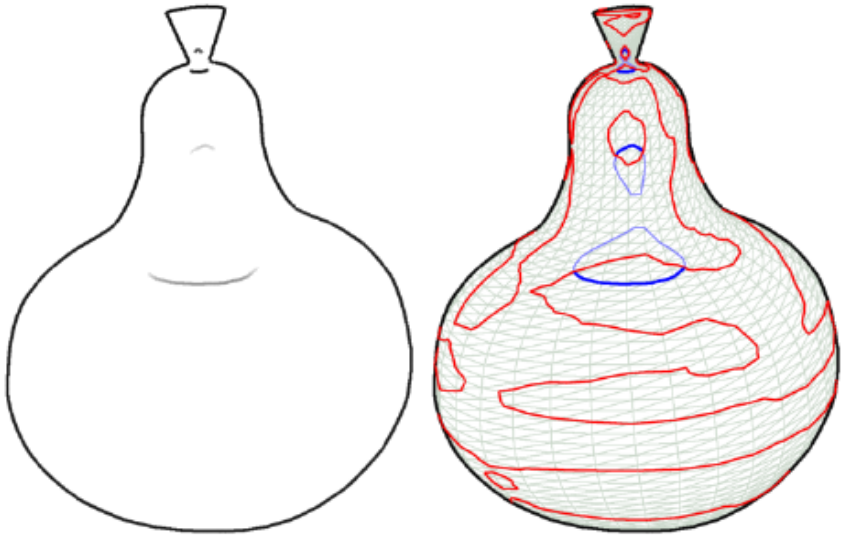


Figure 1: Left: *suggestive contours for line drawings* [DeCarlo et al. 2003] *are a recent example of a driving application for the estimation of curvatures and derivatives of curvature.* ***Right:*** *suggestive contours are drawn along the zeros of curvature in the view direction, shown here in blue, but only where the derivative of curvature in the view direction is positive (the curvature derivative zeros are shown here in red). This paper describes a general*

SECOND FUNDAMENTAL FORM MATRIX

$$II := \begin{pmatrix} D_u N & D_v N \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial N}{\partial u} \cdot \vec{u} & \frac{\partial N}{\partial v} \cdot \vec{u} \\ \frac{\partial N}{\partial u} \cdot \vec{v} & \frac{\partial N}{\partial v} \cdot \vec{v} \end{pmatrix}$$

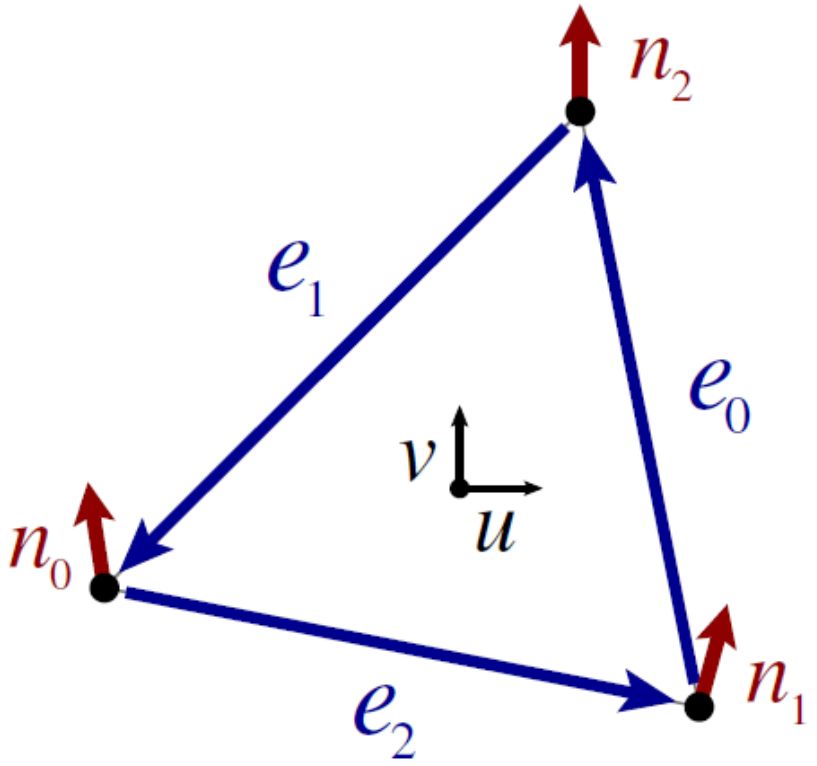
Assume u, v are orthogonal

SECOND FUNDAMENTAL FORM MATRIX

$$\vec{s} = c_1\vec{u} + c_2\vec{v}$$

$$\Longrightarrow II \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = D_{\vec{s}}N$$

FINITE DIFFERENCE PER-FACE



$$\mathbf{II}\begin{pmatrix} e_0 \cdot u \\ e_0 \cdot v \end{pmatrix} = \begin{pmatrix} (n_2 - n_1) \cdot u \\ (n_2 - n_1) \cdot v \end{pmatrix}$$

$$\mathbf{II}\begin{pmatrix} e_1 \cdot u \\ e_1 \cdot v \end{pmatrix} = \begin{pmatrix} (n_0 - n_2) \cdot u \\ (n_0 - n_2) \cdot v \end{pmatrix}$$

$$\mathbf{II}\begin{pmatrix} e_2 \cdot u \\ e_2 \cdot v \end{pmatrix} = \begin{pmatrix} (n_1 - n_0) \cdot u \\ (n_1 - n_0) \cdot v \end{pmatrix}$$

Figure from the paper

Least-squares

AVERAGE FOR PER-VERTEX

- Rotate tangent plane about cross product of normals
- Average using Voronoi weights

COMPLETELY DIFFERENT FORMULA

Consistent Computation of First- and Second-Order Differential Quantities for Surface Meshes

Xiangmin Jiao*
Dept. of Applied Mathematics & Statistics
Stony Brook University

Hongyuan Zha†
College of Computing
Georgia Institute of Technology

Abstract

Differential quantities, including normals, curvatures, principal directions, and associated matrices, play a fundamental role in geometric processing and physics-based modeling. Computing these differential quantities *consistently* on surface meshes is important and challenging, and some existing methods often produce inconsistent results and require *ad hoc* fixes. In this paper, we show that the computation of the gradient and Hessian of a height function provides the foundation for consistently computing the differential quantities. We derive simple, *explicit* formulas for the transformations between the first- and second-order differential quantities (i.e., normal vector and curvature matrix) of a smooth surface and the first- and second-order derivatives (i.e., gradient and Hessian) of its corresponding height function. We then investigate a general, flexible numerical framework to estimate the derivatives of the height function based on local polynomial fittings formulated as weighted least squares approximations. We also propose an iterative fitting

often require *ad hoc* fixes to avoid crashing of the code, and their effects on the accuracy of the applications are difficult to analyze.

The ultimate goal of this work is to investigate a mathematically sound framework that can compute the differential quantities *consistently* (i.e., satisfying the intrinsic constraints) with provable *convergence* on general surface meshes, while being flexible and easy to implement. This is undoubtly an ambitious goal. Although we may have not fully achieved the goal, we make some contributions toward it. First, using the singular value decomposition [Golub and Van Loan 1996] of the Jacobian matrix, we derive explicit formulas for the transformations between the first- and second-order differential quantities of a smooth surface (i.e., normal vector and curvature matrix) and the first- and second-order derivatives of its corresponding height function (i.e., gradient and Hessian). We also give the explicit formulas for the transformations of the gradient and Hessian under a rotation of the coordinate system. These transformations can be obtained without forming the shape operator and the associated computation of its eigenvalues or eigenvectors. We

COMPLETELY DIFFERENT FORMULA

Consistent Computation of First- and Second-Order Differential Quantities for Surface Meshes

Xiangmin Jiao*
Dept. of Applied Mathematics & Statistics
Stony Brook University

Hongyuan Zha†
College of Computing
Georgia Institute of Technology

Abstract

Differential quantities, including normals, curvatures, principal directions, and associated matrices, play a fundamental role in geo-

ible numerical framework to estimate the derivatives of the height function based on local polynomial fittings formulated as weighted least squares approximations. We also propose an iterative fitting scheme to improve accuracy. This framework generalizes polyno-

often require *ad hoc* fixes to avoid crashing of the code, and their effects on the accuracy of the applications are difficult to analyze.

The ultimate goal of this work is to investigate a mathematically sound framework that can compute the differential quantities *con-*

and Hessian under a rotation of the coordinate system. These transformations can be obtained without forming the shape operator and the associated computation of its eigenvalues or eigenvectors. We

Theorem 3 *The mean and Gaussian curvature of the height function* $f(\boldsymbol{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ *are*

$$\kappa_H = \frac{tr(\boldsymbol{H})}{2\ell} - \frac{(\nabla f)^T \boldsymbol{H} (\nabla f)}{2\ell^3}, \text{ and } \kappa_G = \frac{det(\boldsymbol{H})}{\ell^4}. \qquad (16)$$

CONSERVED QUANTITY APPROACH

Discrete Differential-Geometry Operators for Triangulated 2-Manifolds

Mark Meyer[1], Mathieu Desbrun[1,2], Peter Schröder[1], and Alan H. Barr[1]

[1] Caltech
[2] USC

Visualization and Math. III

Summary. This paper proposes a unified and consistent set of flexible tools to approximate important geometric attributes, including normal vectors and curvatures on arbitrary triangle meshes. We present a consistent derivation of these first and second order differential properties using *averaging Voronoi cells* and the mixed Finite-Element/Finite-Volume method, and compare them to existing formulations. Building upon previous work in discrete geometry, these operators are closely related to the continuous case, guaranteeing an appropriate extension from the continuous to the discrete setting: they respect most intrinsic properties of the continuous differential operators. We show that these estimates are optimal in accuracy under mild smoothness conditions, and demonstrate their numerical quality. We also present applications of these operators, such as mesh smoothing, enhancement, and quality checking, and show results of denoising in higher dimensions, such as for tensor images.

DISCRETE DIFFERENTIAL GEOMETRY

Structure preservation:

- Keeping properties from the continuous abstraction exactly true in a discretization.

GAUSS-BONNET THEOREM

$$\int_M K dA + \int_{\partial M} k_g ds = 2\pi(\chi(M) - n_b)$$

Number of boundaries

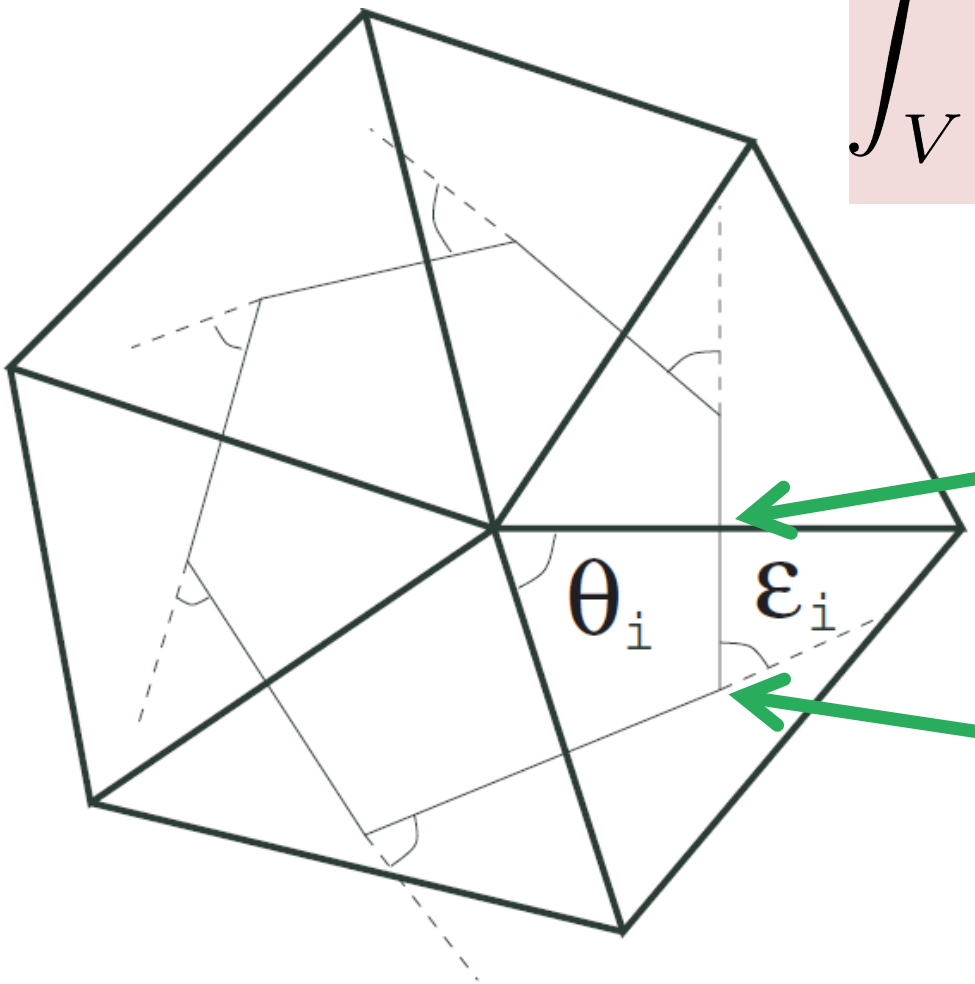
Gaussian curvature

Geodesic curvature (curvature projected on tangent plane)

2-2g

FOR POLYGONAL CELLS

$$\int_V K \, dA = 2\pi - \sum_j \varepsilon_j$$



Change is in normal direction

Turning angle integrated curvature

Figure from the paper

FLIP THINGS BACKWARD

DEFINITION:

Gaussian curvature integrated over region V is given by

$$\int_V K \, dA = 2\pi - \sum_j \theta_j$$

Divide by area for curvature estimate

Recall:

EULER CHARACTERISTIC

$$V - E + F := \chi$$

$$\chi = 2 - 2g$$

$g = 0$

$g = 1$

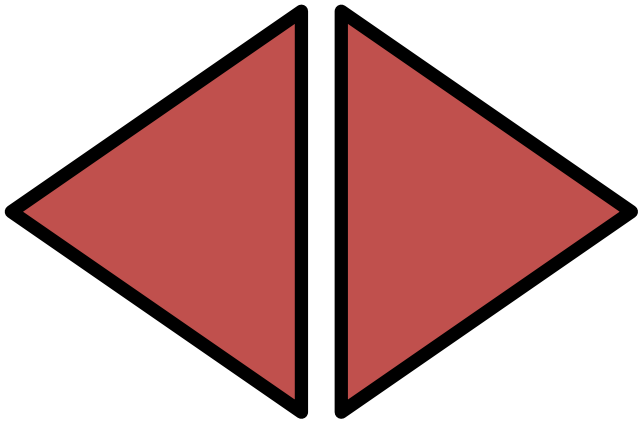
$g = 2$

Recall:

CONSEQUENCES FOR TRIANGLE MESHES

$$V - E + F := \chi$$

"Each edge is adjacent to two faces. Each face has three edges."



$$2E = 3F$$

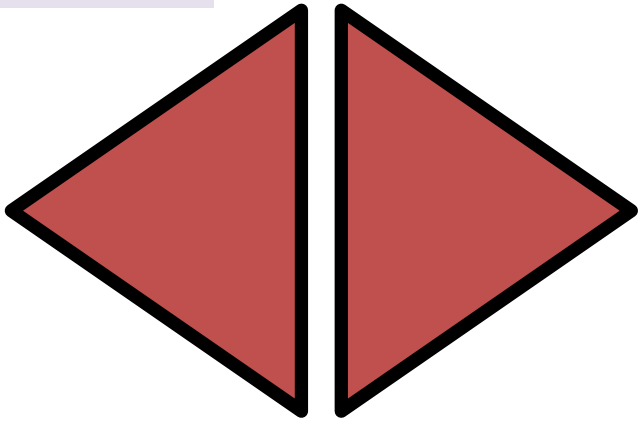
Closed mesh: Easy estimates!

Recall:

CONSEQUENCES FOR TRIANGLE MESHES

$$V - \frac{1}{2}F := \chi$$

"Each edge is adjacent to two faces. Each face has three edges."



$$2E = 3F$$

Closed mesh: Easy estimates!

Recall:

CONSEQUENCES FOR TRIANGLE MESHES

Big number

$$V - \frac{1}{2}F := \chi$$

Small number

$$F \approx 2V$$

Closed mesh: Easy estimates!

DISCRETE GAUSS-BONNET

$$\int_M K \, dA = \sum_i \int_{V_i} K \, dA$$

Partition the surface

DISCRETE GAUSS-BONNET

$$\int_M K\,dA = \sum_i \int_{V_i} K\,dA$$

$$= \sum_i \left(2\pi - \sum_j \theta_{ij} \right)$$

Apply our definition

DISCRETE GAUSS-BONNET

$$\int_M K\,dA = \sum_i \int_{V_i} K\,dA$$

$$= \sum_i \left(2\pi - \sum_j \theta_{ij} \right)$$

$$= 2\pi V - \sum_{ij} \theta_{ij}$$

Pull out constants

DISCRETE GAUSS-BONNET

$$\begin{aligned}\int_M K\,dA &= \sum_i \int_{V_i} K\,dA \\ &= \sum_i \left(2\pi - \sum_j \theta_{ij}\right) \\ &= 2\pi V - \sum_{ij} \theta_{ij} \\ &= 2\pi V - \pi F\end{aligned}$$

Consider sum over triangles

DISCRETE GAUSS-BONNET

$$\int_M K\,dA = \sum_i \int_{V_i} K\,dA$$

$$= \sum_i \left(2\pi - \sum_j \theta_{ij} \right)$$

$$= 2\pi V - \sum_{ij} \theta_{ij}$$

$$= 2\pi V - \pi F$$

$$= \pi(2V - F)$$

$$= 2\pi\chi$$

By definition

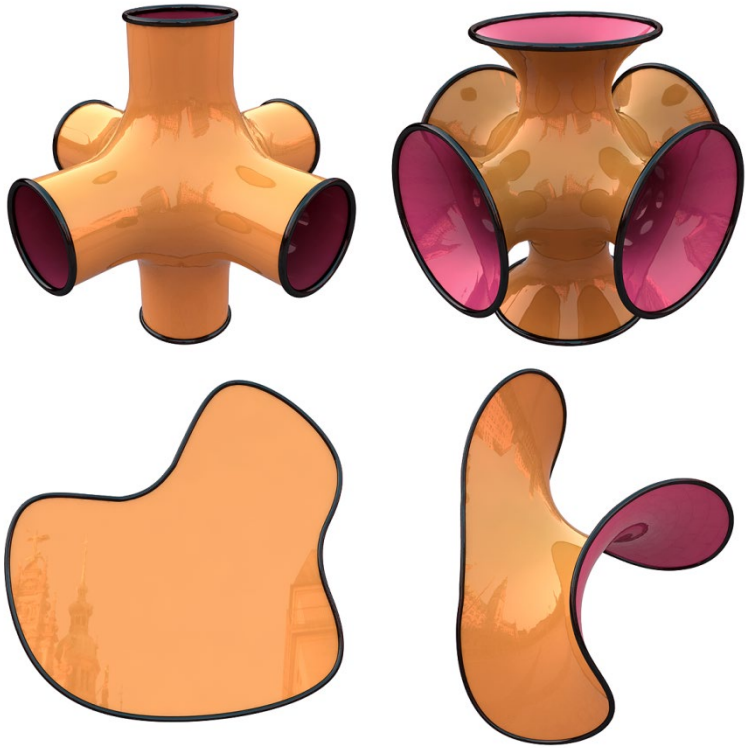
<qed/>

MEAN CURVATURE NORMAL

$$E(M) = \text{Area}(M)$$

$$\nabla E(p) = H\vec{n}$$

“Variational derivative”

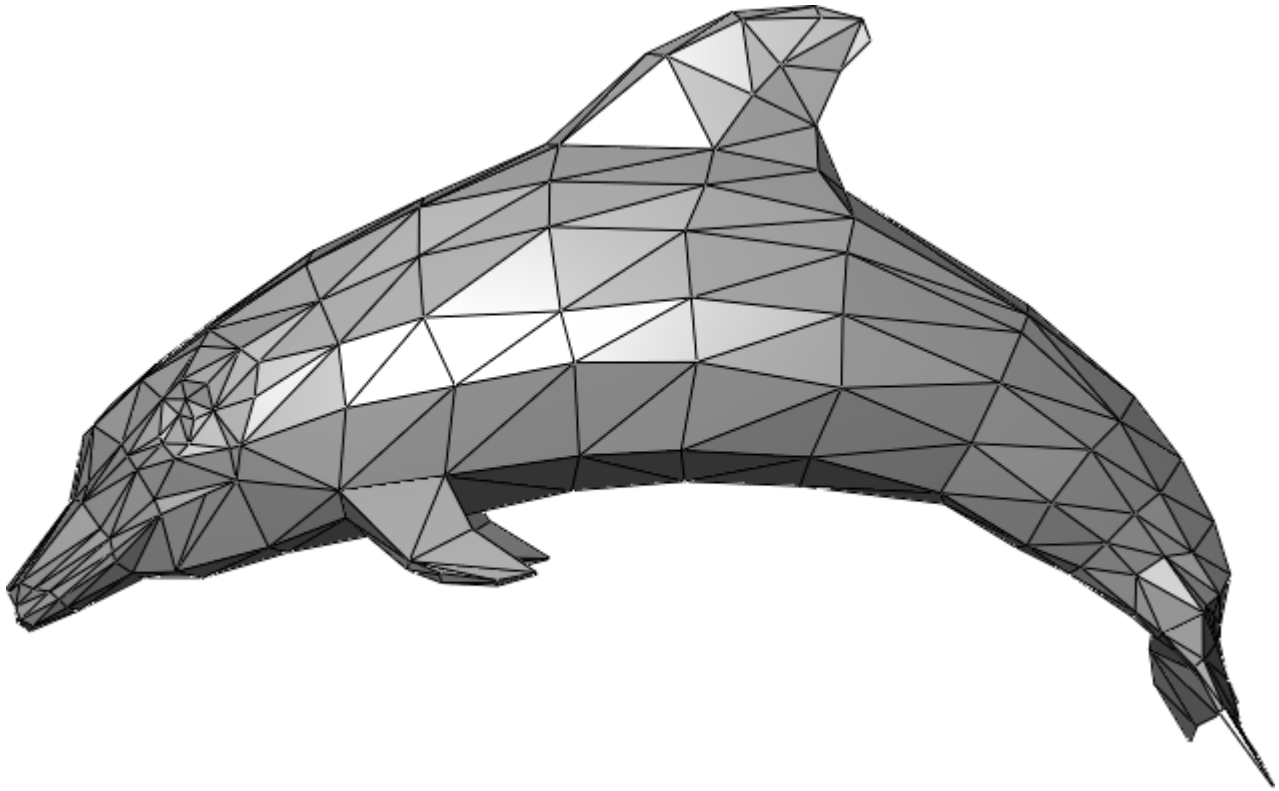


$$\nabla E(p) \equiv 0 \ \forall p \in \text{int}\, M$$

Minimal surfaces

Image courtesy K. Crane

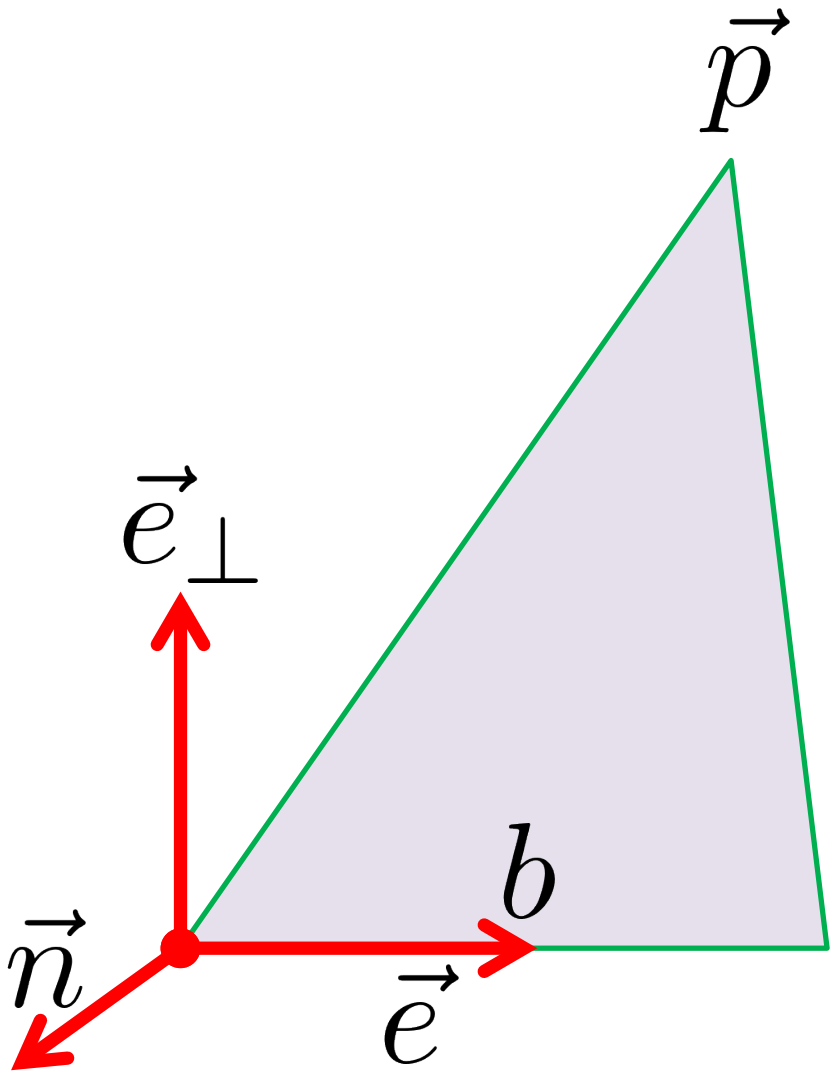
AREA FUNCTIONAL FOR MESHES



$$\text{Area} : \mathbb{R}^{3V} \to \mathbb{R}$$

http://upload.wikimedia.org/wikipedia/commons/f/fb/Dolphin_triangle_mesh.png

SINGLE TRIANGLE



$$\vec{p} = p_n\vec{n} + p_e\vec{e} + p_\perp\vec{e}_\perp$$

$$A = \frac{1}{2}b\sqrt{p_n^2 + p_\perp^2}$$

As a function of $\vec{p}$

SINGLE TRIANGLE: DERIVATIVES

$$\vec{p} = p_n \vec{n} + p_e \vec{e} + p_\perp \vec{e}_\perp$$

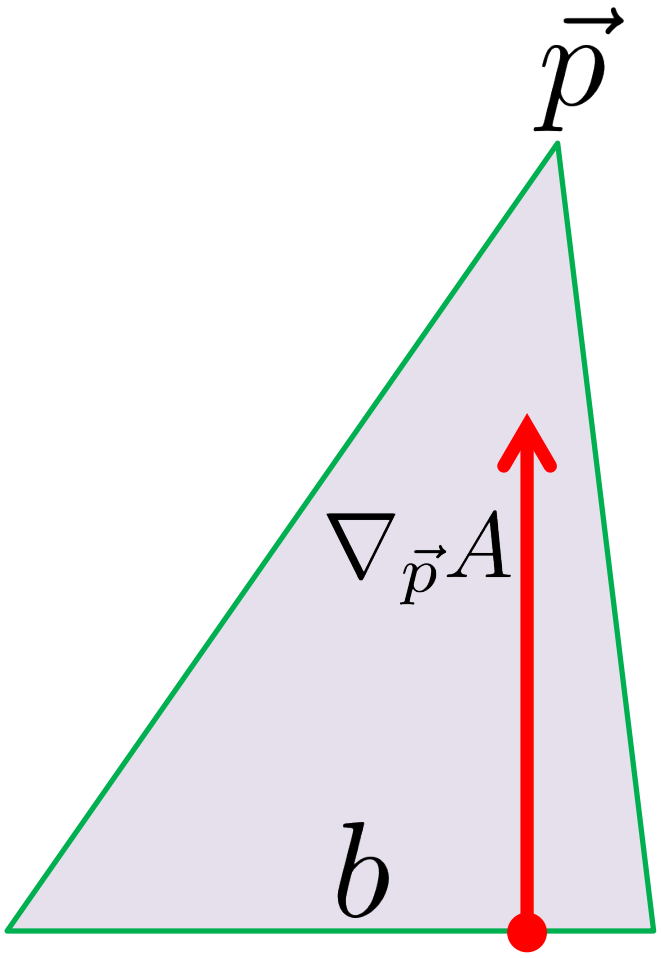
$$A = \frac{1}{2} b \sqrt{p_n^2 + p_\perp^2}$$

$$\frac{\partial A}{\partial p_e} = 0$$

$$\frac{\partial A}{\partial p_n} = \frac{\overset{0}{\cancel{b p_n}}}{2\sqrt{p_n^2 + p_\perp^2}} = 0 \Longrightarrow \nabla_{\vec{p}} A = \frac{1}{2} b \vec{e}_\perp$$

$$\frac{\partial A}{\partial p_\perp} = \frac{b p_\perp}{2\sqrt{p_n^2 + p_\perp^2}}$$

SINGLE TRIANGLE: COMPLETE

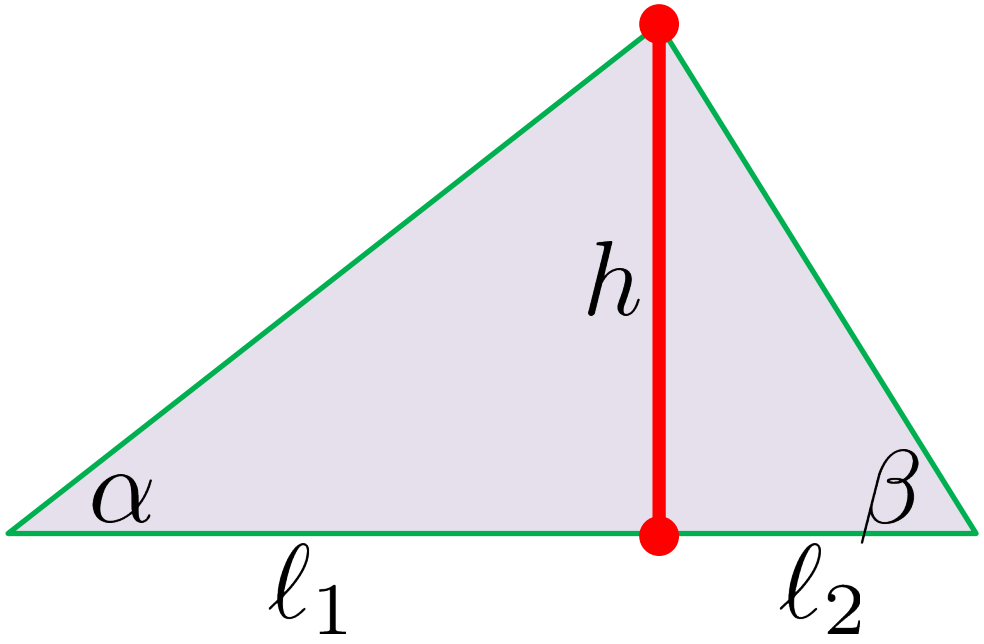


$$\vec{p} = p_n \vec{n} + p_e \vec{e} + p_\perp \vec{e}_\perp$$

$$A = \frac{1}{2} b \sqrt{p_n^2 + p_\perp^2}$$

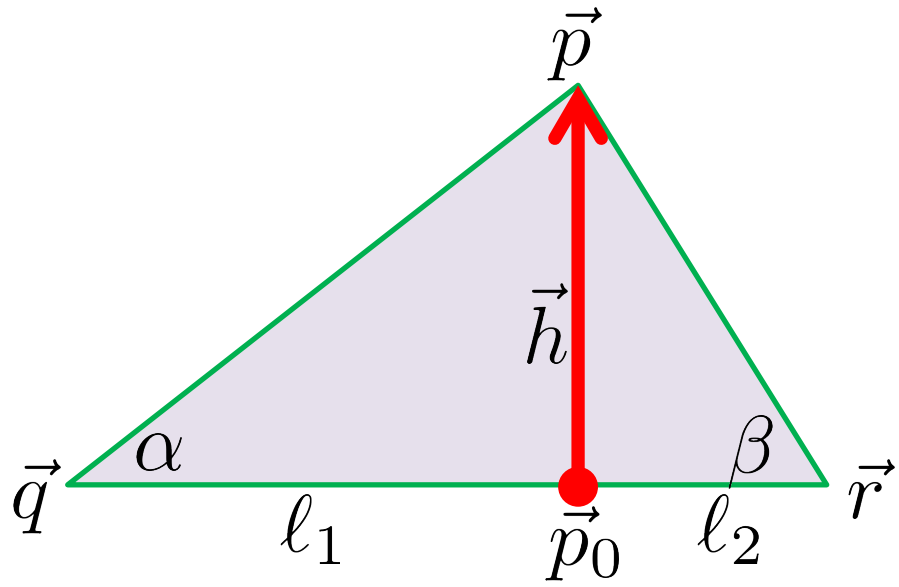
$$\nabla_{\vec{p}} A = \frac{1}{2} b \vec{e}_\perp$$

RATIO OF BASE TO HEIGHT



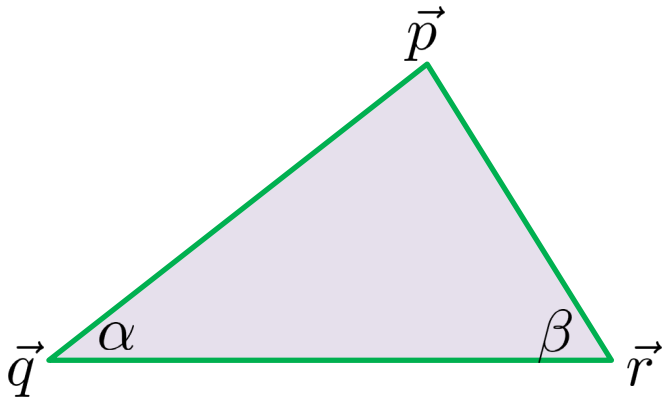
$$\frac{b}{h} = \frac{\ell_1 + \ell_2}{h} = \frac{\ell_1}{h} + \frac{\ell_2}{h} = \cot\alpha + \cot\beta$$

HEIGHT VECTOR



$$\vec{h} = \vec{p} - \vec{p}_0 = \vec{p} - (\vec{r} - \vec{q}) \frac{\tan\alpha}{\tan\alpha + \tan\beta}$$

ALTERNATIVE GRADIENT FORMULA



$$\nabla_{\vec{p}} A = \frac{1}{2} b \vec{e}_{\perp}$$

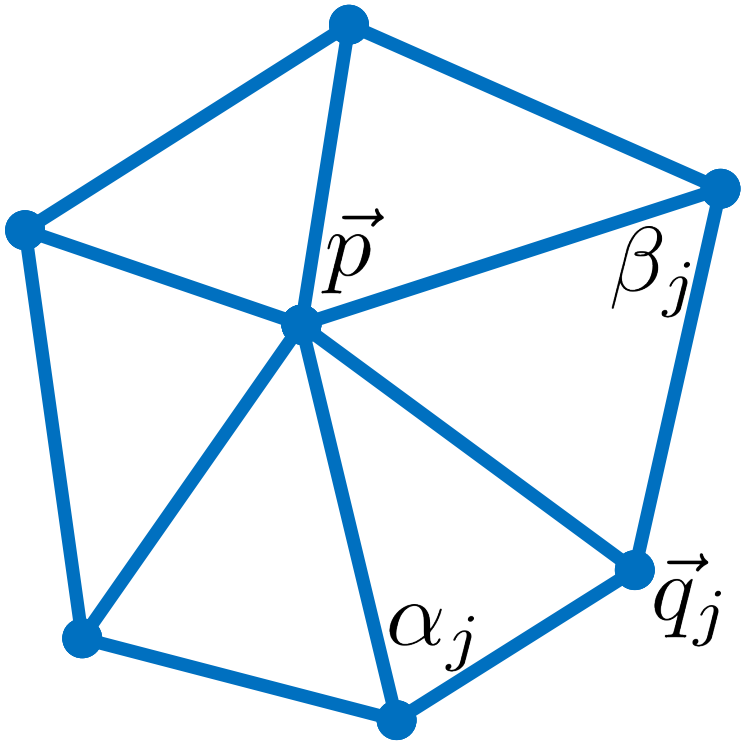
$$= \frac{1}{2} \frac{b}{\|\vec{h}\|} \vec{h}$$

$$= \frac{1}{2} (\cot\alpha + \cot\beta) \left[\vec{p} - (\vec{r} - \vec{q}) \frac{\tan\alpha}{\tan\alpha + \tan\beta} \right]$$

$$= \frac{1}{2} ((\vec{p} - \vec{r}) \cot\alpha + (\vec{p} - \vec{q}) \cot\beta)$$

SUMMING AROUND A VERTEX

$$\nabla_{\vec{p}} A = \frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(\vec{p} - \vec{q}_j)$$



$$\nabla_{\vec{p}} A = \frac{1}{2}((\vec{p} - \vec{r}) \cot \alpha + (\vec{p} - \vec{q}) \cot \beta)$$

Vanishes as you refine the mesh

INTEGRATED MEAN CURVATURE NORMAL

DEFINITION:

The mean curvature normal integrated over region V is given by

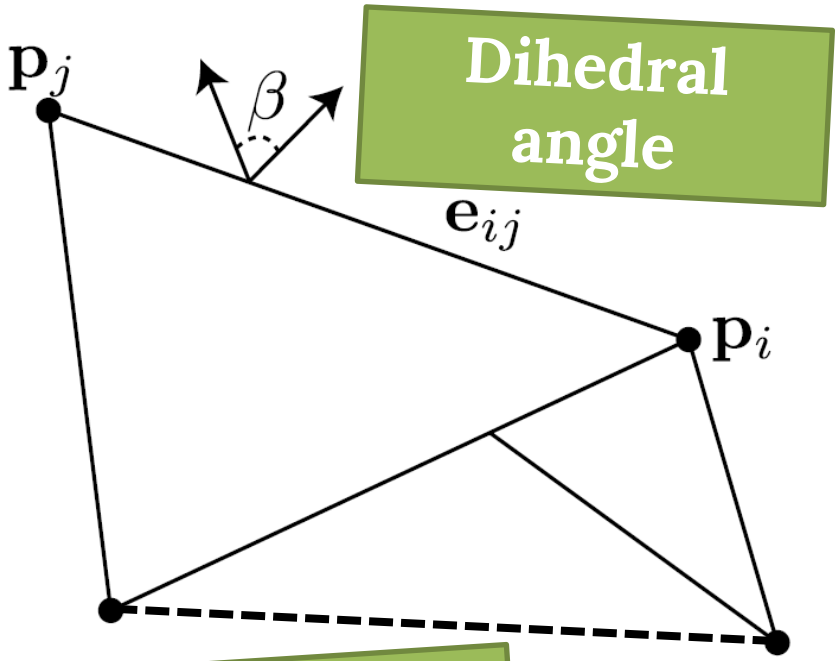
$$\nabla_{\vec{p}} A = \frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(\vec{p} - \vec{q}_j)$$

Divide by area for curvature estimate

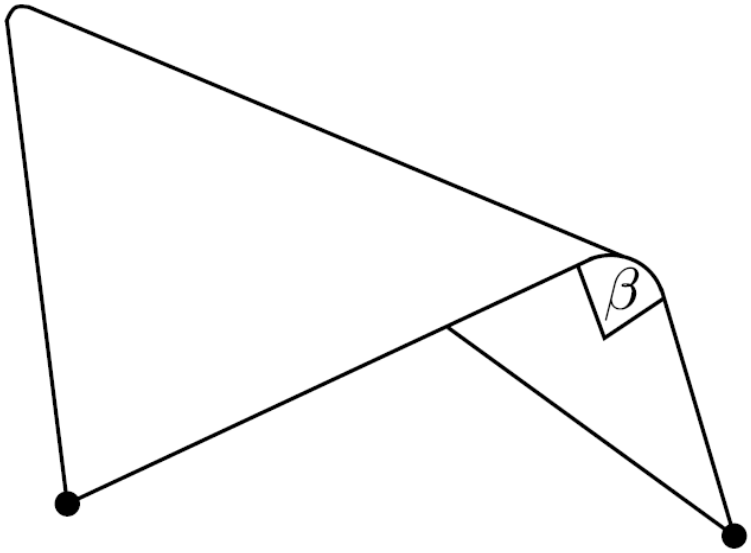
PIPELINE

- Compute integrated H, K
- Divide by area of cell for estimated value

ANOTHER MEAN CURVATURE



Dihedral angle



Same vertices, lower curvature

$$\int_B H = \frac{1}{2}\beta\|\mathbf{e}\|$$

J.A. Bærentzen et al., *Guide to Computational Geometry Processing* (2012)

Used for triangulation applications

TUNED FOR VARIATIONAL APPLICATIONS

Computing discrete shape operators on general meshes

Eitan Grinspun
Columbia University
eitan@cs.columbia.edu

Yotam Gingold
New York University
gingold@mrl.nyu.edu

Jason Reisman
New York University
jasonr@mrl.nyu.edu

Denis Zorin
New York University
dzorin@mrl.nyu.edu

Abstract

Discrete curvature and shape operators, which c... are essential in a variety of applications: simulat... geometric data processing. In many of these appl... approaches for formulating curvature operators ... expensive methods used in engineering applicati... computer graphics.

We propose a simple and efficient formulation for ... degrees of freedom associated with normals. On... curvature operators commonly used in graphics; ... and produces consistent results for different types ...

Cotan

Theirs

TUNED FOR ROBUSTNESS

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

Robust statistical estimation of curvature on discretized surfaces

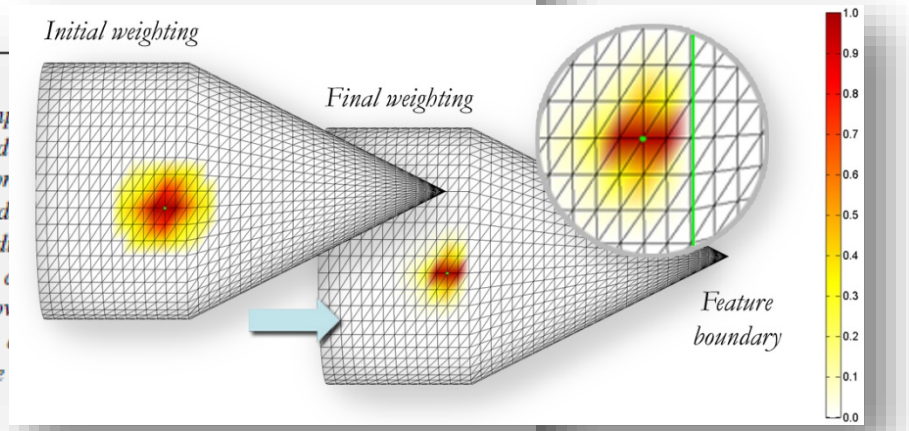
Evangelos Kalogerakis, Patricio Simari, Derek Nowrouzezahrai and Karan Singh

Dynamic Graphics Project, Computer Science Department, University of Toronto

Abstract

A robust statistics approach to curvature estimation on discretely samp
point clouds, is presented. The method exhibits accuracy, stability and
sampled surfaces with irregular configurations. Within an M-estimatio
noise and structured outliers by sampling normal variations in an ad
each point. The algorithm can be used to reliably derive higher order d
surface normals while preserving the fine features of the normal and
with state-of-the-art curvature estimation methods and shown to impro
across ground truth test surfaces under varying tessellation densities
noise. Finally, the benefits of a robust statistical estimation of curvature
applications of mesh segmentation and suggestive contour rendering.

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



ALTERNATIVE STRATEGIES

- Locally fit a smooth surface
  What type of surface? How to fit?

- Different formula
  Function of curvature? Where on mesh?
  Convergence of approximation?

- Learn curvature computation
  Tune for application? Training data?

PRACTICAL ADVICE

Try as many as you can.

Most are easy to implement!