Universität Konstanz

Illustrative Computer Graphics

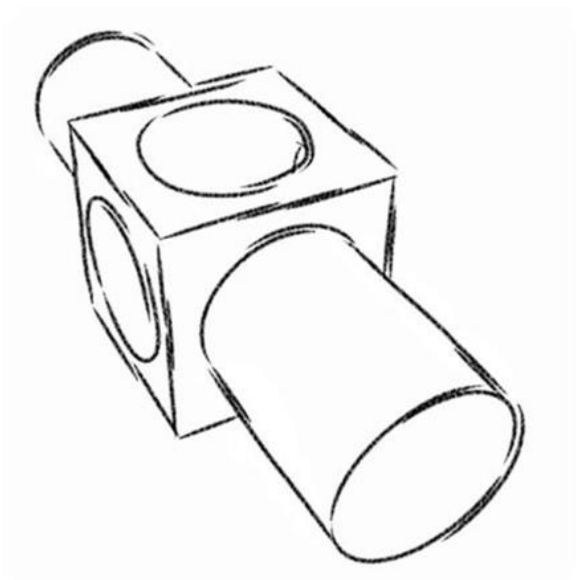
Week 9

3D Lines II – Mathematics of Line

Lecturers:
Oliver Deussen
KC Kwan

Important Lines

- Silhouette lines
- Near silhouette lines (suggestive contours)
- Geometry-based lines
- View-dependent lines

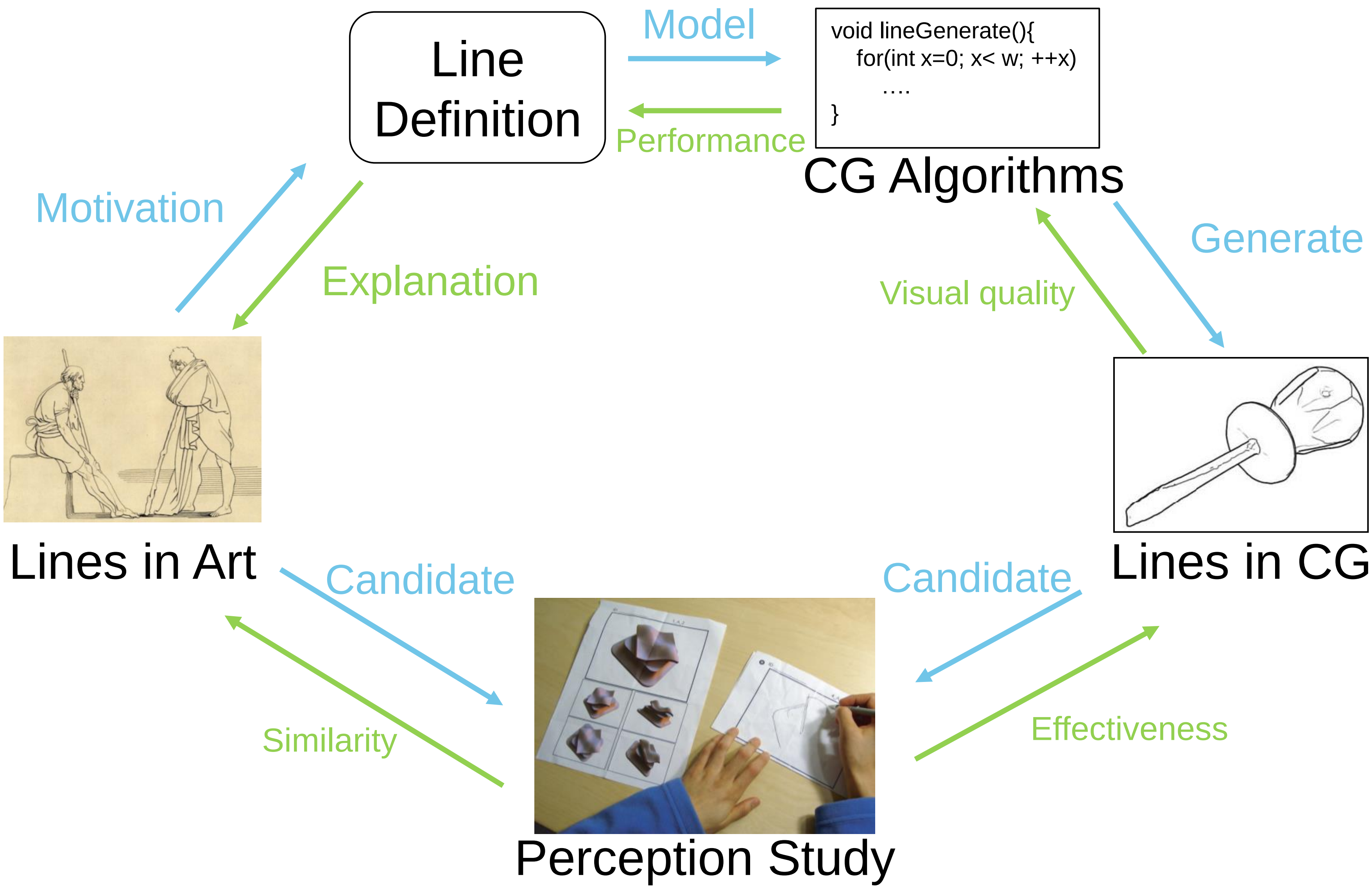


Chapter 9.1 – Line Definitions

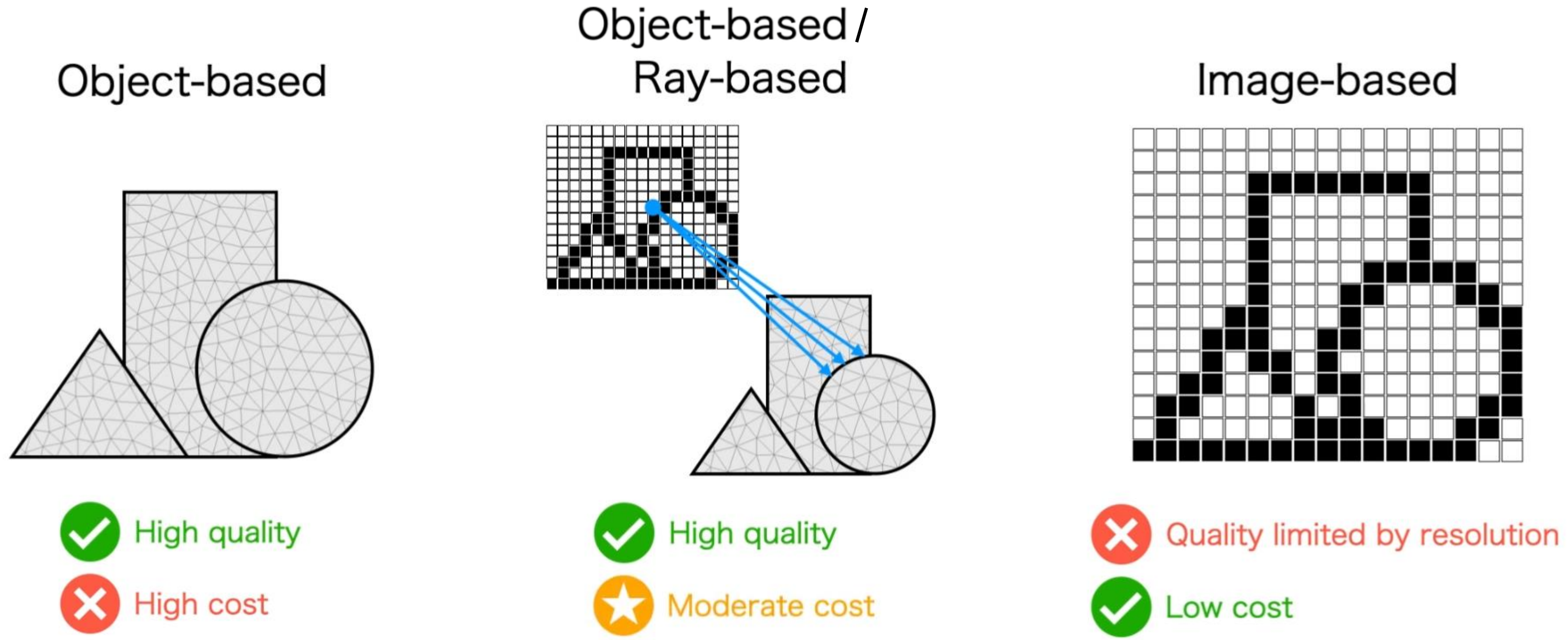
Based on SIGGRAPH 2008 Class:

Line Drawings from 3D Model

Overview


Line Definition
Model
Performance
void lineGenerate(){
for(int x=0; x< w; ++x)
....
}
CG Algorithms
Motivation
Explanation
Generate
Visual quality
Lines in Art
Lines in CG
Candidate
Candidate
Similarity
Effectiveness
Perception Study

Three Types of Definitions



View-Independent	View-dependent

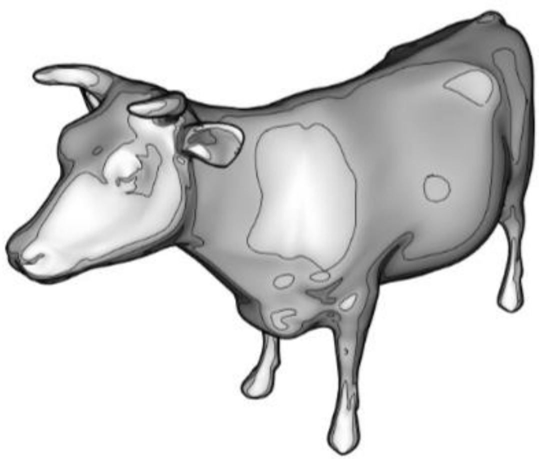
https://www.youtube.com/watch?v=2q6F6gF1XYo

Image-based Lines

Based on the color / intensity of the rendered scene

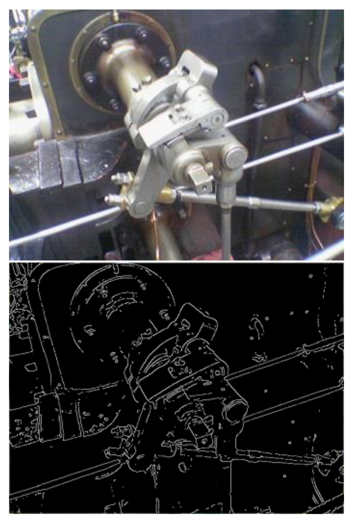
- Pros: Intuitive; good for GPU
- Cons: Difficult to stylize; resolution dependences

Isophote



http://users.csc.calpoly.edu/~zwood/teaching/csc471/finalW16_1/lchang07/index.html

Edge



By Simpsons contributor @ Wikipedia

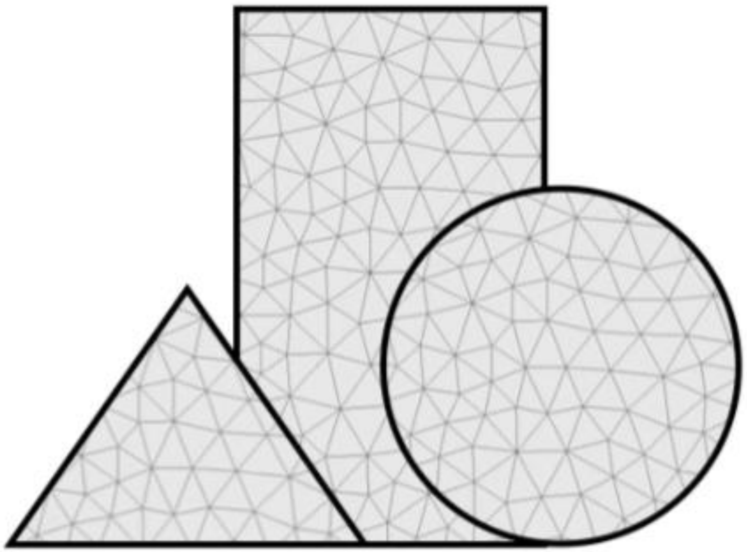
Ridges &Valley

Apparent Ridges for Line Drawing

View-Independent Object-Space Lines

Based on the geometry of the objects

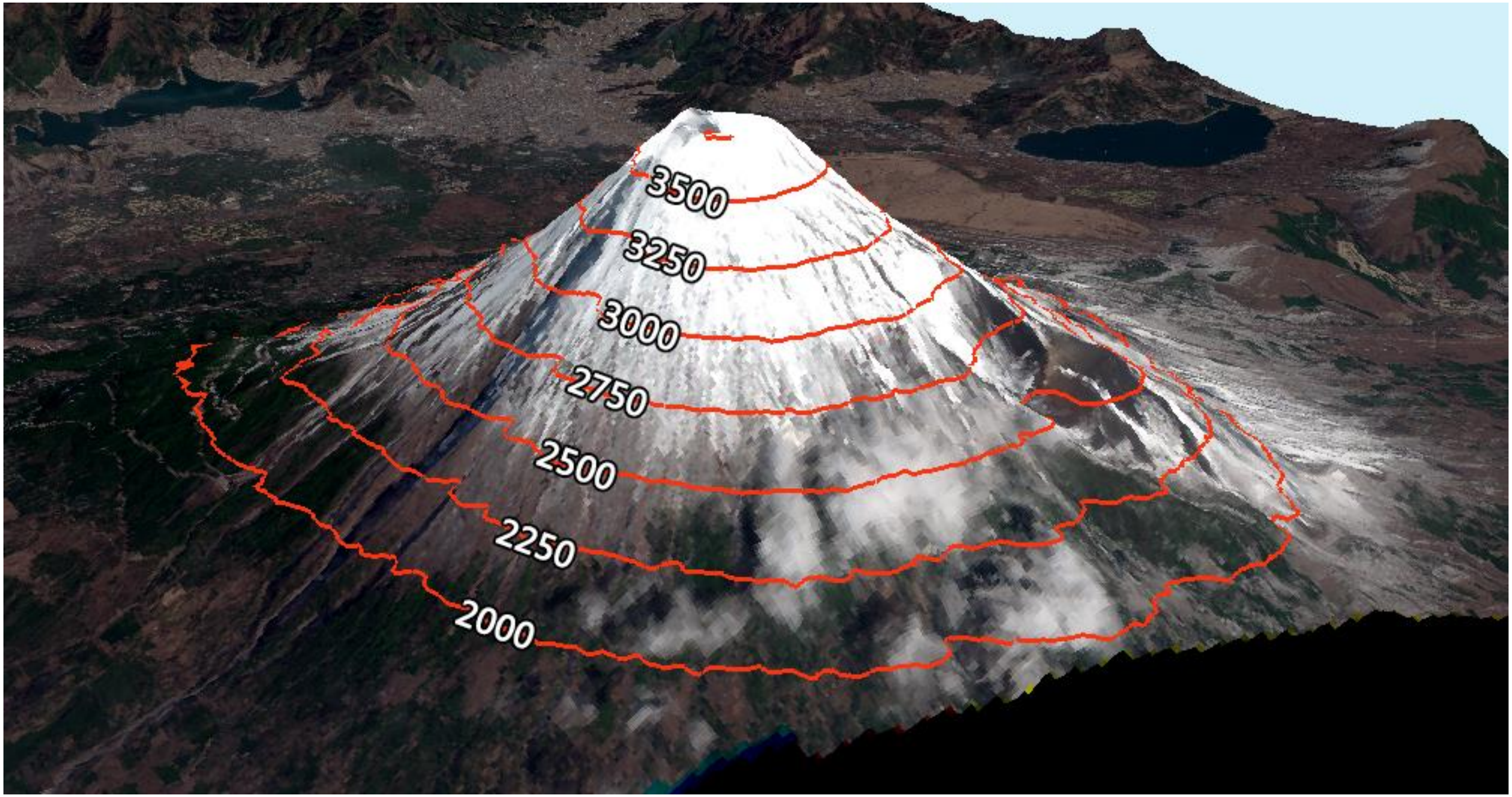
- Pros: Intrinsic properties of shape; can be precomputed
- Cons: Complex; can be misinterpreted as markings



https://www.youtube.com/watch?v=2q6F6gF1XYo

View-Independent Object-Space Lines

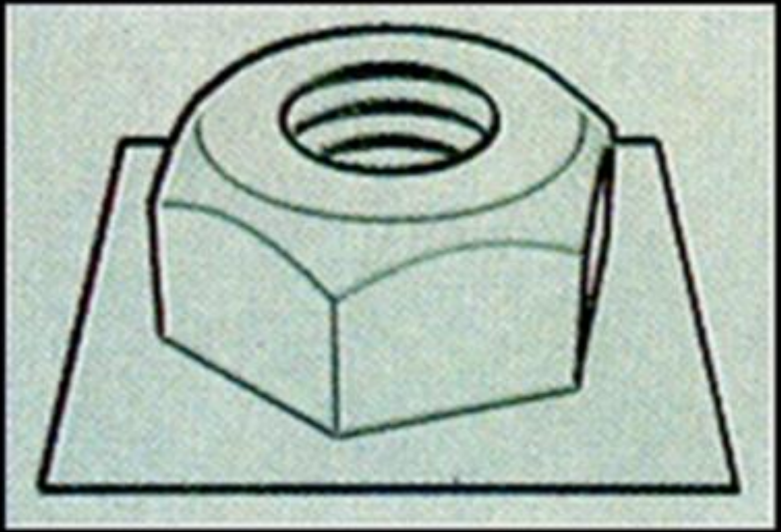
Topo lines:



https://gisgeography.com/contour-lines-topographic-map/

View-Independent Object-Space Lines

Creases: infinitely sharp folds

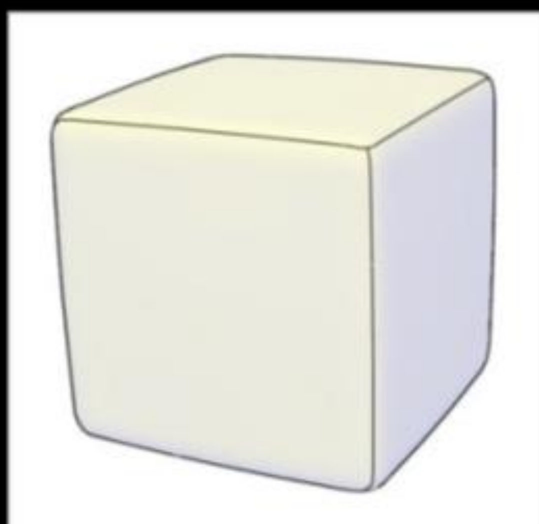


[Saito & Takahashi 90]

View-Independent Object-Space Lines

Ridges and valleys (crest lines)

- Local maxima of curvature
- Sometimes effective, sometimes not

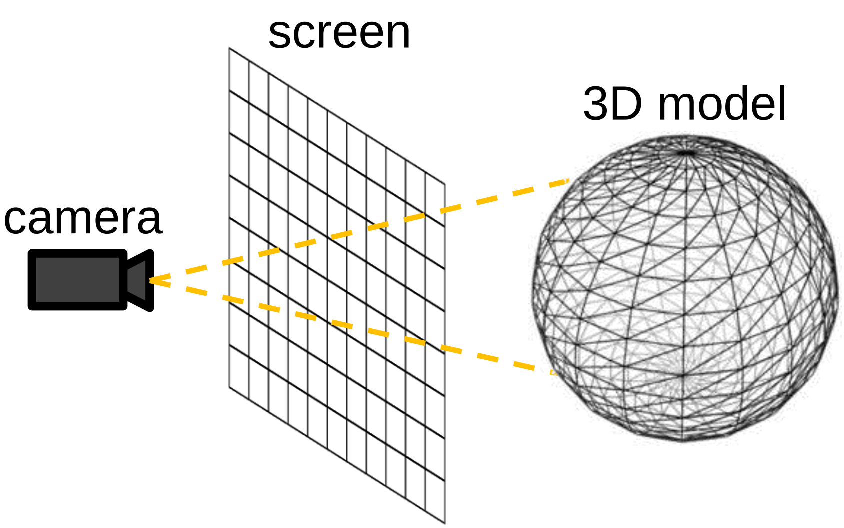


[Thirion 92, Interrante 95, Stylianou 00, Pauly 03, Ohtake 04 …]

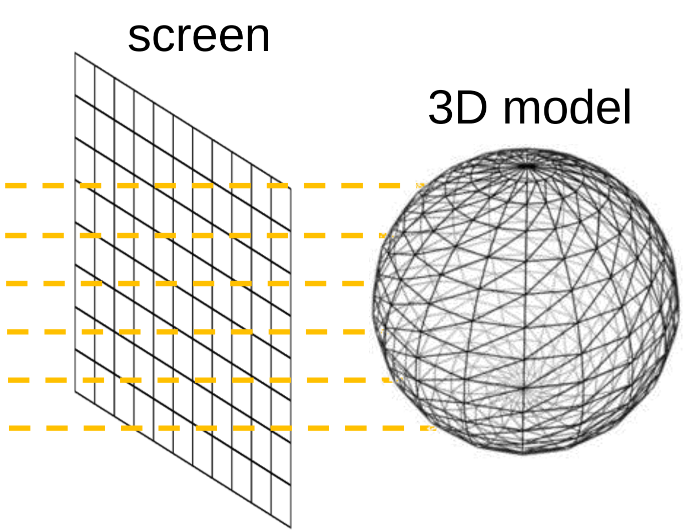
Ray-based Lines

Based on the geometry from current view

- Pros: Seem to be perceived as conveying shape
- Cons: recompute per frame



Perspective

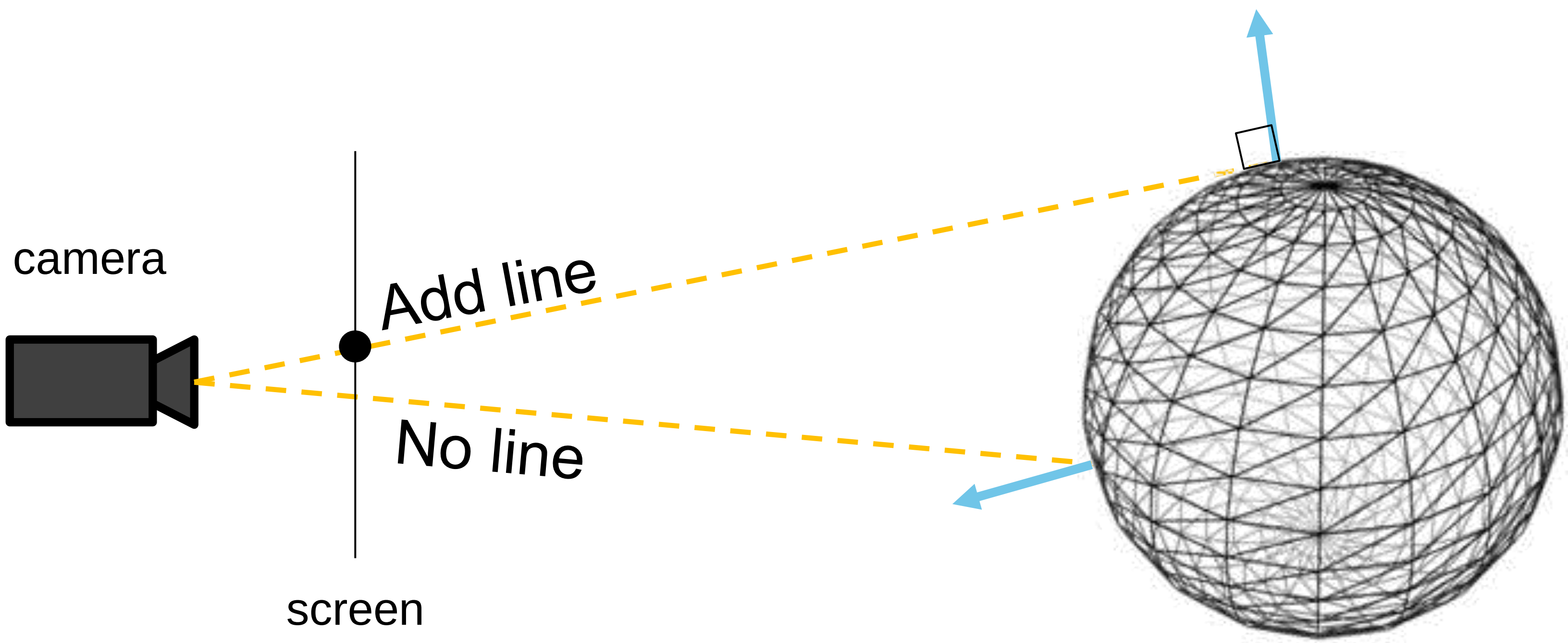


Orthographic

Ray-based Lines

Perpendicular normal

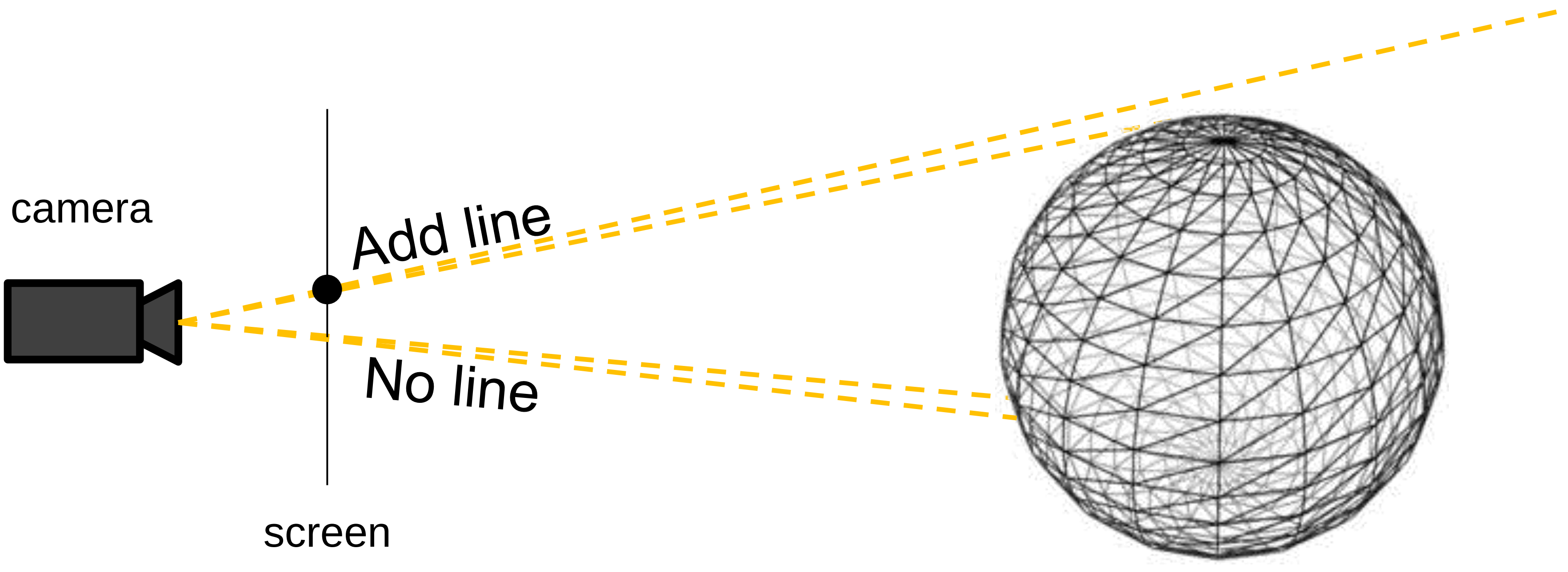
- Add line if the normal is perpendicular to the view
  - Dot product = 0



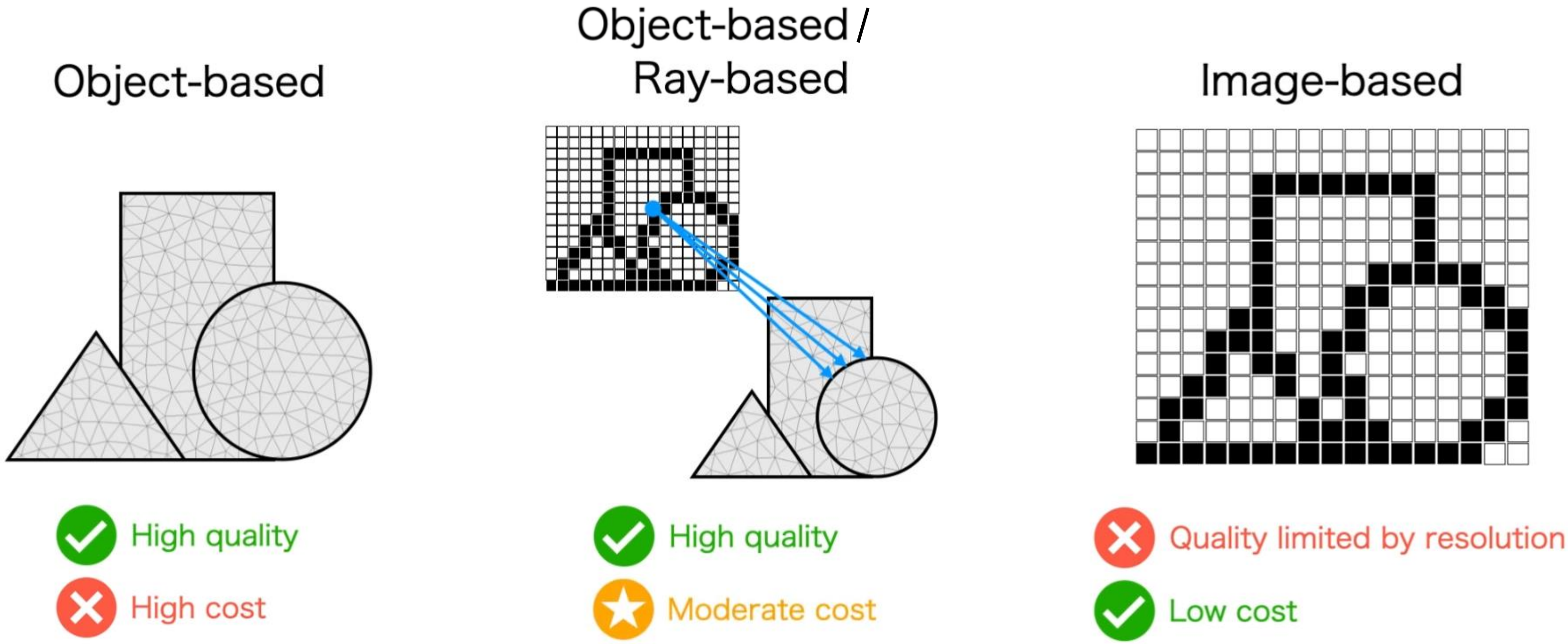
Ray-based Lines

Depth discontinuous

- Add line if there is sudden change of depth



Three Types of Definitions



View-Independent

View-dependent

https://www.youtube.com/watch?v=2q6F6gF1XYo

Chapter 9.2 – More Maths

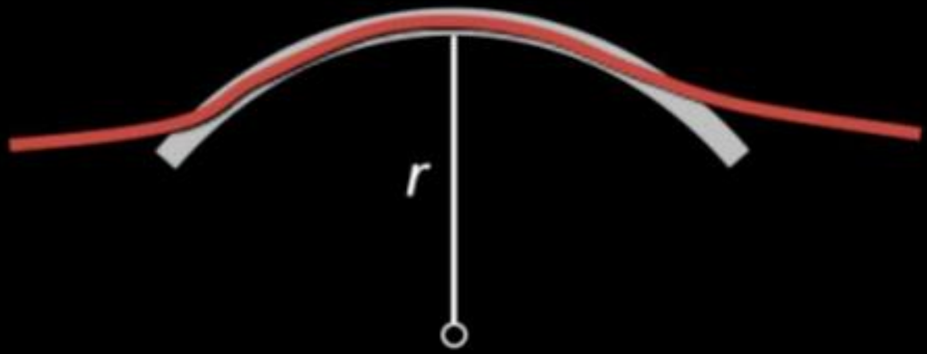
Differential Geometry

- Many shape-conveying lines based on surface differentials
- First-order: surface normals
  - Example: occluding contours occur where normal perpendicular to view direction
- Other lines need higher-order derivatives

Differential Geometry

Many lines based on curvatures

- Second-order differential properties of surface
- For a curve: reciprocal of radius of circle that best approximates it locally



$$\kappa = 1/r$$

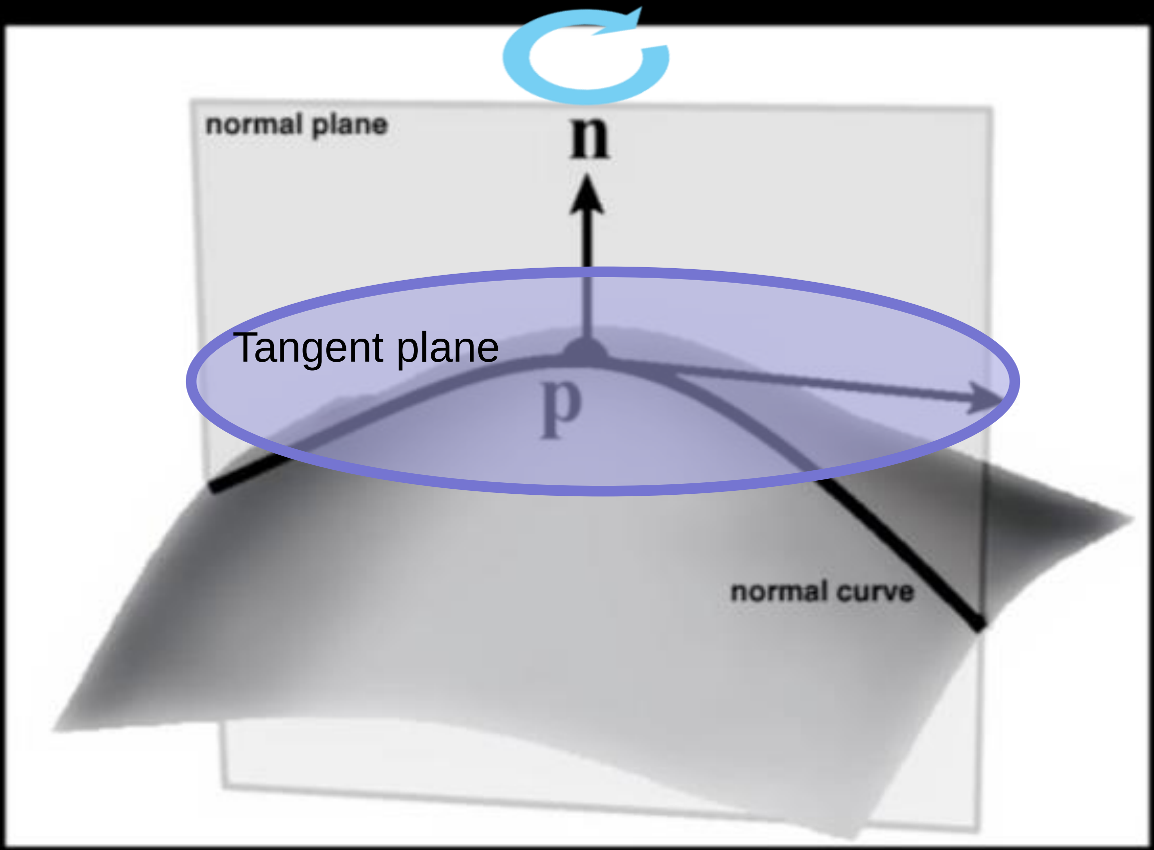
- For a surface: ?

Surface Curvature

- Normal Curvature K_n
- Radial Curvature K_r
- Principle Curvature K_1 K_2
- Gaussian Curvature K
- Mean Curvature H

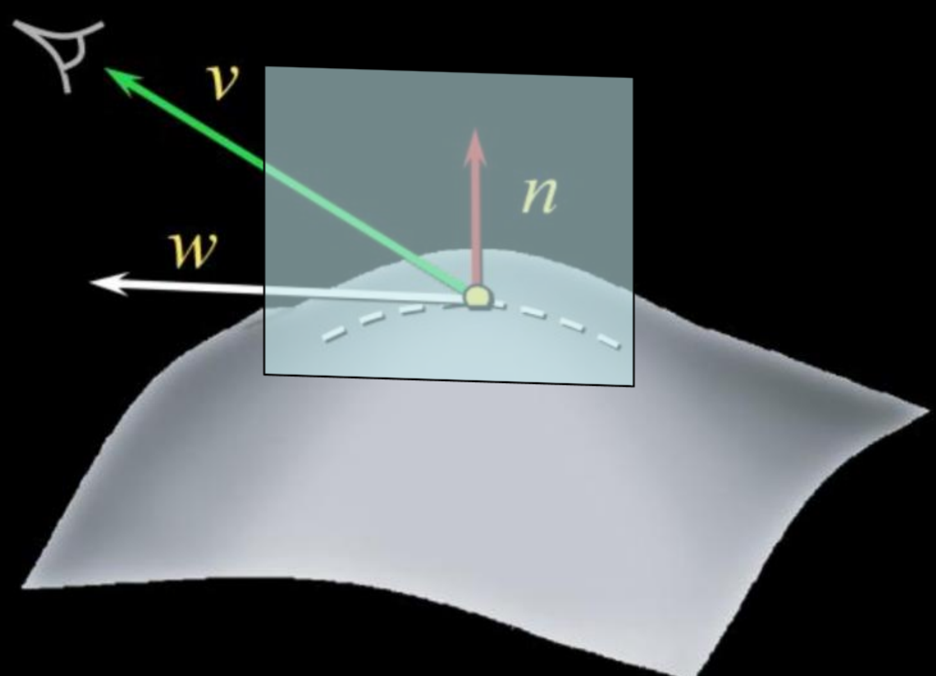
Normal Curvature

Curvature of a normal curve



Radial Curvature κ_r

Curvature in projected view direction, w:

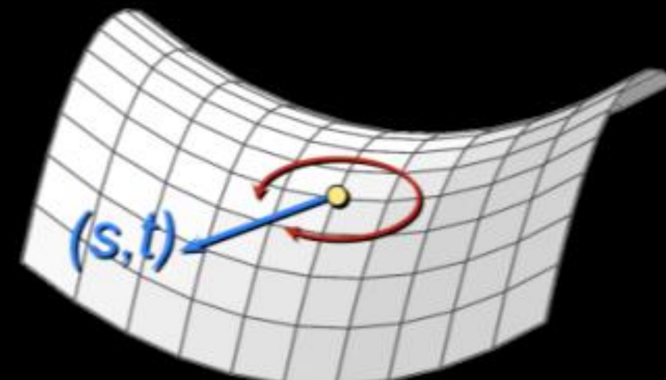


Curvature on a Surface

Normal curvature varies with direction, but for a smooth surface satisfies

$$\kappa_n = \begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix}$$

$$= \begin{pmatrix} s & t \end{pmatrix} \mathbf{II} \begin{pmatrix} s \\ t \end{pmatrix}$$

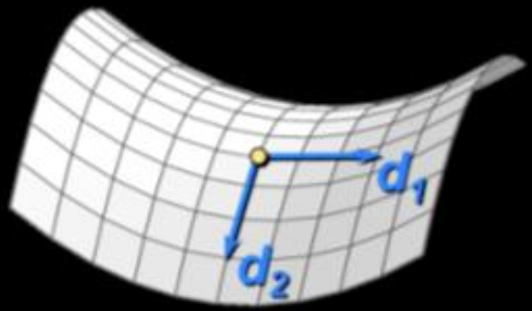


for a direction (s,t) in the tangent plane and a symmetric matrix $\mathbf{II}$

Principal Curvatures and Directions

Can always rotate coordinate system so that $\mathbf{II}$ is diagonal:

$$\mathbf{II} = \mathbf{R}^{\mathrm{T}} \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \mathbf{R}$$



κ_1 and κ_2 are *principal curvatures*, and are minimum and maximum of normal curvature

Associated directions are *principal directions*

Eigenvalues and eigenvectors of $\mathbf{II}$

Gaussian and Mean Curvature

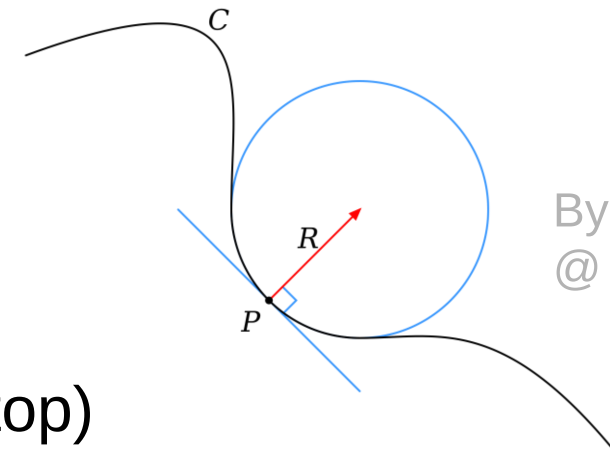
The Gaussian curvature $K = \kappa_1 \kappa_2$

The mean curvature $H = ½ (\kappa_1 + \kappa_2)$

Equal to the determinant and half the trace, respectively, of the curvature matrix

Enable qualitative classification of surfaces

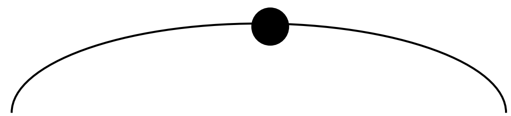
Assumption



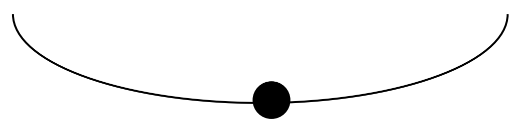
By Emperorhoney
@ Wikipedia

Curvature

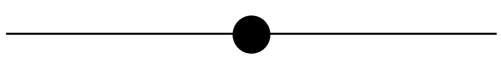
Positive → Convex (to the top)

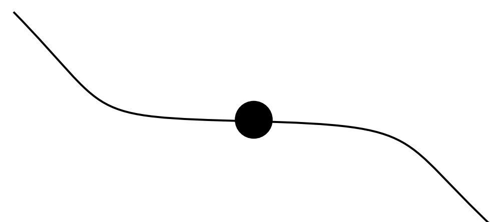


Negative → Concave (to the bottom)

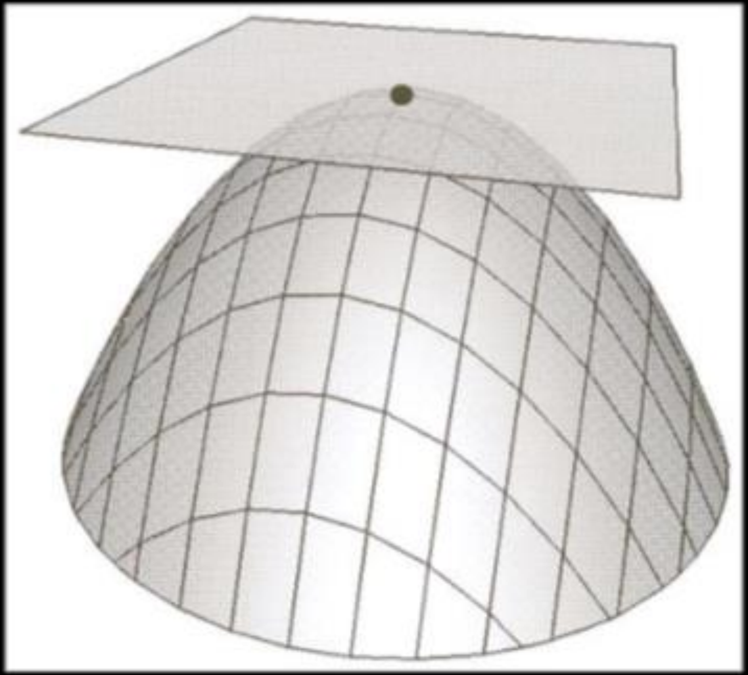


Zero → Flat





Positive Gaussian Curvature: Elliptic Points



Convex/concave depending on sign of H

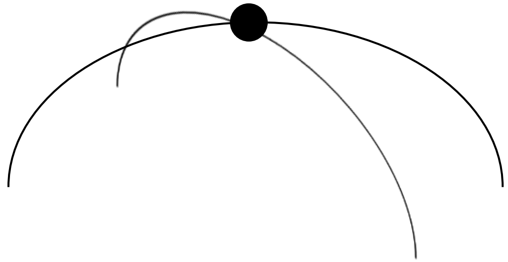
Tangent plane intersects surface at 1 point

Explanation

Gaussian curvature $K = K_1 \times K_2$

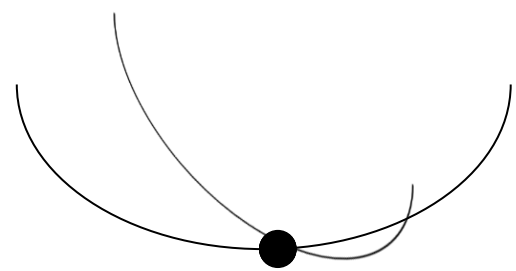
If K is positive:

K_1 and K_2 are both positive



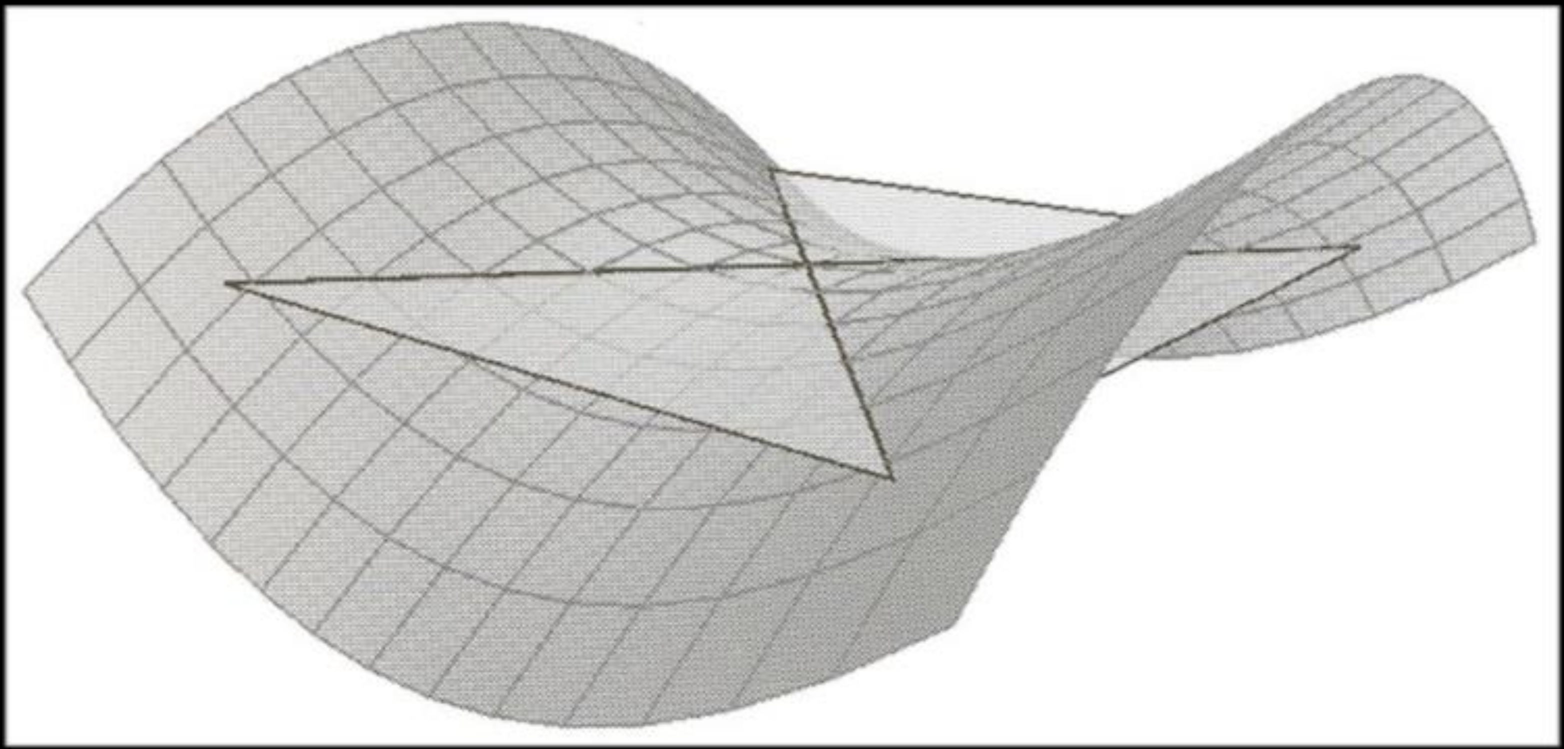
All are convex

K_1 and K_2 are both negative



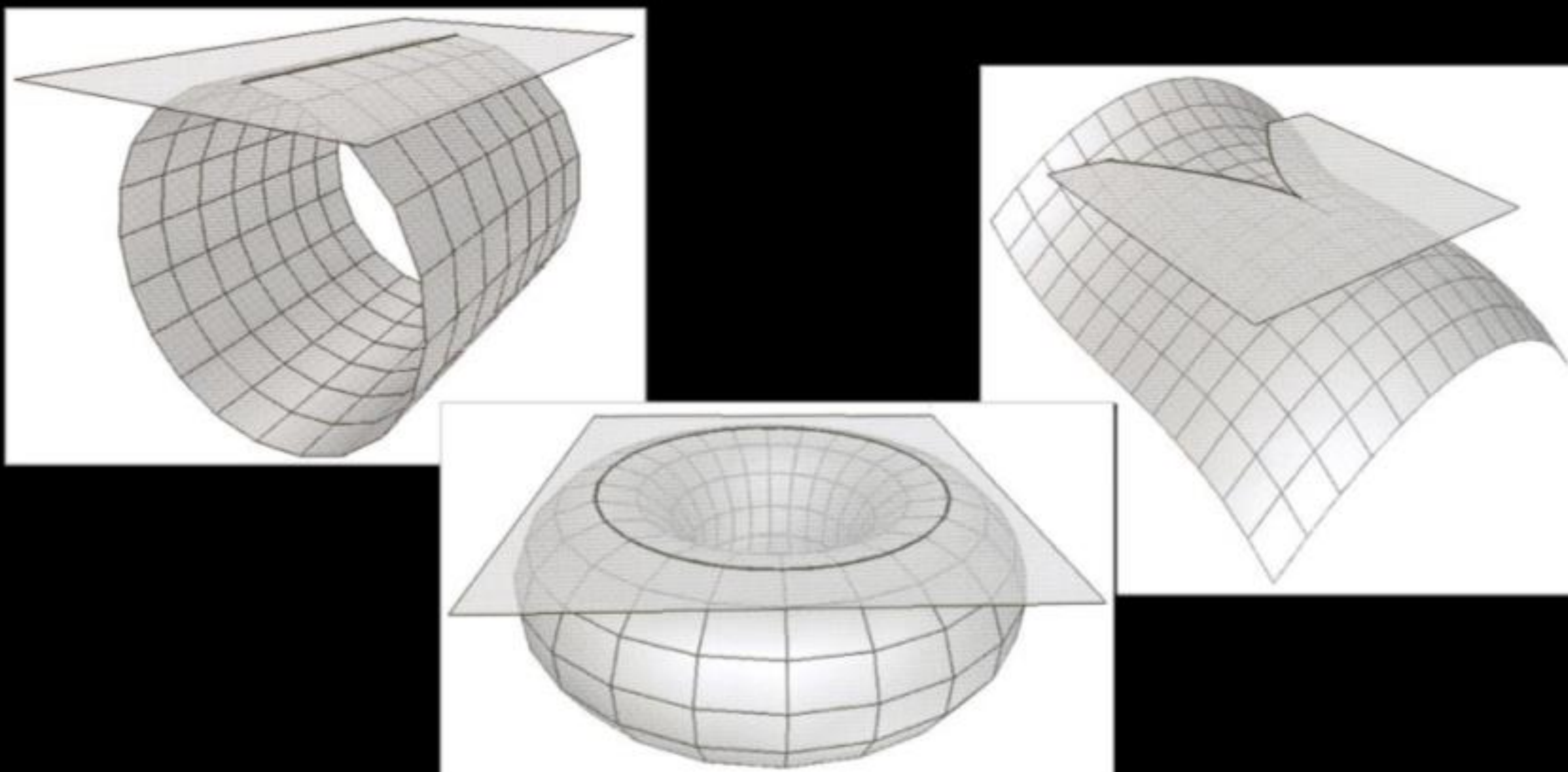
All are concave

Negative Gaussian Curvature: Hyperbolic Points



Tangent plane intersects surface along 2 curves

Zero Gaussian Curvature: Parabolic Points



Tangent plane intersects surface along 1 curve

Historical Note

Mathematician Felix Klein was convinced that parabolic lines held the secret to a shape's aesthetics, and had them drawn on the Apollo of Belvedere…

He soon abandoned the idea…

[Hilbert & Cohn-Vossen]

Zeros of κ_r, H, and K

$\kappa_r = 0$ $\qquad$ $H = 0$ $\qquad$ $K = 0$

Zeros of κ_r, H, and K (with derivative tests)

$\kappa_r = 0$
$D_w \kappa_r > 0$

$H = 0$
$D_w H > 0$

$K = 0$
$D_w K > 0$

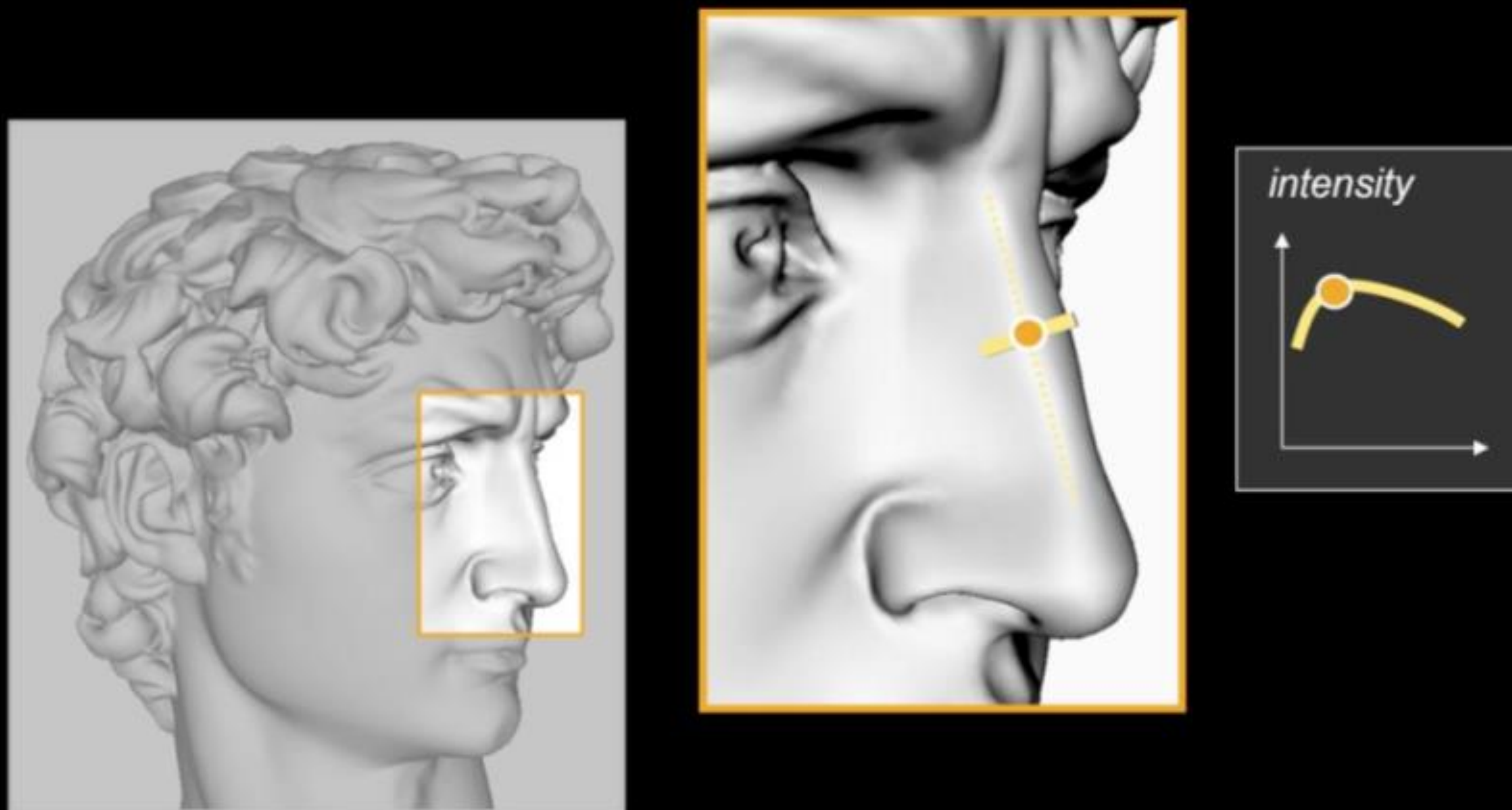
Line Drawings with Shading

from Frank Miller's *Sin City* (1991)

Line Drawings with Shading

from Frank Miller's *Sin City* (1991)

Intensity Ridges in Images



Intensity Ridges in Images

Assume:

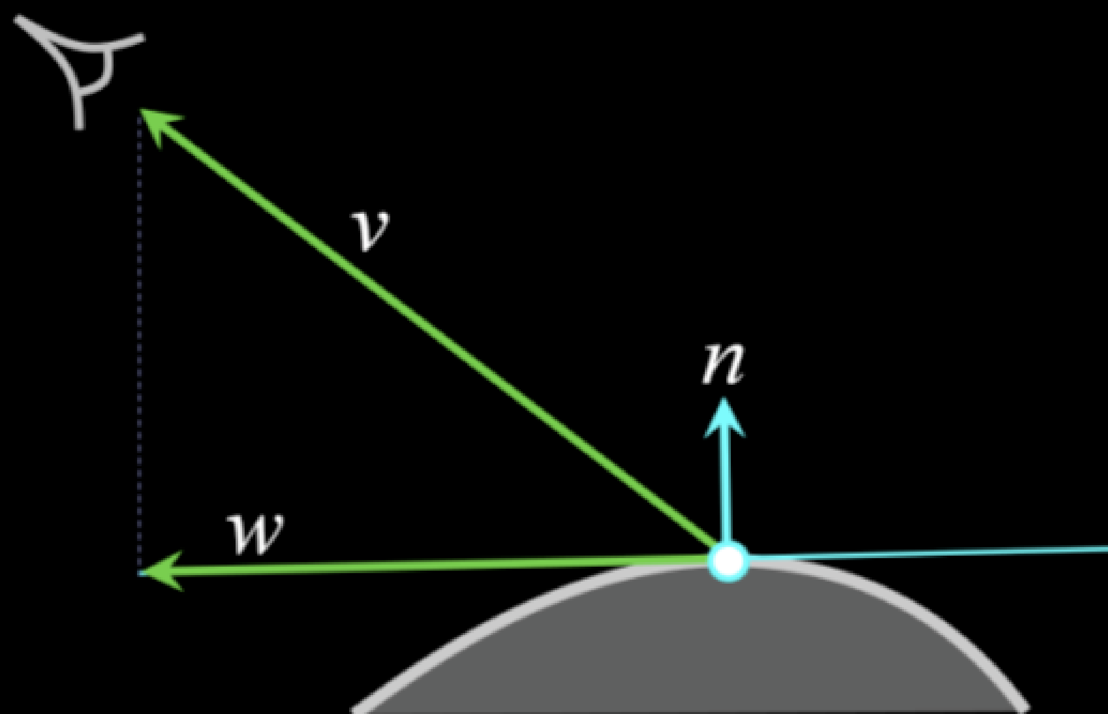
- Light at camera
- Lambertian material

What lines on the surface correspond to intensity ridges?

- Depends on how a ridge is defined (Saint-Venant, principal curvature extrema, …)
- Exact answer very messy

Surface Coordinates: w and $w_\perp$

w is the projected viewing direction



$w_\perp = n \times w$ (comes out of the screen)

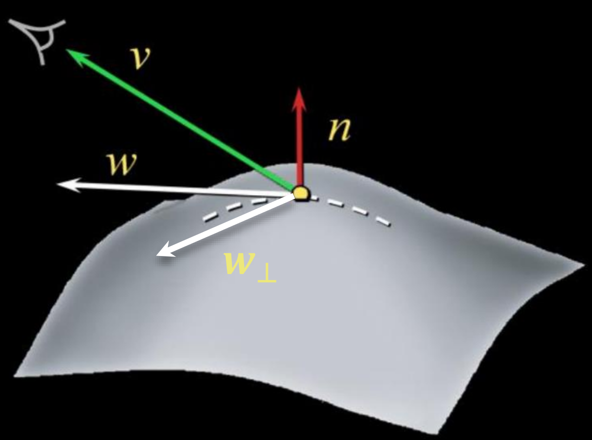
Highlight Lines

Suggestive highlights

– Maxima of $n \cdot v$ along w

Principal highlights

– Maxima of $n \cdot v$ along $w_\perp$

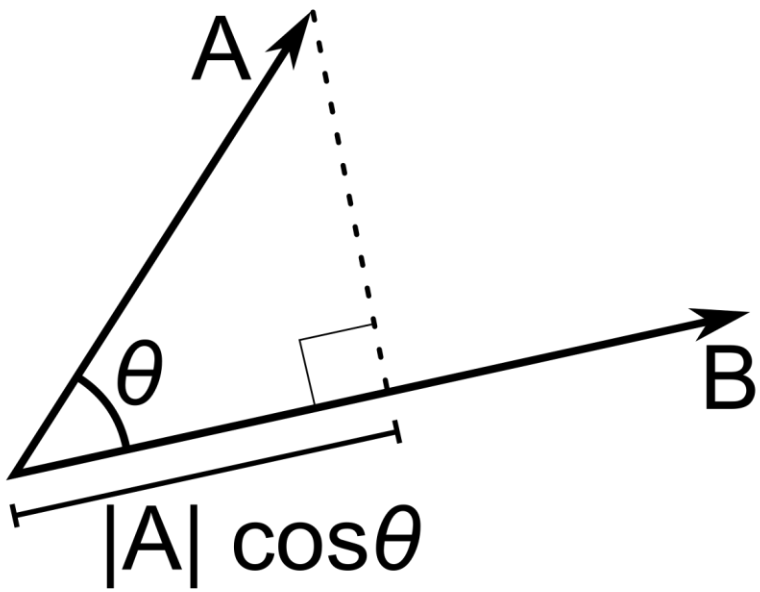


Lines are drawn in white

– In practice only draw strong maxima

Explanation

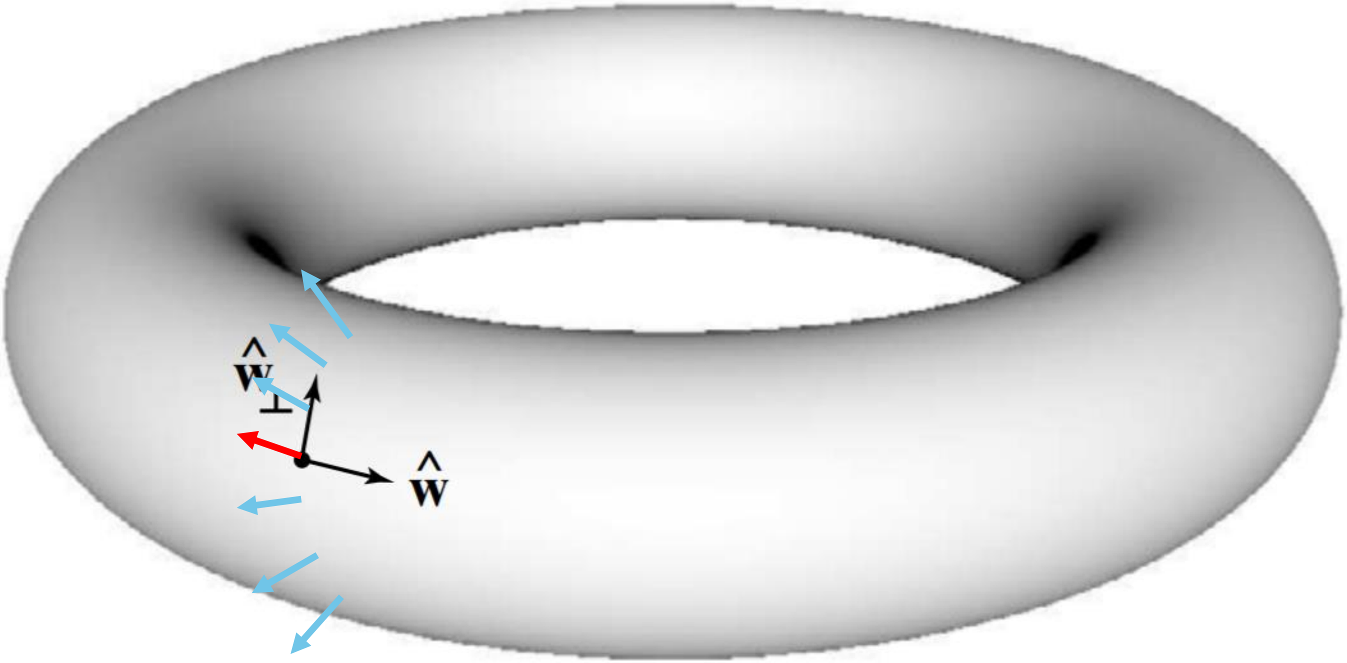
A dot B



It is maxima if the angle is small

Explanation

- Is this normal have most similar direction to the view vector compared to other normal along $w_{\perp}$

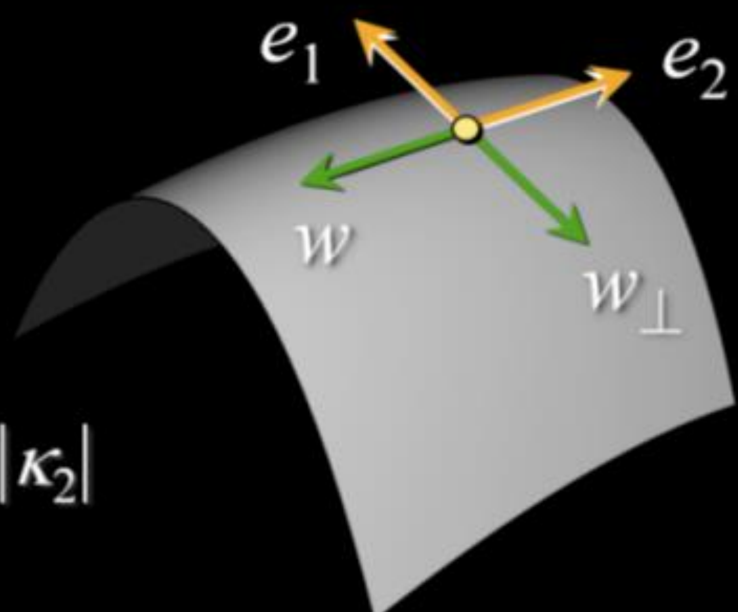


Principal Highlights

(Strong) maxima of $n \cdot v$ along $w_\perp$

$-D_{w_\perp} n \cdot v \sim \tau_r$ (radial torsion)

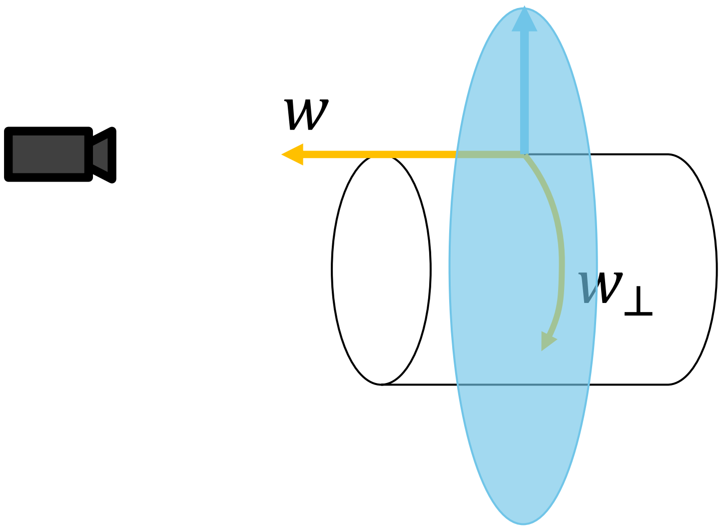
– Zeros of τ_r occur where w and $w_\perp$ are principal directions

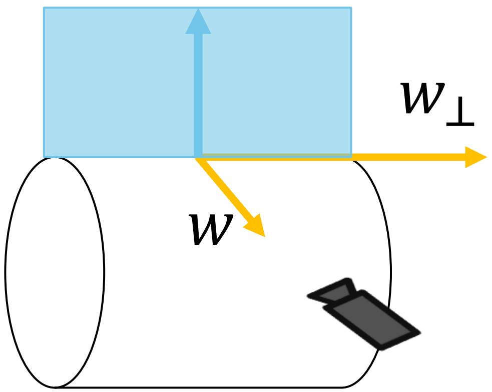


$|\kappa_1| > |\kappa_2|$

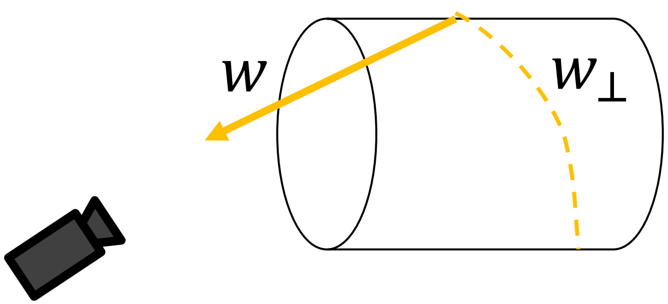
Explanation

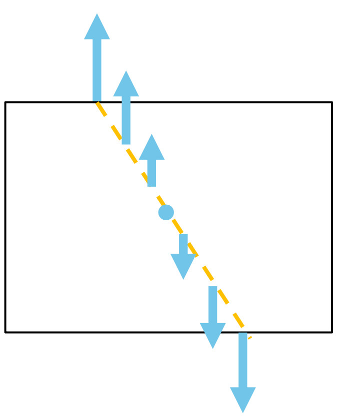
- Zeros of τ_r radial torsion





- Non-zeros of τ_r radial torsion



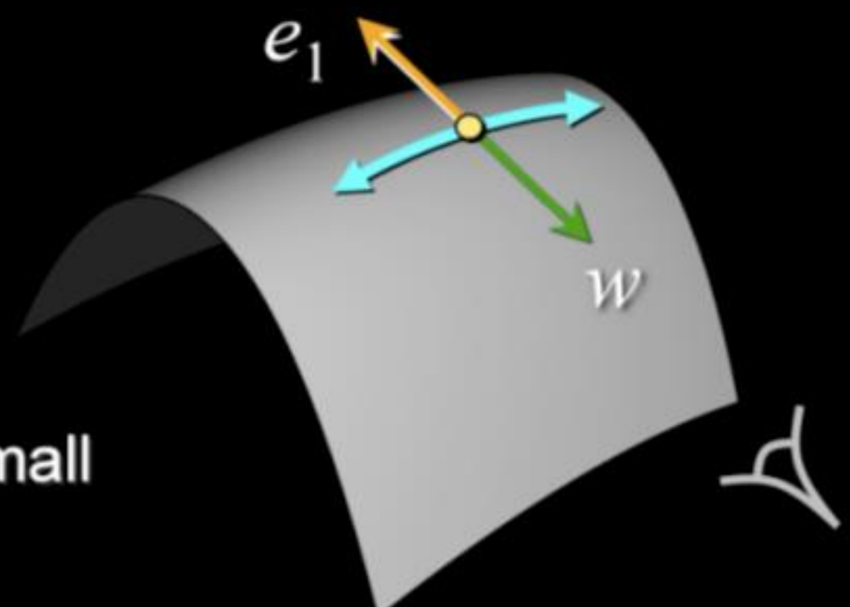


Principal Highlights

(Strong) maxima of $n \cdot v$ along $w_\perp$

$$-D_{w_\perp} n \cdot v \;\sim\; \tau_r$$ (radial torsion)

– Zeros of τ_r occur where w and $w_\perp$ are principal directions



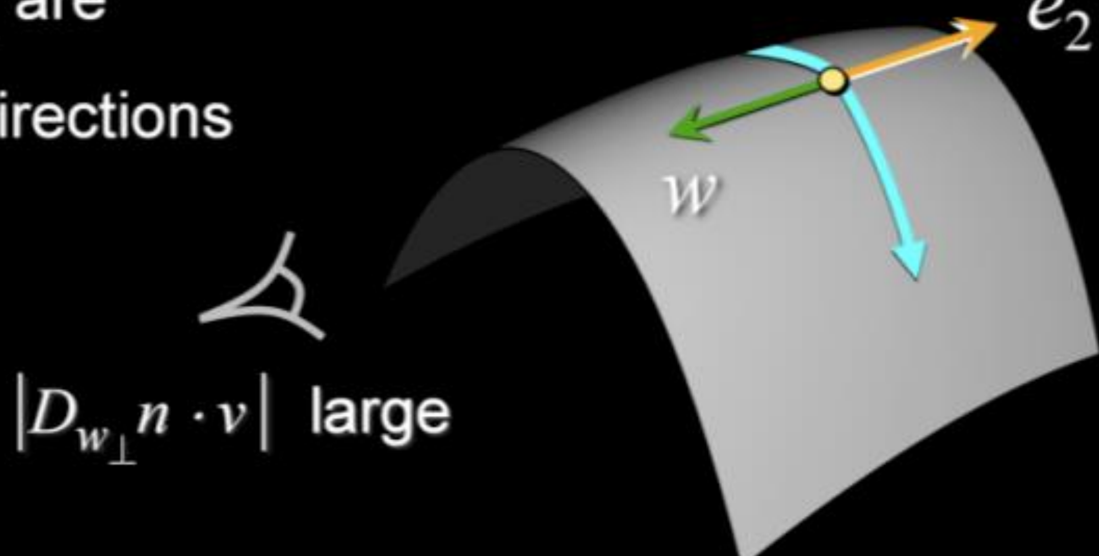
$\left|D_{w_\perp} n \cdot v\right|$ small

Principal Highlights

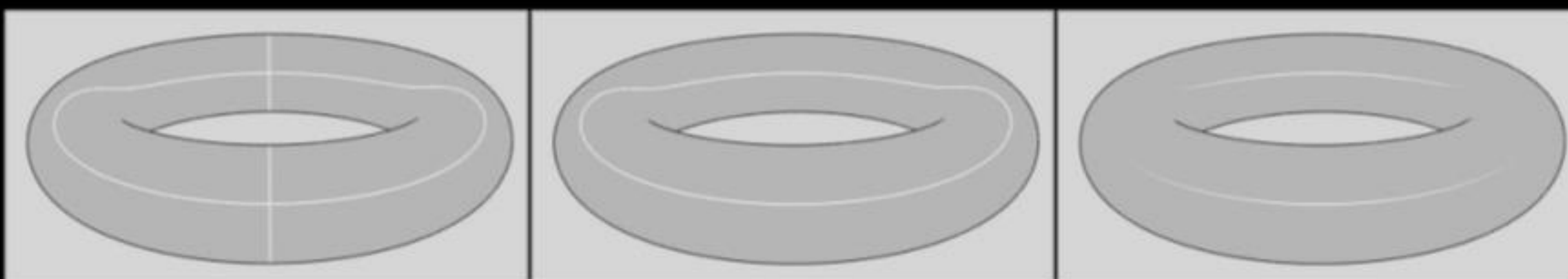
(Strong) maxima of $n \cdot v$ along $w_\perp$

$-D_{w_\perp} n \cdot v \;\sim\; \tau_r$ (radial torsion)

– Zeros of τ_r occur where w and $w_\perp$ are principal directions



Principal Highlights



$\tau_r = 0$

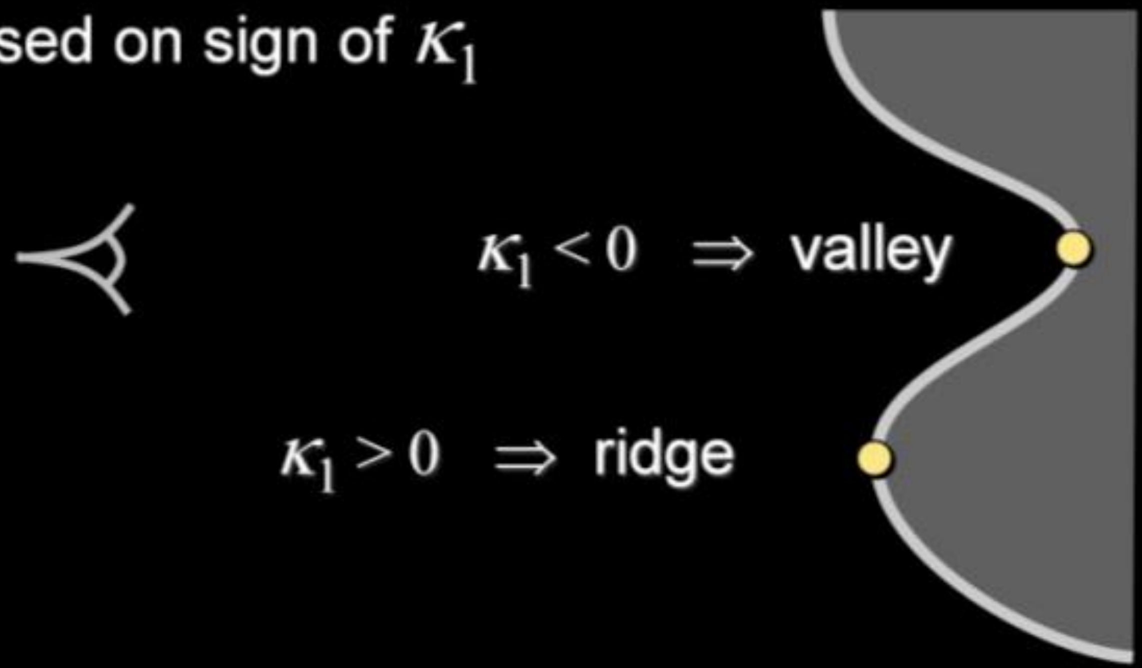
$w \cdot e_1 = 0$
(potentially strong extrema)

$w \cdot e_1 = 0$
and
$D_{w_\perp} \tau_r < -\varepsilon < 0$
(strong maxima)

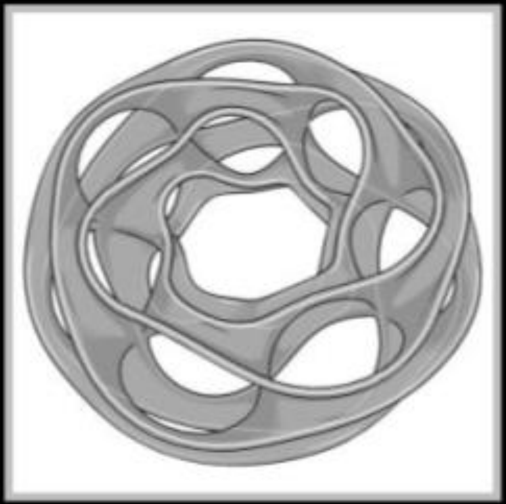
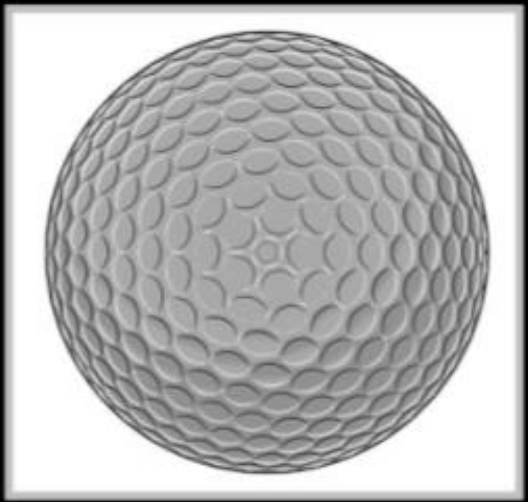
Principal Highlights

Points where $w \cdot e_1 = 0$ and $D_{w\perp} \tau_r < 0$

- equivalent to Saint-Venant creases in depth
- classify based on sign of κ_1



Results

Suggestive Contour (SC) + Principal Highlight (PH)

Important

- Examination
  - 10 Feb 2022 (Thu) whole day
  - 11 Feb 2022 (Fri) whole day

- You need to **register** examination on **Zeus**!

- A form to select your timeslot for oral examination
  - Please pack together :)
  - Only morning (currently)

- If you cannot sign the timeslot in lecture, please phone call our secretary: **Ingrid Baiker** **+49 07531 88-4233**

Sketch 10

Sketch 10

- Task:
  - Find pixels with normal perpendicular to view vector
  - Combine with depth buffer-based lines and toon shading
- Use the mouse position to modify the parameters that find more or less points (normal threshold)

Sketch 10 Hints

- Use epsilon = mouseX / width
  - Warning: mouseX is an Integer!
- Get the normal n from the RGB image
- The view Vector is (0, 0, 1)
- PVector.dot() gives you the dot product of two vectors, which is related to their angle
- Depending on the angle, color each pixel
- Make sure your window actually has focus
  - If you get no output, fix epsilon to 0.5 for debugging.

Sketch 10

Intended result

Move your mouse to change the result!