Surface Curvature Extraction
for 3D Image Analysis or Surface Rendering

John Rugis & Reinhard Klette

Department of Computer Science
The University of Auckland
Auckland, New Zealand

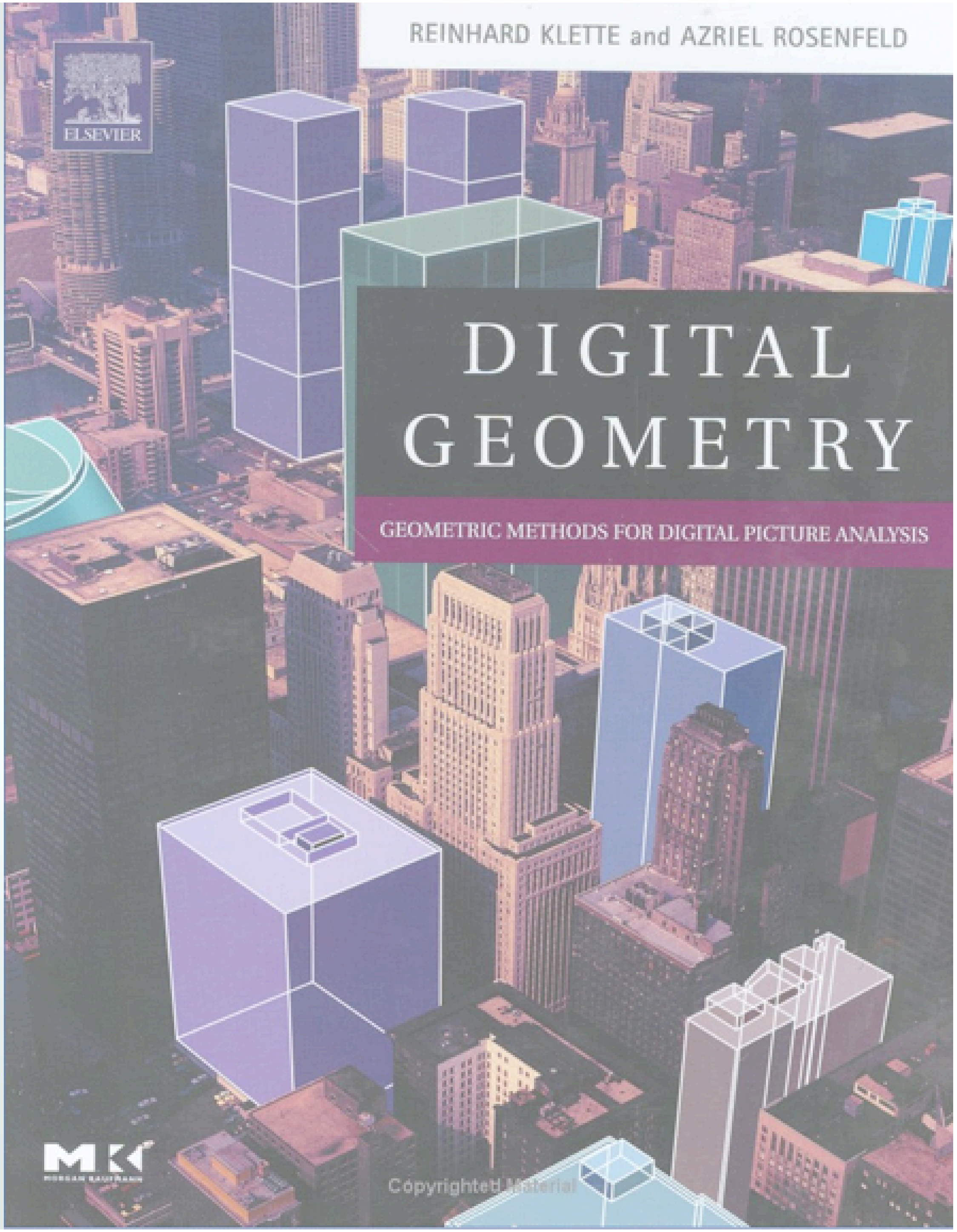
Outline

- Curvature Estimators for Digital Curves
- Curvature Estimators for Digital Surfaces
- Scale Invariance
- Curvature Maps
- Application (Digital Michelangelo Project)
- Projecting Surface Curvature Maps
- Conclusions

Curvature Estimators (for Digital Curves)

joint book with
Azriel Rosenfeld
… 2004

…
joint work with
Simon Hermann
2002-2006



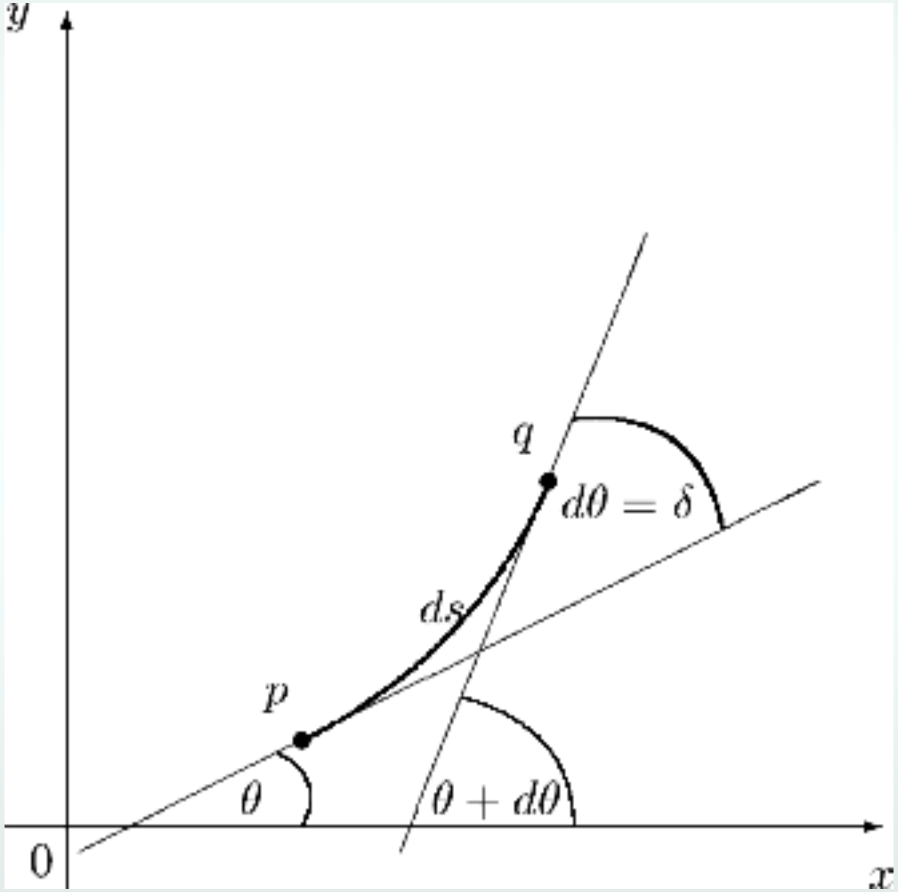
QUESTIONS raised in this book

- Do high resolution images support curvature estimation that is based on definitions in differential geometry? (majority of corner detectors is based on heuristics)

- Do these estimates converge towards the true curvature value assuming an increase in grid resolution?

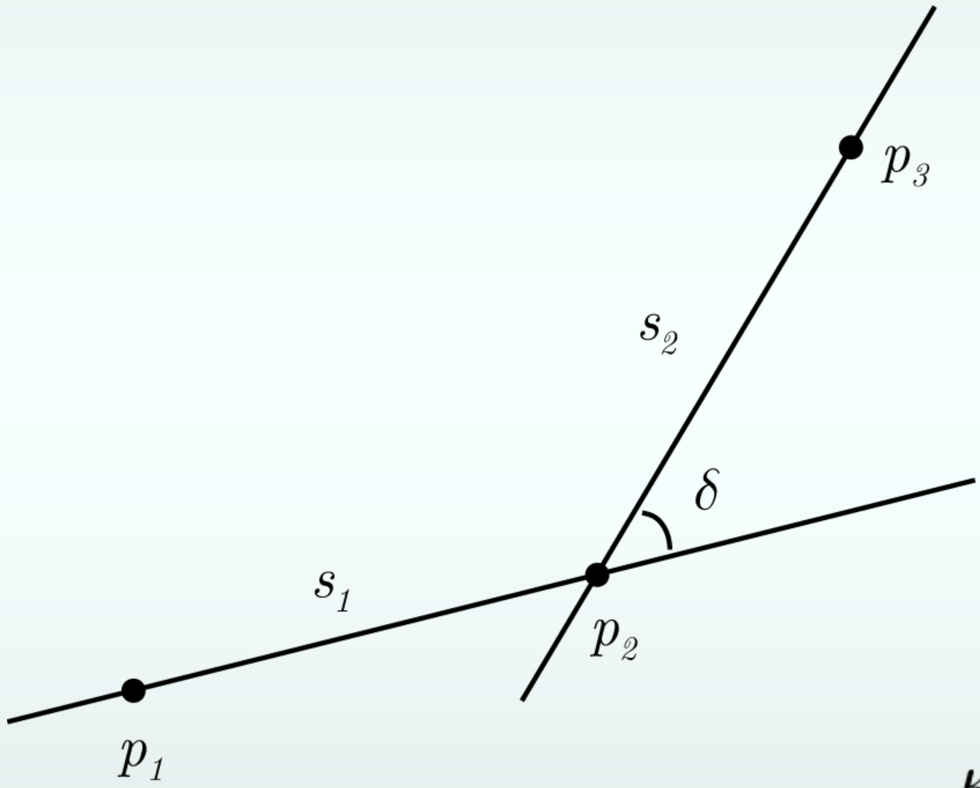
- What kind of applications are supported by (or: require) curvature estimations?

Option 1: derivative of tangent angle



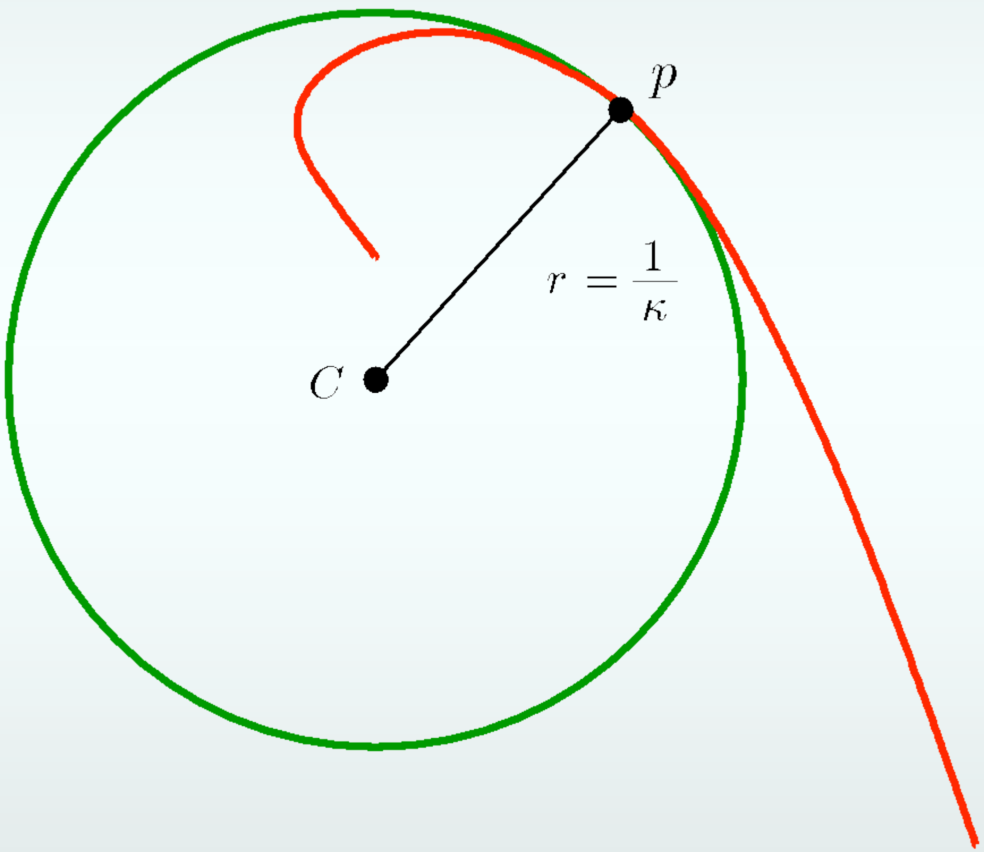
$$\kappa(p) = \lim_{pq \to 0} \frac{\delta}{pq}$$

A: Curvature Estimation from points

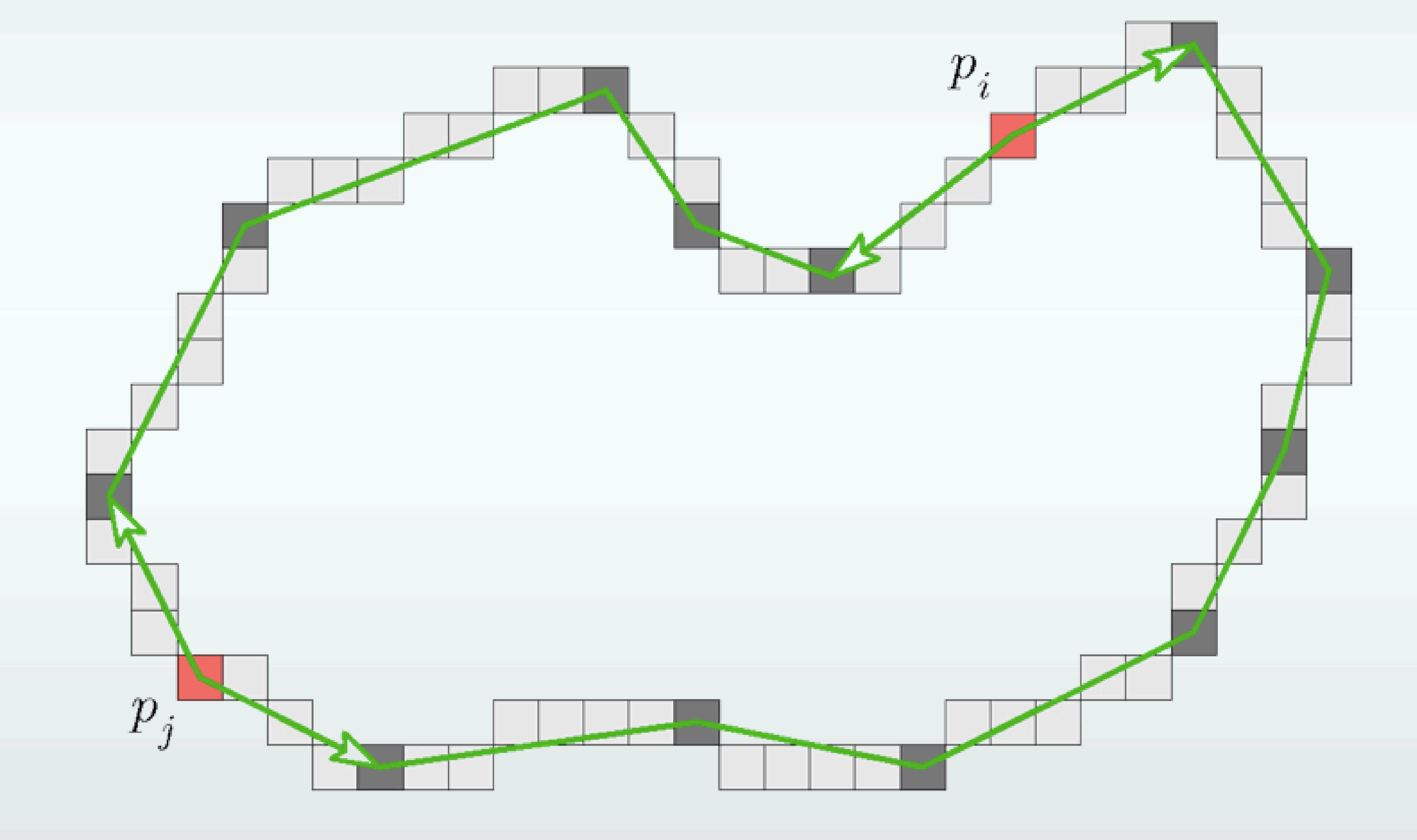


$$\kappa(p_2) = \frac{\delta}{(s_1 + s_2)/2}$$

Option 2: radius of osculating circle

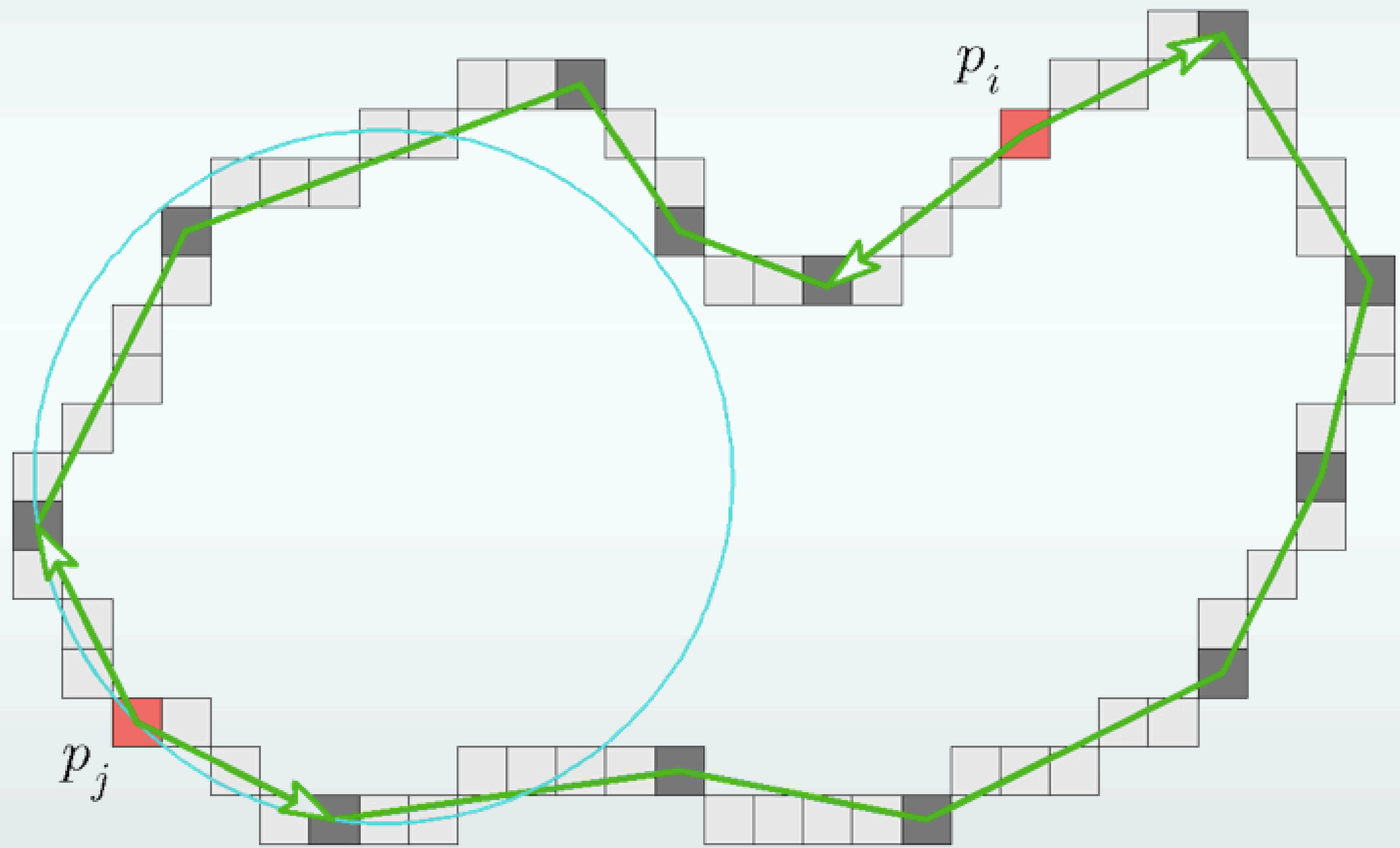


B: Curvature Estimation using DSSs

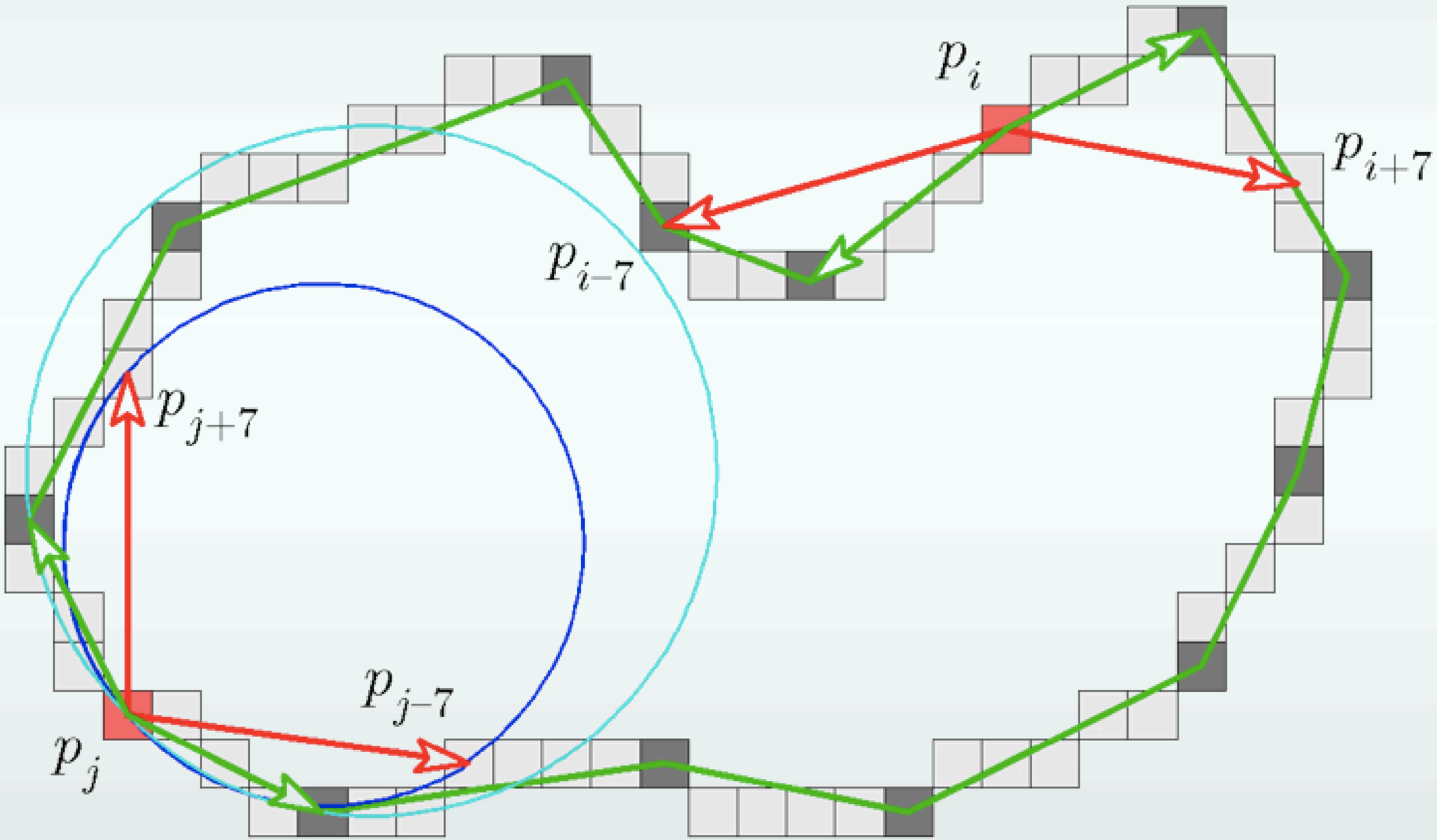


Approximation points defined by maximum-length DSSs

B: Curvature Estimation using DSSs



C: Using a Global Constant $k = a{\cdot}n$



Approximation points defined by constant *a* and length *n*

Option 3: derivative of the curve

$$\kappa = \frac{\begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}}{\left(x'^2 + y'^2\right)^{\frac{3}{2}}}$$

Requires a parametric representation $(x(t), y(t))$ of curves

D: second order curves

$$\boxed{\begin{aligned} x(t) &= a_2 t^2 + a_1 t + a_0 \\ y(t) &= b_2 t^2 + b_1 t + b_0 \end{aligned}}$$

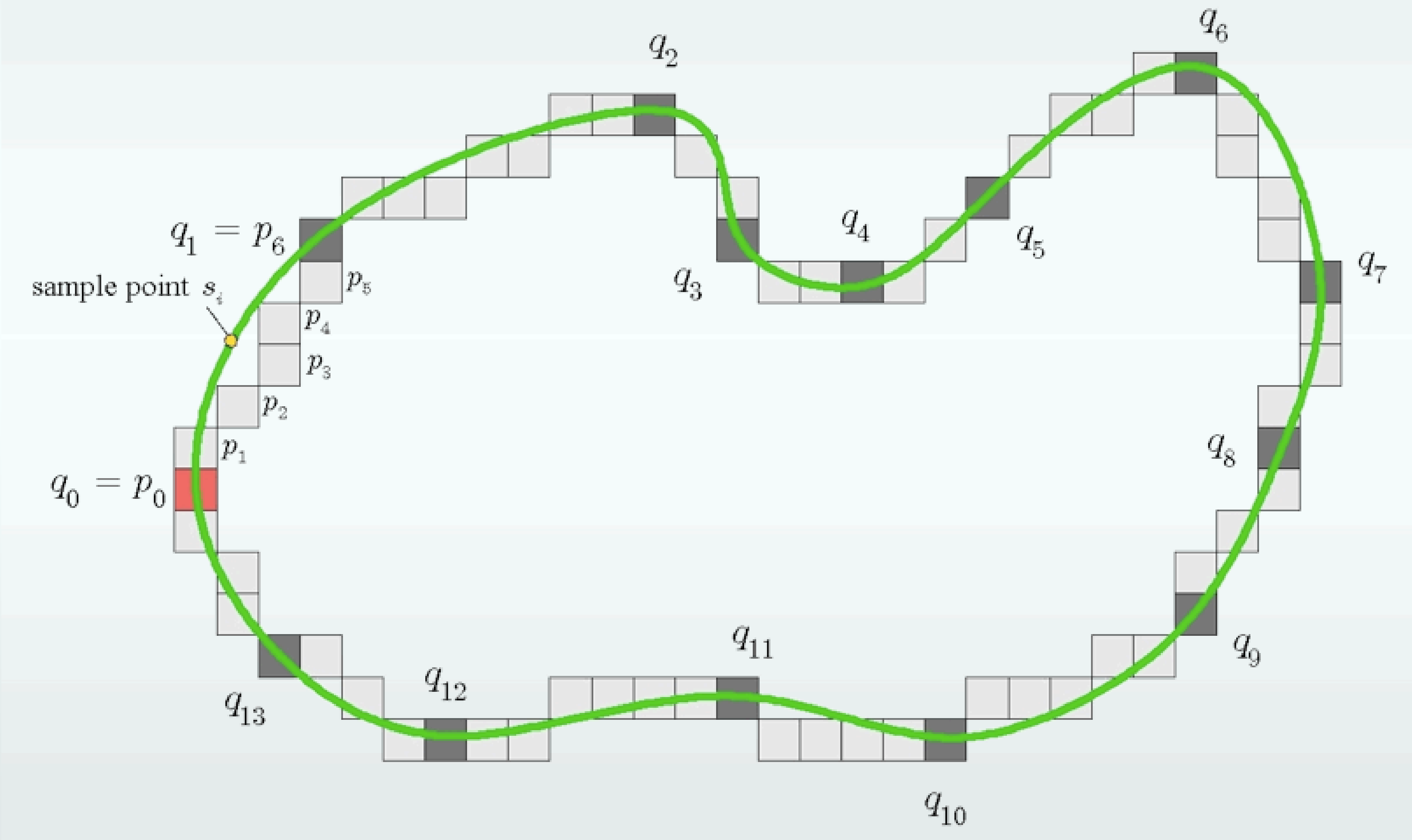
<u>Algorithm **M2003**</u>
by Majed Marji

Let $t = 0$ at p_i, and $t = 1$ and $t = -1$

at successor and predecessor.

$$\boxed{\kappa = \frac{2(a_1 b_2 - a_2 b_1)}{(a_1^2 + b_1^2)^{\frac{3}{2}}}}$$

E: spline interpolation



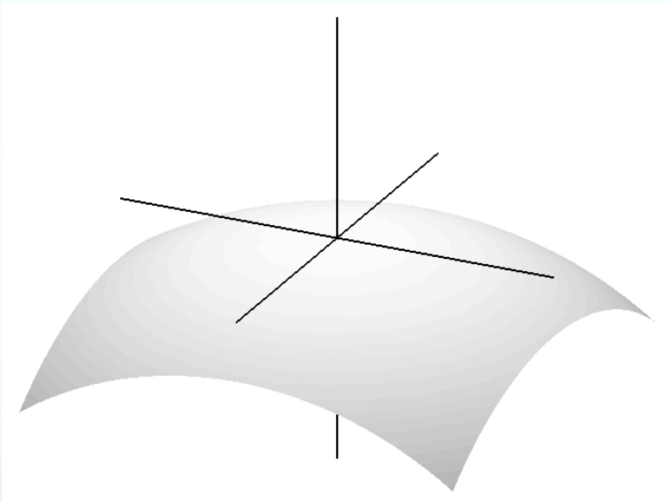
Curvature Estimation for 2D Curves

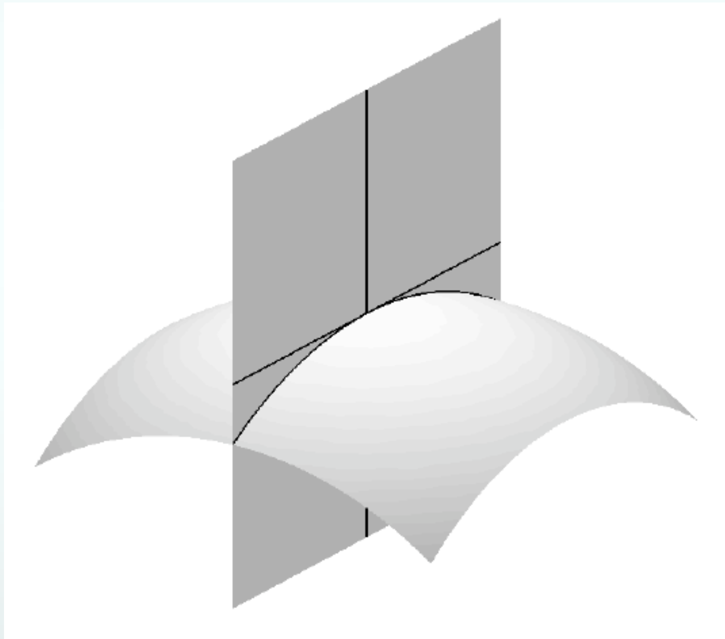
- DSS curvature estimators have in general a good overall performance, even for low resolutions,

  but fail for multigrid convergence especially in cases of convex curves (in difference to DSS based length estimation!)
- Using a global constant seems to support multigrid convergence, but there is no proof
- Spline interpolation using DSS seems to be convergent, and it converges faster than when using a global constant (thus: our recommendation), but no proofs either

Curvature Estimators for Digital Surfaces

Normal Curvature – use any cutting plane that is aligned with the surface normal at a point, then calculate the planar curvature…





NOTE: There is a minimum and a maximum normal curvature associated with each point on a C^1 surface (Gauss).

Curvature for Digital Surfaces

$$\kappa_1 = \text{minimum normal curvature}$$

$$\kappa_2 = \text{maximum normal curvature}$$

Mean Curvature: $$H = \frac{\kappa_1 + \kappa_2}{2}$$

Gaussian Curvature: $$K = \kappa_1 \, \kappa_2$$

F: Two-Cut Mean Theorem

Take the mean of the estimated curvature for any two orthogonal normal curvature cut planes (basic theorem).

$$H = \frac{\kappa_a + \kappa_b}{2}$$

This is valid for any two orthogonal normal cut planes!

G: Three Cut Mean Approach

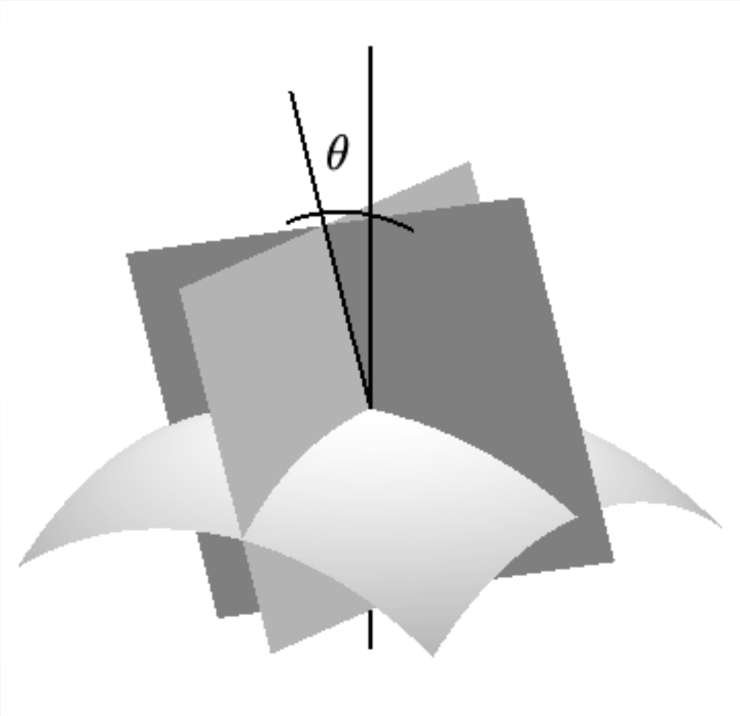
Take the mean of the estimated curvature for any three equally spaced (i.e. 60 degree angle) normal cut planes (theorem?; it works).

$$H = \frac{\kappa_a + \kappa_b + \kappa_c}{3}$$

It works for any three equally spaced normal cut planes!

Compensated Two-Cut Approach

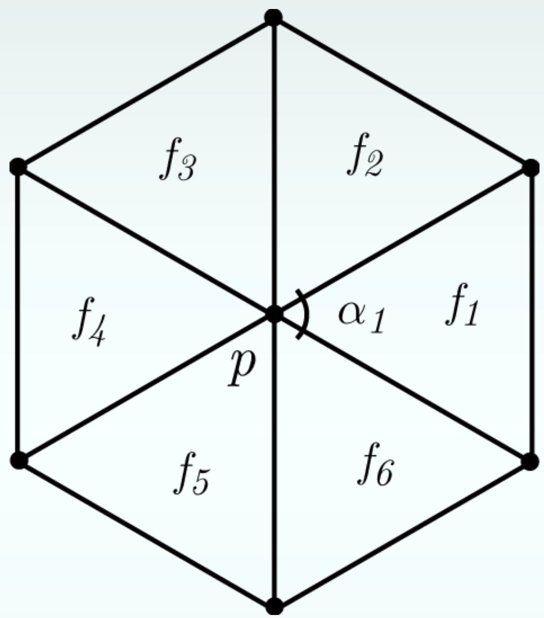
We can compensate for cut planes that do not align with the surface normal… (Meusnier, 18th century)



$$H = \left(\frac{\kappa_a + \kappa_b}{2} \right) \cos \theta$$

This technique works for both the two and three cut method.

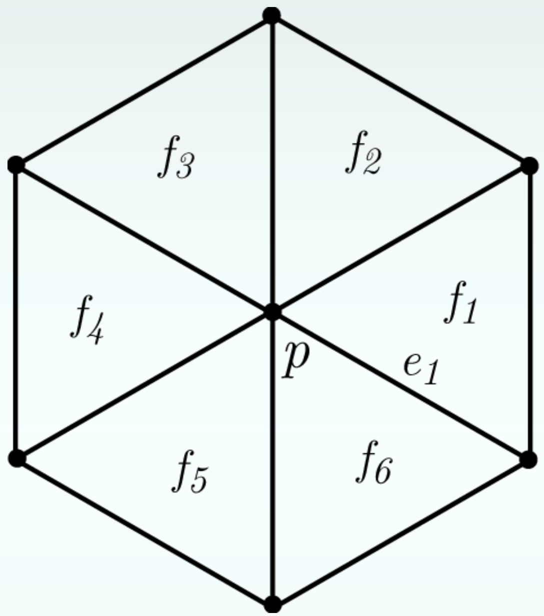
H: Gaussian Curvature & Triangle Mesh

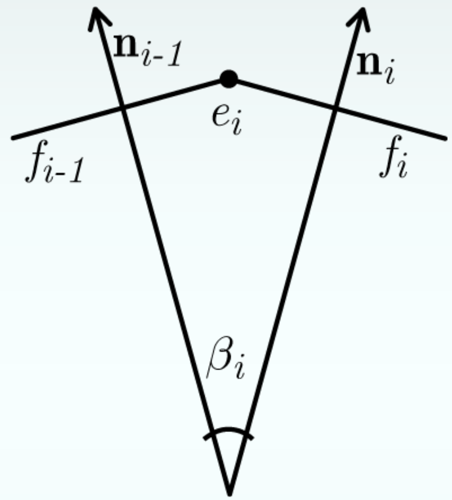


$$K(p) = \frac{3\,(2\pi - \sum \alpha_i)}{\sum \mathcal{A}(f_i)}$$

This known estimator applies for all adjacency counts greater than two!

I: Mean Curvature & Triangle Mesh





$$H(p) = \frac{3\sum ||e_i||\,\beta_i}{4\sum \mathcal{A}(f_i)}$$

This known estimator applies for all adjacency counts greater than two!

Scale Invariance

• Gaussian and Mean curvature are translation and rotation invariant, but *not scale invariant*.

• However, in shape analysis we are often interested in scale invariance.

• We introduced a scale invariant measure, *similarity curvature*.

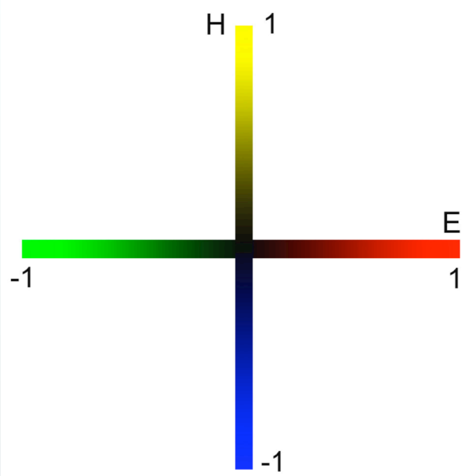
Similarity Curvature

The *similarity curvature* is given by:

$$R(p) = \begin{cases} (\kappa_3, 0) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ are both positive,} \\ (-\kappa_3, 0) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ are both negative,} \\ (0, \kappa_3) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ differ, and } |\kappa_2| \geq |\kappa_1|, \\ (0, -\kappa_3) & \text{if signs of } \kappa_1 \text{ and } \kappa_2 \text{ differ, and } |\kappa_1| > |\kappa_2|. \end{cases}$$

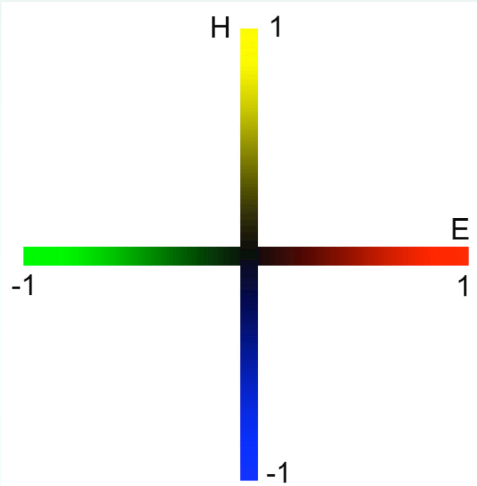
$$\kappa_3 = \frac{\min(|\kappa_1|, |\kappa_2|)}{\max(|\kappa_1|, |\kappa_2|)}$$

Shading code for similarity curvature (while noting that the first term in $R(p)$ represents *ellipsoidal patches* and the second term represents *hyperbolic patches*):

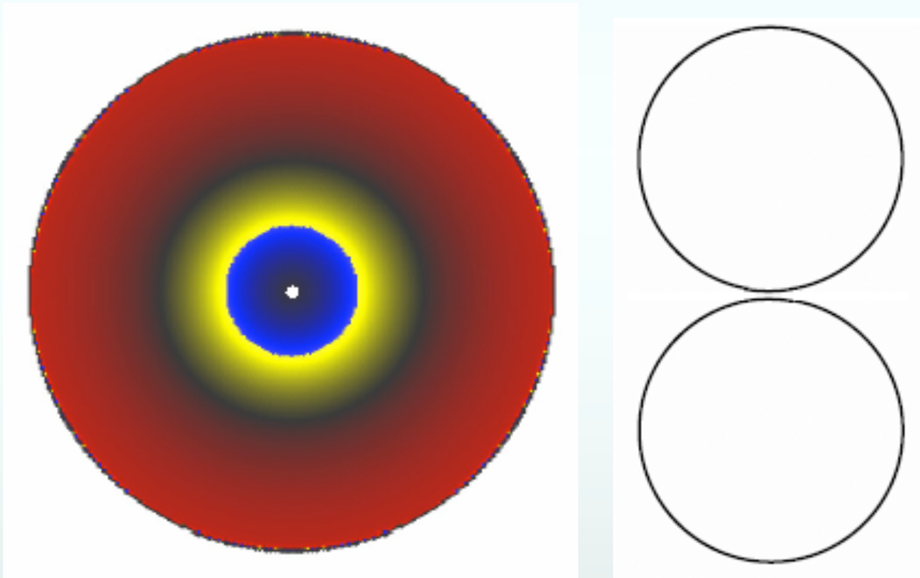


J. Rugis and R. Klette, PSIVT 2006

Example:
Similarity Curvature for Torus



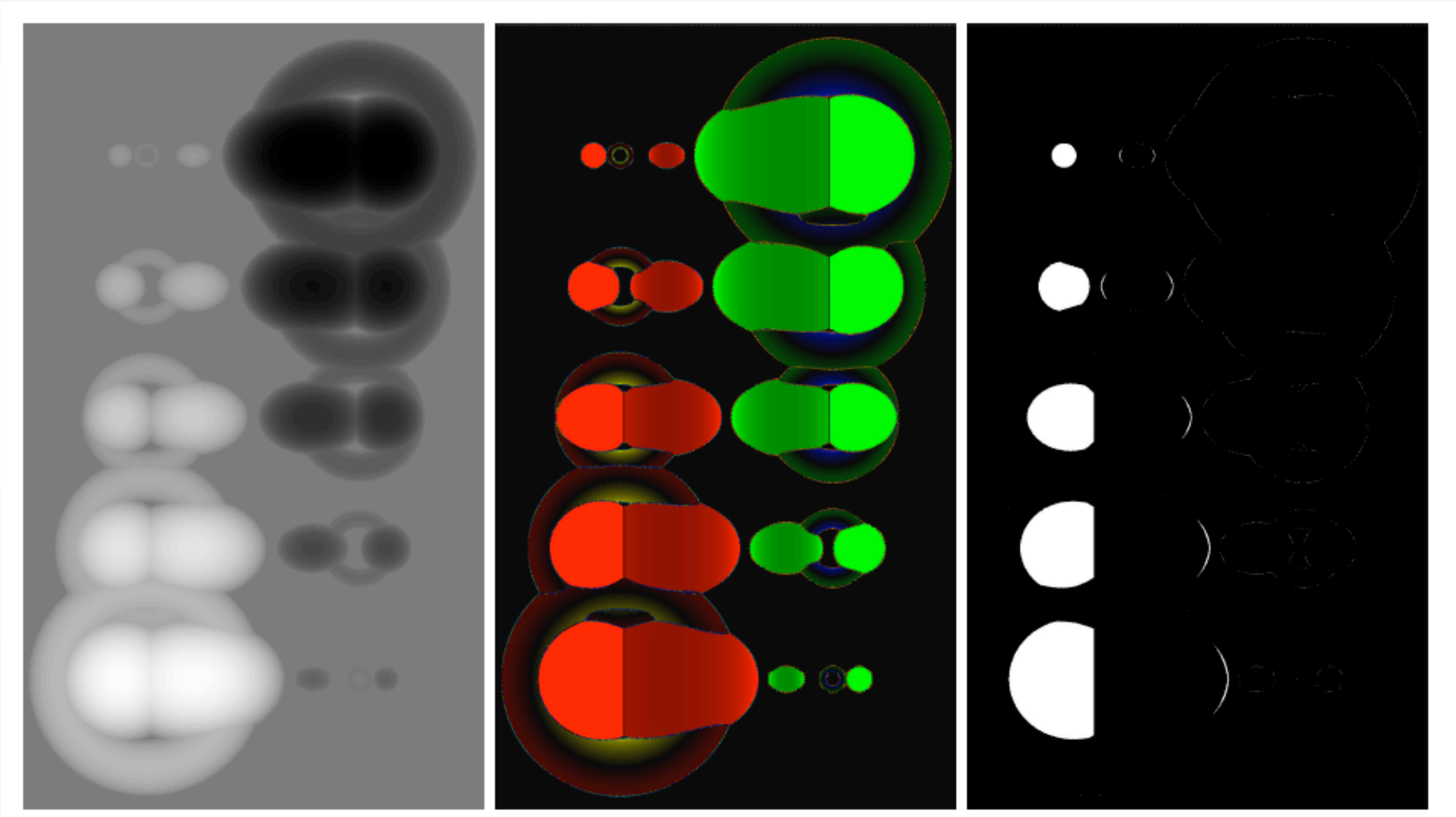
Note: The outer region of a torus is ellipsoidal while the inner region is hyperbolic.



A shaded torus and its cross-section.

Curvature Maps

We produce *curvature maps* by projecting the shading coded curvature values onto a 2D image plane. Similarity curvature maps:



Test scene depth map, similarity curvature map, and extracted spherical bump patches.

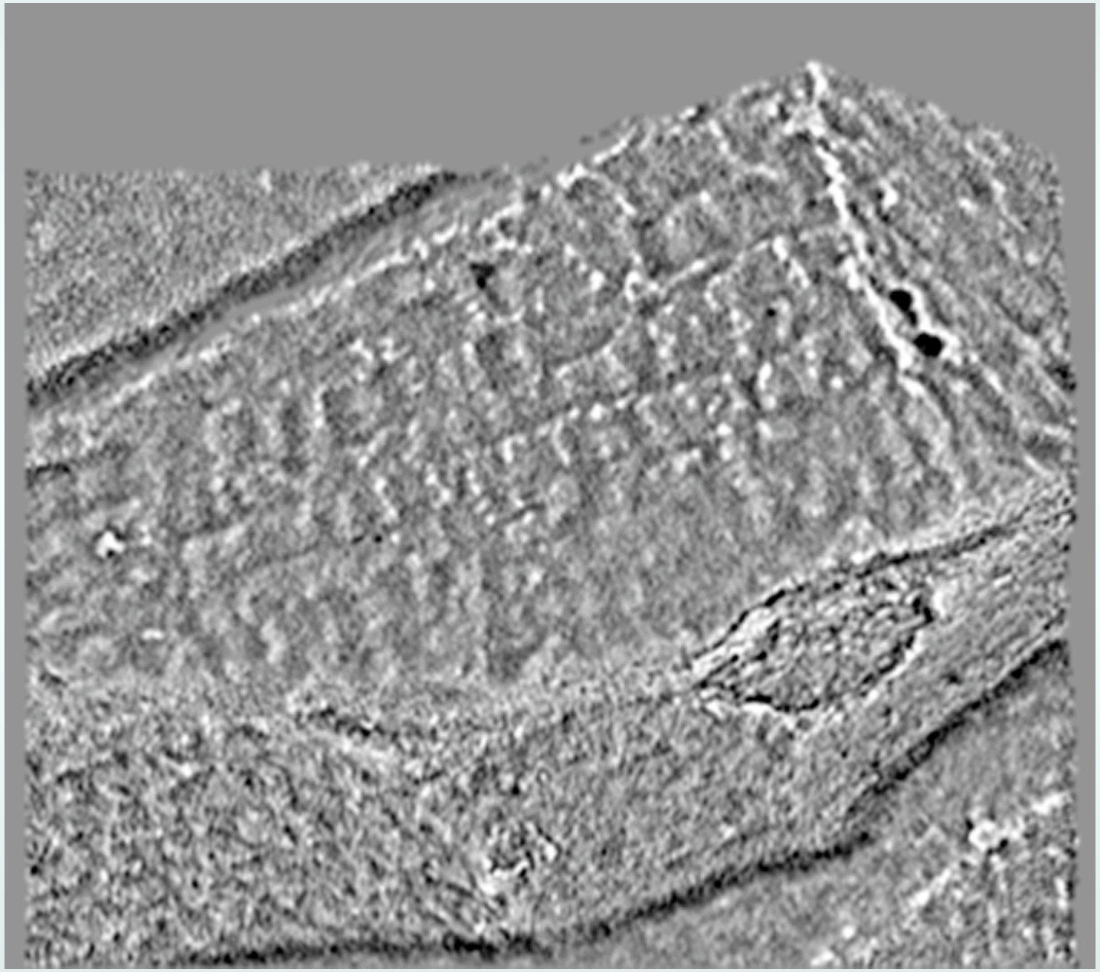
Application: The Digital Michelangelo Project

A number of Michelangelo’s statues, including the David, were digitized by a team from Stanford University in ≤ 2000. Since then, those data are a popular research subject (see SIGGRAPHs).

- The David dataset contains ~ 1.1×10^9 points.

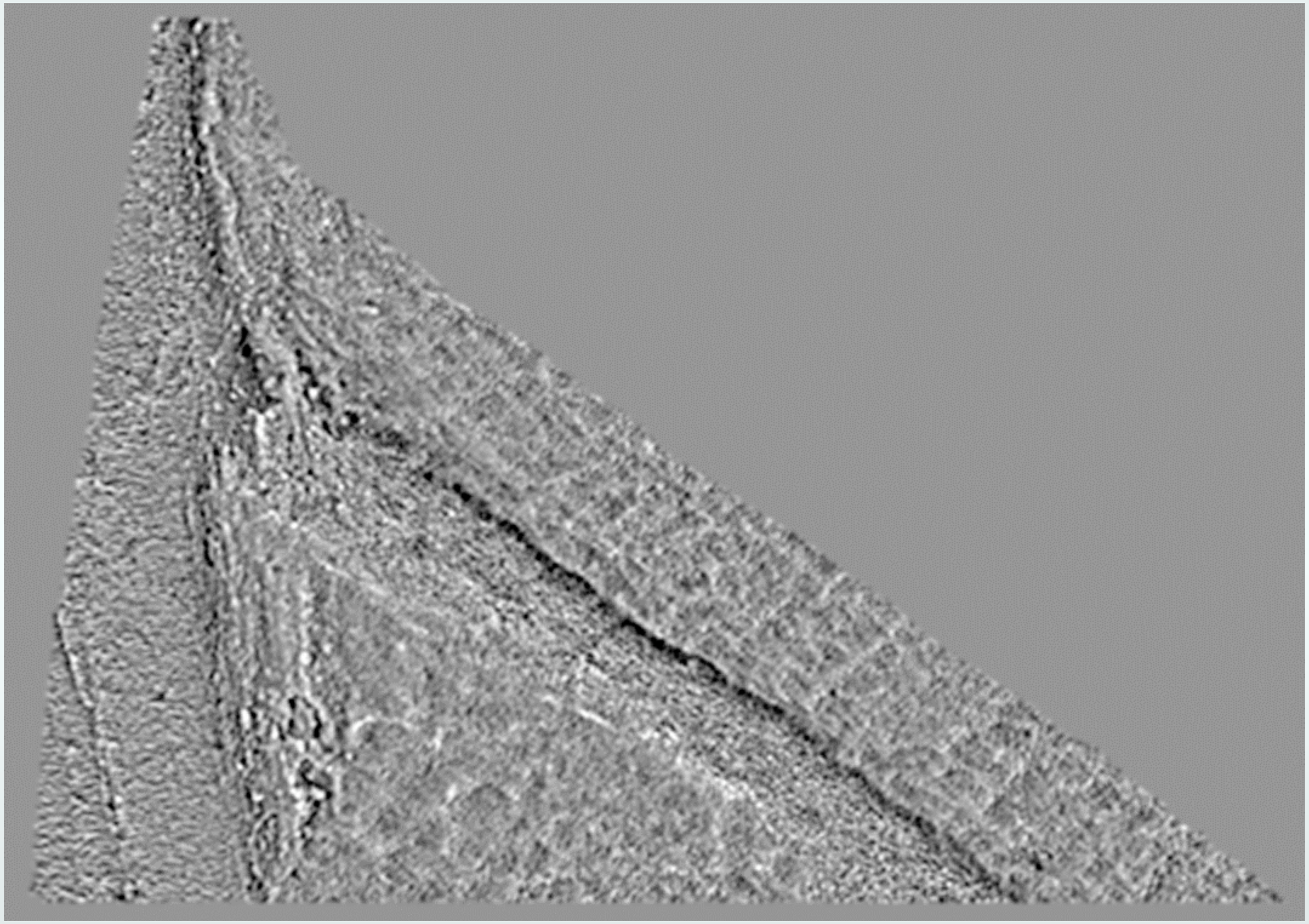
- We created *mean curvature map images*, one for each of the over 6,300 individual scans of the statue. Original intention: support alignments. It works. But surface rendering was more exciting:

Mean Curvature Map Images of David



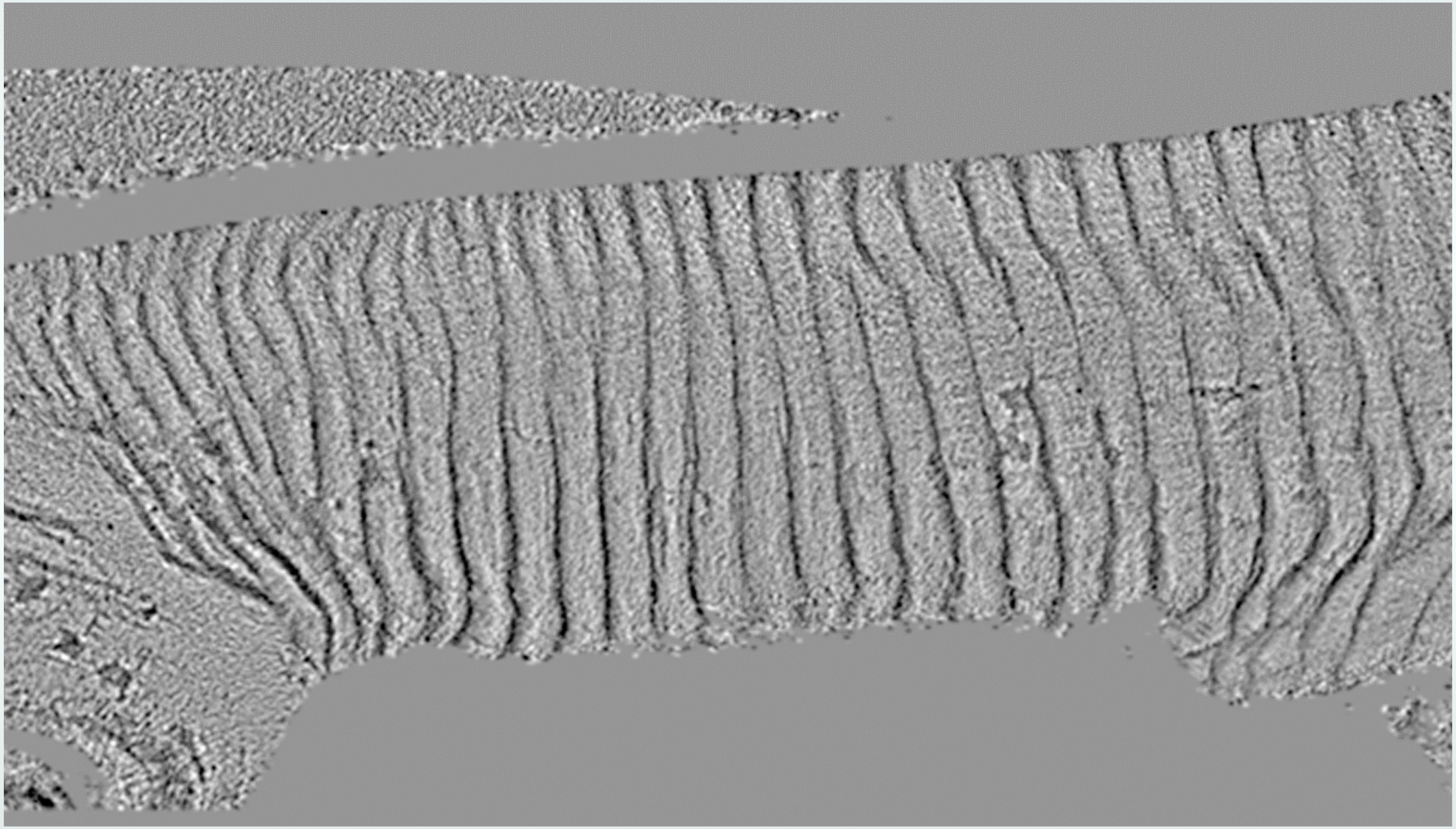
Dark: concave. Light: convex. Gray: planar.

Compensated 3 Cut or Mean Curvature & Triangle Mesh: about same



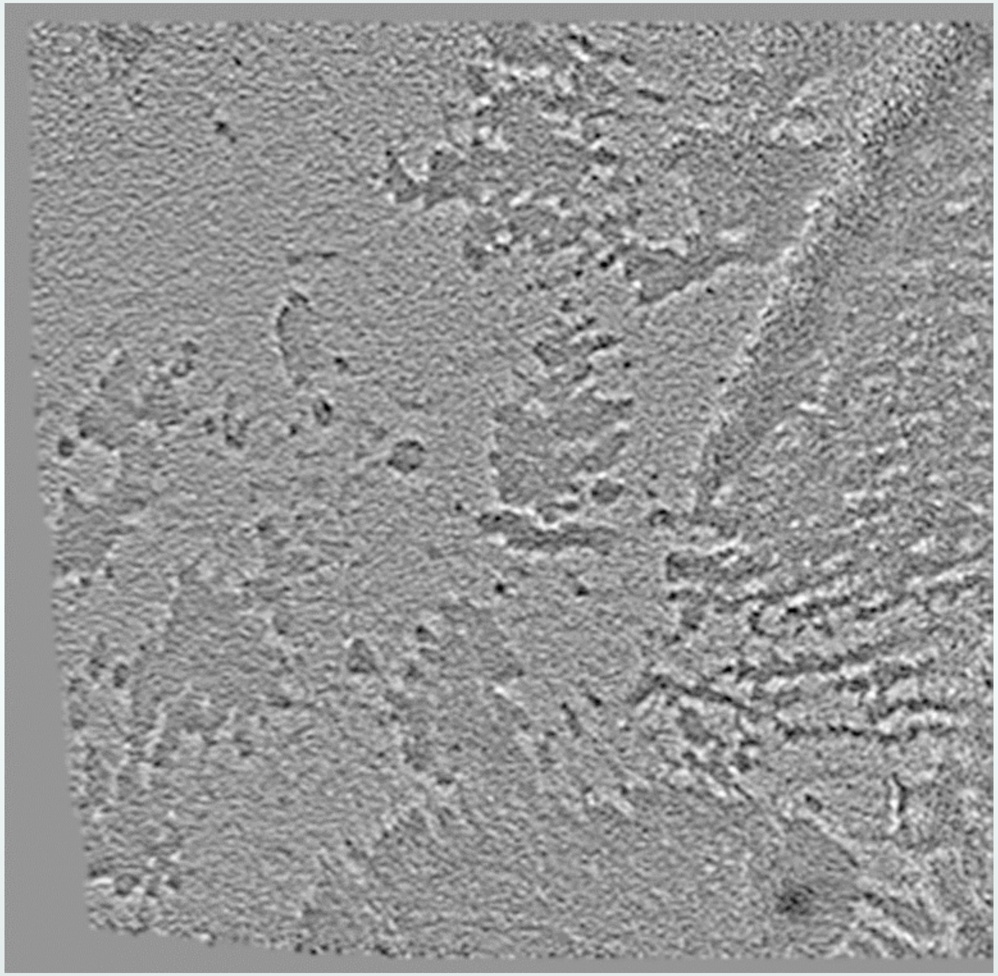
Curvature texture (rough chisel marks) in the base of the statue.

But: our Compensated 3 Cut Approach is faster



Piece of tree trunk in the statue (back of one leg).

Scan height: ~140mm   Scan resolution: ~0.3mm.

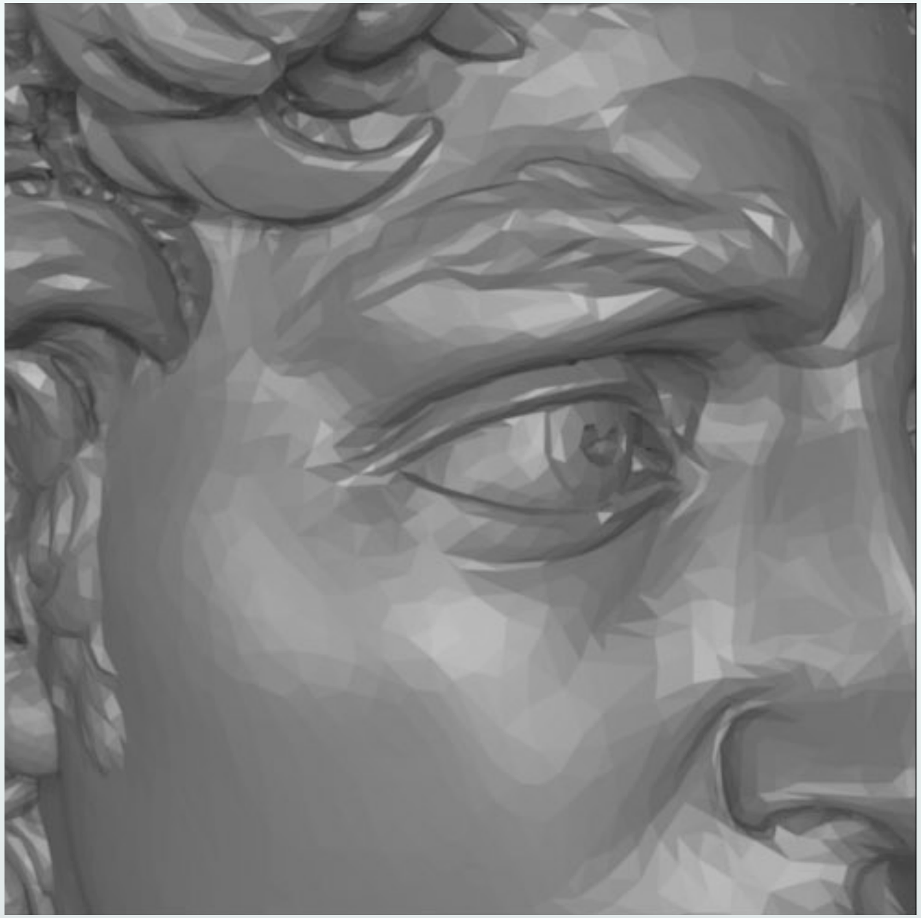


Curvature texture in the base of the statue.

Simplified Common Model

Simplification: about 15 : 1 reduction in resolution

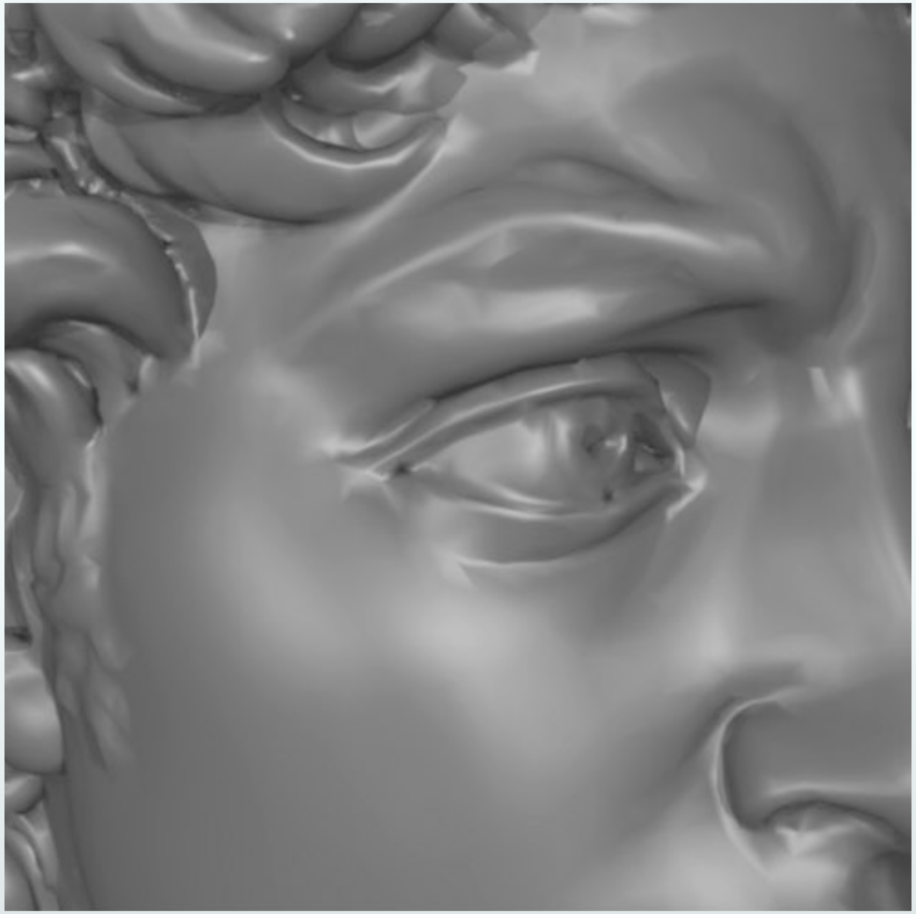
Flat shading of triangle mesh.



Simplified Common Model

Simplification: about 15 : 1 reduction in resolution

Smooth shading of triangle mesh.



Why Curvature Maps for Surface Rendering?

3D surface model: provides accurate surface geometry (note: unification of scans also based on matching curvature maps) and the “basic” surface rendering

Curvature maps provide additional surface micro-geometry information (basically at pixel or even subpixel level)

Mapping such curvature maps into “valleys” or “hills” for enhancing normal-based surface rendering (e.g. for “a shadow of a 2-pixel-diameter hill” is impractical at this level of resolution.

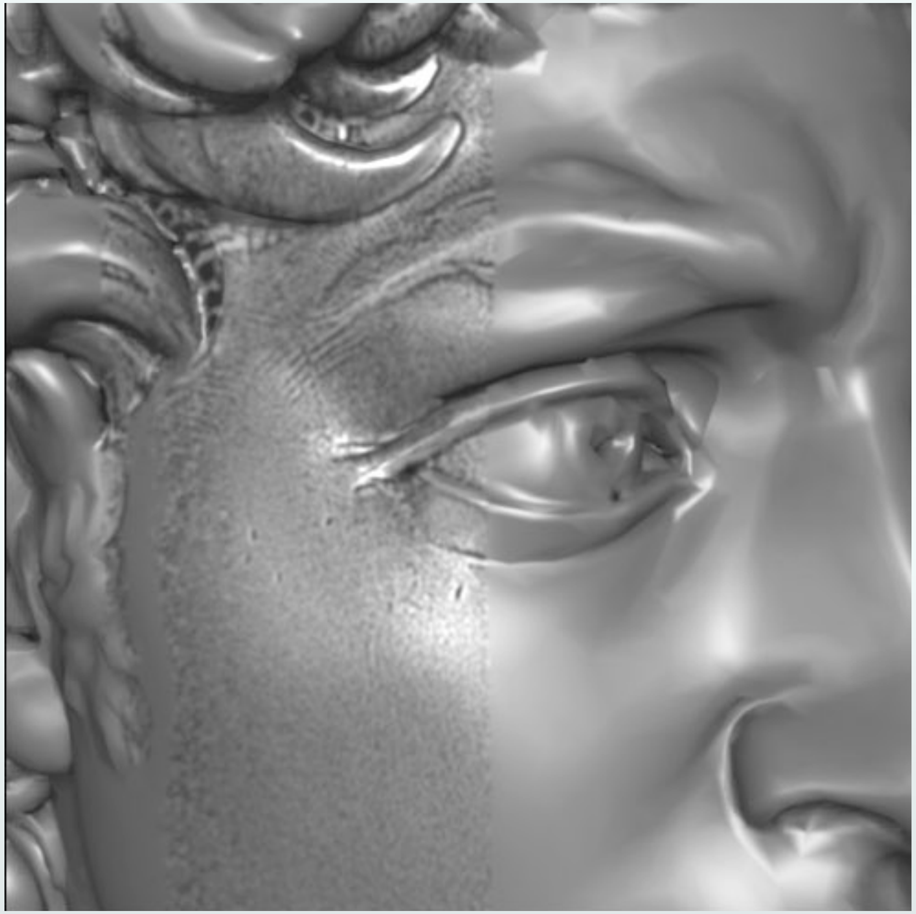
We simply add curvature maps as “surface texture” for enhanced pseudo-photorealism.

Projecting Surface Curvature Maps

We project curvature map images onto a simplified mesh model.

With a strip of projected
surface curvature.

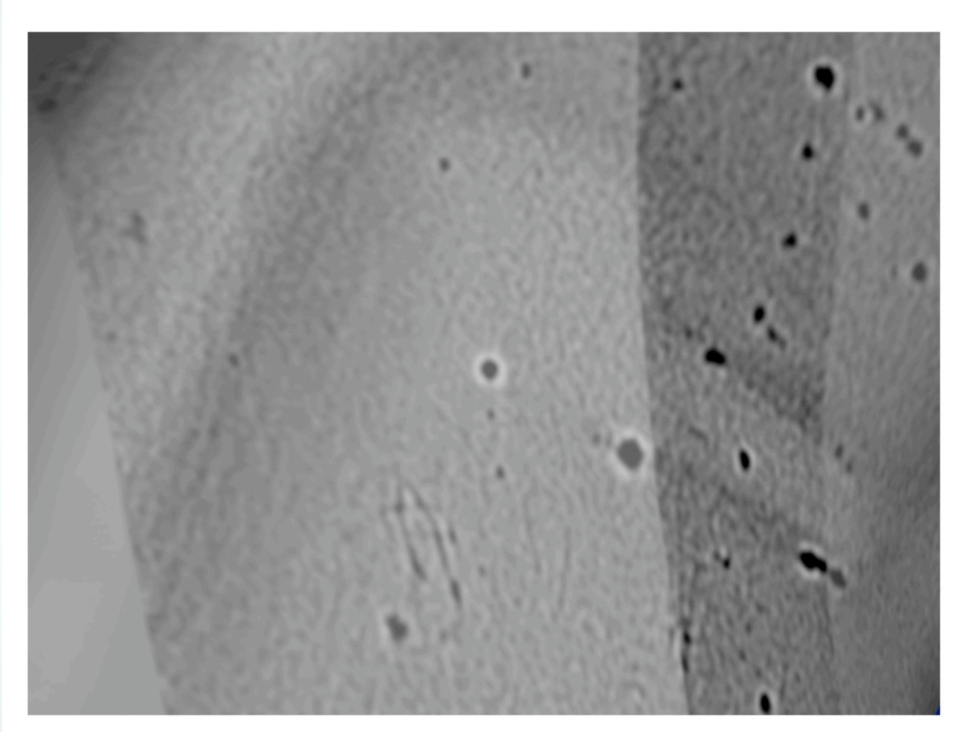
J. Rugis, SIGGRAPH 2006



Projecting Surface Curvature Maps

What about overlapping scans with the standard lighting model?

The standard lighting model (single point light source for each scans) with two overlapping scans.

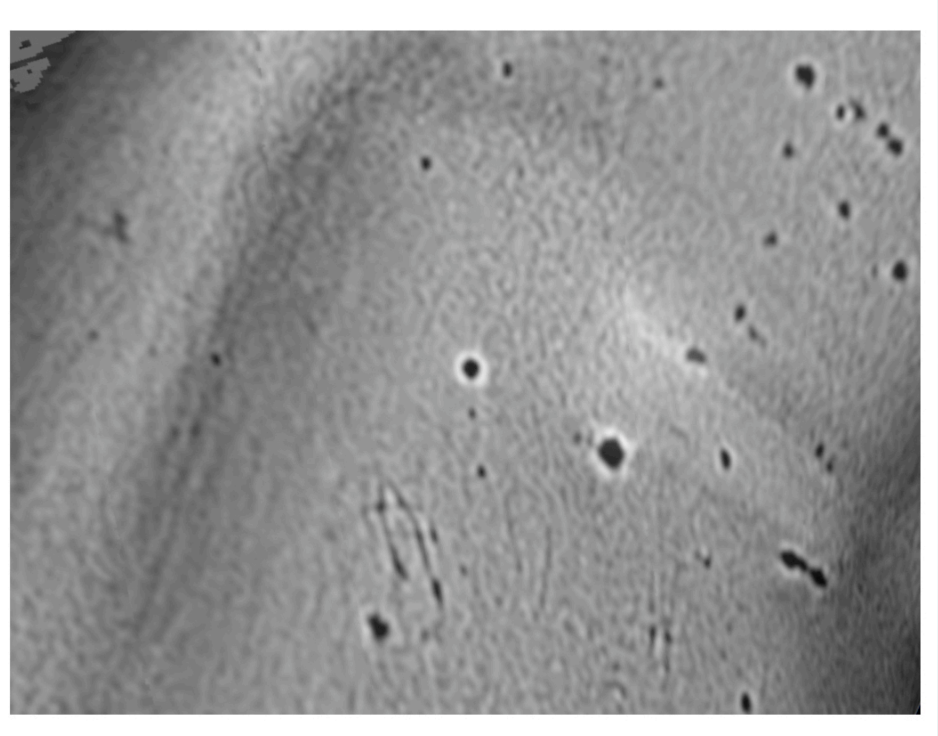


Projecting Surface Curvature Maps

We handle overlapping scans by using a new lighting model.

The new lighting model gives seamless scan strip overlap.

J. Rugis, SIGGRAPH 2007



$$I_\lambda = Ambient + \sum Specular + \left(\frac{\sum_{i=1}^{n} Projection}{n} \right) \sum Diffuse$$

Conclusions

Curvature is a valuable image analysis property. Still there are open issues to be analyzed (e.g., proof of n-cut method, $n \geq 3$).

But: curvature maps calculated via image analysis are also very useful for improved near photo-realistic and accurate 3D surface visualization.

And: calculated curvature maps add new knowledge to the historic analysis (kind of chisel used by Michelangelo etc.).