Curvature-Based Transfer Functions

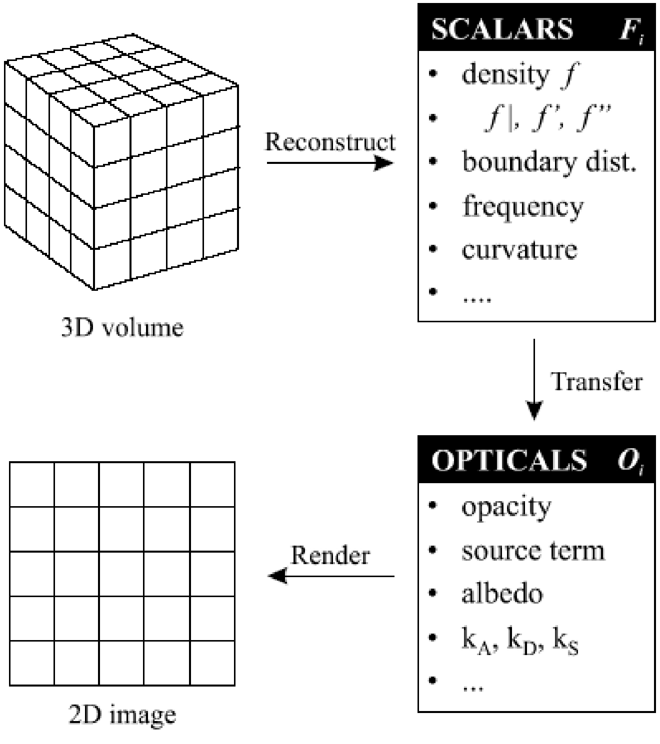
Michal Červeňanský
cervenansky@sccg.sk

Outline

- Introduction
  - TF domain
  - TF design
- Curvature-Based TF
  - definitions
  - design of curvature TF
  - curvature estimation
- Non-photorealistic rendering
  - contour thickness
  - valleys and ridges emphasizing

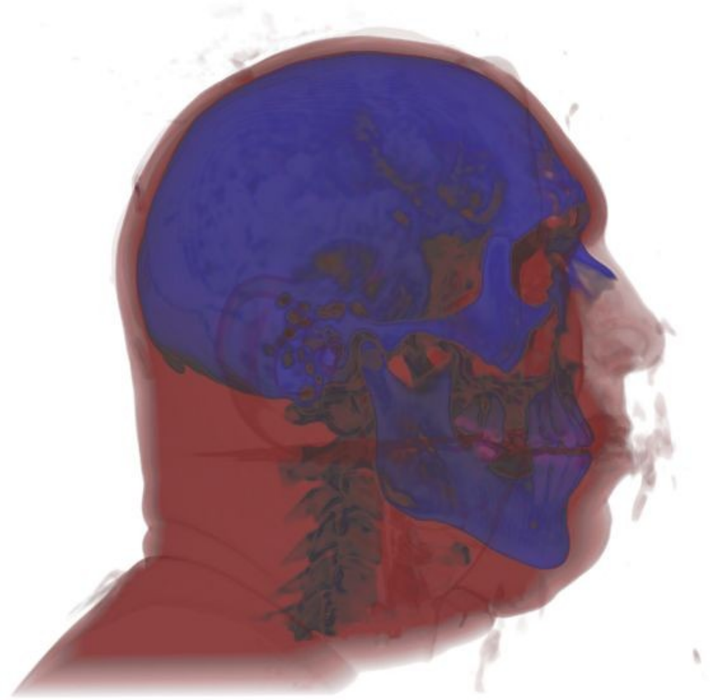
Introduction

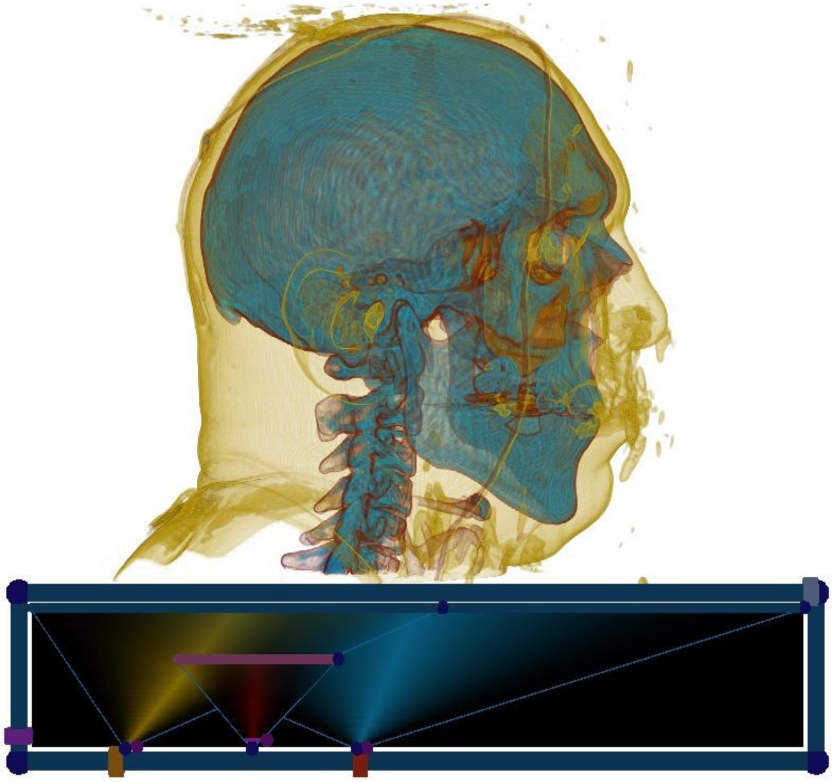
- Mapping data properties to optical properties
  - data properties: density, gradient magnitude, second-order derivative, curvature, ...
  - optical properties: opacity, color, luminance, index of refraction, ...
- Responsible for classification set



Introduction (TF domain)

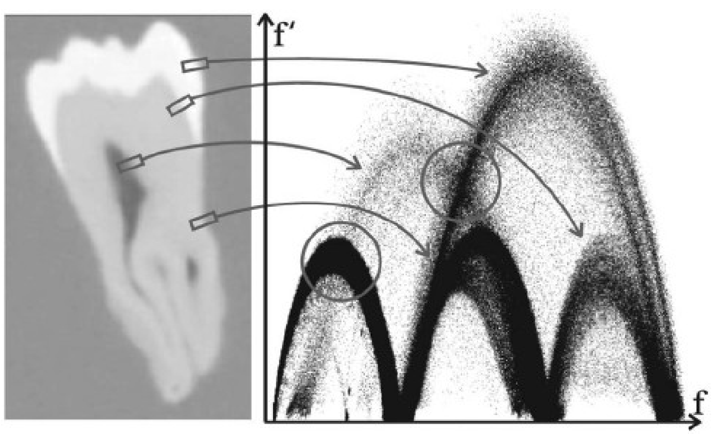
- One-dimensional - density
- Multi-dimensional – f, f', f'', curvature, frequency,...





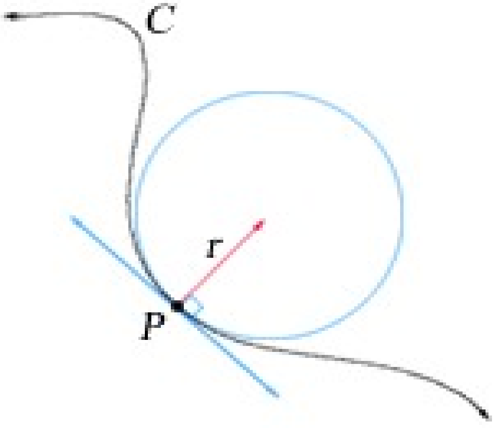
Introduction (TF design)

- Trial-error
- Data-centric (contour spectrum, iso-surfaces, curvature, boundary detection, LH histogram…)
- Image-centric (genetic alg., design galleries, …)
- Others (learning classifier, …)



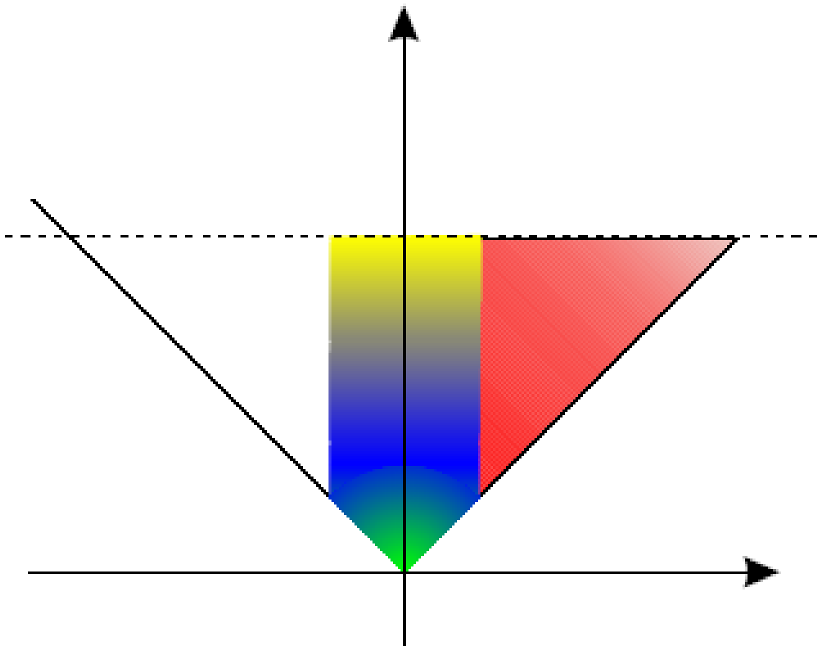
Curvature-Based TF

- Normal curvature – $k_n(t)$
  - plane (P,n,t) ∩ surface → curve → curvature
- Principal directions – s_1, s_2
  - directions where $k_n(t)$ for all t is max, min
- Principal curvature – $k_1 = k_n(s_1)$, $k_2 = k_n(s_2)$



Design of curvature TF

- k_1, k_2 defined on $R \to RxR$
- some restriction allowed:
  - $k_1 \geq k_2$ convexity/concavity determined by sign
  - $k_1 \geq 0$, $k_1 \geq |k_2|$
  - radius $\geq 1/2 \to k \leq 2$
- $|k_2| \leq k_1 << 2$



Curvature estimation

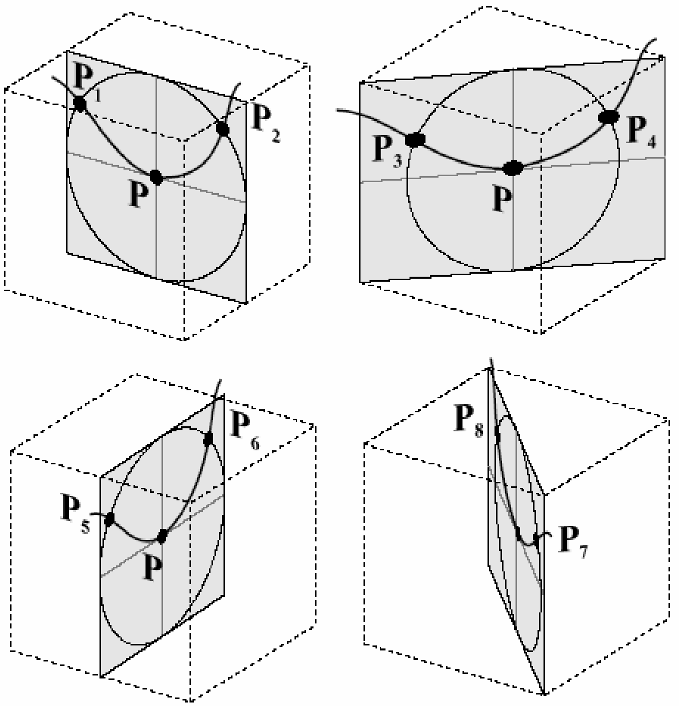
- By derivative estimation
  - sensible to noise
  - low-pass filtering
- Local fitting
  - quadratic patches
  - analytical estimation of curvature

Curvature estimation local fitting

- Dupin indicatrix $Lx^2 + 2Mxy + Ny^2 = \pm 1$
- One/pair of conics in the tangent plane
- By euler theorem $k_n(t) = k_1\cos^2(fi)+k_2\sin^2(fi)$
- $fi = fi(t)$ angle between t, s_1
- $k \geq 3$ normal curvature $\rightarrow$ k points on DI
- Linear equations system

Curvature estimation local fitting(2)

- Plane curvature estimation
- Reconstruction of a curve and normal vector
- Not necessarily normal-section curves
- Meusiner theorem $k_n(t_i) = k(t_i)\cos(fi)$
- Osculating circle fitting
- P, P_u, P_v collinear $k_n(t_i) = 0$
- Approximate OC with C
- $k(t_i)=1/|(C-P)|$
- $t_i=((P_u-P)x(P_v-P))x(C-P)$



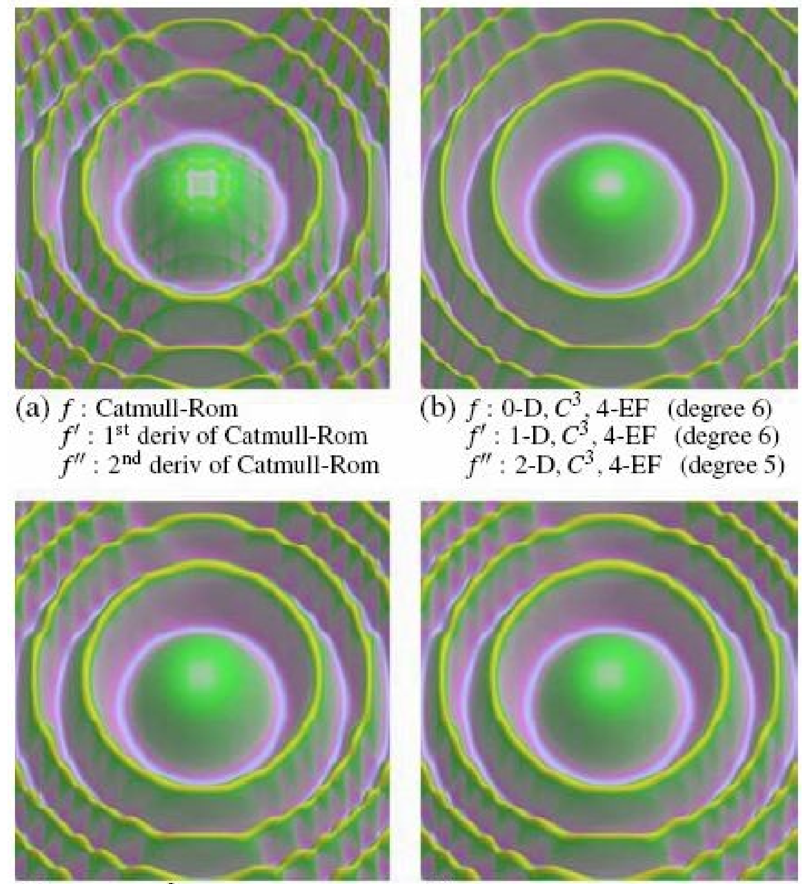
Curvature estimation by derivtive estimation

- Necessary steps for curvature estimation:
  - n = -g/|g|, P = I-nn^T
  - Hessian matrix H, geometry tensor G = -PHP/|g|
  - trace T of G
  - Frobenius norm F of G   (trance(GG^T))$^{(1/2)}$

$$\kappa_1 = \frac{T + \sqrt{2F^2 - T^2}}{2},\ \kappa_2 = \frac{T - \sqrt{2F^2 - T^2}}{2}$$

Curvature estimation by derivtive estimation (2)

- Partial derivatives by convolution (g, H)
- Piecewise polynomial filters
- 200 filter combination
- Accuracy > continuity
- Smoothing for noisy data



(a) f : Catmull-Rom
f' : 1st deriv of Catmull-Rom
f'' : 2nd deriv of Catmull-Rom

(b) f : 0-D, C^3, 4-EF (degree 6)
f' : 1-D, C^3, 4-EF (degree 6)
f'' : 2-D, C^3, 4-EF (degree 5)

(c) f : 0-D, C^3, 3-EF (degree 7)
f' : 1-D, C^2, 2-EF (degree 4)
f'' : 2-D, C^0, 2-EF (degree 1)

(d) f : Catmull-Rom
f' : 1st deriv of B-spline
f'' : 2nd deriv of B-spline

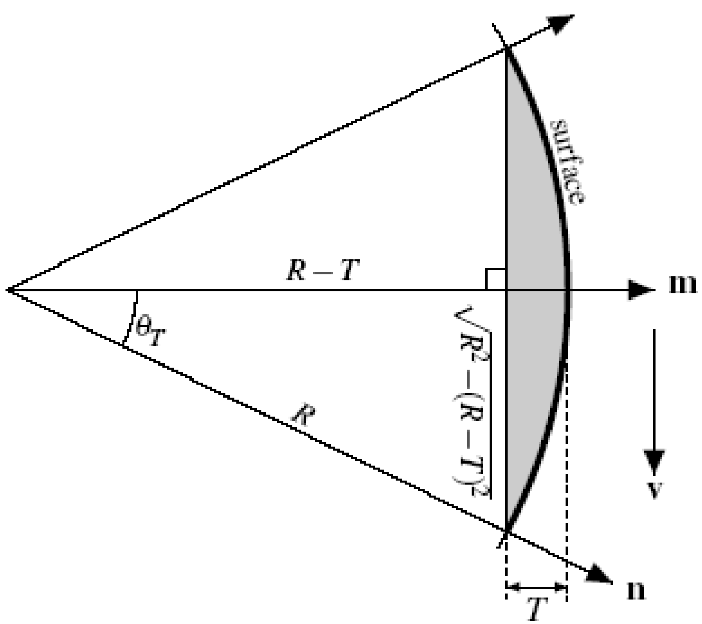
Non-photorealistic rendering contour thickness

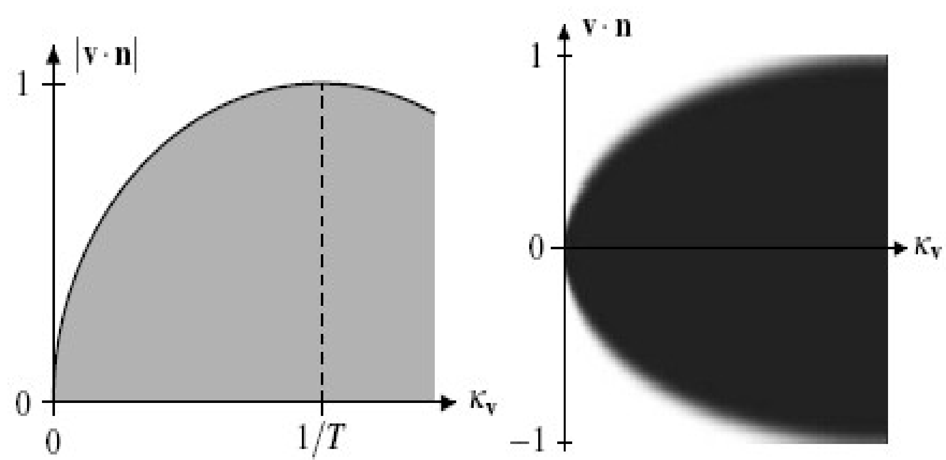
- Transition between front/back-facing surface
- Contours based on v.n   0
- Nearly flat surface → contours too thick

Non-photorealistic rendering contour thickness (2)

- Curvature along the view direction k_v

$$\kappa_{\mathbf{v}} = \frac{(\mathbf{P}\mathbf{v})^{\mathrm{T}}}{|\mathbf{P}\mathbf{v}|} \mathrm{G} \frac{\mathbf{P}\mathbf{v}}{|\mathbf{P}\mathbf{v}|} = \frac{\mathbf{v}^{\mathrm{T}} \mathbf{G}\, \mathbf{v}}{|\mathbf{P}\mathbf{v}|^2} = \frac{\mathbf{v}^{\mathrm{T}} \mathbf{G}\, \mathbf{v}}{\mathbf{v}^{\mathrm{T}} \mathbf{P}\, \mathbf{v}}$$

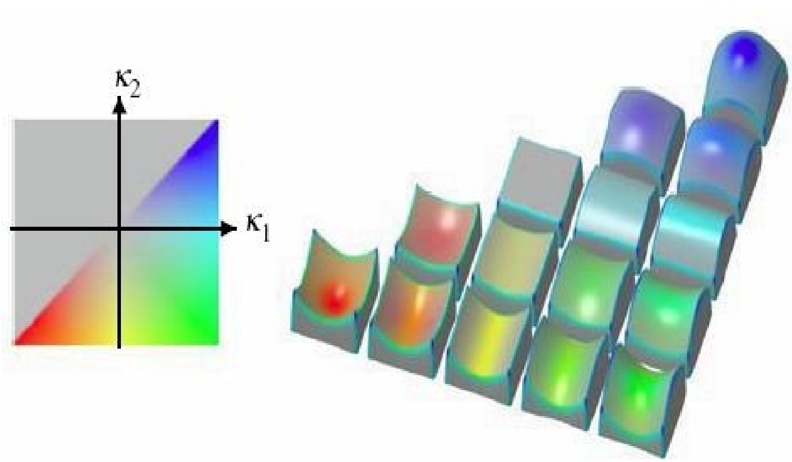


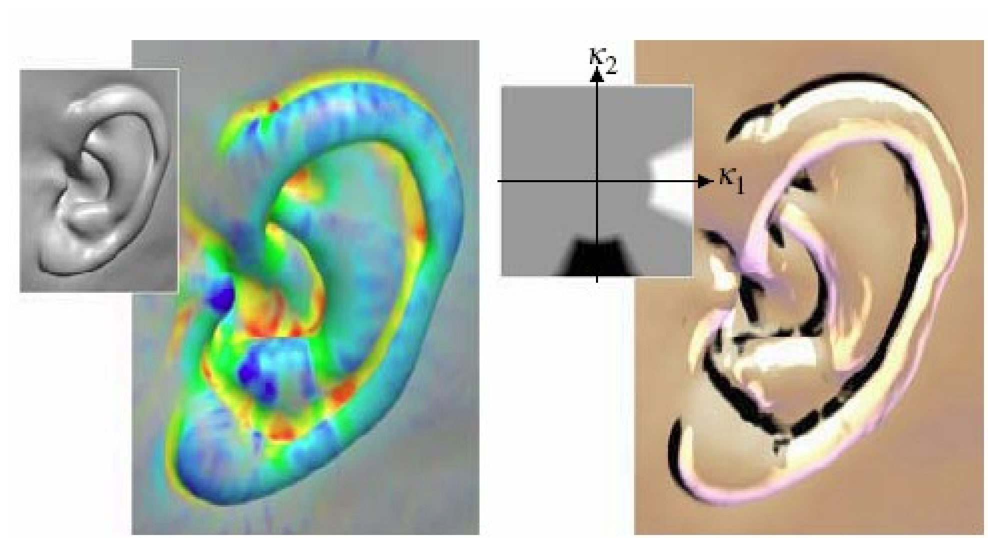


(a) Contour function (b) 2-D lookup table

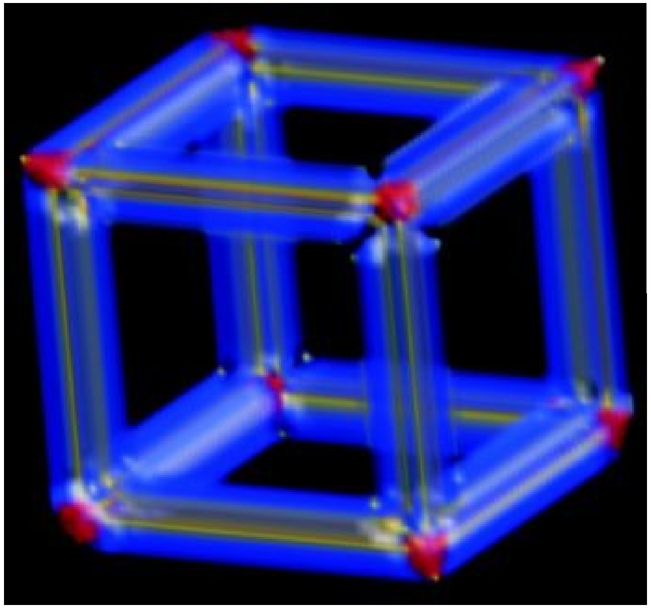
Non-photorealistic rendering valleys and ridges emphasizing

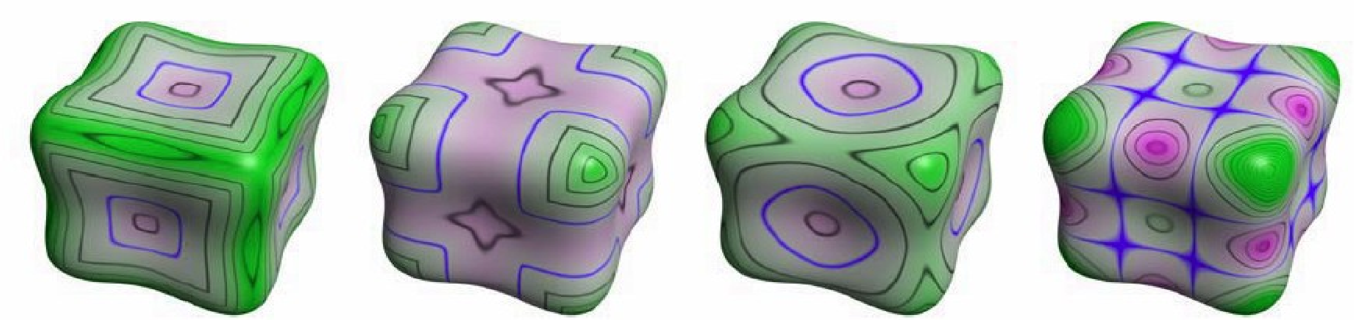
- Used for depiction of surface details
- Two-dimensional space of k_1, k_2 – lookup table

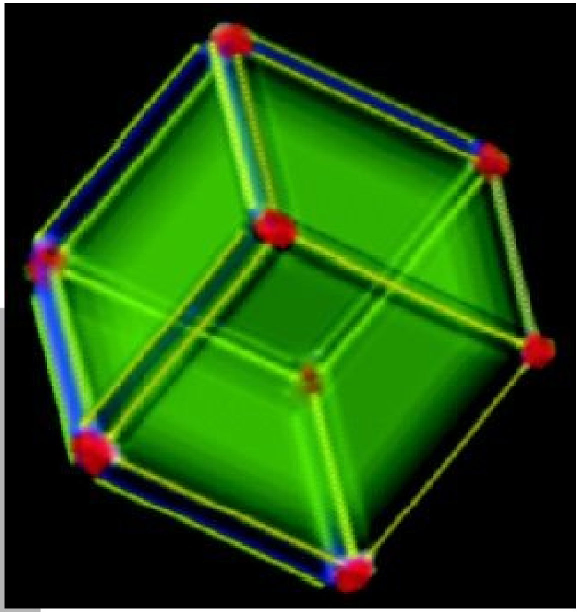


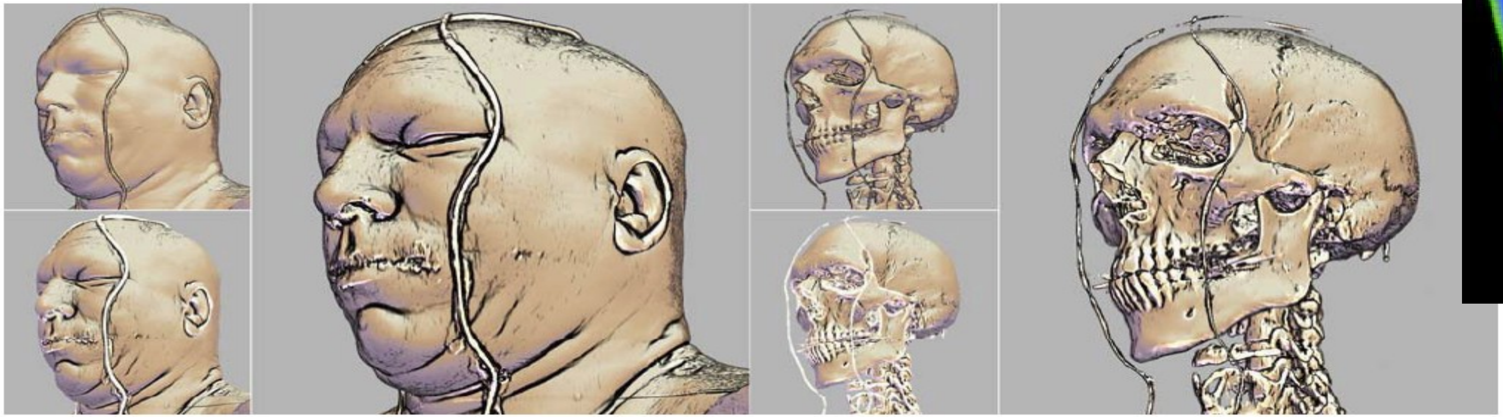


Results









Thank you
for your attention.