Overview

- Related Work
- Tensor Voting in 2-D
- Tensor Voting in 3-D
- Tensor Voting in N-D
- Application to Vision Problems
- Stereo
- Visual Motion
- Binary-Space-Partitioned Images
- 3-D Surface Extraction from Medical Data
- Epipolar Geometry Estimation for Non-static Scenes
- Image Repairing
- Range and 3-D Data Repairing
- Video Repairing
- Luminance Correction
- Conclusions

Tensor Voting in 3-D

- Representation with tensors
- Tensor voting and voting fields
- First order voting
- Vote analysis and structure inference
- Examples
- Curvature

3-D Tensor Voting

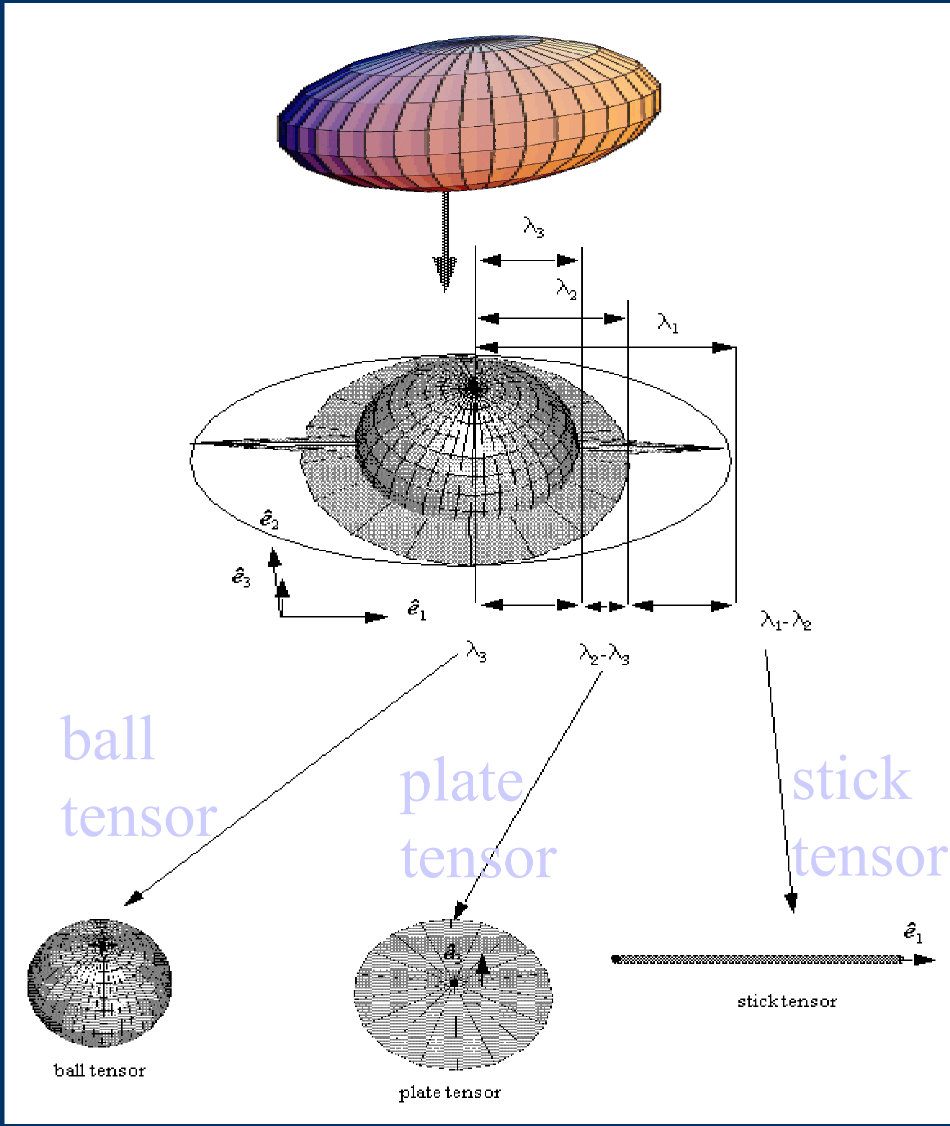
- Representation: **3-D Tensors**
- Constraints: **3-D Voting Fields**
- Data communication: **Voting**

3-D Tensors

The input may consist of

point curvel surfel

3-D Tensor Decomposition



3 eigenvalues (λ_{max} λ_{mid} λ_{min})

3 eigenvectors ($\mathbf{V}_{max}$ $\mathbf{V}_{mid}$ $\mathbf{V}_{min}$)

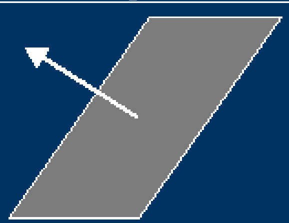
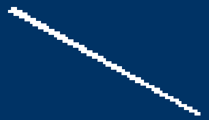
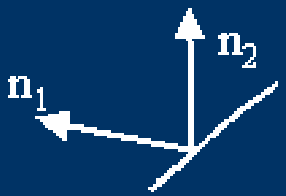
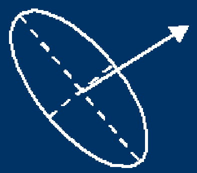
3-D second order Tensors

Equivalent to:

- Ellipsoid
  - Special cases: *"stick"*, *"plate"* and *"ball"*
- 3x3 matrix

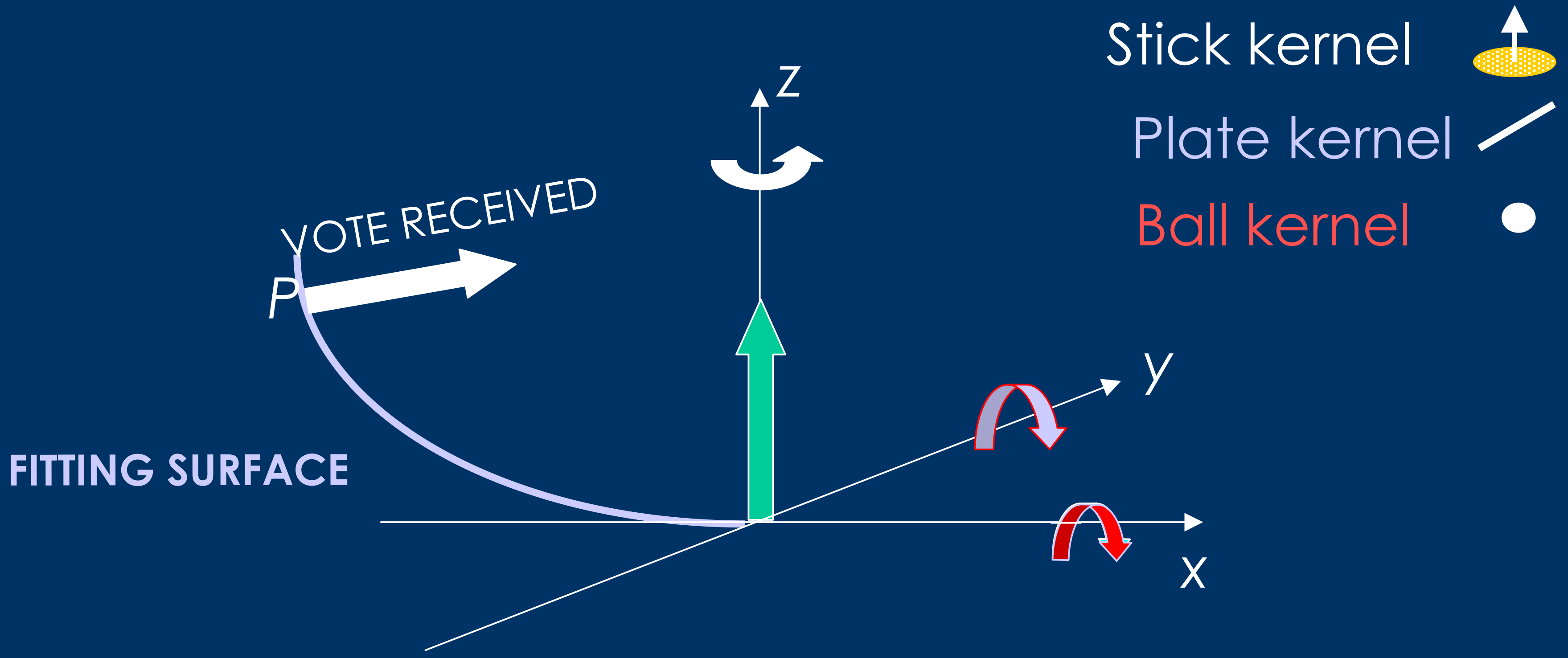
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T + \lambda_3 \cdot e_3 e_3^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) + \lambda_3 (e_1 e_1^T + e_2 e_2^T + e_3 e_3^T)$$

Representation

Input	Second Order Tensor	Eigenvalues	Quadratic Form
		$\lambda_1=1$ $\lambda_2=\lambda_3=0$	$\begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_x n_y & n_y^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix}$
		$\lambda_1=\lambda_2=1$ $\lambda_3=0$	$\begin{bmatrix} n_{1x}^2+n_{2x}^2 & n_{1x}n_{1y}+n_{2x}n_{2y} & n_{1x}n_{1z}+n_{2x}n_{2z} \\ n_{1x}n_{1y}+n_{2x}n_{2y} & n_{1y}^2+n_{2y}^2 & n_{1y}n_{1z}+n_{2y}n_{2z} \\ n_{1x}n_{1z}+n_{2x}n_{2z} & n_{1y}n_{1z}+n_{2y}n_{2z} & n_{1z}^2+n_{2z}^2 \end{bmatrix}$
		$\lambda_1=\lambda_2=\lambda_3=1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3-D Voting Fields

Derived from the Fundamental 2-D Stick Field



Voting Fields in 3-D

- 2-D stick fields are cuts of the 3-D ones containing the voting stick
  - 3-D first and second order stick fields derived by rotating the *fundamental 2-D stick field*
- Plate and Ball fields derived by integrating contributions of rotating stick voter
  - Stick spans disk and sphere respectively

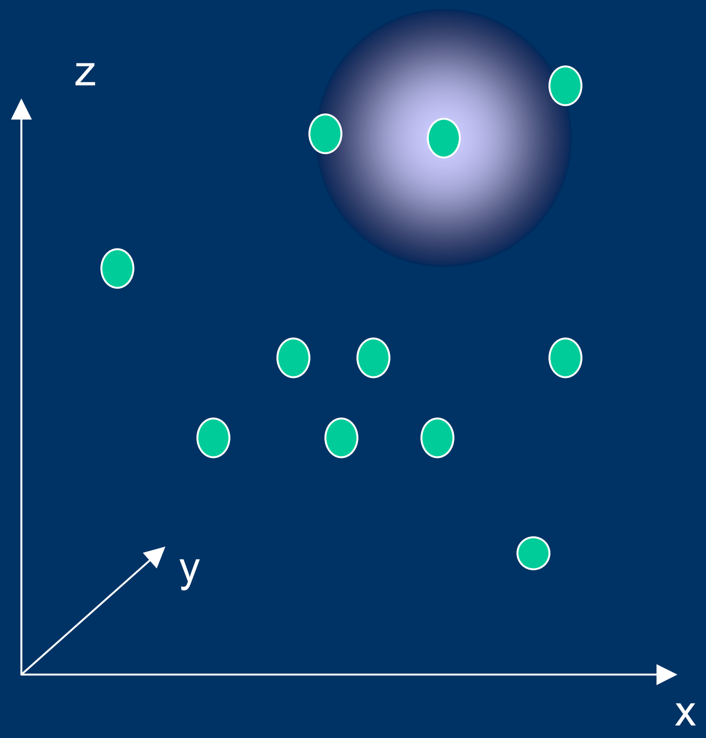
Pre-computed Voting Fields

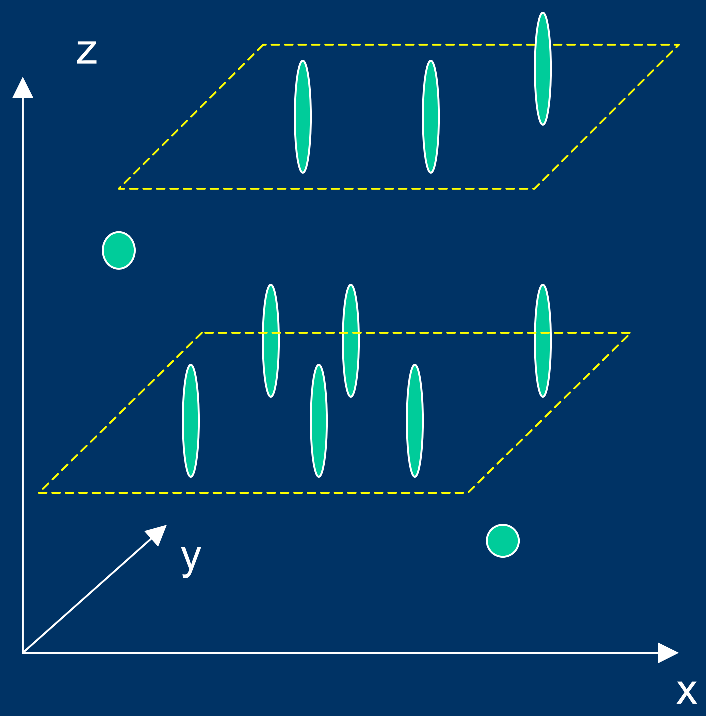
- All fields are computed at grid locations once
- When voting takes place
  - Fields aligned with voting tensors
  - Used as look-up tables
  - Votes at receivers not on grid computed by tri-linear interpolation
- Small trade-off in accuracy for considerable improvement in speed

Tensor Voting in 3-D

- Input tensors are decomposed into:
  - Stick
  - Plate
  - Ball
- Each component casts first and second order votes
- Each token accumulates all votes cast by its neighbors

Second Order Voting

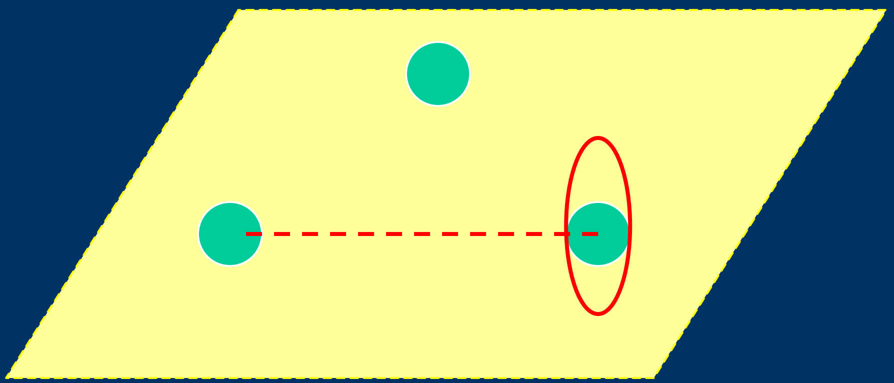




- Tokens in the same structure reinforce each other
- Isolated tokens receive little or contradicting support

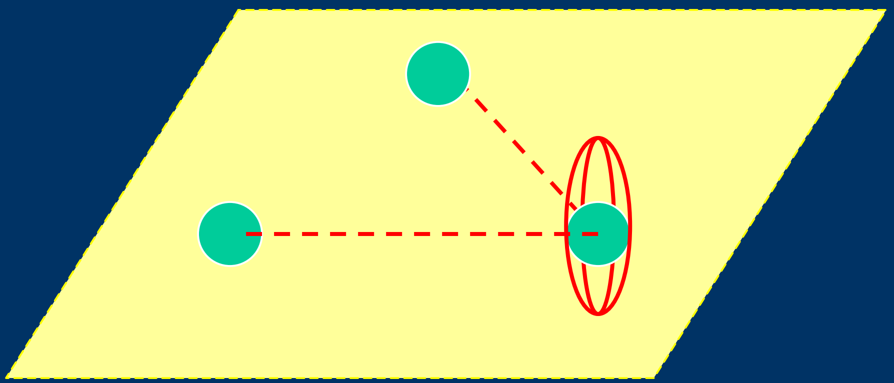
Surface Normal Inference from Unoriented Tokens

- Three unoriented tokens define plane, but voting operates pairwise
- Two tokens define a straight line and the voter casts a *plate vote*



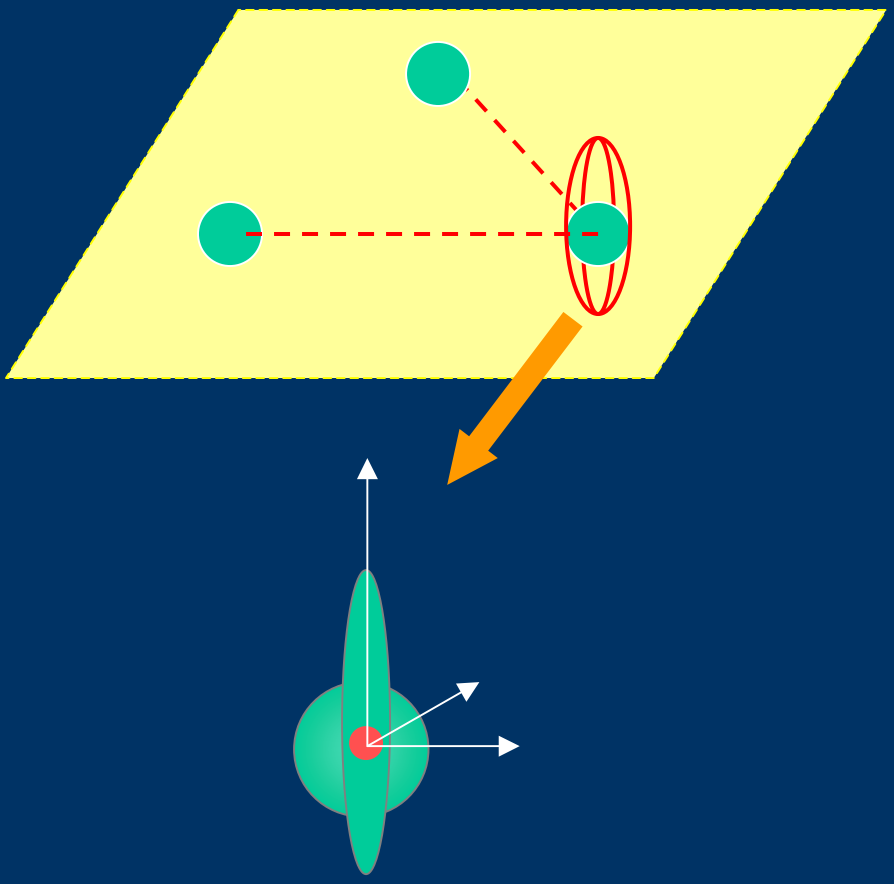
Surface Normal Inference from Unoriented Tokens

- Three unoriented tokens define plane, but voting operates pairwise
- Two tokens define a straight line and the voter casts a *plate vote*

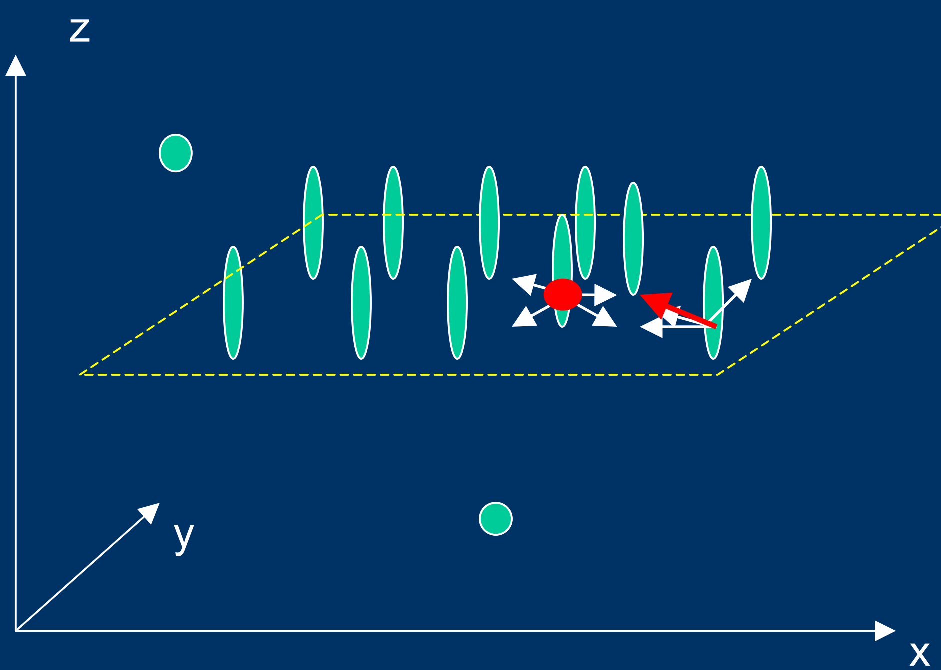


Surface Normal Inference from Unoriented Tokens

- Three unoriented tokens define plane, but voting operates pairwise
- Two tokens define a straight line and the voter casts a *plate vote*
- Accumulation of plate votes with a common axis results in *salient stick component*



First Order Voting



• Tokens in the interior of a structure receive first order votes from all directions

• Tokens at boundaries receive first order votes from one side of a half-space

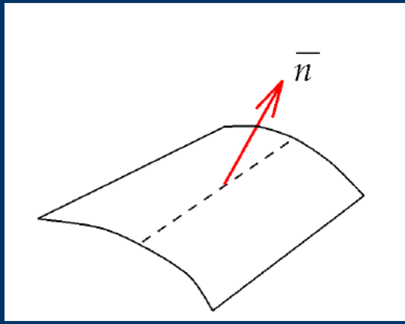
Interpretation of Resulting Tensors

Structure Type	**Saliency**	**Tensor Orientation**	**Polarity**	**Polarity orientation**
Surface inlier	High λ_1- λ_2	Normal: $\mathbf{e}_1$	Low	-
Surface boundary	High λ_1- λ_2	Normal: $\mathbf{e}_1$	High	Normal to $\mathbf{e}_1$ and boundary
Curve inlier	High λ_2- λ_3	Tangent: $\mathbf{e}_3$	Low	-
Curve endpoint	High λ_2- λ_3	Tangent: $\mathbf{e}_3$	High	Parallel to $\mathbf{e}_3$
Volume inlier	High λ_3	-	Low	-
Volume boundary	High λ_3	-	High	Normal to bounding surface
Junction	Distinct locally max λ_3	-	Low	-
Outlier	Low	-	Indifferent	-

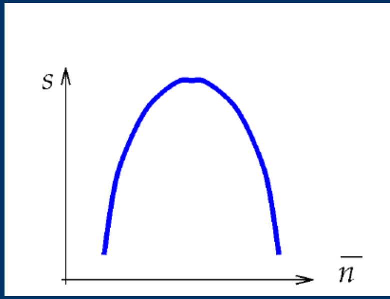
Structure Inference in 3-D

- Surfaces and curves extracted as local maxima of surface and curve saliency
- Perform **Dense Vote**, where votes are collected at all locations
- Detect *zero-crossings* of first derivative of saliency
- Extract surfaces using *Marching Cubes*
- Extract curves similarly

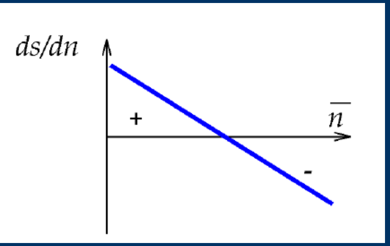
Surface Extraction



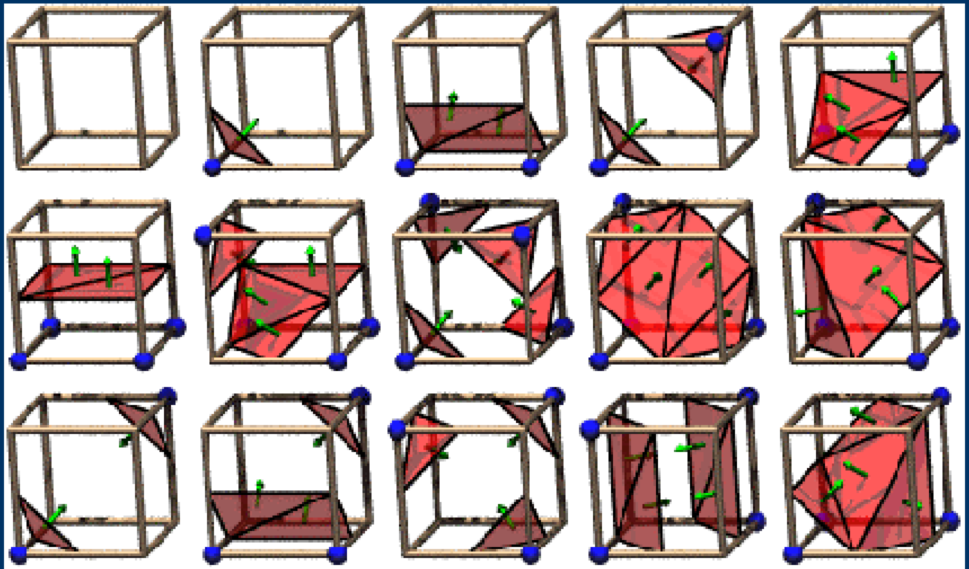
Surface Patch



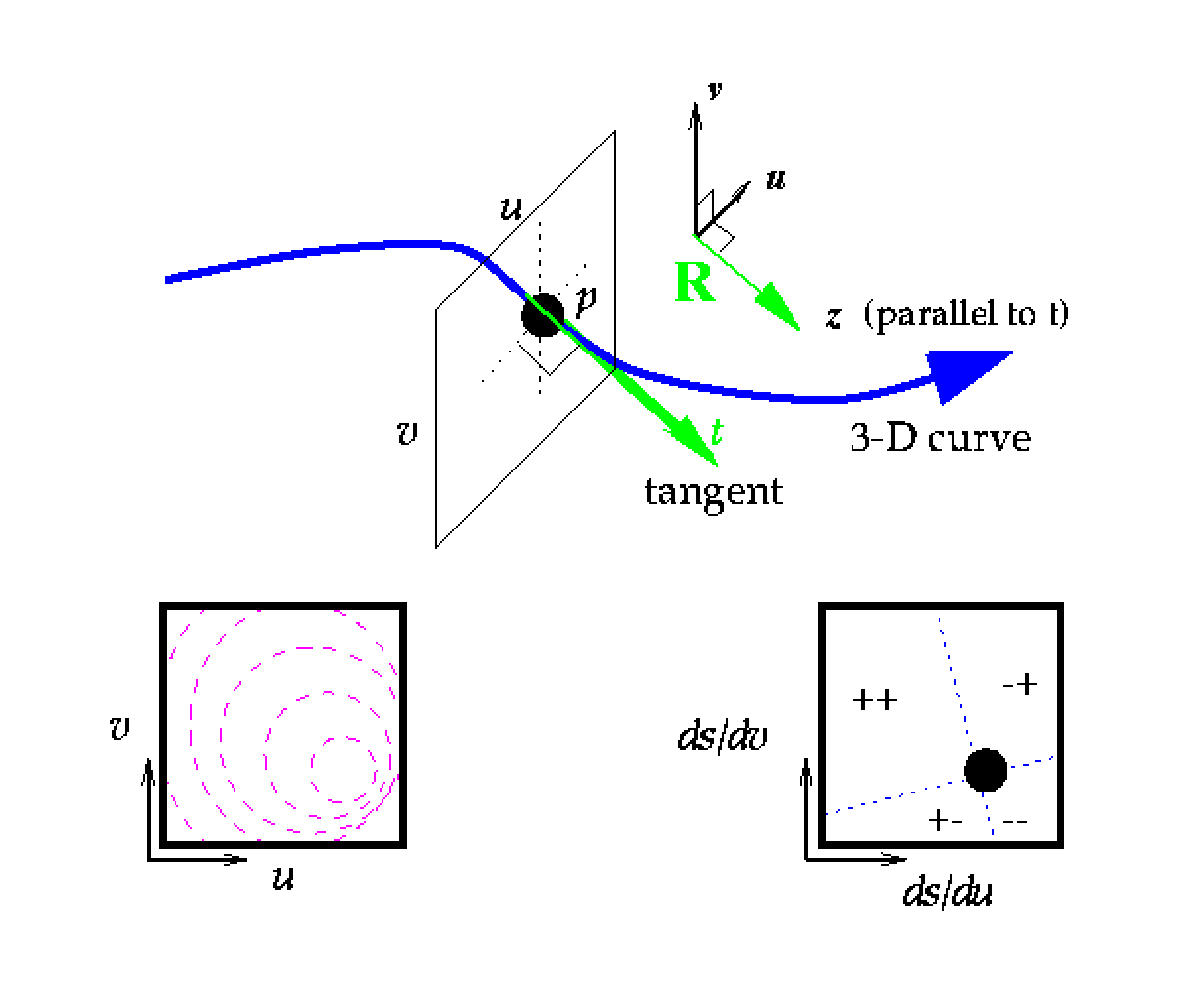
Surface Saliency along normal direction



First derivative of Surface Saliency

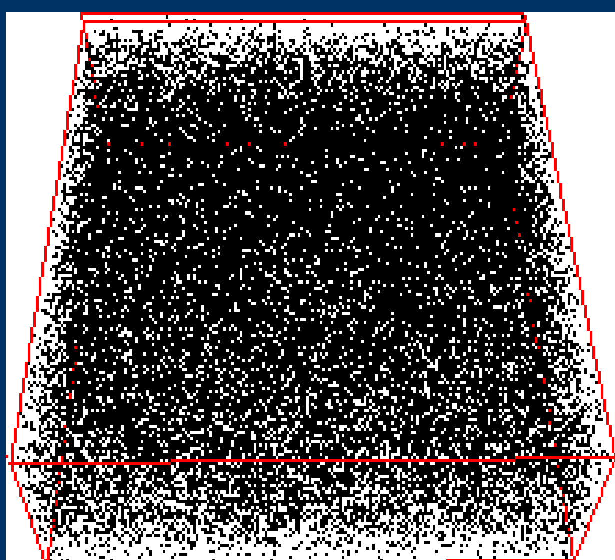


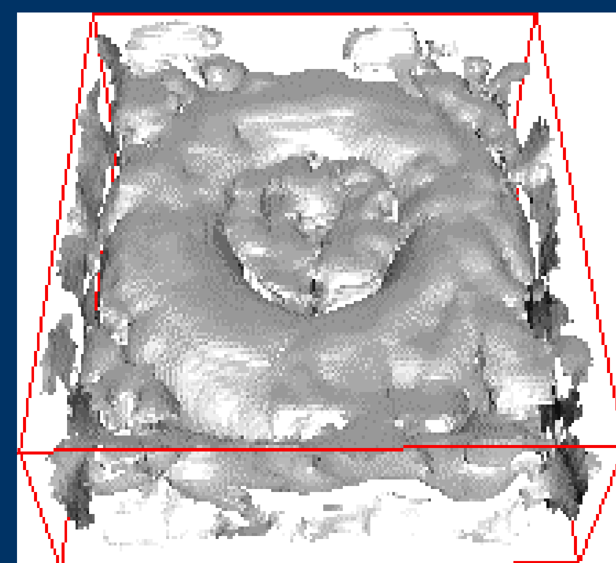
Curve Extraction



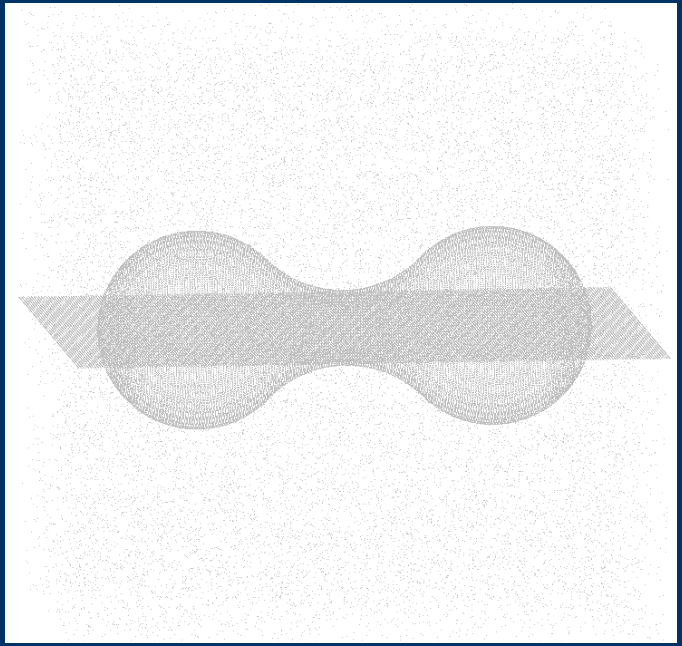
Graceful Degradation with Noise

1200% noise

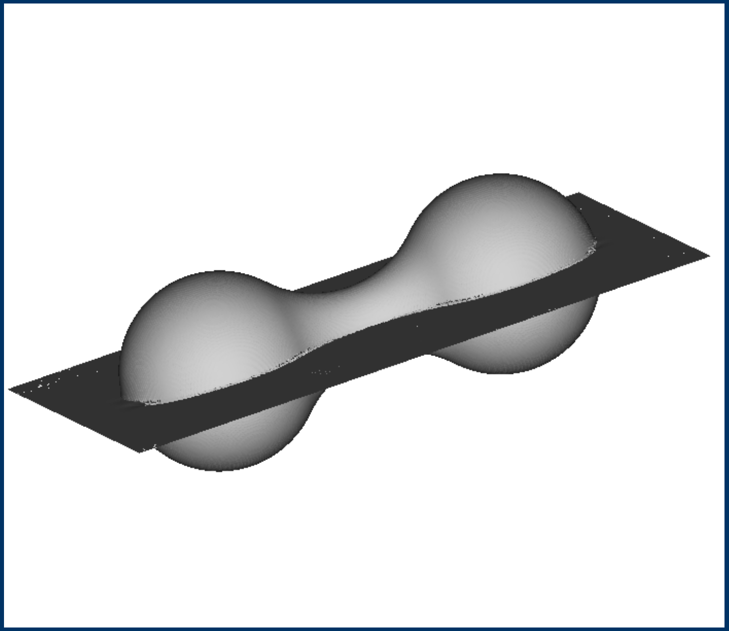




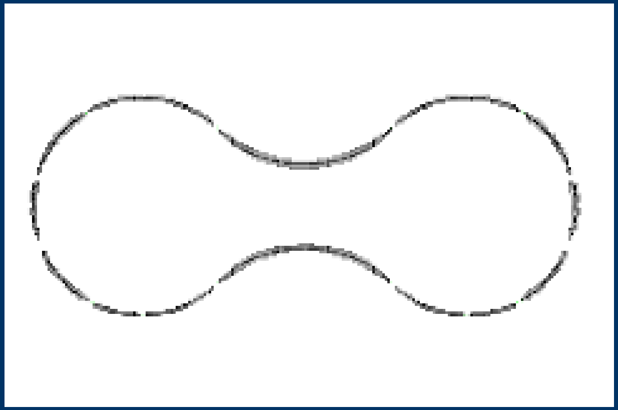
Examples in 3-D



Input

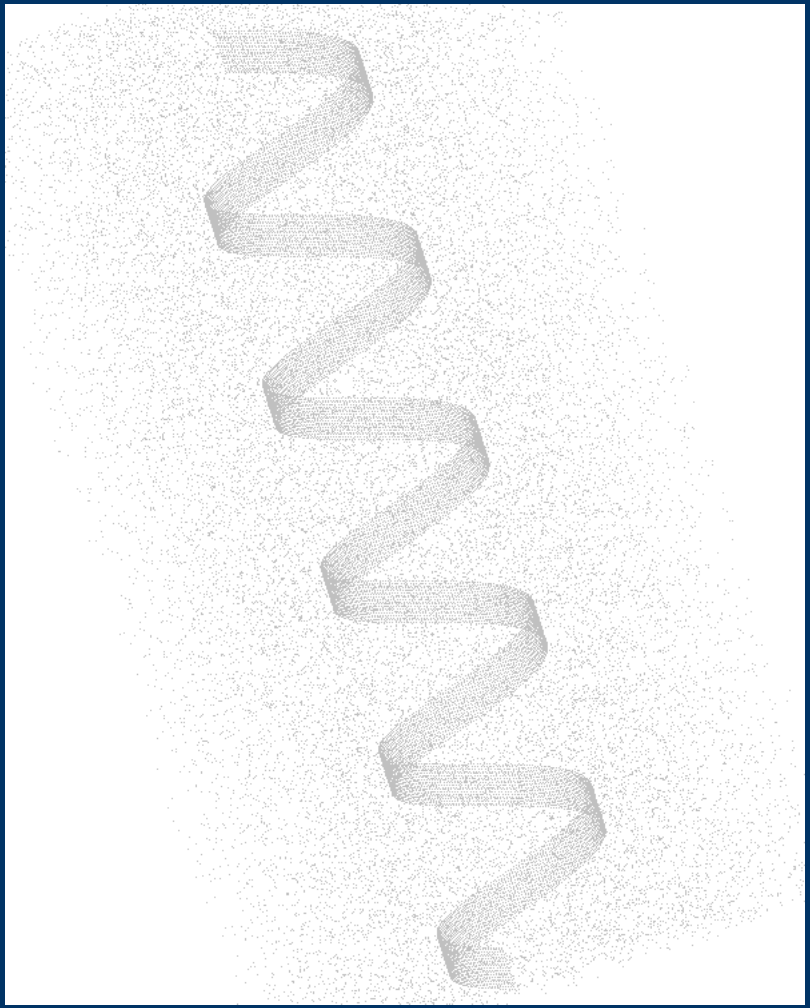


Surfaces

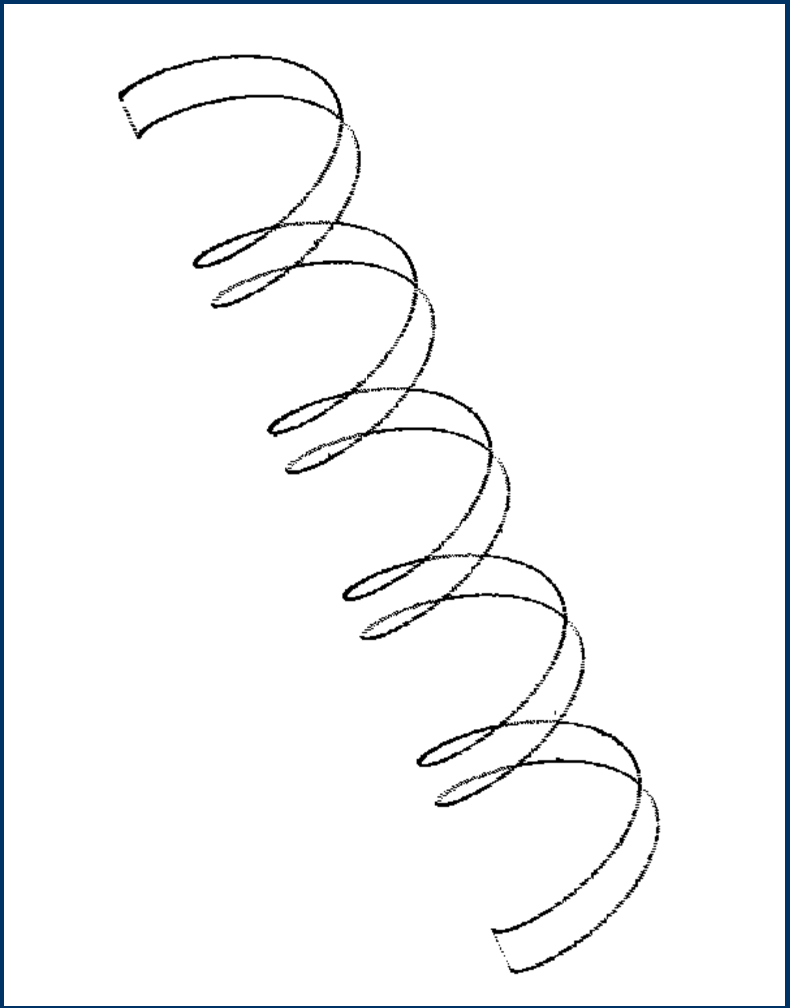


Surface Intersections

Examples in 3-D

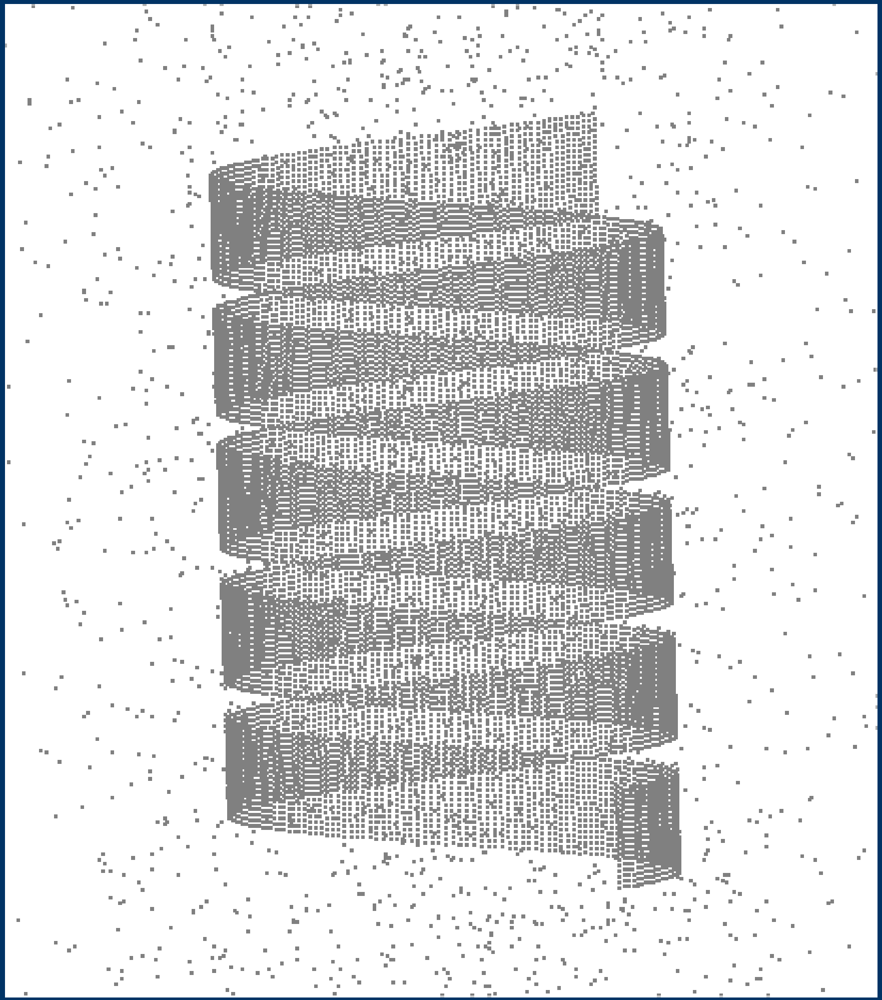


Input

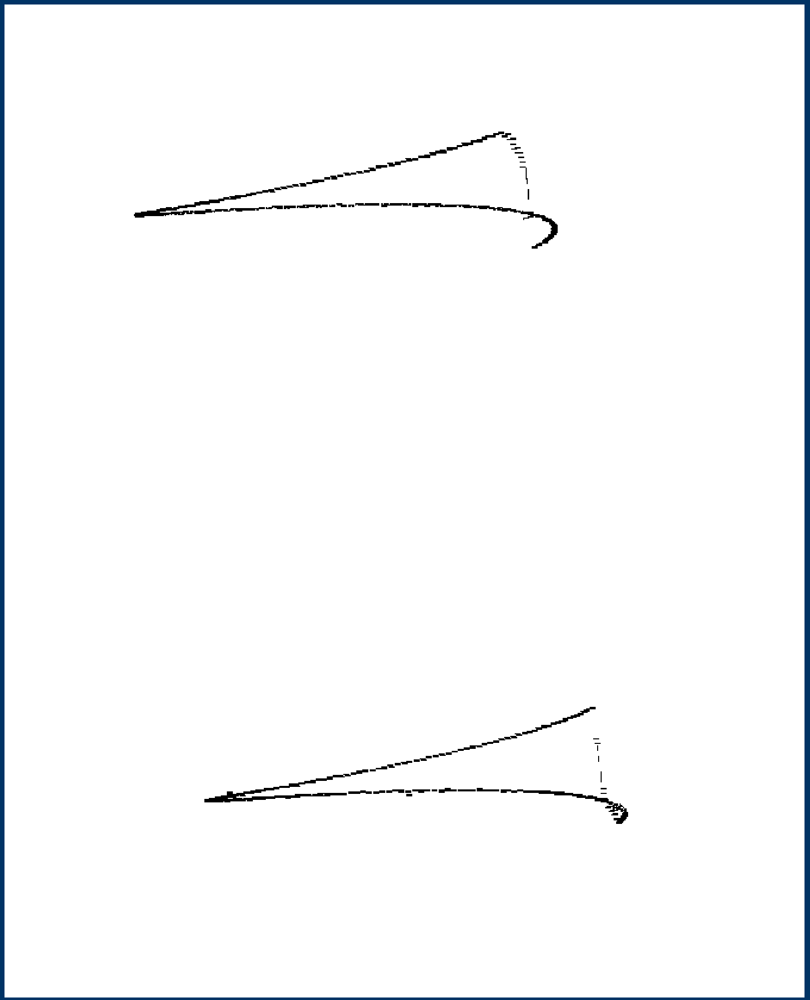


Surface Boundaries

Examples in 3-D

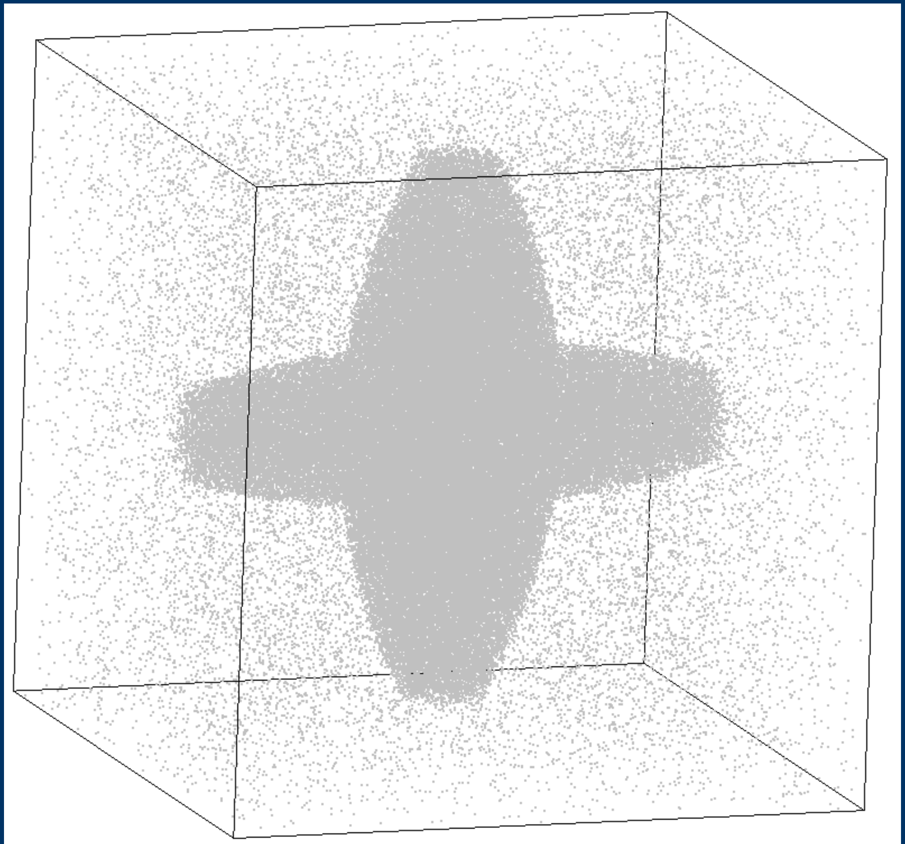


Input

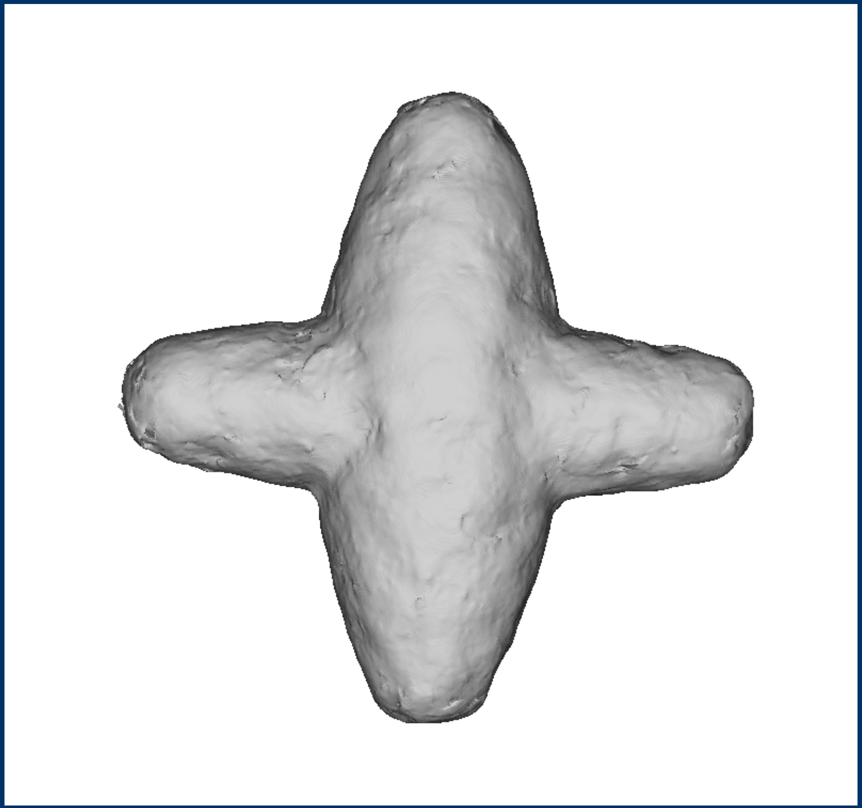


Surface Boundaries

Examples in 3-D



Input



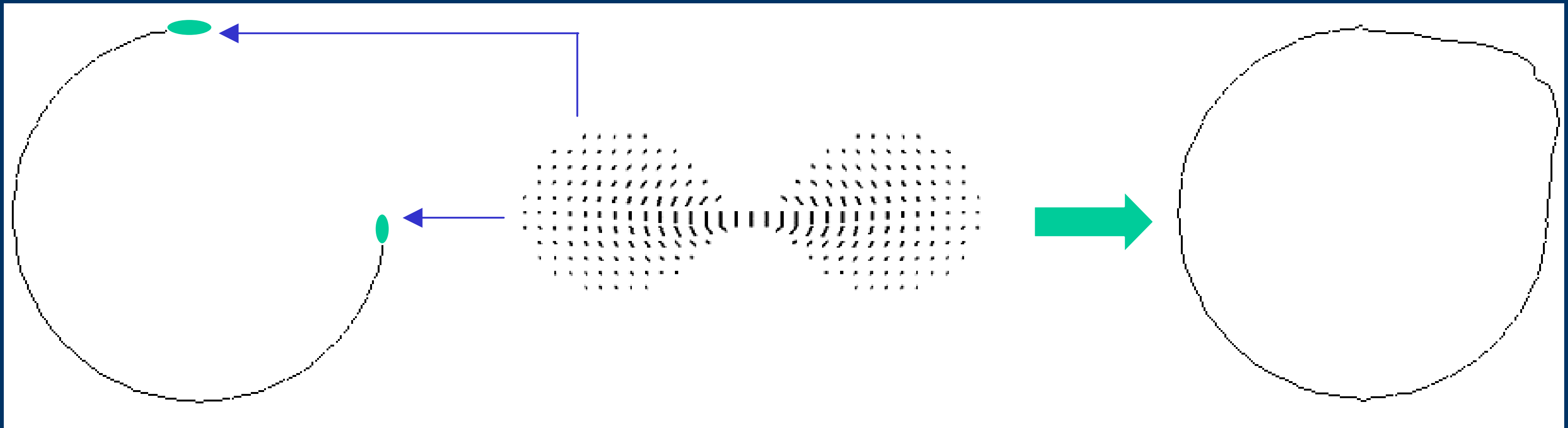
Volume Boundaries

Curvature

- Useful shape descriptor
  - Viewpoint invariant
  - Can guide reconstruction
- Accurate quantitative estimation is difficult
  - Unavoidable outliers
  - Unstable second order properties

Why Curvature?

voting with regular field



a circle will not be produced

Our approach on Curvature Estimation

- No partial derivative computation
- No local surface fitting
- Zero curvature is handled uniformly
- Robust to outliers
- Sign and direction of principal curvatures are accurately estimated

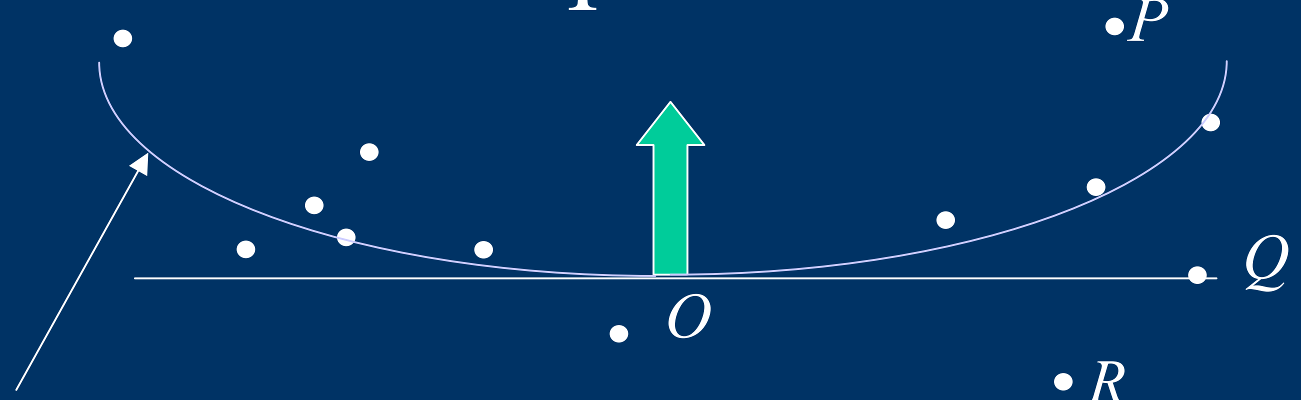
Two Estimations

- Sign of principal curvature
- Principal direction

Sign of Principal Curvature

- In 3-D each input site is labeled as locally
  - planar
  - elliptic
  - parabolic
  - hyperbolic, an outlier, or a discontinuity
- Then, we know which **side,** w.r.t. the estimated stick, the surface should **locally curve to**

Vote Representation



Smooth connection prescribed by the stick kernel

	P (above)	*Q* (on)	*R* (below)
vote **direction**	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$	$\begin{bmatrix}0\\1\end{bmatrix}$	$-\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

vote **strength** inversely proportional to distance from *O*

Vote collection at *O*

- compute **mean** μ
  - preferred side

$$M = \begin{bmatrix} M_x \\ M_y \end{bmatrix} = \frac{1}{n} \sum_{P \in nbhd(O)} \vec{v}_P, \mu = \frac{M_x}{M_y}$$

- compute total **variance** Σ
  - deviation from "mean"

$$S = \frac{1}{n-1} BB^T, \sum = trace(S)$$

- $\| \mu \|, \Sigma$ together indicate which side w.r.t. the input stick the curve should curve to

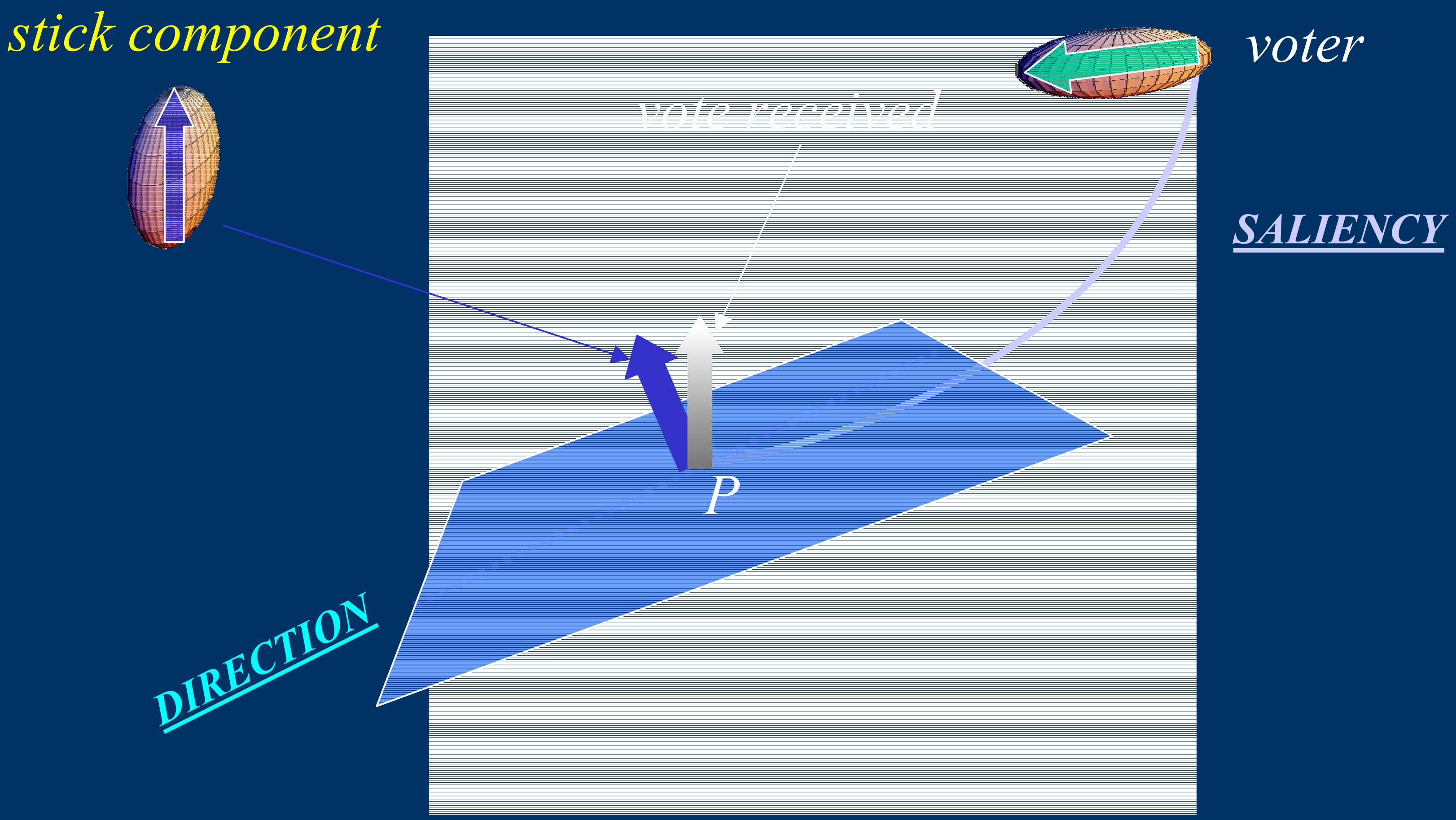
Geometric Interpretation

$\lvert \mu \rvert \approx 0$?	$\Sigma \approx 0$?	
✓	✓	**planar**
✗	✓	**elliptic**
✓	✗	**hyperbolic**
✗	✗	**parabolic**

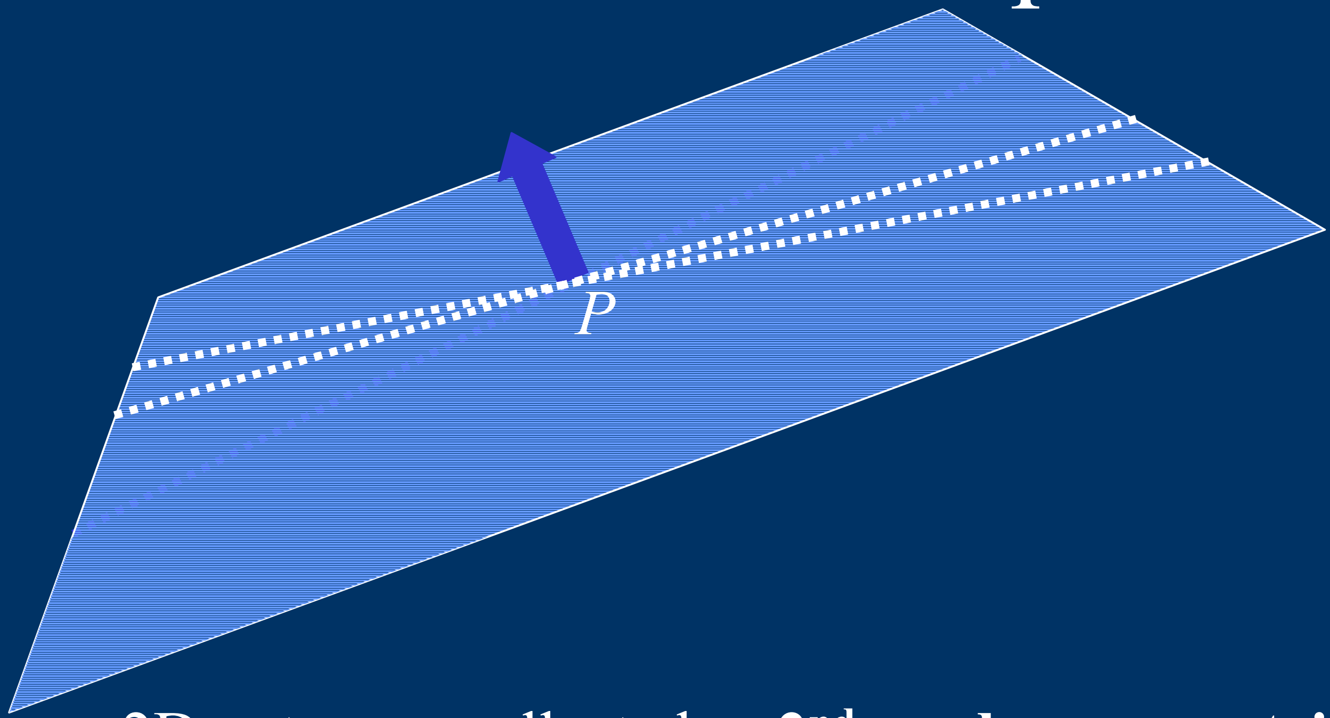
Two Estimations

- Sign of principal curvature
- Principal direction

Principal Direction



Vote Collection & Interpretation



2D votes are collected as **2nd oreder symmetric tensors**

V_{max} = **maximum** direction

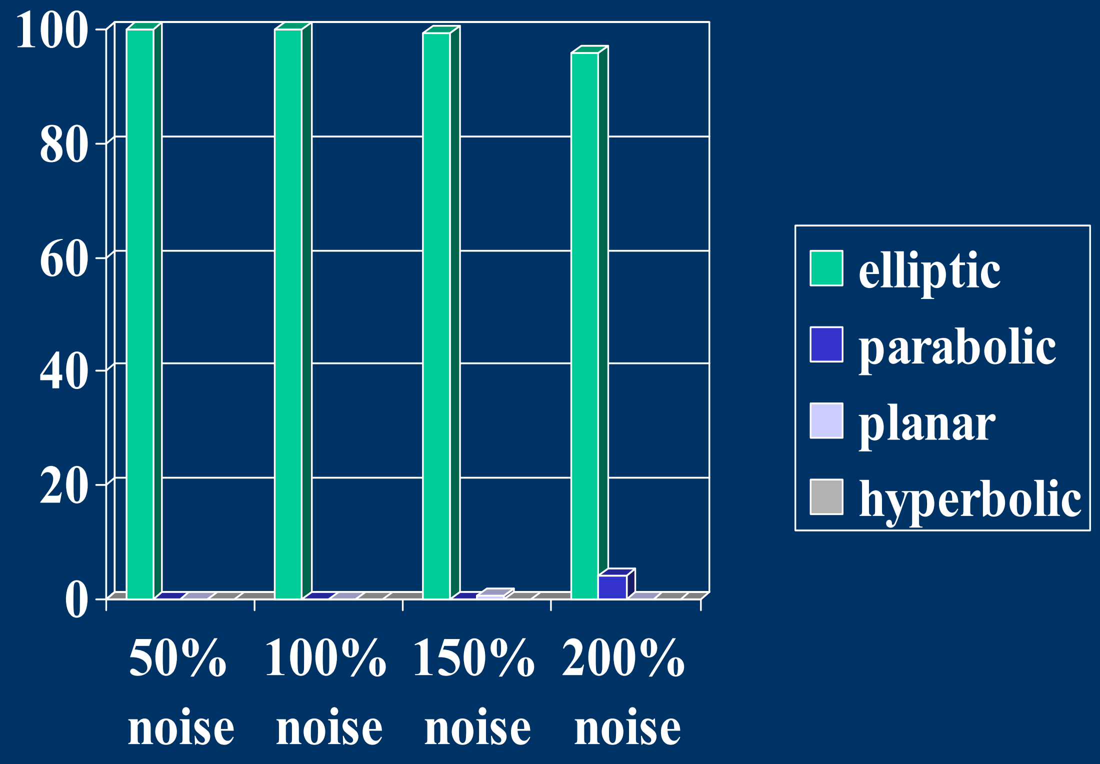
V_{min} = **minimum** direction

Curvature-Based Stick Kernel

- hyperbolic
  - original
- planar
  - very thin
  - more decay with high curvature
- parabolic or elliptic
  - one side of stick

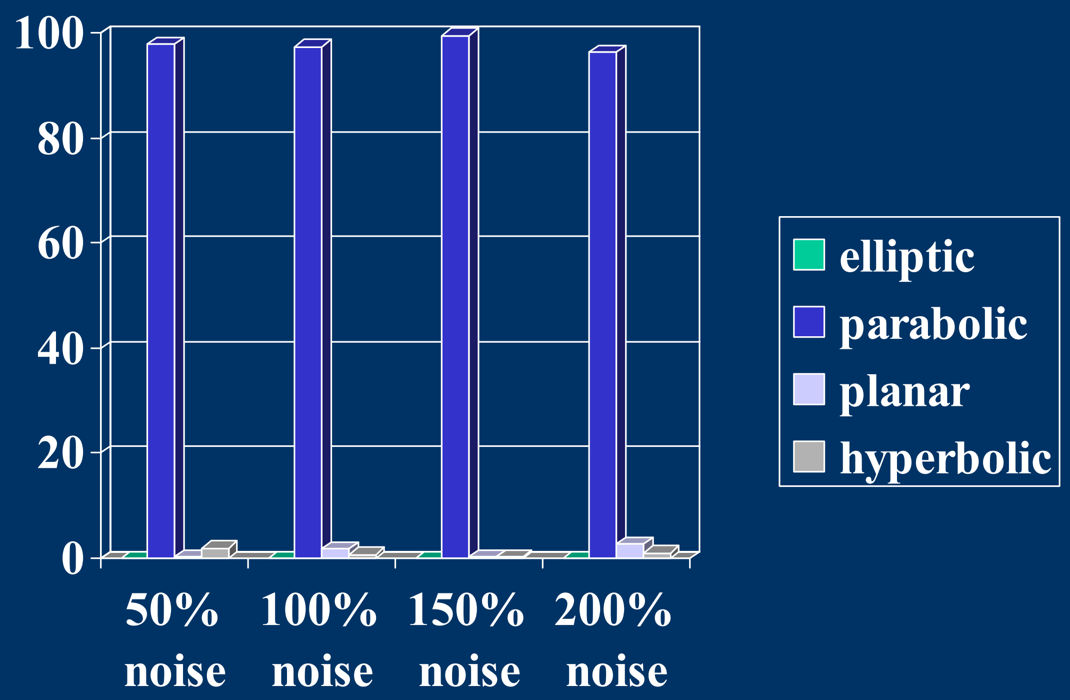
Accuracy of Labeling

Sphere (489 points)



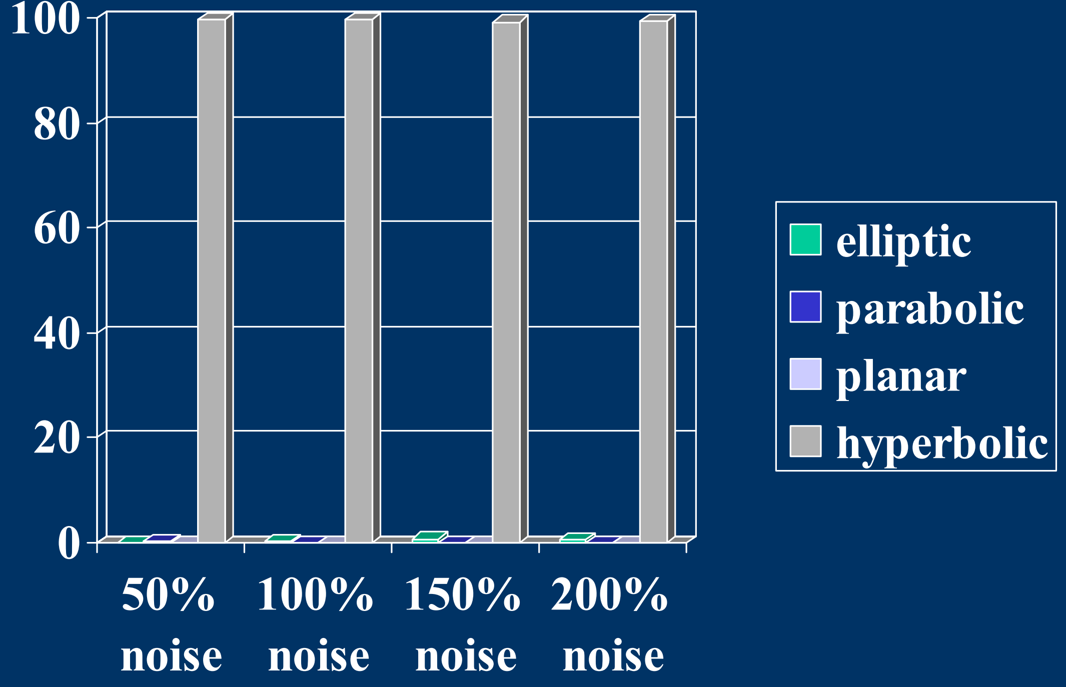
Accuracy of Labeling

Cylinder (3844 points)

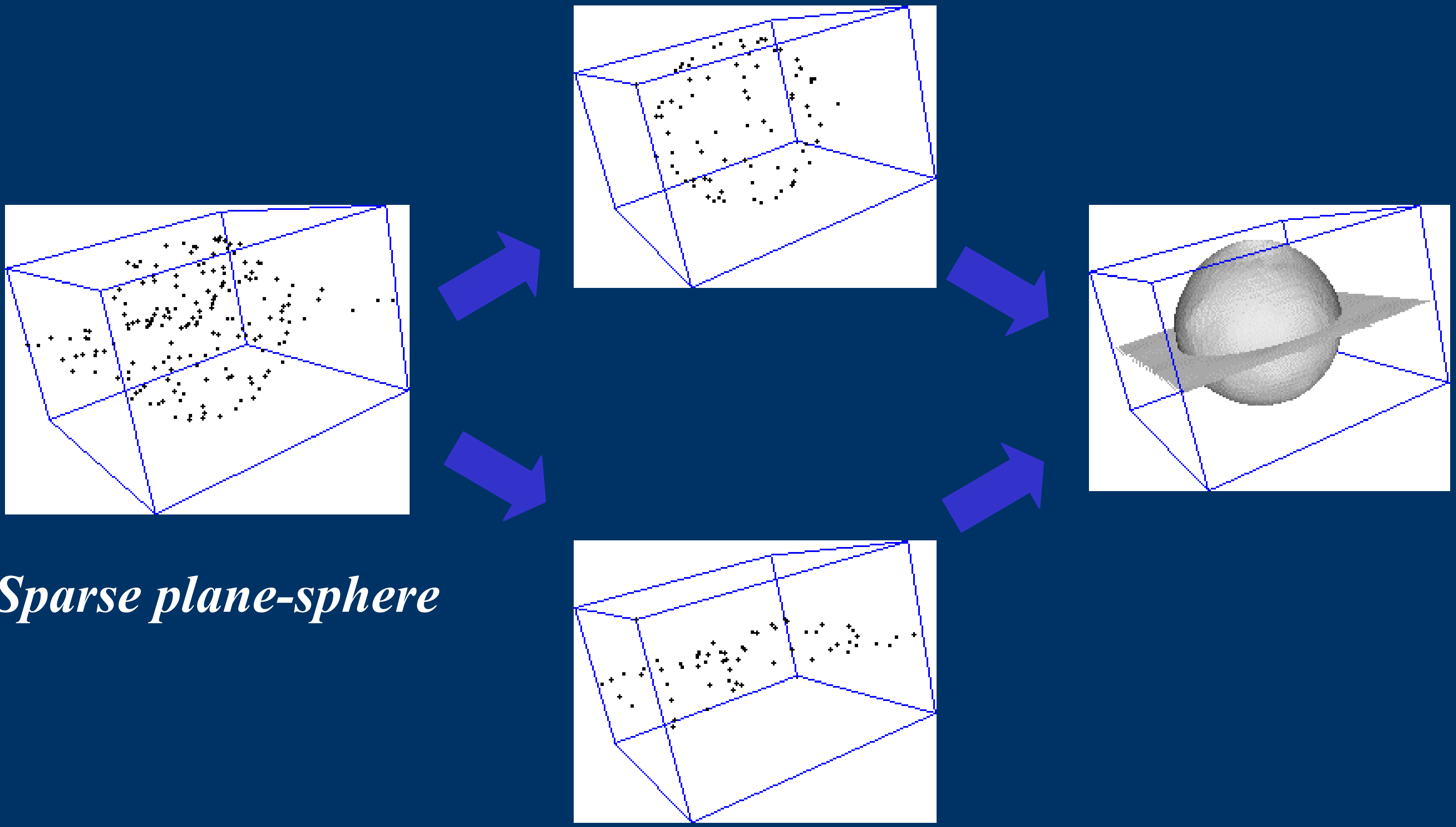


Accuracy of Labeling

Saddle (605 points)

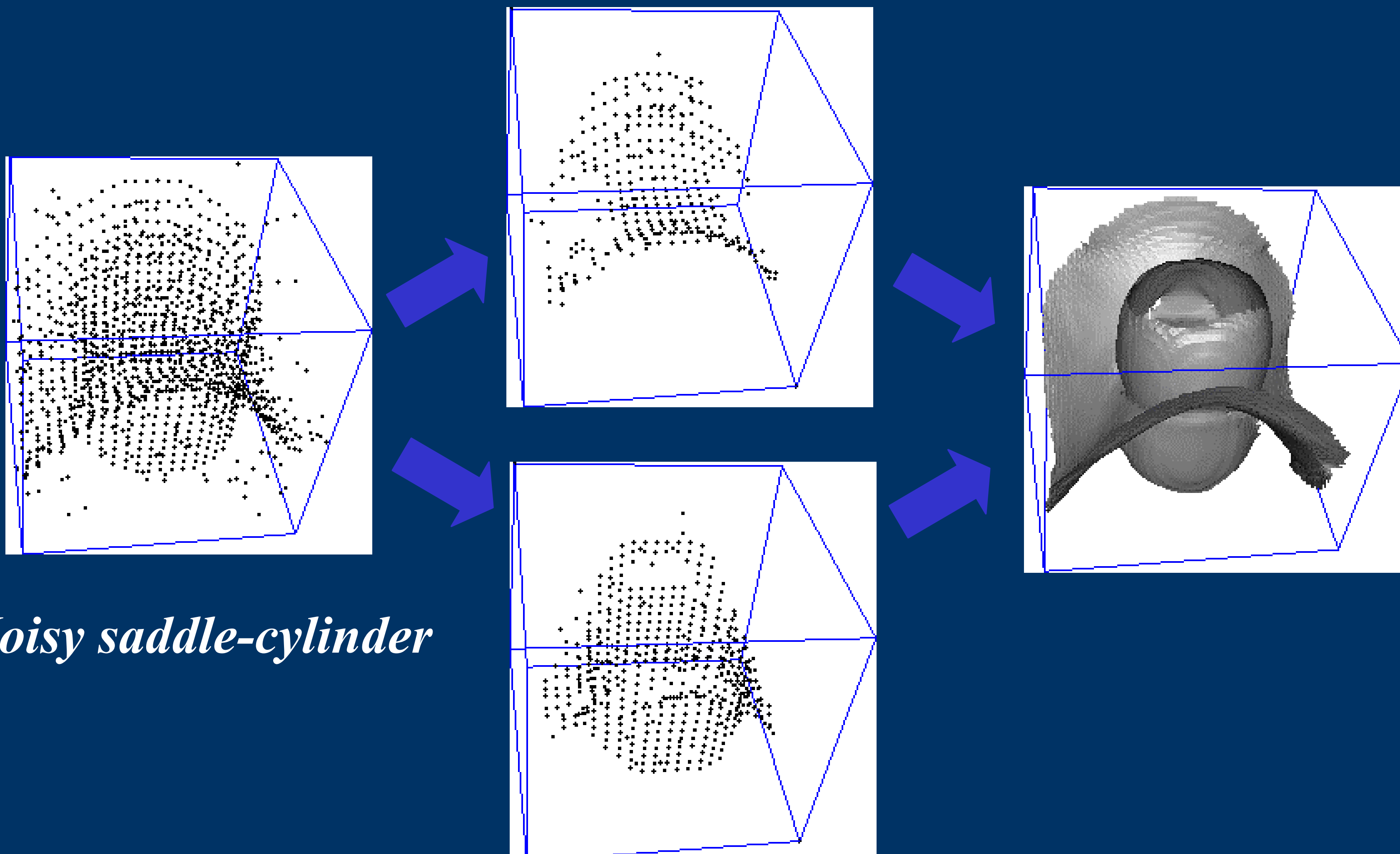


Grouping by Curvature



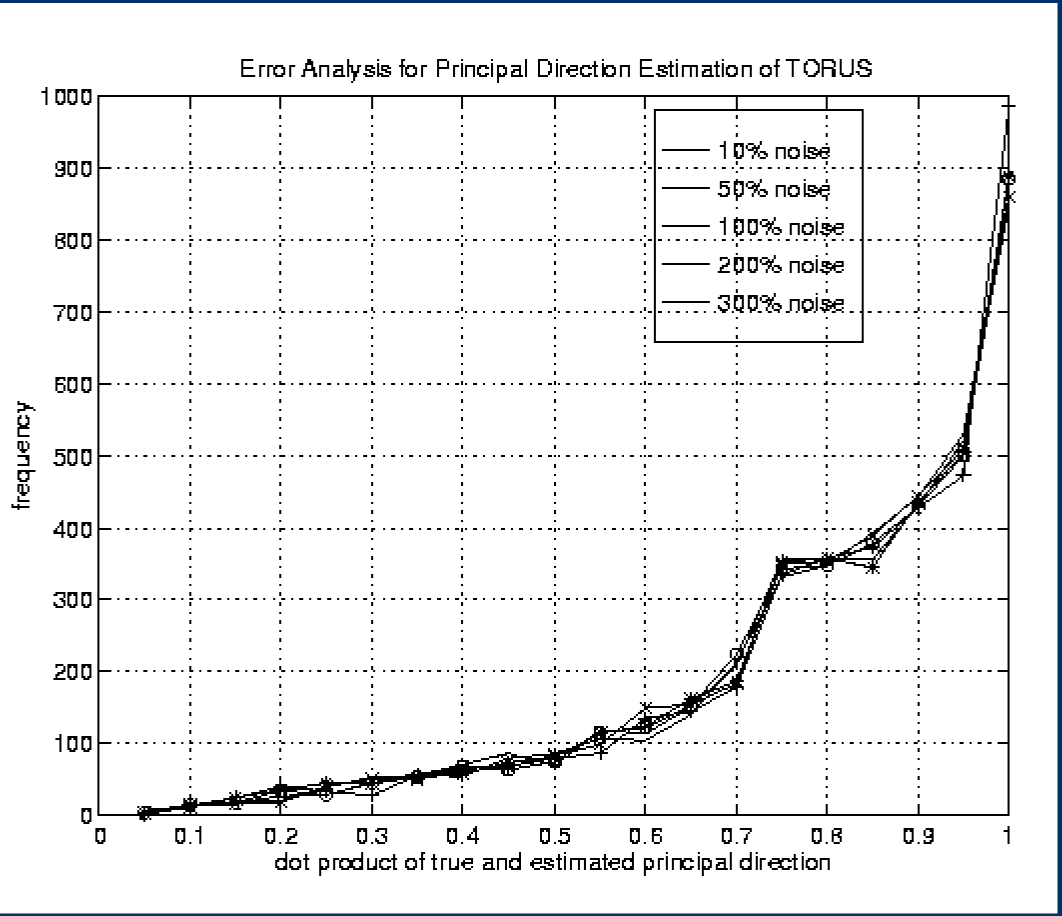
Sparse plane-sphere

Grouping by Curvature



Noisy saddle-cylinder

Robustness of Principal Curvature Estimation



Estimated principal direction does not adversely affected by noise

Overview

- Related Work
- Tensor Voting in 2-D
- Tensor Voting in 3-D
- Tensor Voting in N-D
- Application to Vision Problems
- Stereo
- Visual Motion
- Binary-Space-Partitioned Images
- 3-D Surface Extraction from Medical Data
- Epipolar Geometry Estimation for Non-static Scenes
- Image Repairing
- Range and 3-D Data Repairing
- Video Repairing
- Luminance Correction
- Conclusions

Tensor Voting in N-D

Direct generalization from 2-D and 3-D cases

– Tensors become second order, N-dimensional, symmetric, non-negative definite
– Polarity vectors become N-D vectors
– There are N+1 structure types (0-D junction to N-D hyper-volume)
– N second order and N first order fields are required

Voting Fields in N-D

- Vote generation from unit stick is the same
  - Voter, receiver and voting stick define a plane in any dimension
- Other fields can be derived as shown in previous sections

Applications in N-D

- Motion segmentation in 4-D space (x, y, v_x, v_y)
- Epipolar geometry estimation in 4-D Joint Image Space
- Affine motion parameter estimation in 4-D space
- Epipolar geometry estimation in 8-D space

Applications in N-D

- Voting in intensity / color space:
  - Image repairing
  - 3-D data repairing
  - Video repairing
  - Luminance correction

Issues in N-D

- Space must be Euclidean
  - Distances in voting space must be meaningful
- Data structures
  - Efficient search for neighbors
- Voting fields
  - Pre-computation becomes inefficient when grid positions are comparable to number of tokens