The Tensor Voting Framework

Gérard Medioni

University of Southern California

Chi-Keung Tang

Hong Kong University of Science and Technology

Motivation

- Computational framework to address a wide range of computer vision problems
- Computer Vision attempts to infer scene descriptions from one or more images
  - Primitives and constraints might vary from problem to problem
  - Many problems can be formulated as *perceptual organization* problems in an appropriate space

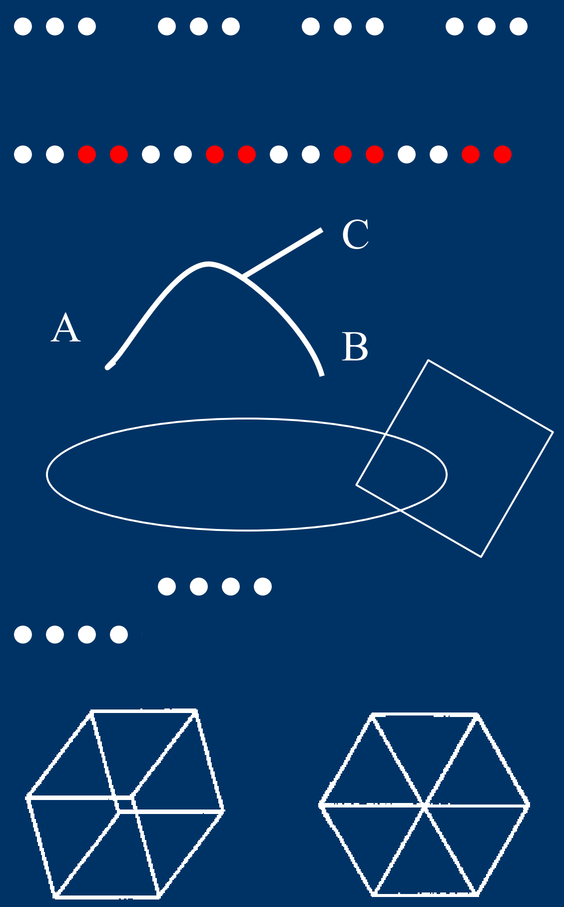
Need for Constraints

- Since the problem has infinite number of solutions, constraints need to be imposed
- Constraints may be
  - non-consistent
  - difficult to implement

Perceptual Organization

Gestalt principles:

- Proximity
- Similarity
- Good continuation
- Closure
- Common fate
- Simplicity



The Smoothness Constraint

Matter is cohesive → Smoothness

Difficult to implement, as true "almost everywhere" only

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

Related Work

Regularization

- Computer vision problems are inverse problems and ill-posed
- Constraints needed to derive solution
- Can be formulated as optimization

➢ Selection of objective function is not trivial

➢ Iterative

Relaxation Labeling

- Problems posed as the assignment of labels to tokens
- Remove labels that violated constraints and iteratively restrict solution space
- Continuous, discrete, deterministic and stochastic implementations

➢ Iterative

Robust Methods

- Model fitting based on robust statistics of data
- Classification of data as inliers and outliers
- Deterministic: M-estimators, LMedS etc.
- Stochastic: RANSAC etc.
- Very robust to noise

➢ Can operate only with limited and known a priori models

Level Set Methods

- Solutions represented implicitly as zero-level iso-contours or iso-surfaces of multivariate functions
- Evolve according to optimization criterion

➢ Sensitive to initialization

➢ Iterative

Clustering

- Group or partition the data according to affinity measures
  - Affinities encoded as edges of graph
- Partition the data by cutting the graph in a way that results in minimum disassociation between clusters (global decision)
  - Generalized eigenvalue problem
- Stochastic variants also exist

Structural Saliency

- Structural Saliency is a property of the structure as a whole
  - Parts of the structure are not salient in isolation
- Shashua and Ullman defined saliency measure based on proximity and curvature variation
- Large number of methods from Computer Science and Neuroscience
- Based on local interactions between tokens

➢ Saliency defined as scalar

Grouping with Cooperative and Inhibitive Fields

- Grossberg, Mingolla, Todorovic: Boundary Contour System and Feature Contour System
- Heitger and von der Heydt: computational model of Neural Contour Processing
- Williams and Jacobs: Stochastic Completion Fields
- Other recent techniques using fields or kernels that facilitate feature cooperation and inhibition have been reported

Grouping with Cooperative and Inhibitive Fields

- Similarities with Tensor Voting:
  - Influence decays with distance and curvature
  - Gaussian attenuation
  - 8-shaped fields
  - Circular arcs as smooth paths between tokens
- Representation with tensors is richer
- Tokens of different structure types can be simultaneously processed and interact with each other within the Tensor Voting Framework

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 2-D

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

The Tensor Voting Framework

- *Data Representation:* Tensors
- *Constraint Representation:* Voting fields
  - enforce smoothness
- *Communication:* Voting
  - non-iterative
  - no initialization required

Desirable Properties of the Representation

- Local
- Layered
- Object-centered

Our Approach in a Nutshell

- Each input site propagates its information in a neighborhood
- Each site collects the information cast there
- Salient features correspond to local extrema

Properties of Tensor Voting

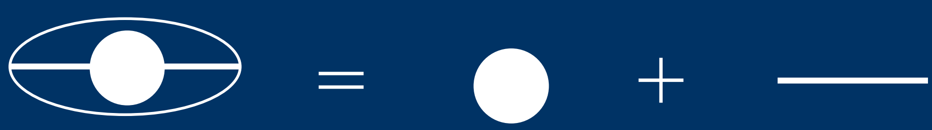
- Non-Iterative
- Can extract all features *simultaneously*
- One parameter (scale)
- Non-critical thresholds
- Efficient

Second Order Symmetric Tensors

- Second order, symmetric non-negative definite tenosrs
- Equivalent to:
  - Ellipse
    - Special cases: *"ball" and "stick"* tensors
  - 2x2 matrix

$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T =$$

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

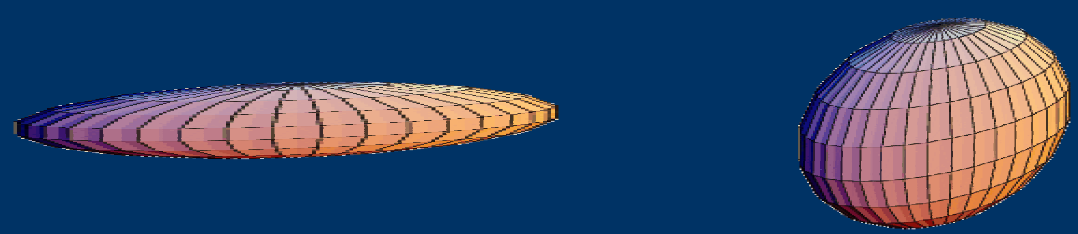


$$\begin{bmatrix} a^2 + b^2 & a^2 \\ a^2 & a^2 \end{bmatrix} = \begin{bmatrix} a^2 & a^2 \\ a^2 & a^2 \end{bmatrix} + \begin{bmatrix} b^2 & 0 \\ 0 & 0 \end{bmatrix}$$

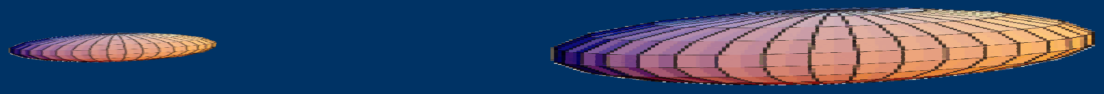
Second Order Symmetric Tensors

Properties captured by second order symmetric Tensor

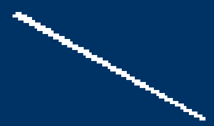
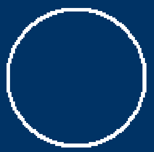
– shape: orientation certainty



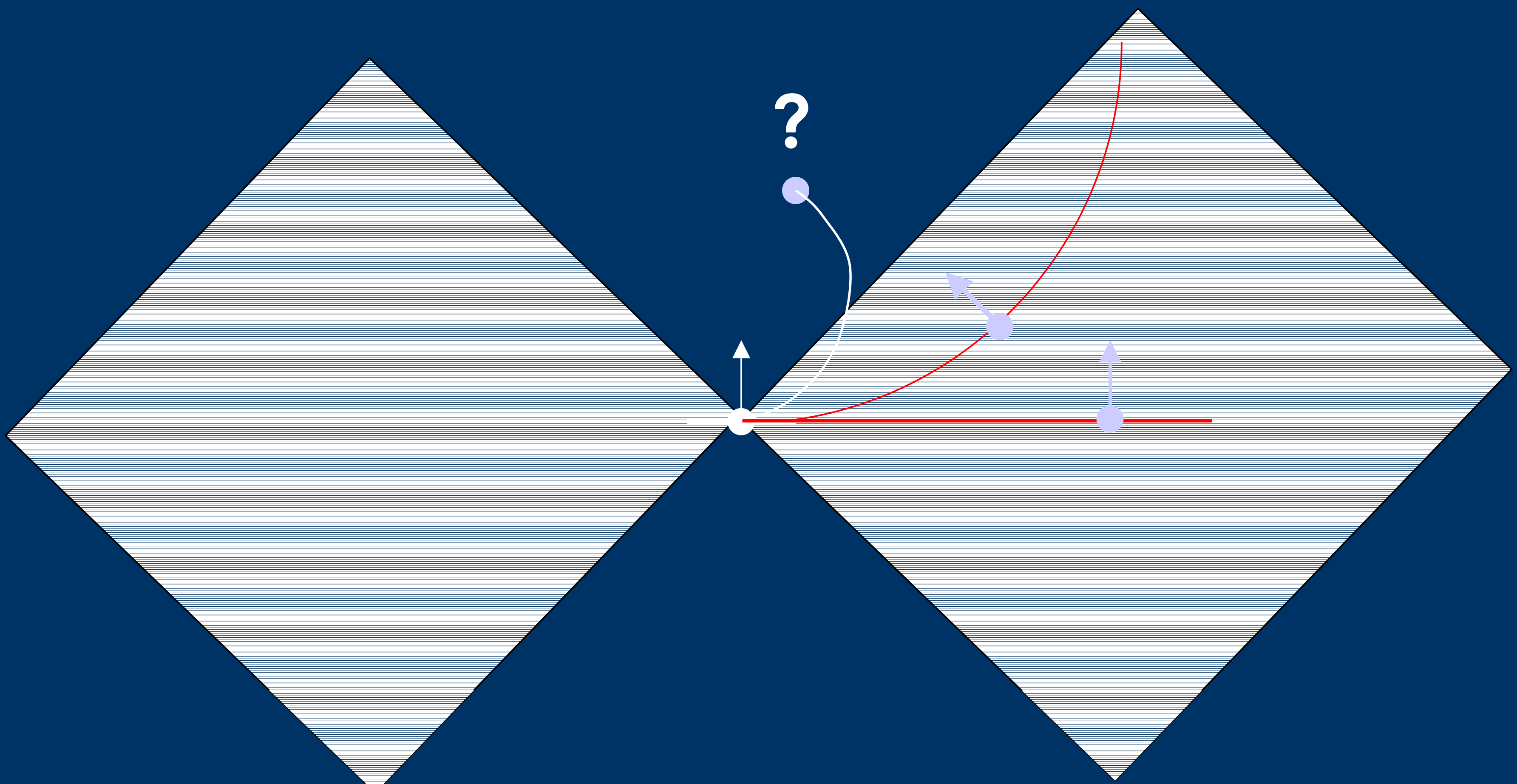
– size: feature saliency



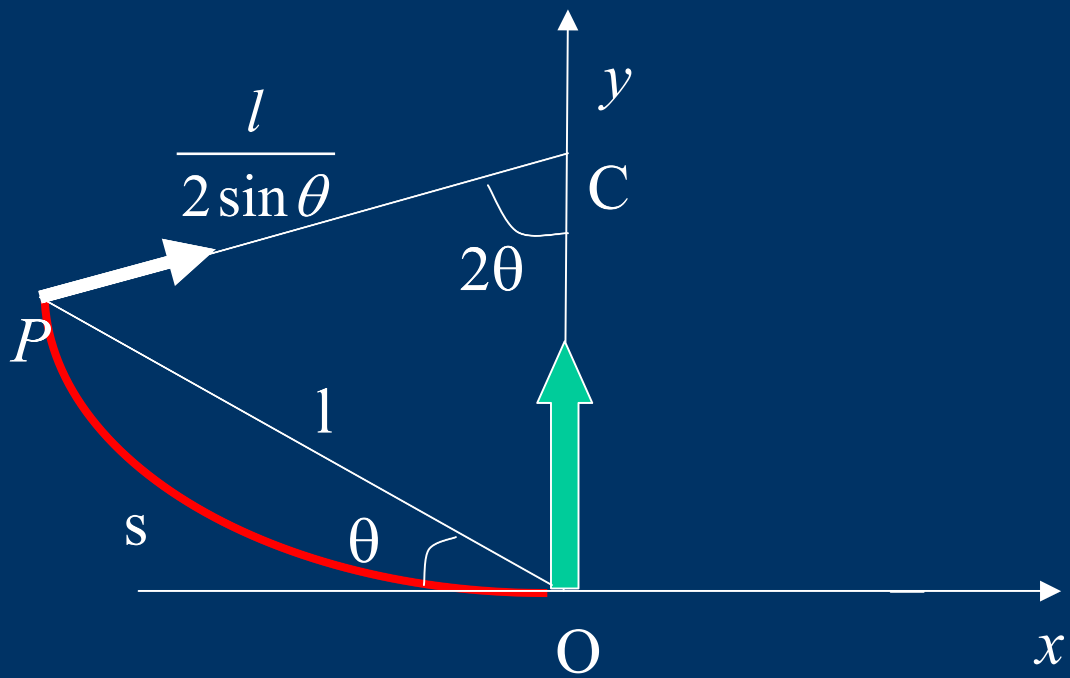
Representation with Second Order Symmetric Tensors

Input	Second Order Tensor	Eigenvalues	Quadratic Form
		$\lambda_1=1 \quad \lambda_2=0$	$\begin{bmatrix} n_x^2 & n_x n_y \\ n_x n_y & n_y^2 \end{bmatrix}$
		$\lambda_1=\lambda_2=1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Design of the Voting Field



Saliency Decay Function



$$S(s,\kappa) = e^{-\left(\frac{s^2 + c\kappa^2}{\sigma^2}\right)}$$

$$s = \frac{\theta l}{\sin\theta}$$

$$\kappa = \frac{2\sin\theta}{l}$$

- σ: scale of voting, s: arc length, κ: curvature
- Votes attenuate with length of smoothest path
- Straight continuation is favored over curved

Scale of Voting

- The Scale of Voting is the single critical parameter in the framework
- Essentially defines size of voting neighborhood
  - Gaussian decay has infinite extend, but it is cropped to where votes remain meaningful (e.g. 1% of voter saliency)

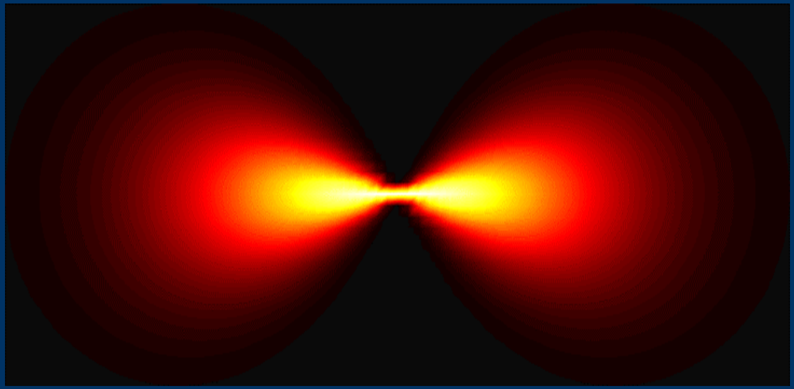
Scale of Voting

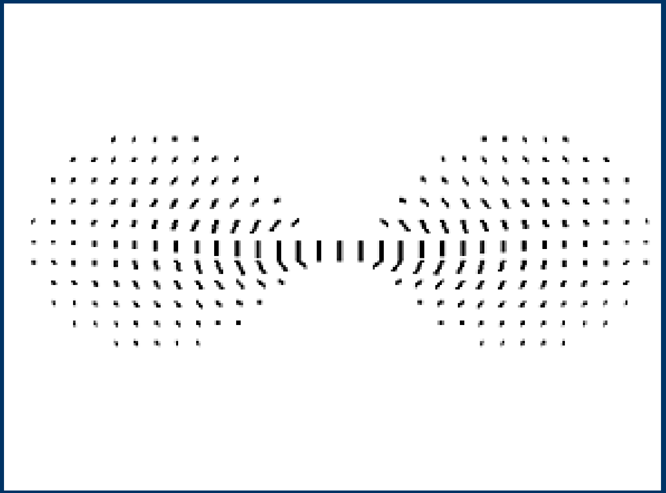
- The Scale is a measure of the degree of Smoothness
- Smaller scales correspond to small voting neighborhoods, fewer votes
  - Preserve details
  - More susceptible to outlier corruption
- Larger scales correspond to large voting neighborhoods, more votes
  - Bridge gaps
  - Smooth perturbations
  - Robust to noise

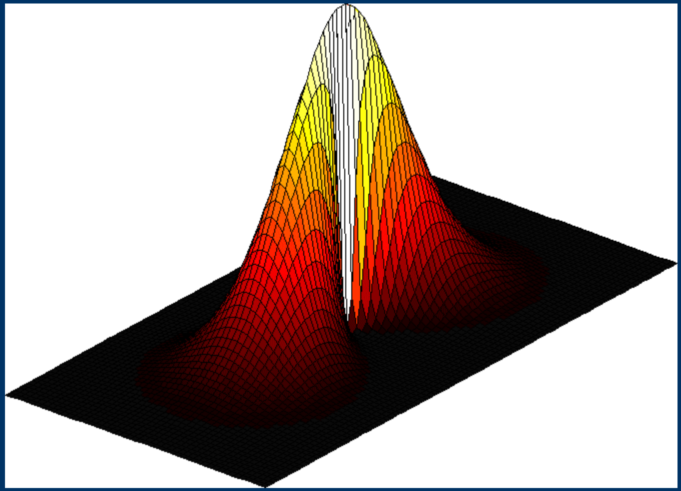
Scale of Voting

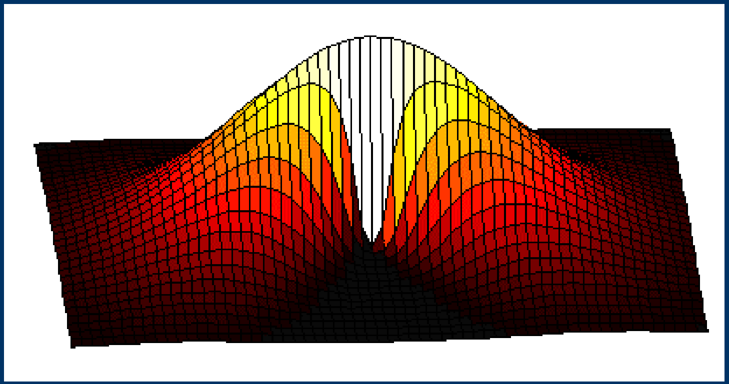
- Results are not sensitive to reasonable selections of scale
- Quantitative evaluations in the remainder

Fundamental Stick Voting Field







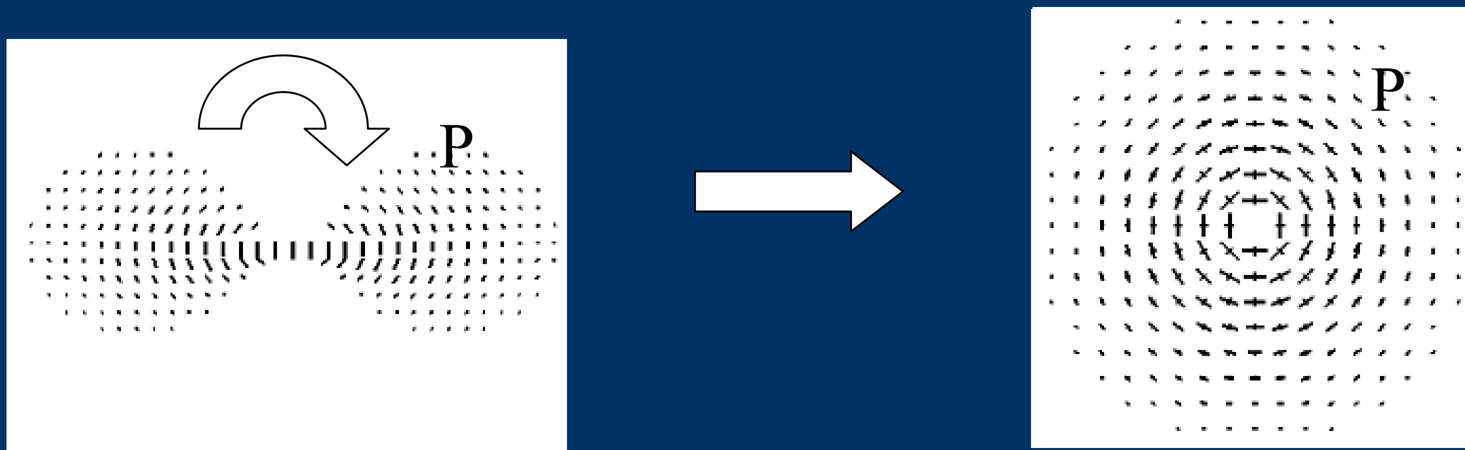


Fundamental Stick Voting Field

All other fields in *any* N-D space are generated from the *Fundamental Stick Field*:

- Ball Field in 2-D
- Stick, Plate and Ball Field in 3-D
- Stick, …, Ball Field in N-D

2-D Ball Field



S(P) **B**(P)

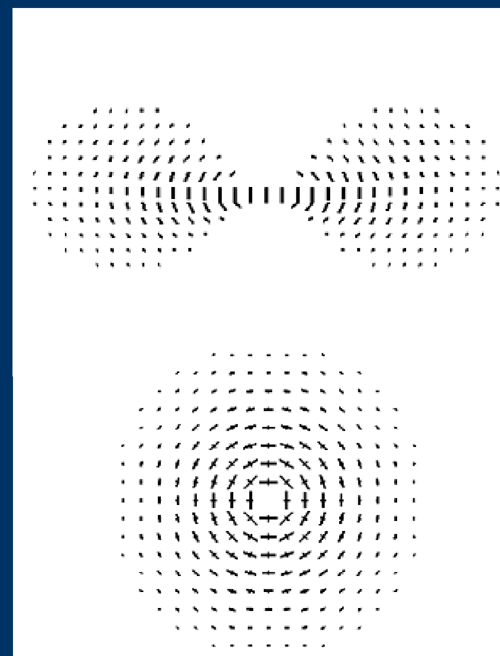
Ball field computed by integrating the contributions of rotating stick

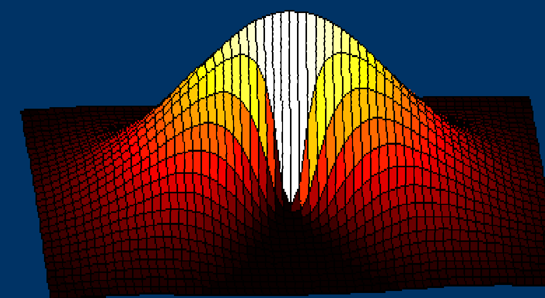
$$\mathbf{B}(P) = \int \mathbf{S}(P)\, d\theta$$

2-D Voting Fields

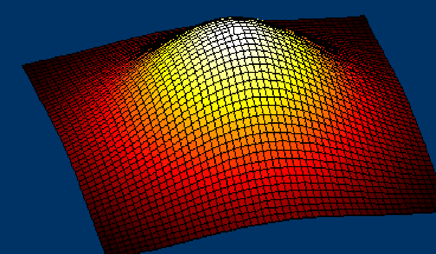
Each input site **propagates** its **information** in a **neighborhood**

 votes with





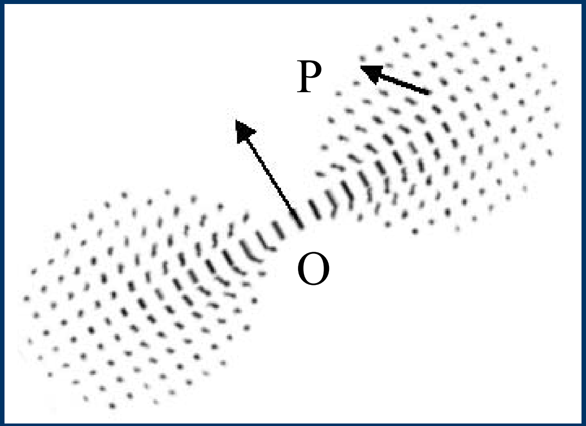
 votes with



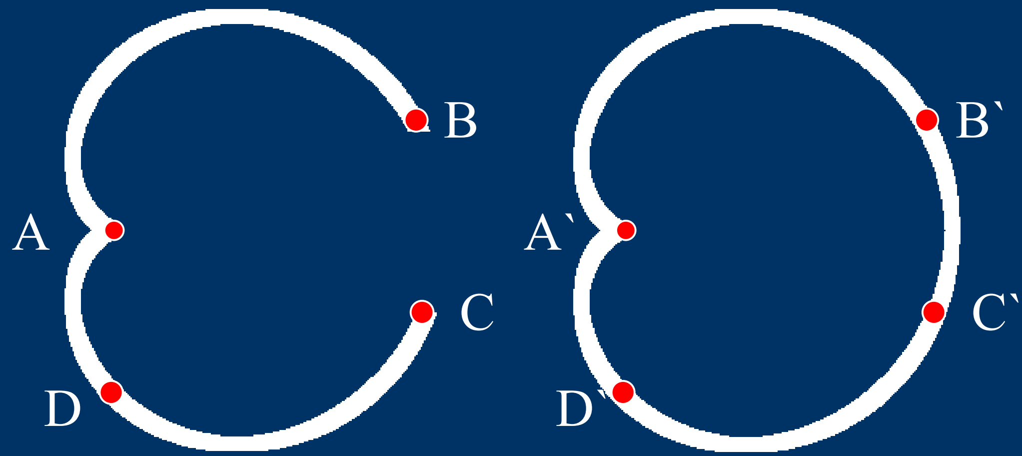
 votes with  + 

Voting

- Voting from a *ball* tensor is isotropic
  – Function of distance only
- The stick voting field is aligned with the orientation of the *stick* tensor



Need for First Order Information



Tensors
A: ball
B, C, D, E: stick

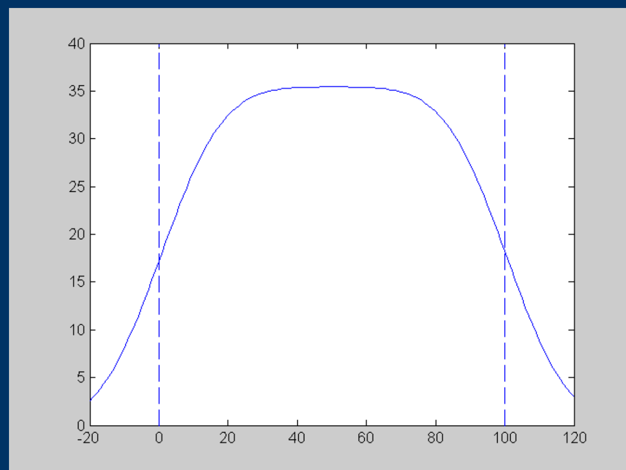
- Second order tensors are insensitive to *signed* orientation
- They cannot discriminate between interior points and terminations of perceptual structures

Polarity Vectors

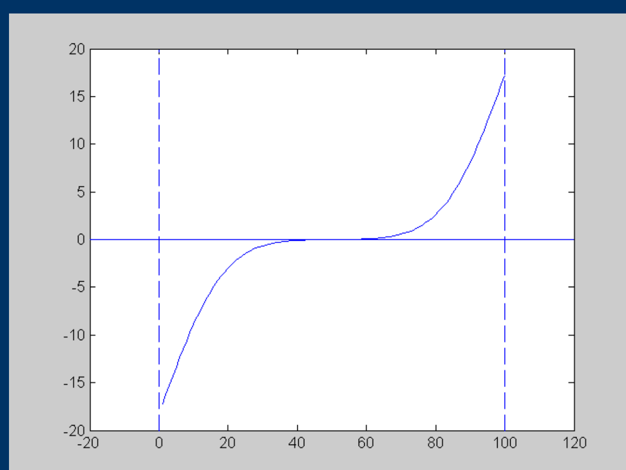
- Representation augmented with Polarity Vectors (first order tensors)
- Sensitive to direction from which votes are received
- Exploit property of boundaries to have all their neighbors on the same side of the half-space

Polarity

Input



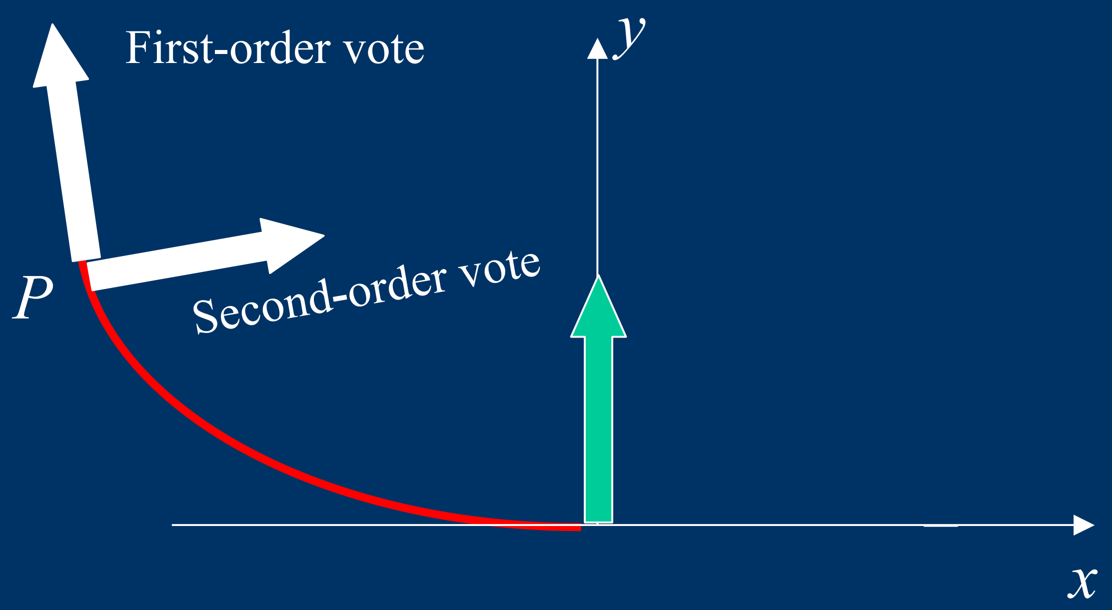
Saliency



Polarity

First Order Voting

- Votes are cast along the tangent of the smoothest path
- Vector votes instead of tensor votes
- Accumulated by vector addition



First Order Voting Fields

- Magnitude is the same as in the second order case

$$S(s,\kappa) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)}$$

- First-order Ball field can be derived from the first-order Stick Field after integration

First and Second Order Voting

- Both votes are based on the *second order information* (first order vector has to be initialized as zero)
- The second order tensor is decomposed into the Stick and Ball components
- Each component casts a first and second order vote

Vote Collection

Each site **collects** the information cast there

- By **tensor** addition (for second order votes):

$$V_{SO} = \sum \mathrm{V}_{\mathrm{i}}$$

- By **vector** addition (for first order votes):

$$V_{FO} = \sum v_i$$

Tensor Addition

Each site accumulates second order votes by tensor addition:

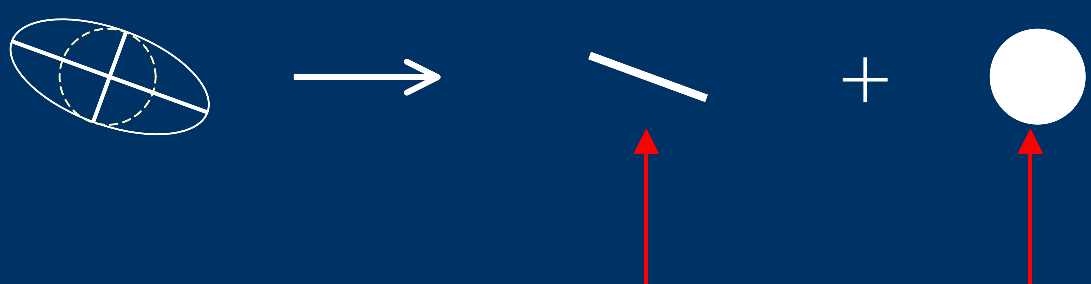
Results of accumulation are usually *generic tensors*

Second Order Vote Interpretation

Salient features correspond to local extrema

At each site

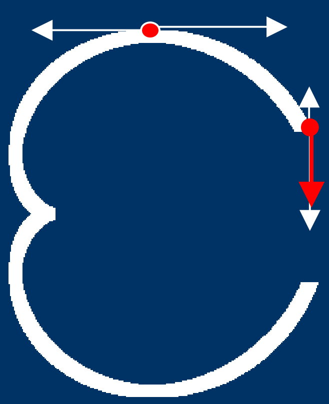
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + \lambda_2 (e_1 e_1^T + e_2 e_2^T)$$



$\Longrightarrow$ Saliency maps SMap BMap

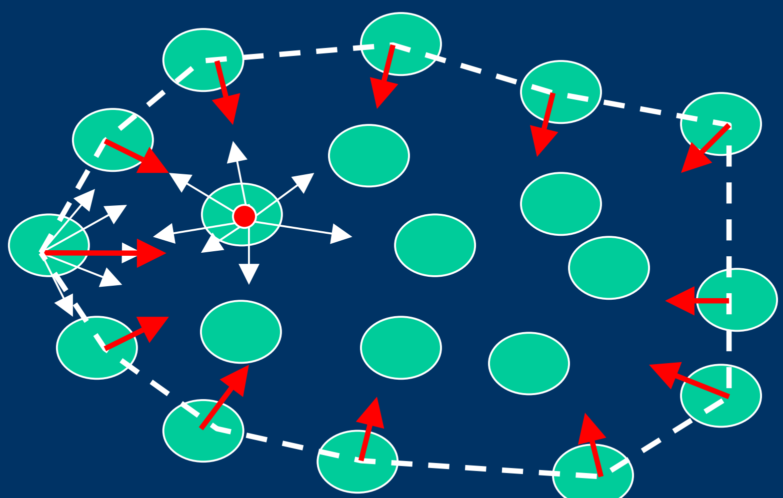
First Order Vote Interpretation (curves)

- Tokens near terminations accumulate first order votes from consistent direction
- Tokens along smooth structures receive opposite votes that cancel out



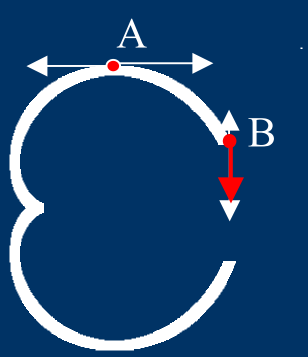
First Order Vote Interpretation (regions)

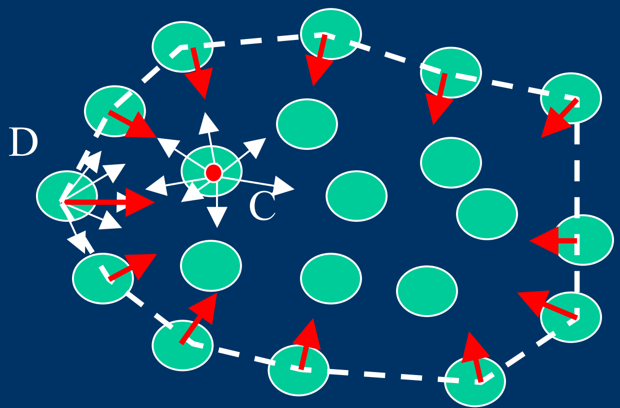
- Tokens near discontinuities accumulate first order votes from consistent direction
- Tokens in the interior of smooth structures receive contradicting votes that cancel out

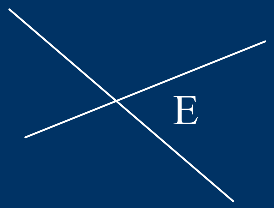


Structure Inference in 2-D

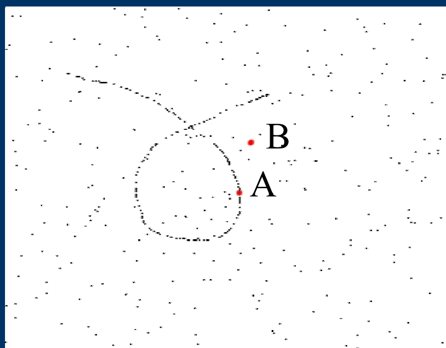
Structure Type	**Saliency**	**Tensor Orientation**	**Polarity**	**Polarity orientation**
Curve inlier **A**	High λ_1- λ_2	Normal: $\mathbf{e}_1$	Low	-
Curve endpoint **B**	High λ_1- λ_2	Normal: $\mathbf{e}_1$	High	Normal to $\mathbf{e}_1$
Region inlier **C**	High λ_2	-	Low	-
Region boundary **D**	High λ_2	-	High	Normal to boundary
Junction **E**	Distinct locally max λ_2	-	Low	-
Outlier	Low	-	Indifferent	-







Sensitivity to Scale



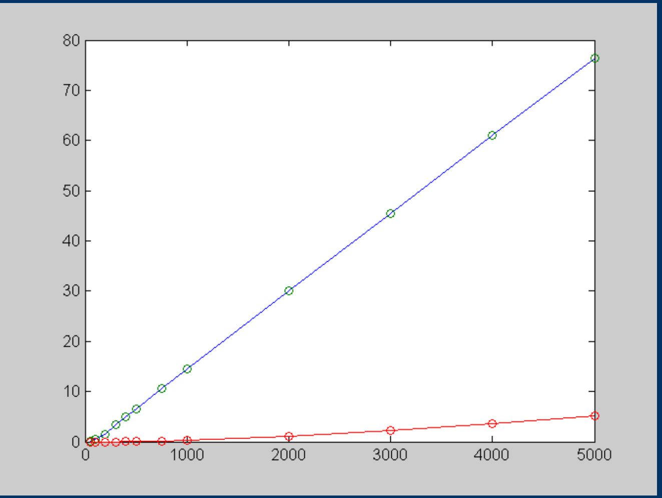
Input

σ = 50

σ = 500

σ = 5000

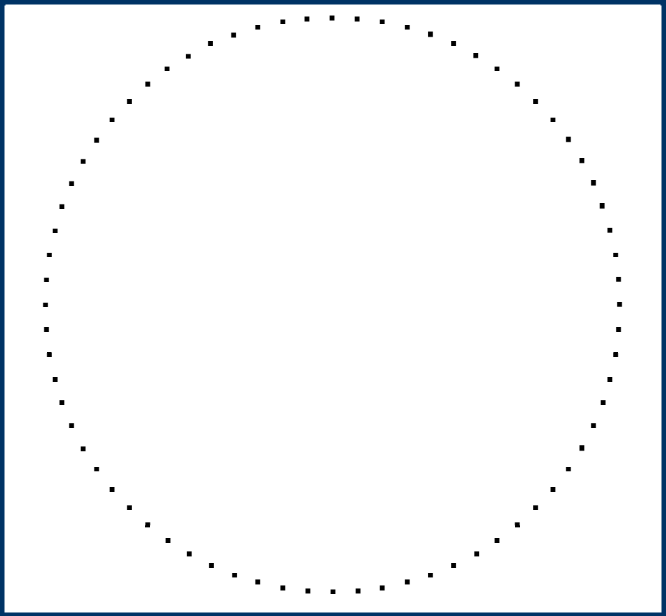
Input: 166 un-oriented inliers, 300 outliers
Dimensions: 960x720
Scale ∈ [50, 5000]
Voting neighborhood ∈ [12, 114]



Curve saliency as a function of scale
Blue: curve saliency at A
Red: curve saliency at B

Sensitivity of Orientation to Scale

Circle with radius 100 (unoriented tokens)

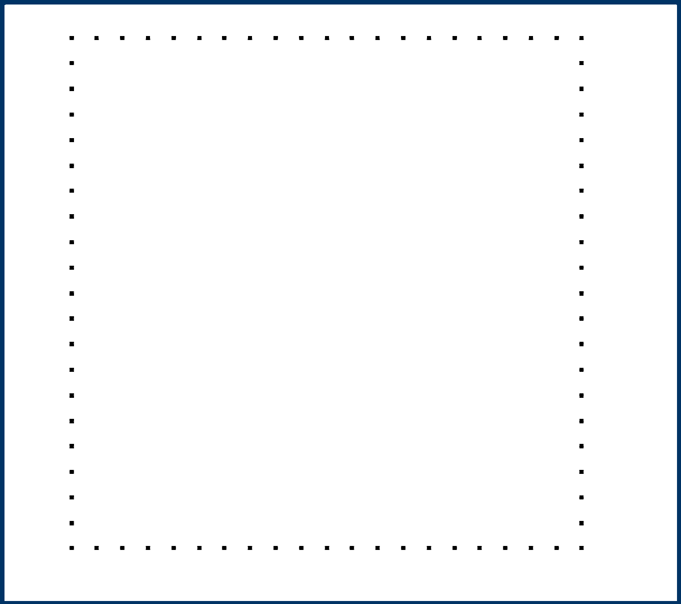


Scale	Average angular error (degrees)
50	1.01453
100	1.14193
200	1.11666
300	1.04043
400	0.974826
500	0.915529
750	0.813692
1000	0.742419
2000	0.611834
3000	0.550823
4000	0.510098
5000	0.480286

As more information is accumulated, the tokens better approximate circle

Sensitivity of Orientation to Scale

Square 200x200 (unoriented tokens)

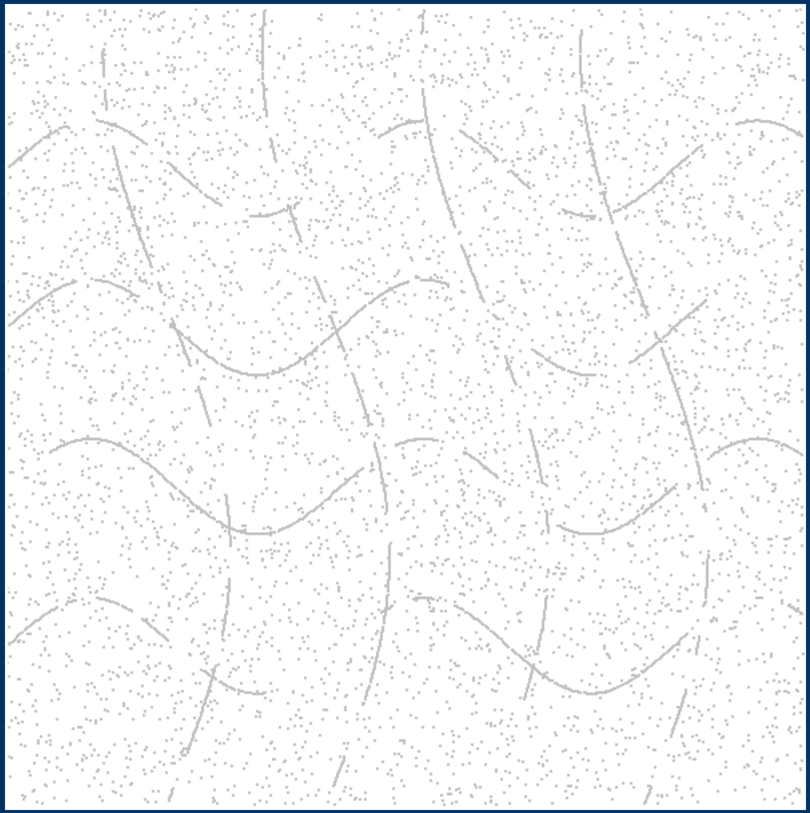


Scale	Average angular error (degrees)
50	1.11601e-007
100	0.138981
200	0.381272
300	0.548581
400	0.646754
500	0.722238
750	0.8893
1000	1.0408
2000	1.75827
3000	2.3231
4000	2.7244
5000	2.98635

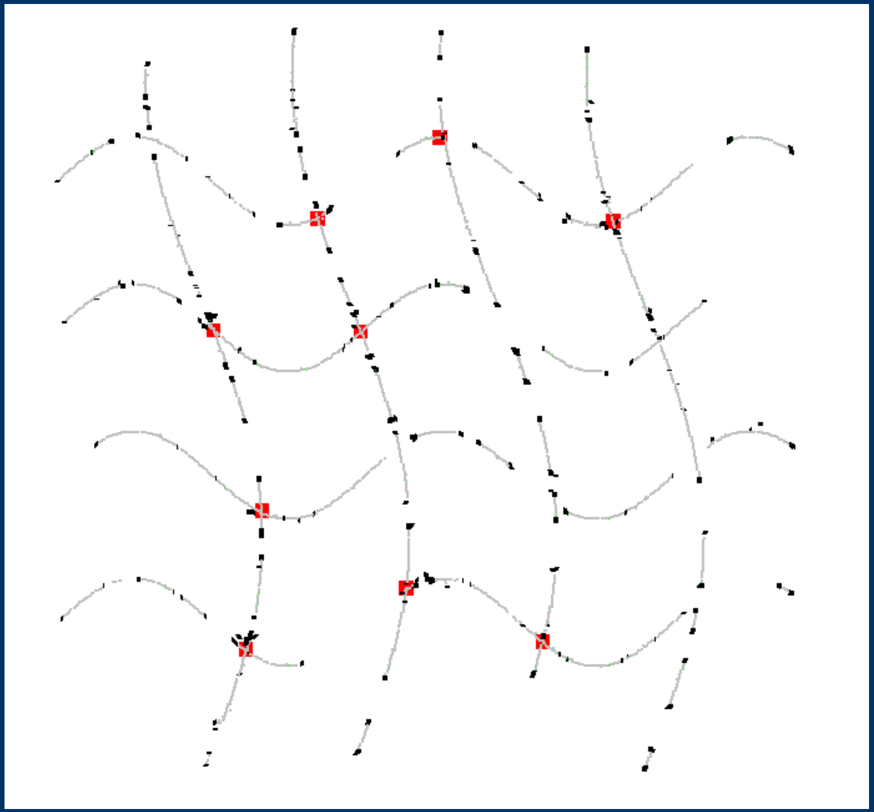
Junctions are detected and excluded

As scale increases to unreasonable levels (>1000) corners get rounded

Examples in 2-D

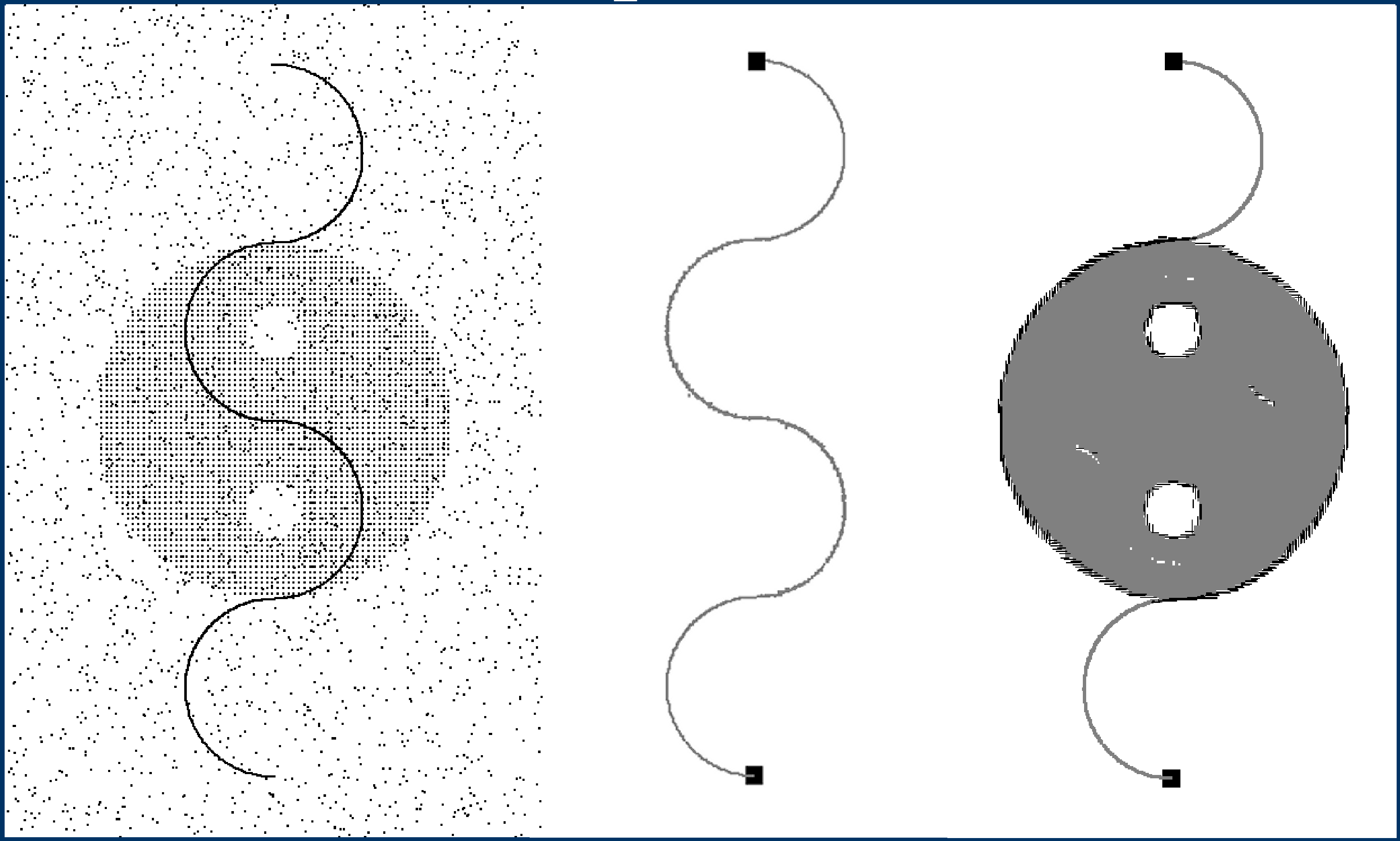


Input



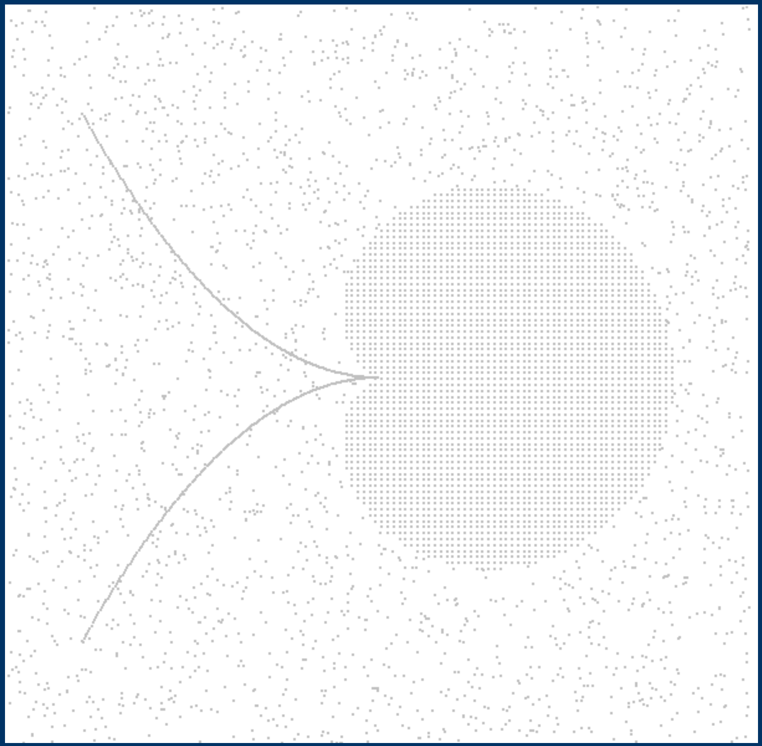
Gray: curve inliers
Black: curve endpoints
Red: junctions

Examples in 2-D

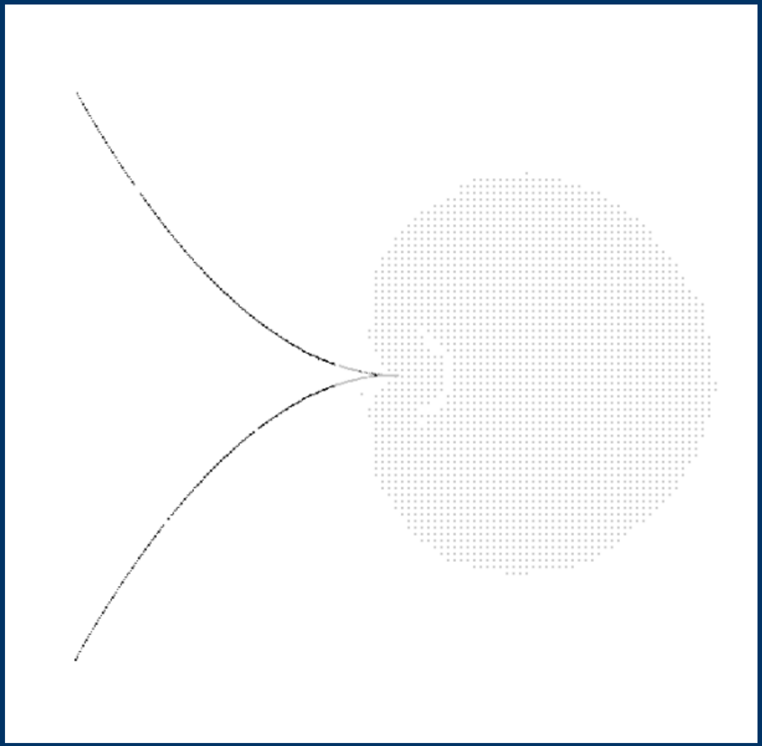


Input Curves and endpoints only Curves, endpoints and regions

Examples in 2-D



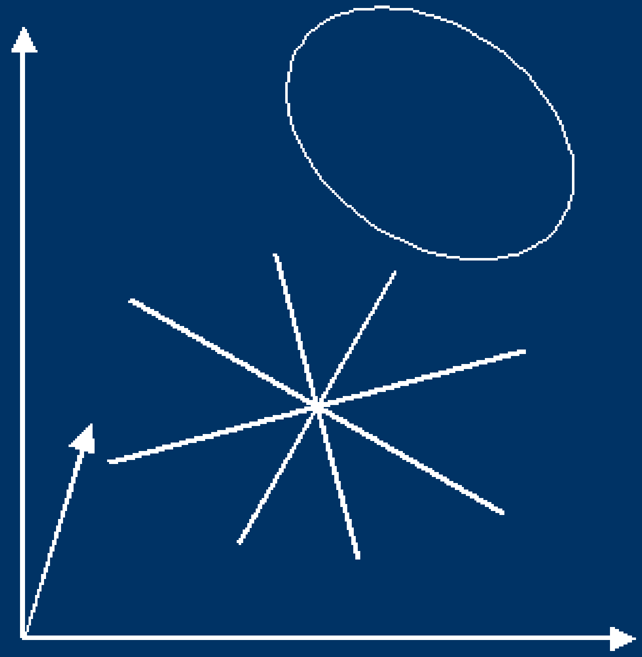
Input



Curve and region inliers

Illusory Contours

- Aligned endpoints interpreted by HVS as forming illusory contours
- Layered scene interpretation



Illusory Contours in the Tensor Voting Framework

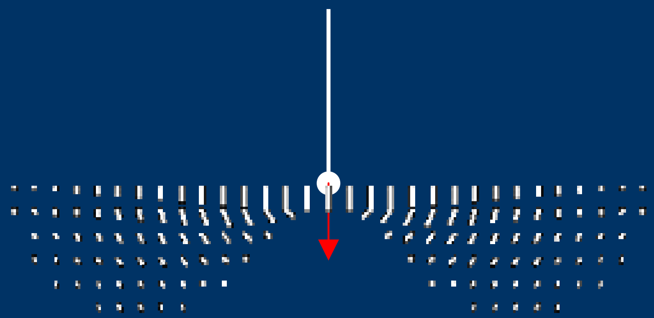
- Endpoint detection
- Used as inputs for illusory contour inference
- Use polarity vector (parallel curve’s tangent) as curve normal

Illusory Contours and Voting Fields

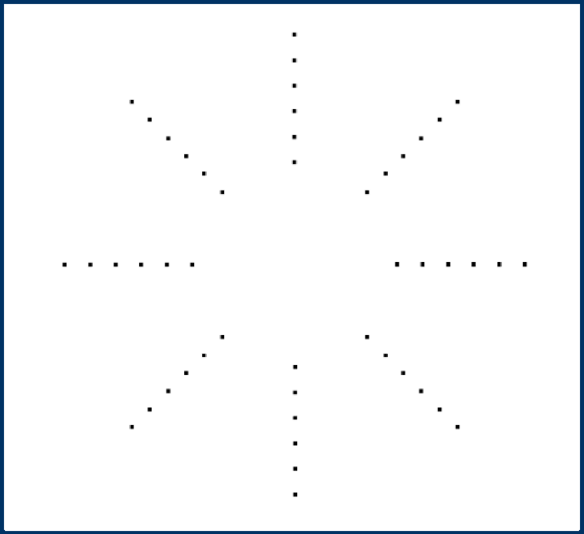
- Since polarity vector (tangent of detected curve segment) serves curve normal

=> fields orthogonal to regular ones

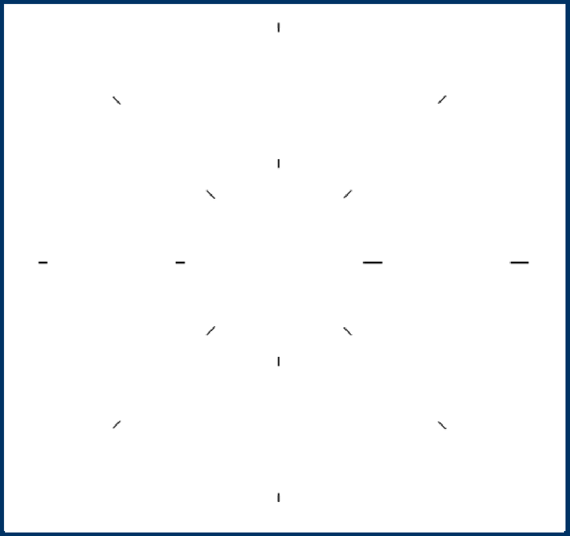
- Illusory contours are convex
  - Votes cast only to half-space away from original curve segments



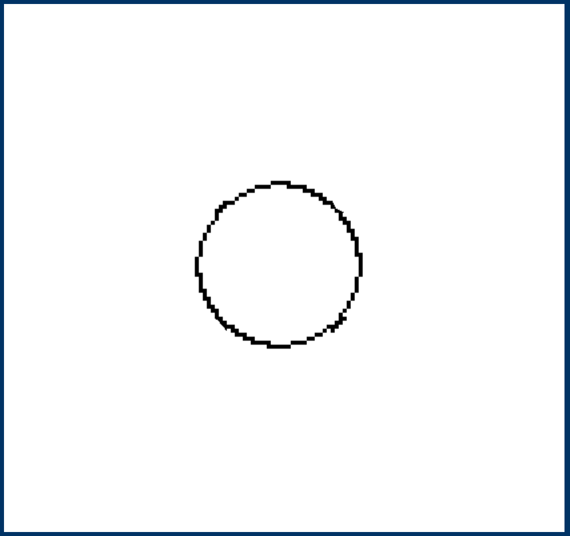
Illusory Contour Example



Input



Detected endpoints and polarity vectors



Illusory contour