Tensor Voting:

A Perceptual Organization Approach to Computer Vision and Machine Learning

Philippos Mordohai

University of North Carolina, Chapel Hill

http://cs.unc.edu/~mordohai

Note: this is a preliminary version. Check my webpage for the most recent presentation

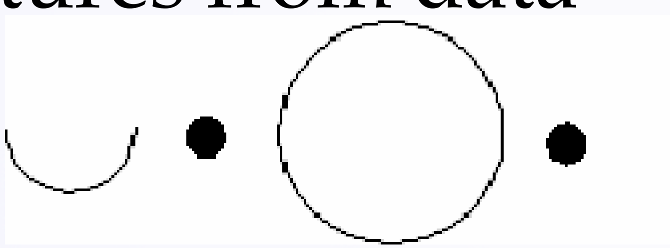
Objectives

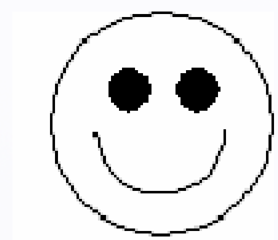
- Unified framework to address wide range of problems as perceptual organization

- Applications:
  - Core computer vision problems
  - Instance-based learning

Motivation

- Develop an approach that is:
  - General
  - Data-driven
- Axiom: the whole is greater than the sum of the parts
- Employ Gestalt principles of proximity and good continuation to infer salient structures from data

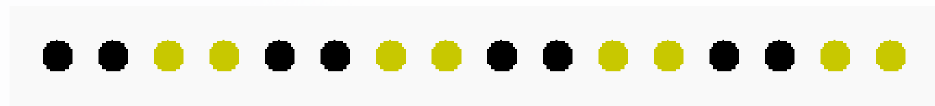




Gestalt Principles

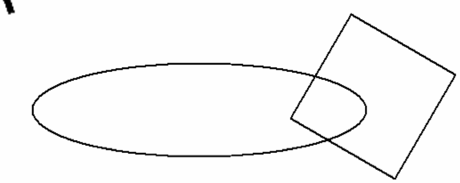
- Proximity

- Similarity

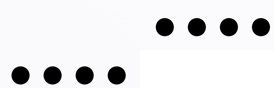


- Good continuation

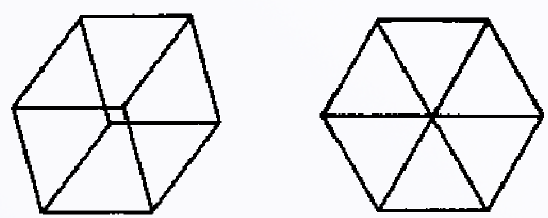
- Closure



- Common fate



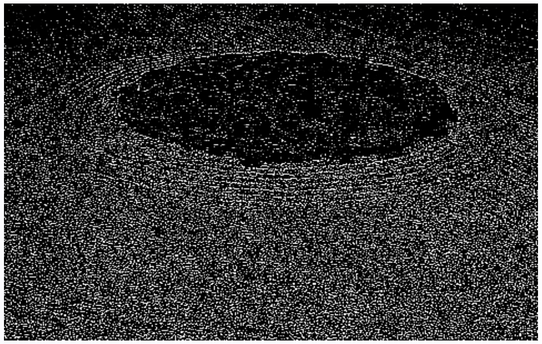
- Simplicity



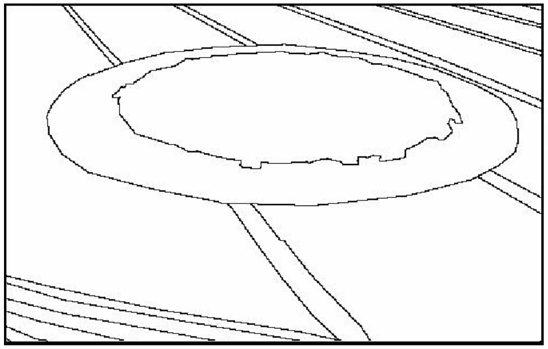
Structural Saliency

- Property of structures to stand out due to *proximity* and *good continuation*

Input



Edge detector



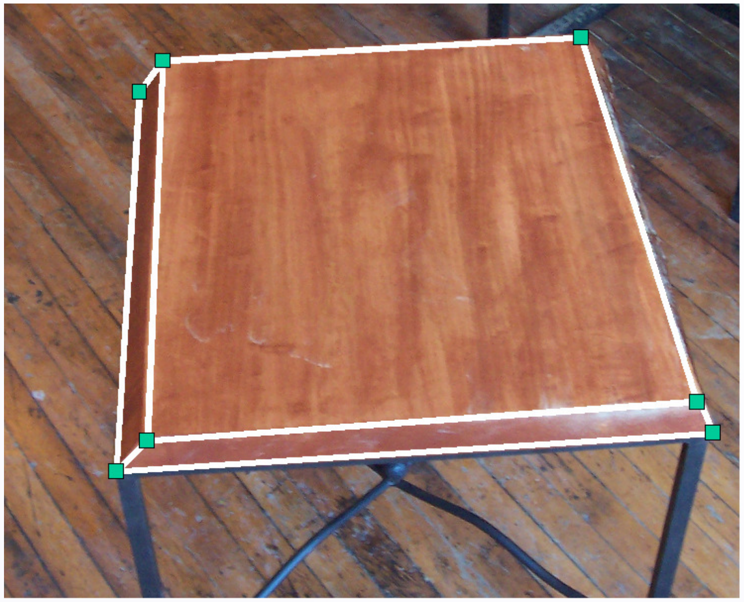
Human observer

- Local responses are not enough
- Need aggregation of support

➢ *The smoothness constraint*: applies almost everywhere

Integrated Descriptions

- Different types of structures interact
  - Junctions are intersections of curves that do not exist in isolation
- Structures have boundaries

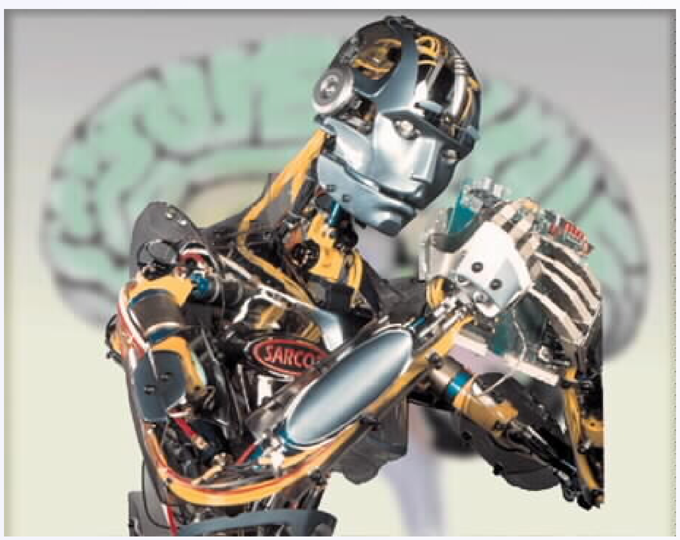


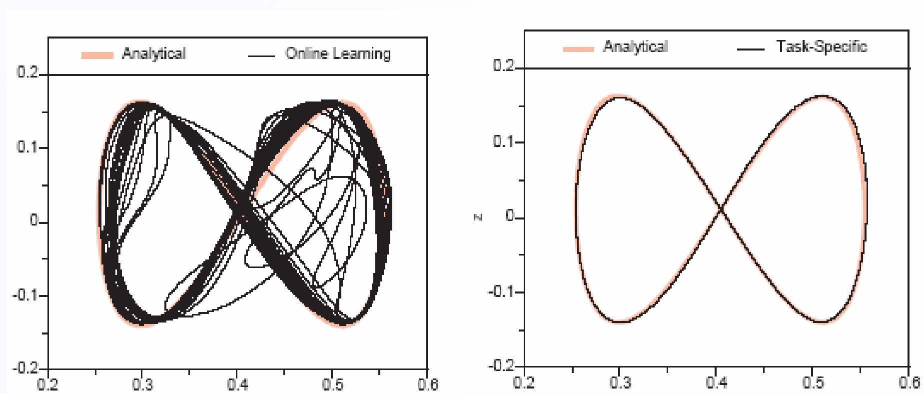
Desired Properties

- Local
- Data-driven (model-free)
- Able to represent *all* structure types and their interactions
- Able to process large amounts of data
- Robust to noise

Beyond 2- and 3-D

- Gestalt principles can be applied in any dimension
- Coherent data form smooth, salient structures
- Positions, velocities and motor commands form manifolds in N-D





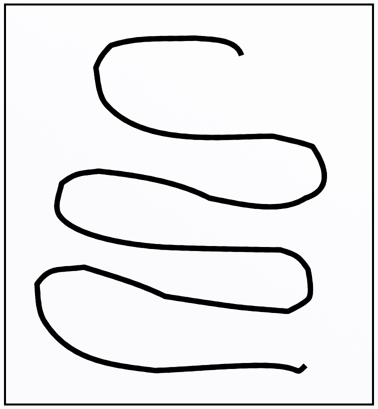
Vijaykumar *et al.* 2002

Perceptual Organization Approaches

- Symbolic methods [Marr 1982, Lowe 1985, Jacobs 1996]
- Clustering [Jain 1988, Shi 2000, Boykov 2001]
- Local interactions [Shashua 1988, Parent 1989, Guy 1996]
- Inspired by human visual system [Grossberg 1985, Heitger 1993, Wiliams 1997]

Differences with our Approach

- Infer all structure types simultaneously and allow interaction between them
- Can begin with oriented or unoriented inputs
- No prior model
- *No objective/cost function*
- Solution emerges from data



Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

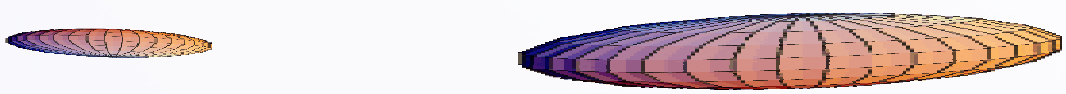
The Original Tensor Voting Framework

- Perceptual organization of generic tokens [Medioni, Lee, Tang 2000]
- Data representation: second order, symmetric, nonnegative definite tensors
- Information propagation: tensor voting
- Infers saliency values and preferred orientation for each type of structure

Second Order Tensors

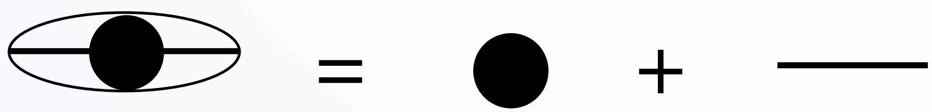
- *Symmetric, non-negative definite*
- Equivalent to:
  - Ellipse in 2-D or ellipsoid in 3-D
  - 2x2 or 3x3 matrix
- Properties that can be encoded:
  - shape: orientation certainty

  - size: feature saliency



Second Order Tensors in 2-D

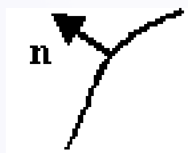
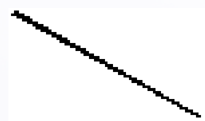
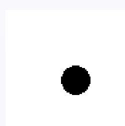
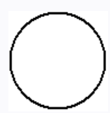
- 2×2 Matrix or Ellipse can be decomposed:
  - ***Stick*** component
  - ***Ball*** component



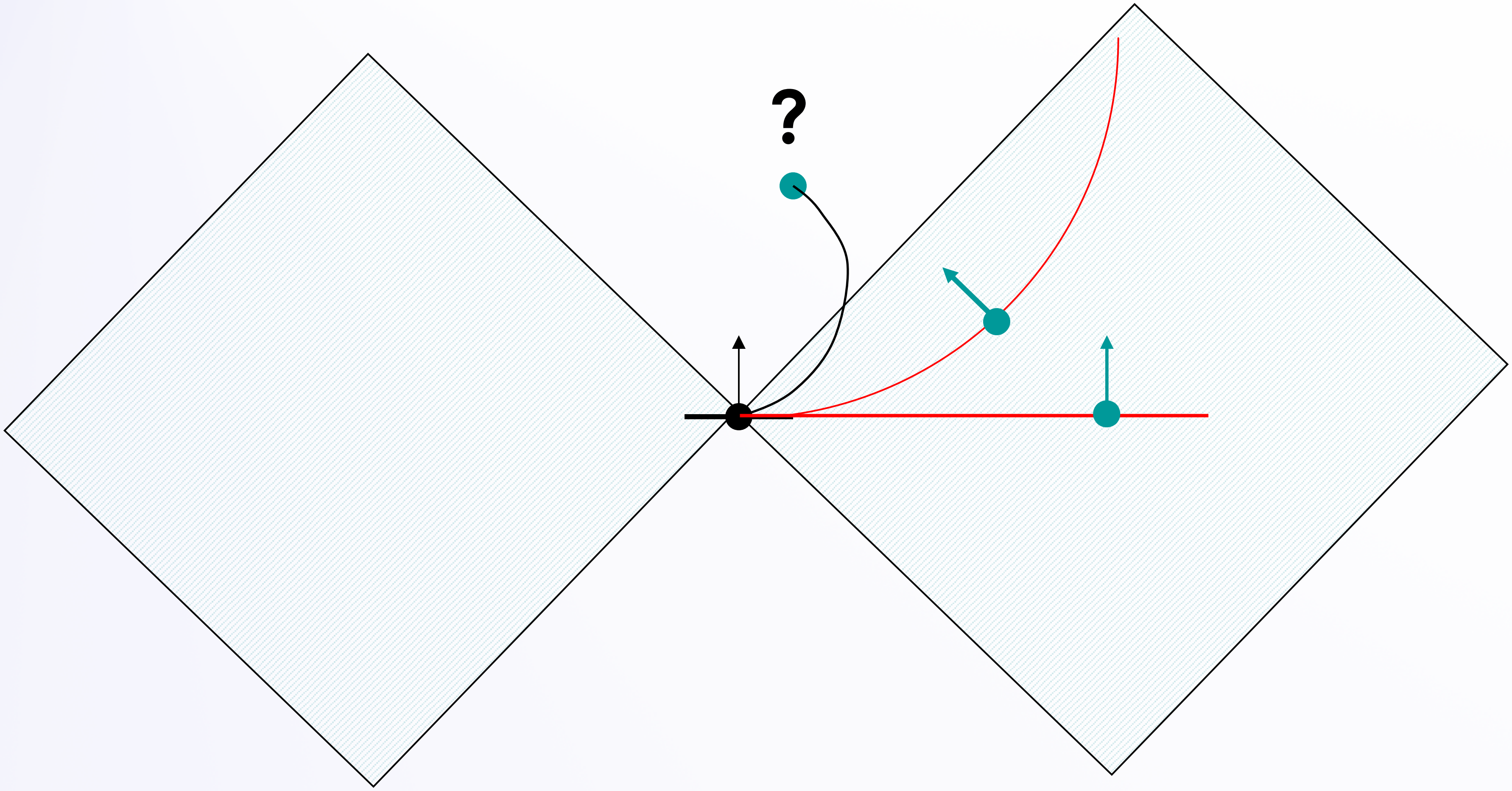
$$\begin{bmatrix} a^2+b^2 & a^2 \\ a^2 & a^2 \end{bmatrix} = \begin{bmatrix} a^2 & a^2 \\ a^2 & a^2 \end{bmatrix} + \begin{bmatrix} b^2 & 0 \\ 0 & 0 \end{bmatrix}$$

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

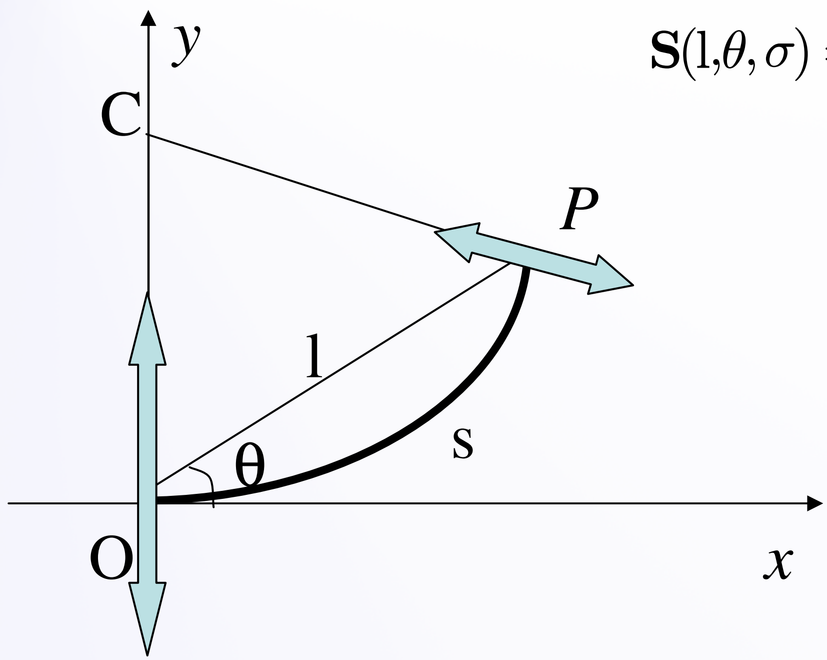
Representation with Tensors

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

Tensor Voting



Saliency Decay Function



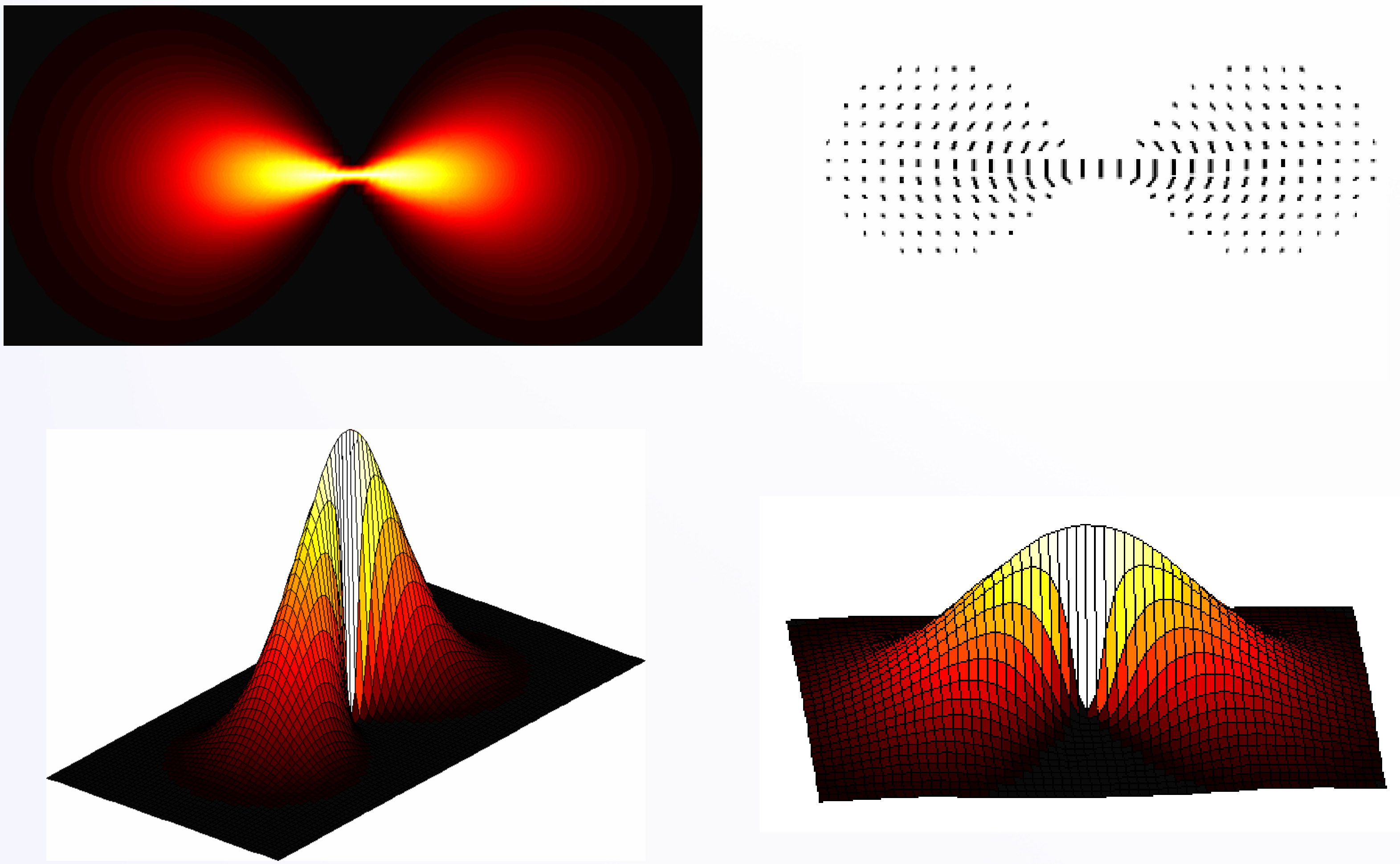
$$\mathbf{S}(l,\theta,\sigma) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)} \begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix} [-sin(2\theta) \quad cos(2\theta)]$$

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

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

- Votes attenuate with length of smooth path and curvature
- Stored in pre-computed *voting fields*

Fundamental Stick Voting Field

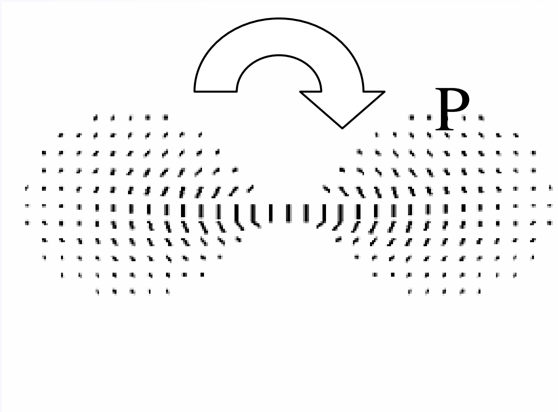


Fundamental Stick Voting Field

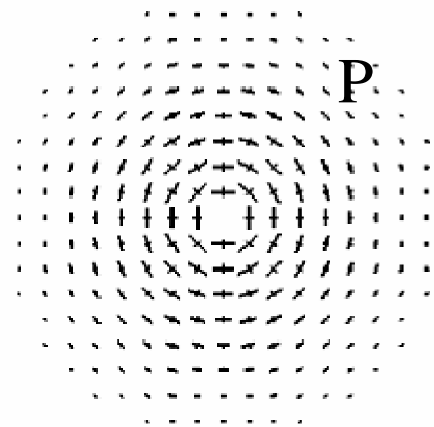
All other fields in *any* N-D space are generated from the *Fundamental Stick Field:*

- 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)



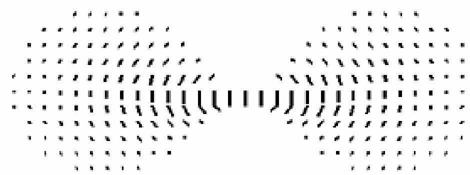
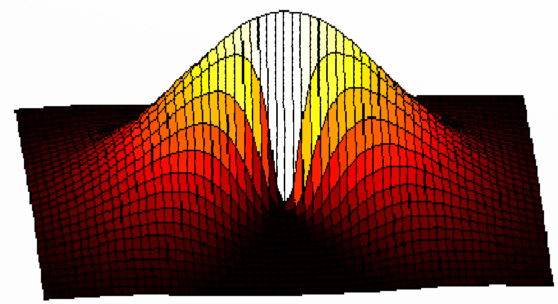
B(P)

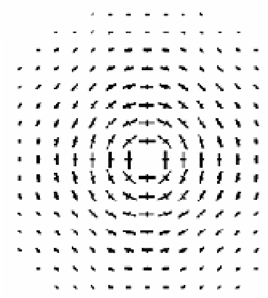
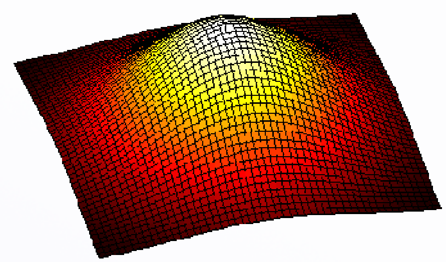
Ball field computed by integrating the contributions of rotating stick

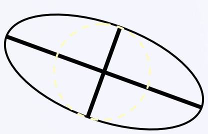
$$\mathbf{B}(P) = \int \mathbf{S}(P)\,d\theta$$

2-D Voting Fields

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

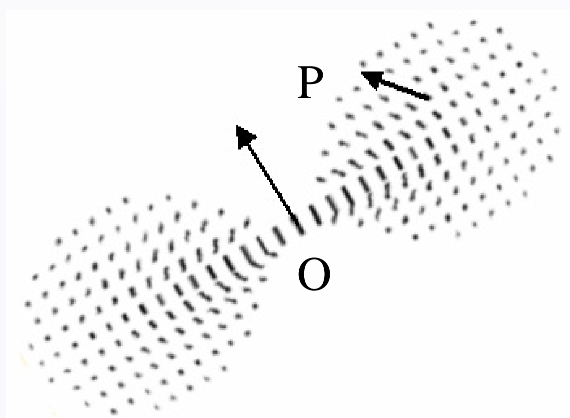
— votes with  

● 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



Scale of Voting

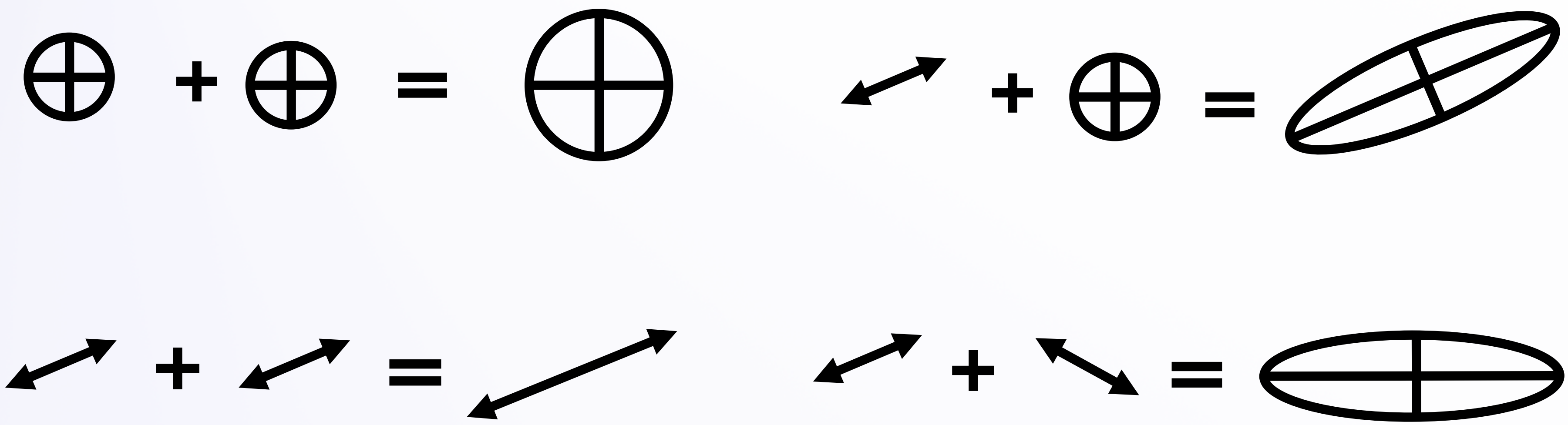
- The Scale of Voting is the single critical parameter in the framework
- Essentially defines size of voting neighborhood
  - Gaussian decay has infinite extend, but it is cropped to where votes remain meaningful (e.g. 1% of voter saliency)

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

Vote Accumulation

Each site accumulates second order votes by tensor addition:



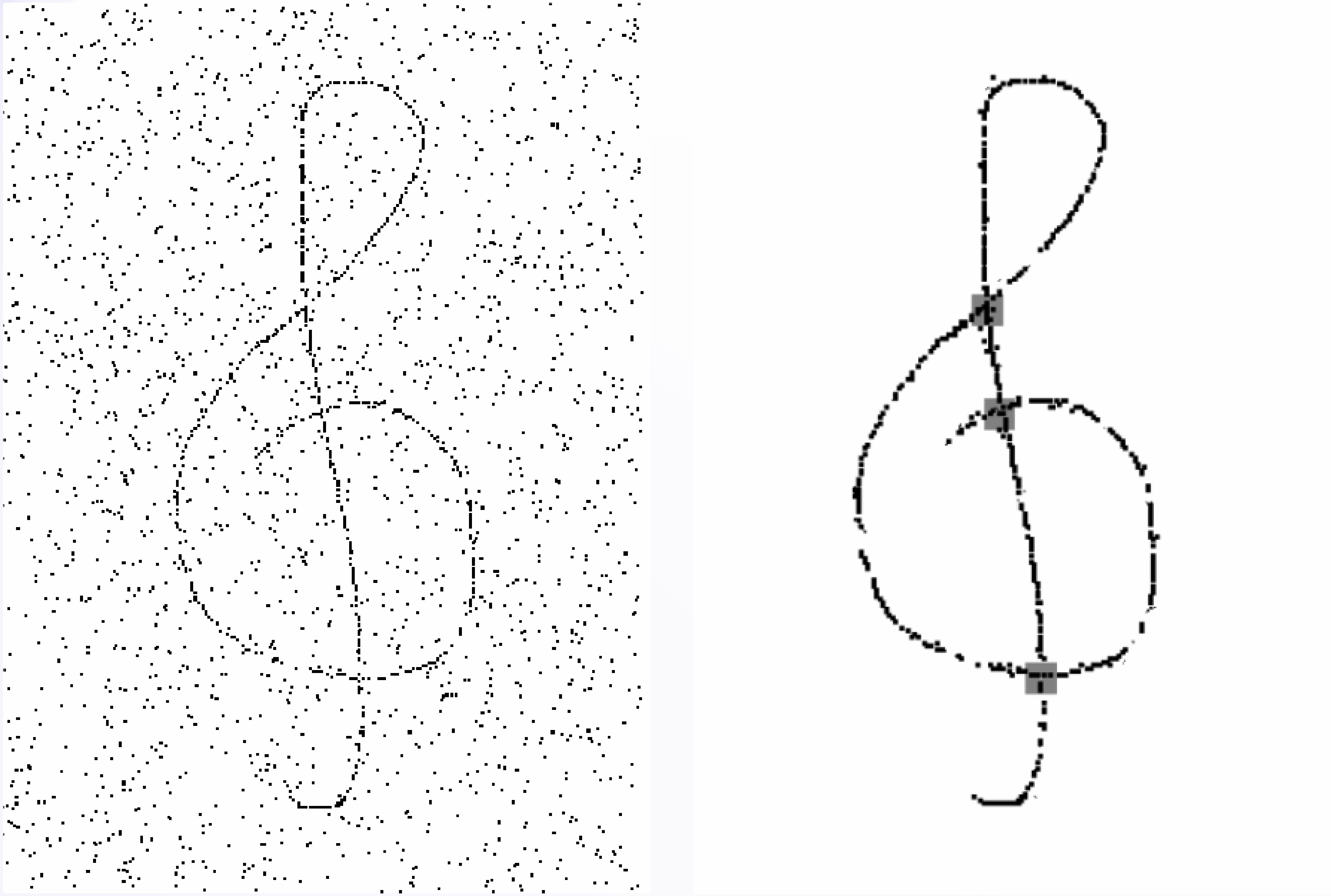
Results of accumulation are usually *generic tensors*

Vote Analysis

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

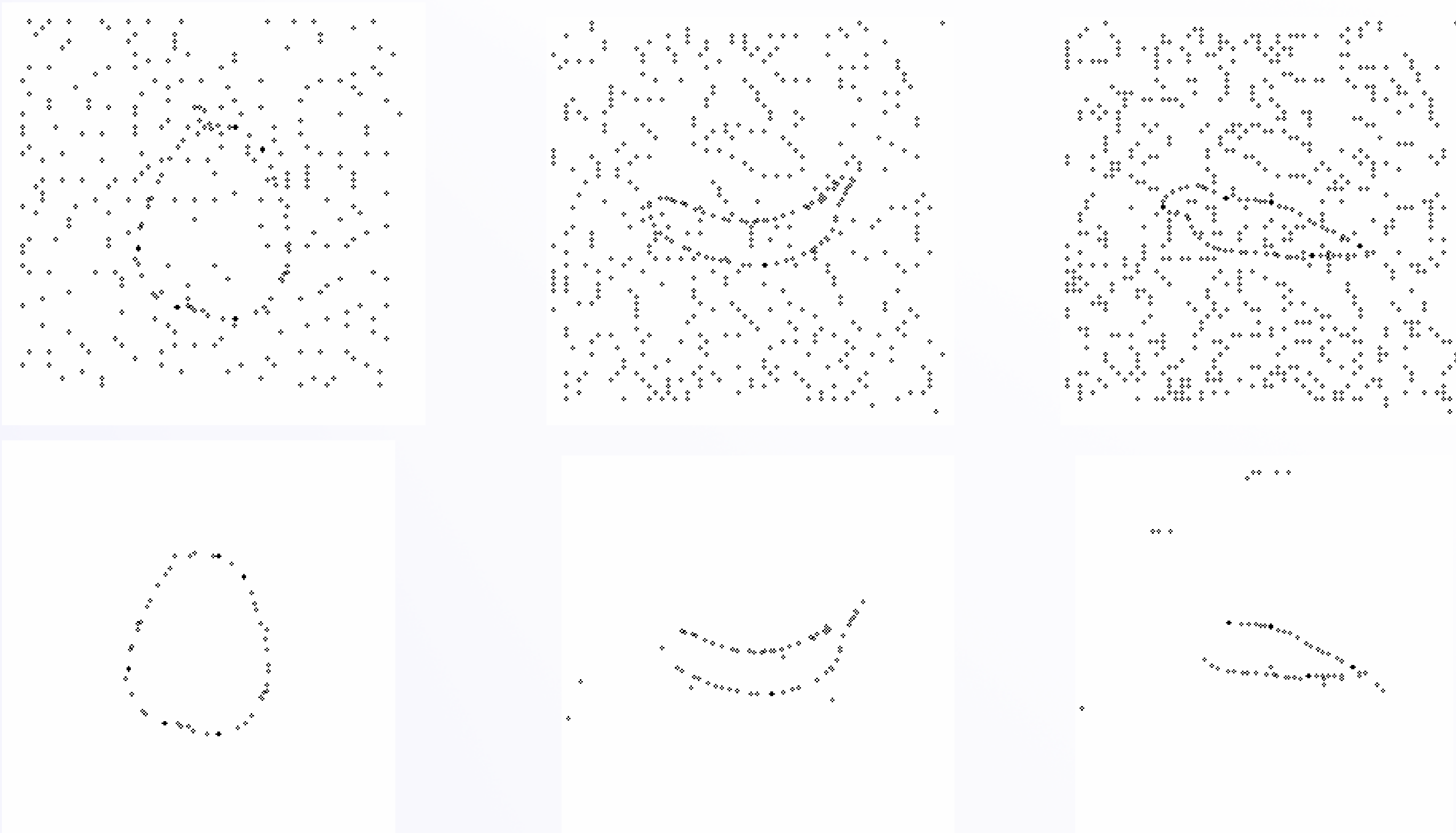
- $\lambda_1 - \lambda_2 > \lambda_2$: stick saliency is larger than ball saliency. Likely on curve.
- $\lambda_1 \approx \lambda_2 > 0$: ball saliency larger than stick saliency. Likely junction.
- $\lambda_1 \approx \lambda_2 \approx 0$: Low saliency. Outlier.

Results in 2D



Structural Saliency Estimation

- Data from [Williams and Thornber IJCV 1999]
- Foreground objects (*N* edgels) on background clutter
- Detect *N* most salient edgels



Structural Saliency Estimation

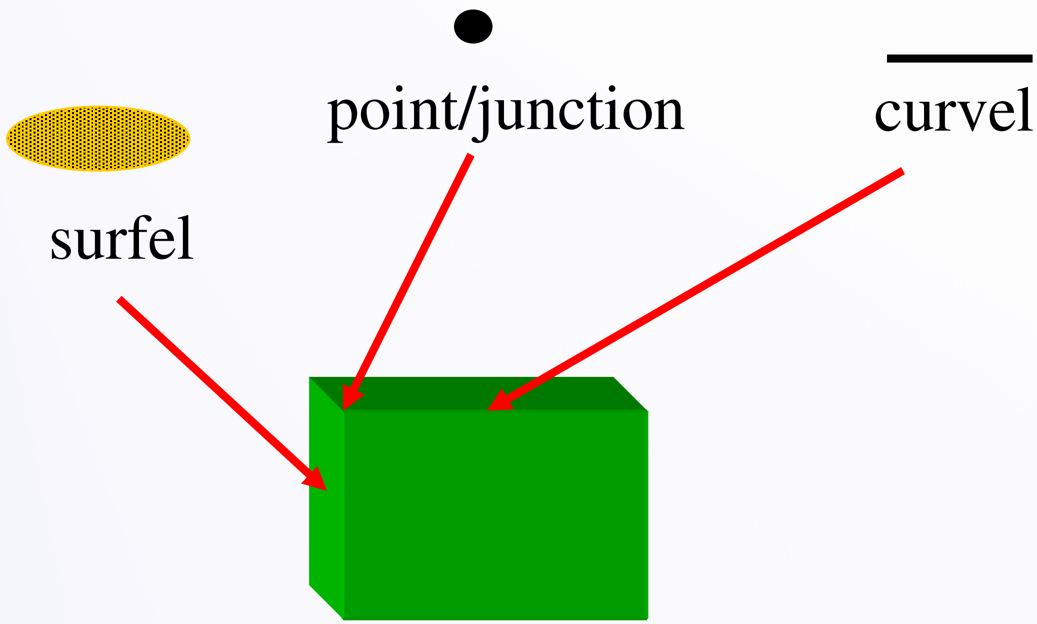
- SNR: ratio of foreground edgels to background edgels
- FPR: false positive rate for foreground detection

- Our results outperform all methods evaluated in [Williams and Thornber IJCV 1999]

SNR	25	20	15	10	5
FPR	10.0%	12.4%	18.4%	35.8%	64.3%

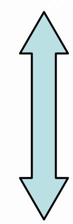
Structure Types in 3-D

The input may consist of

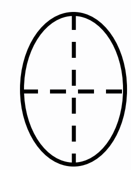


3-D second order Tensors

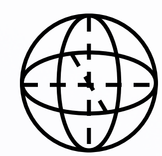
- Encode **normal** orientation in tensor
- Surfel: 1 normal → "stick" tensor



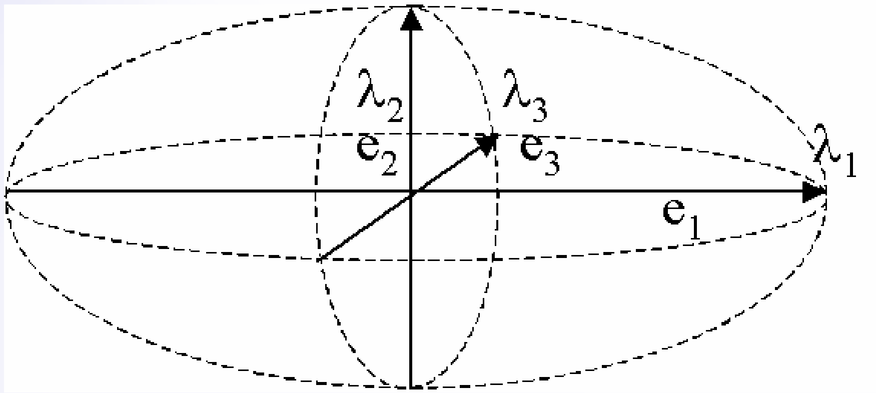
- Curvel: 2 normals → "plate" tensor

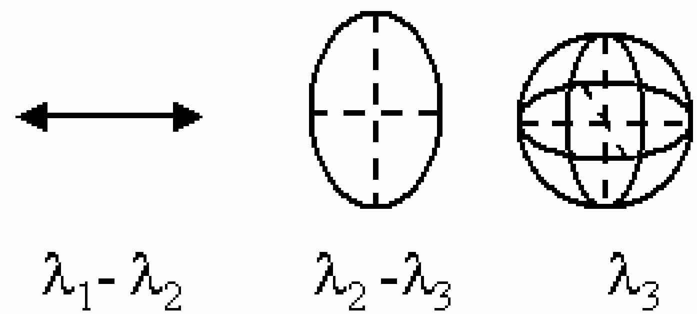


- Point/junction: 3 normals → "ball" tensor



3-D Tensor Analysis





$$\mathbf{T} = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T + \lambda_3 \cdot e_3 e_3^T =$$

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

- Surface saliency: $\lambda_1 - \lambda_2$ normal: $\mathbf{e}_1$
- Curve saliency: $\lambda_2 - \lambda_3$ normals: $\mathbf{e}_1$ and $\mathbf{e}_2$
- Junction saliency: λ_3

3-D Voting Fields

- 2-D stick field is a cut of the 3-D field containing the voter
- Plate and ball fields derived by integrating contributions of rotating stick voter
  - Stick spans disk and sphere respectively

Vote Analysis in 3-D

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

- $\lambda_1 - \lambda_2 > \lambda_2 - \lambda_3$ and $\lambda_1 - \lambda_2 > \lambda_3$: stick saliency is maximum. Likely surface.
- $\lambda_2 - \lambda_3 > \lambda_1 - \lambda_2$ and $\lambda_2 - \lambda_3 > \lambda_3$: plate saliency is maximum. Likely curve or surface intersection
- $\lambda_3 > \lambda_1 - \lambda_2$ and $\lambda_3 > \lambda_2 - \lambda_3$: ball saliency is maximum. Likely junction
- $\lambda_1 \approx \lambda_2 \approx \lambda_3 \approx 0$: Low saliency. Outlier.

Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

Approach for Stereo

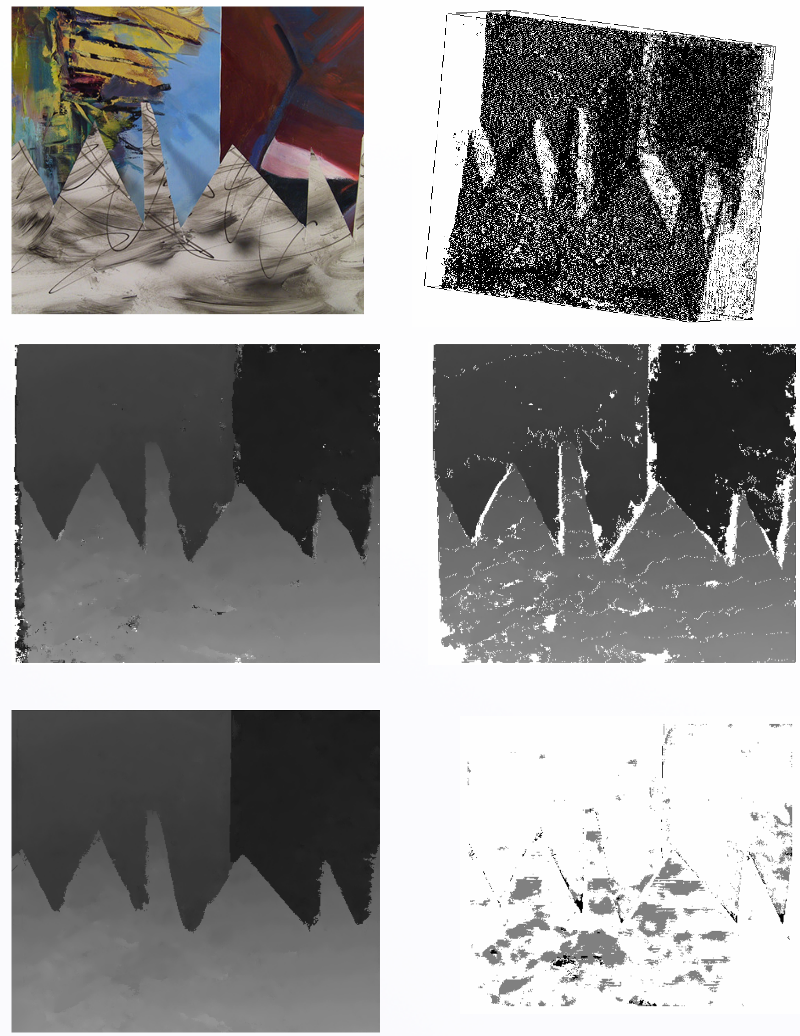
- Problem can be posed as perceptual organization in 3-D
  - Correct pixel matches should form smooth, salient surfaces in 3-D
  - 3-D surfaces should dictate pixel correspondences
- Infer matches and surfaces by tensor voting
- Use monocular cues to complement binocular matches

Challenges

- Major difficulties in stereo:
  - occlusion
  - lack of texture
- Local matching is not always reliable:
  - False matches can have high scores

Algorithm Overview

- Initial matching
- Detection of correct matches
- Surface grouping and refinement
- Disparity estimation for unmatched pixels



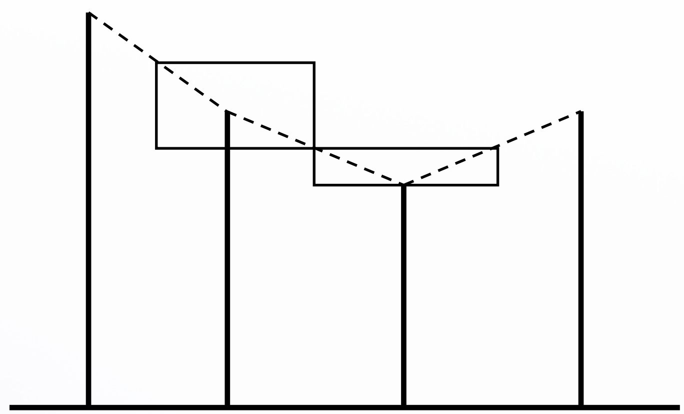
Initial Matching

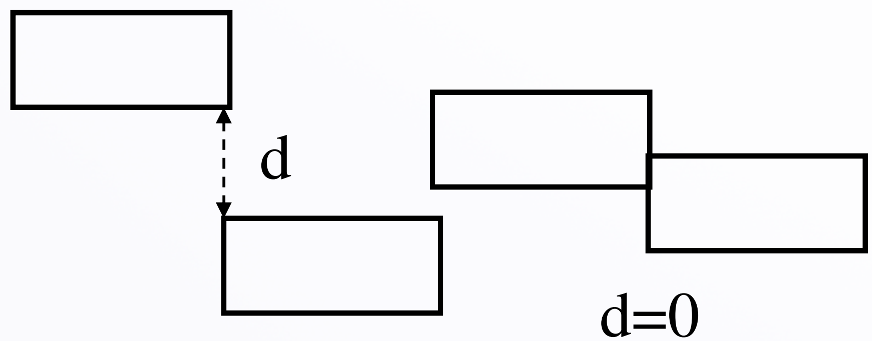
- Each matching technique has different strengths
- Use multiple techniques and both images as reference:
  - 5×5 normalized cross correlation (NCC) window
  - 5×5 shiftable NCC window
  - 25×25 NCC window for pixels with very low color variance
  - 7×7 symmetric interval matching window with truncated cost function

Note: small windows produce random (not systematic) errors (reduced foreground fattening)

Symmetric Interval Matching

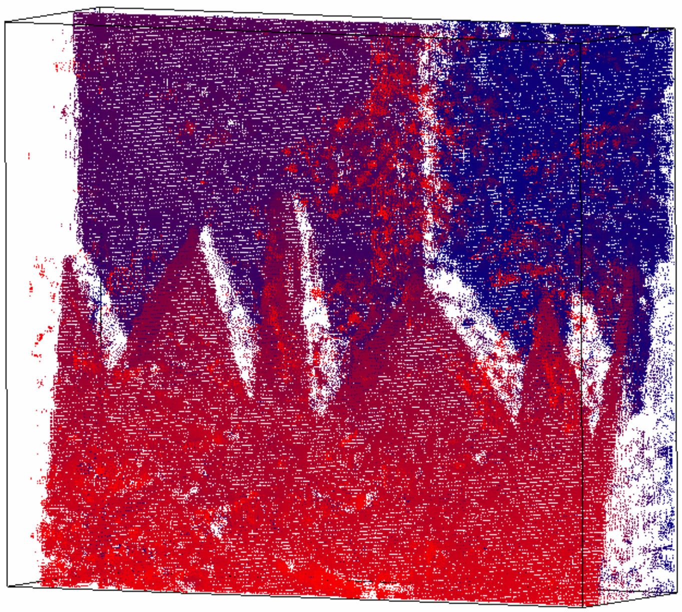
- Upsample scanlines
- Represented the color of pixel (x,y) as the interval (x-½, y) to (x+½, y)
- Dissimilarity measure: distance between intervals
- Truncated to increase performance at discontinuities



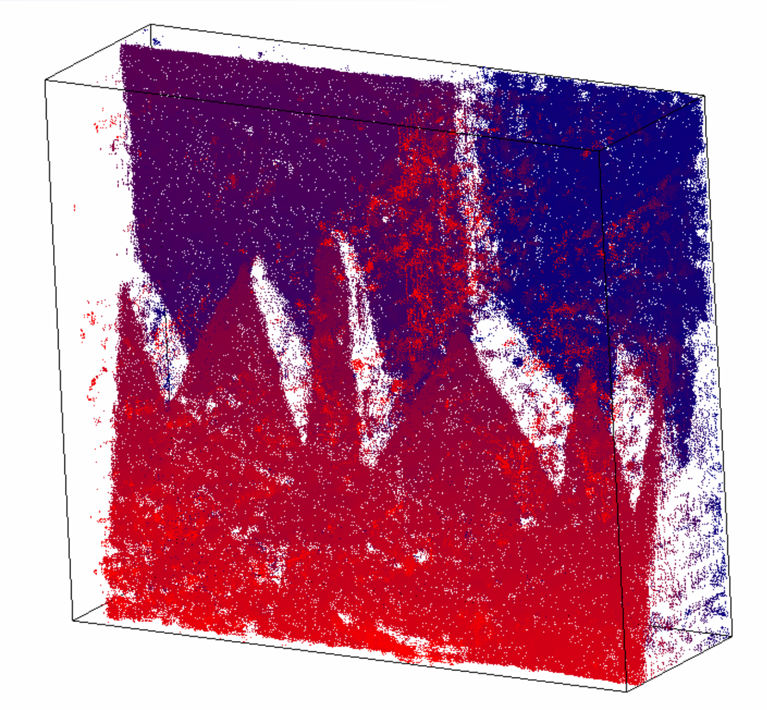


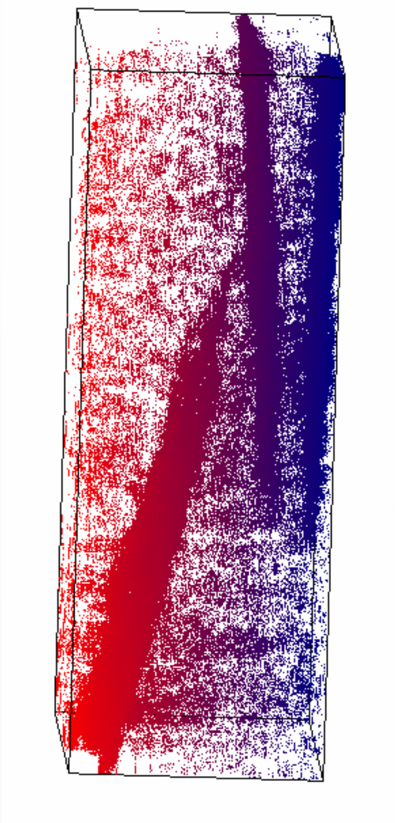
Candidate Matches

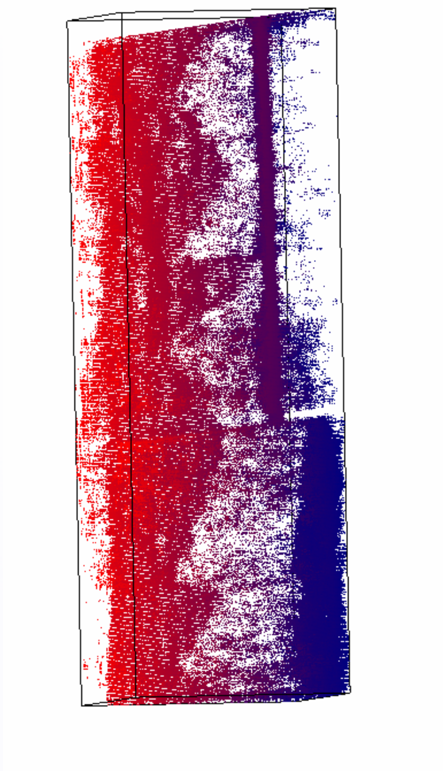
- Compute sub-pixel estimates (parabolic fit)
- Keep all good matches (peaks of NCC)
- Drop scores
  - depend on texture properties
  - Hard to combine across methods



Candidate Matches

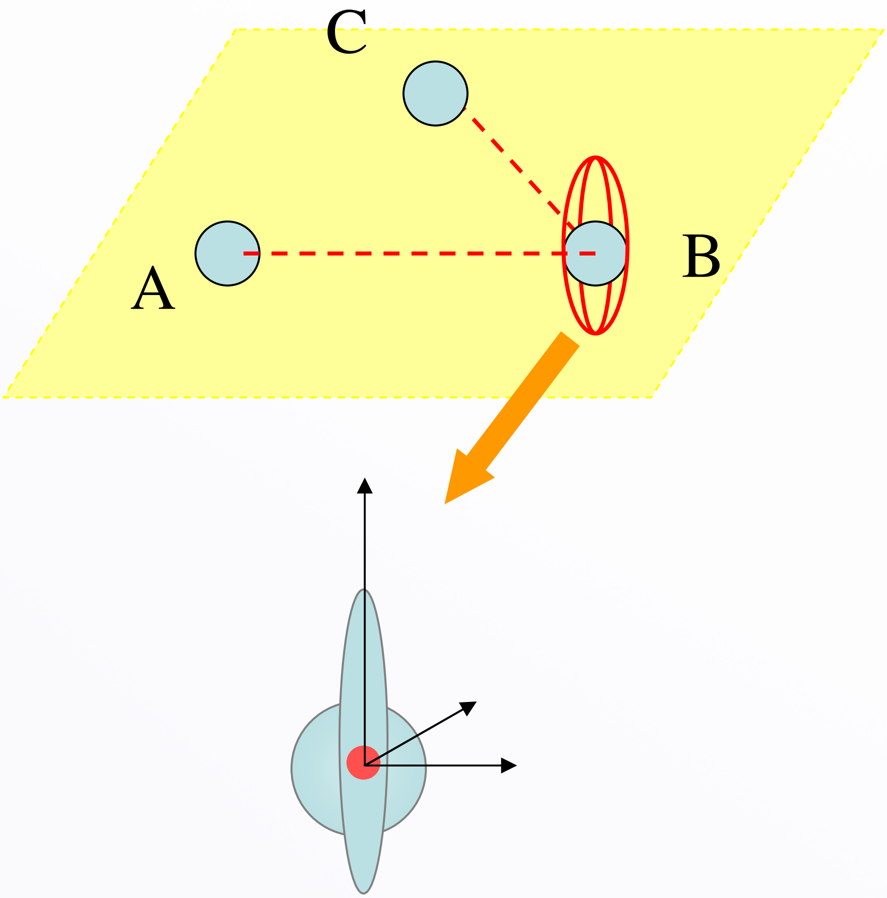






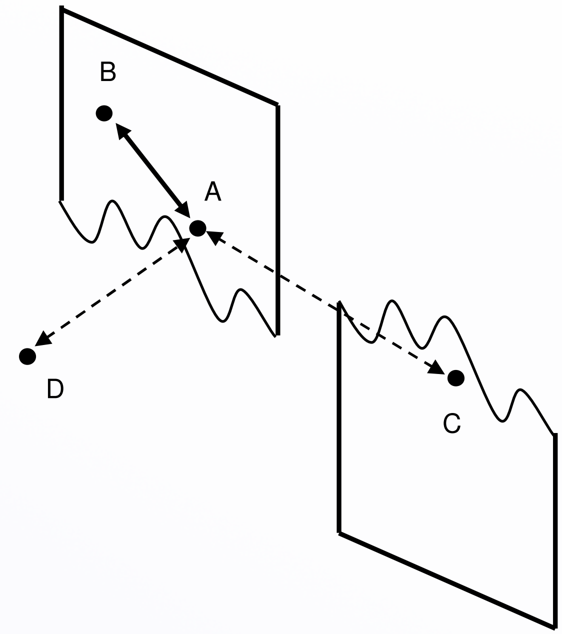
Surfaces from Unoriented Data

- Voting is pair-wise
- Two unoriented tokens define a path and the voter casts a vote (normal spans plane)
- Accumulation of votes with a common axis results in a ***salient surface normal***



Detection of Correct Matches

- Tensor voting performed in **3-D**
- Saliency used as criterion to disambiguate matches instead of aggregated matching cost or global energy
- *Visibility constraint* enforced along rays with respect to surface saliency



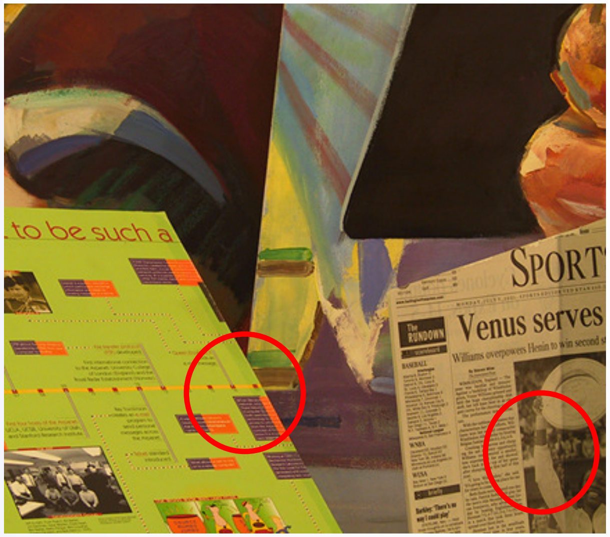
Uniqueness vs. Visibility

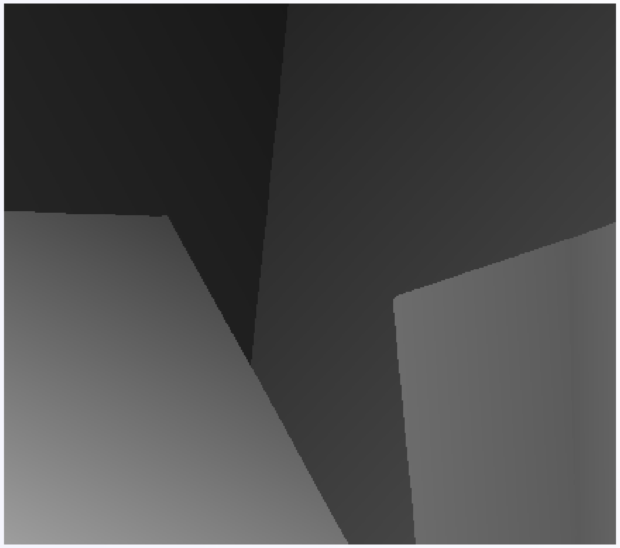
- Uniqueness constraint: One-to-one pixel correspondence
  - Exact only for fronto-parallel surfaces
- Visibility constraint : *M*-to-*N* pixel correspondences
  - [Ogale and Aloimonos 2004][Sun et al. 2005]
  - One match per ray of each camera

Surface Grouping

- Image segmentation has been shown to help stereo
  - Not an easier problem
- Instead*, group candidate matches in 3-D based on geometric properties*
  - Pick most salient candidate matches as seeds
  - Grow surfaces
- Represent surfaces as collections of colors

Nonparametric Color Model





- Does not fail for different surfaces that include similar colors
  - Grouping done in 3-D
- Can represent surfaces with any color distribution
  - No need to compute GMM
- Each pixel has been assigned to a surface now

Surface Refinement

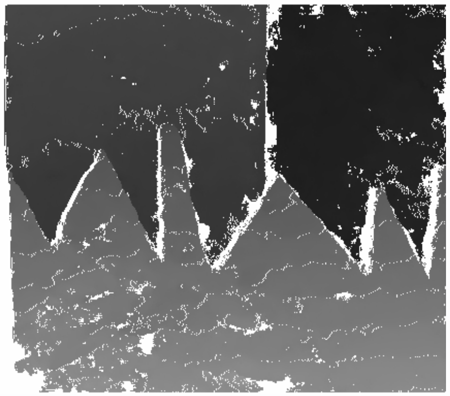
- Re-visit pixels and verify label assignments
- Find all neighboring surfaces including current assignment *s*
- Compute $R_i(x_0,y_0)$ for all surfaces
  - Ratio of pixels of surface *i* within neighborhood *N* similar in color to $I_L(x_0,y_0)$ over all pixels in *N* labeled as *i*
- Remove match if $R_s(x_0,y_0)$ is not maximum

➢ *Set of reliable matches* (reduced foreground fattening)

$$\mathrm{R}_i(x_0,y_0) = \frac{\sum_{(x,y)\in N} T(lab(x,y)=i \; AND \; dist(I_L(x,y), I_L(x_0,y_0) < c_{thr}))}{\sum_{(x,y)\in N} T(lab(x,y)=i))}$$

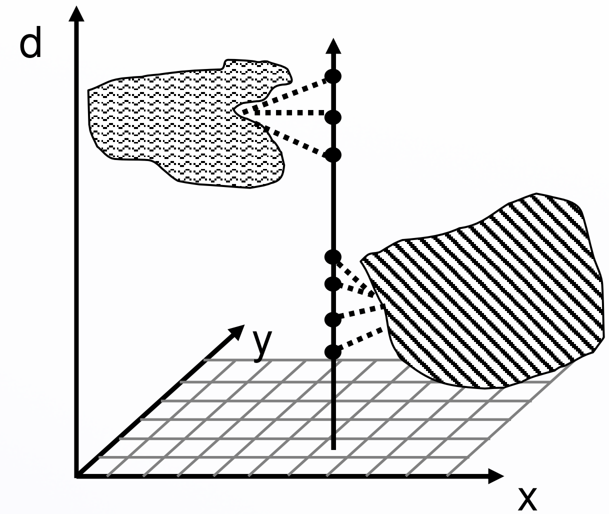
Disparity Hypotheses for Unmatched Pixels

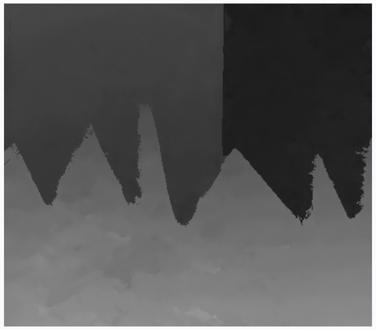
- Check color consistency with nearby layers (on both images if not occluded)
- Generate hypotheses for membership in layers with similar color properties
  - Find disparity range from neighbors (and extend)
  - Allow occluded hypotheses
  - Do not allow occlusion of reliable matches



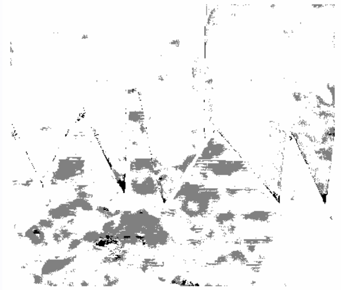
Disparities for Unmatched Pixels

- Vote from neighbors of same surface
- Keep most salient
  - Update occlusion information



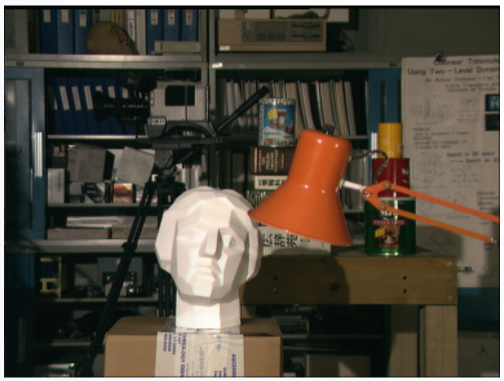


Disparity Map



Error Map

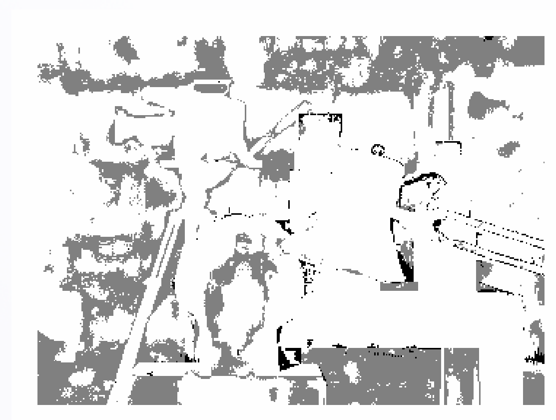
Results: Tsukuba



Left image

Ground truth

Disparity Map

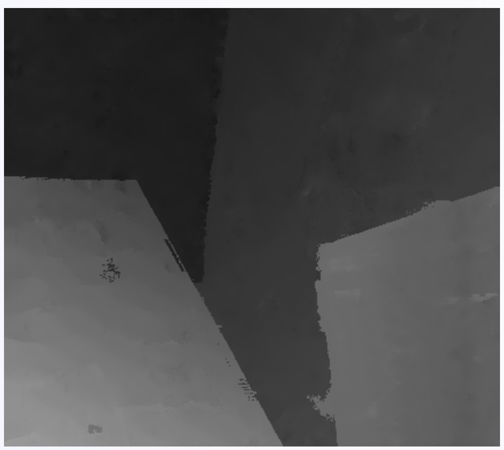


Error Map

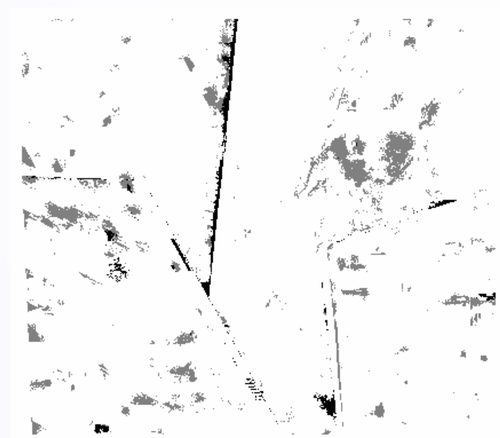
Results: Venus

Left image

Ground truth



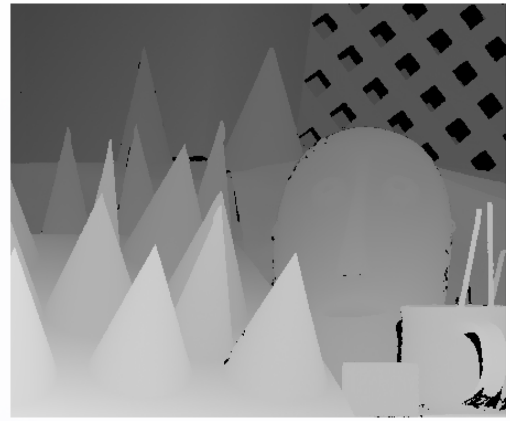
Disparity Map



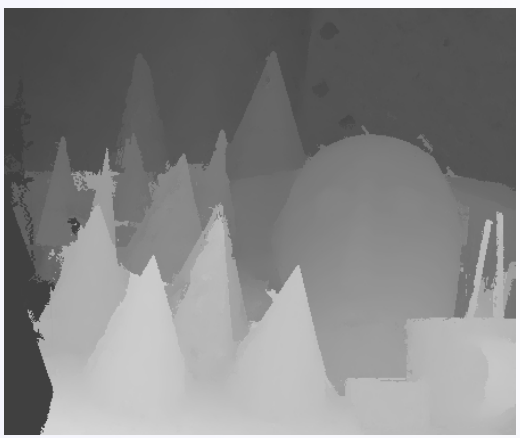
Error Map

Results: Cones

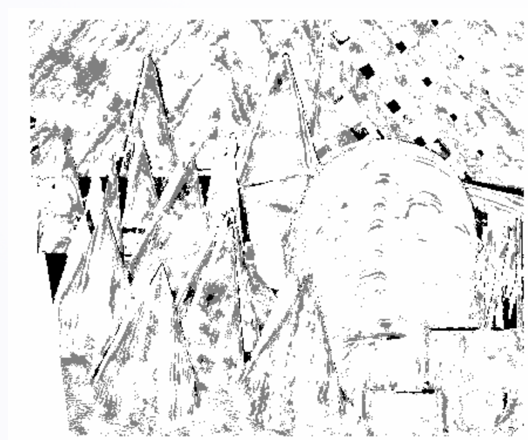
Left image



Ground truth



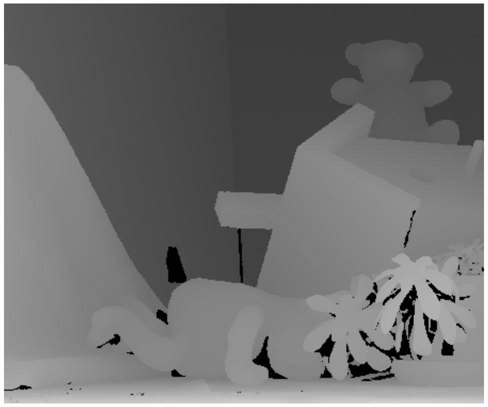
Disparity Map



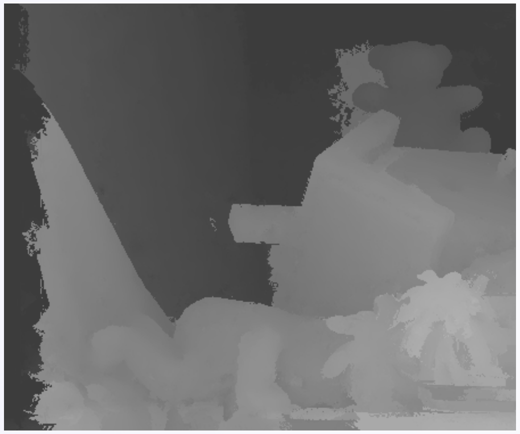
Error Map

Results: Teddy

Left image



Ground truth



Disparity Map

Error Map

Quantitative Results

Error Threshold = 0.5		Sort by nonocc			Sort by all						Sort by disc		
Error Threshold...													
Algorithm	**Avg. Rank**	**Tsukuba** ground truth			**Venus** ground truth			**Teddy** ground truth			**Cones** ground truth		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
C-SemiGlob [19]	3.7	13.9 8	14.7 8	18.9 12	3.30 2	**3.82** 1	10.9 4	**9.82** 1	17.4 3	**22.8** 1	5.37 2	**11.7** 1	**12.8** 1
SemiGlob [6]	5.1	13.4 7	14.3 7	20.3 14	4.55 4	5.38 5	15.7 8	11.0 3	18.5 4	26.1 4	**4.93** 1	12.5 2	13.5 2
AdaptingBP [17]	5.3	19.1 13	19.3 11	17.4 5	4.84 6	5.08 4	**7.84** 1	12.8 5	16.7 2	26.3 5	7.02 4	13.2 5	14.0 3
Segm+visib [4]	6.0	12.7 5	12.9 5	15.8 4	10.4 13	11.0 13	19.5 12	11.0 2	**13.2** 1	23.7 2	8.12 6	13.1 4	17.3 5
DoubleBP [15]	7.6	18.7 11	19.1 10	15.8 3	7.85 10	8.38 9	11.6 5	14.3 6	19.9 5	24.3 3	11.9 10	18.1 9	19.9 10
GenModel [20]	8.4	7.89 4	10.0 4	18.5 9	4.59 5	6.03 6	23.5 19	14.8 8	22.8 9	31.8 8	10.2 7	20.2 13	19.0 9
SymBP+occ [7]	9.1	20.7 16	21.6 16	19.5 13	5.96 7	6.27 7	10.2 2	15.7 9	20.9 7	31.7 7	11.4 9	17.5 8	18.9 8
CostRelax [11]	9.8	26.3 24	27.3 24	33.5 22	**2.92** 1	4.06 2	20.8 14	12.3 4	20.2 6	32.4 10	6.33 3	13.1 3	16.7 4
TensorVoting [9]	9.8	25.5 23	26.2 23	21.2 15	3.32 3	4.12 3	14.6 7	14.6 7	21.8 8	33.3 12	7.05 5	14.5 6	17.4 6
GC+occ [2]	10.3	**6.10** 1	**7.11** 1	**14.6** 1	10.7 14	11.3 14	16.9 10	23.7 16	30.1 16	34.6 15	12.2 12	19.2 11	21.9 13

Summary of Approach to Stereo

- Binocular and monocular cues are combined
- Novel initial matching framework
- No image segmentation
- Occluding surfaces do not over-extend because of color consistency requirement
- Textureless surfaces are inferred based on surface smoothness
  - When initial matching fails

Multiple-View Stereo

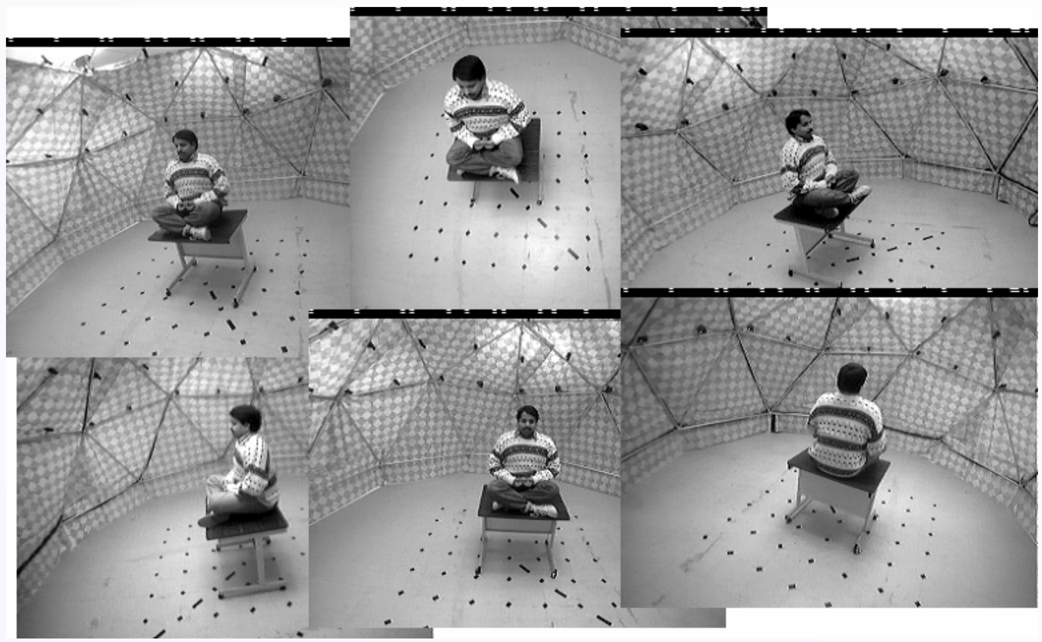
- Approach to dense multiple-view stereo
  - Multiple views: more than two
  - Dense: attempt to reconstruct all pixels
- Process all data simultaneously
  - Do not rely on binocular results
  - Only binocular step: detection of potential pixel correspondences
- Correct matches form coherent salient surfaces
  - Infer them by Tensor Voting

Desired Properties

- General camera placement
  - As long as camera pairs are close enough for automatic pixel matching
- No privileged images
- Features required to appear in no more than two images
- Reconstruct background
  - Do not discard
- Simultaneous processing
  - Do not merge binocular results

Input Images

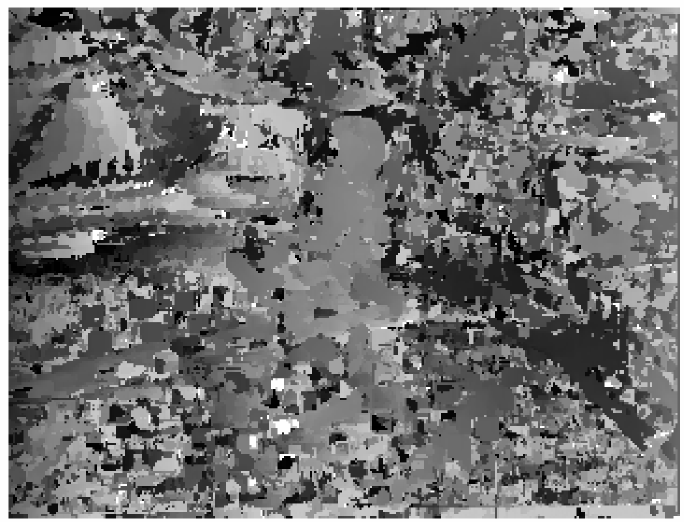
Captured at the CMU dome for the *Virtualized Reality* project



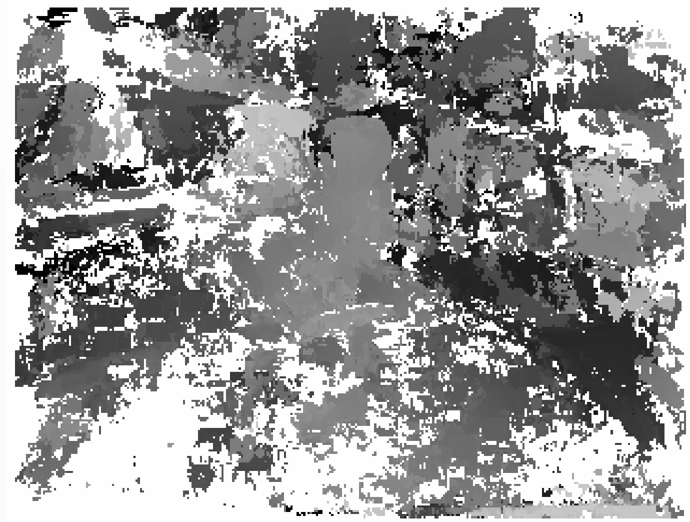
Limitations of Binocular Processing

Errors due to:

- Occlusion
- Lack of texture



Matching candidates in disparity space



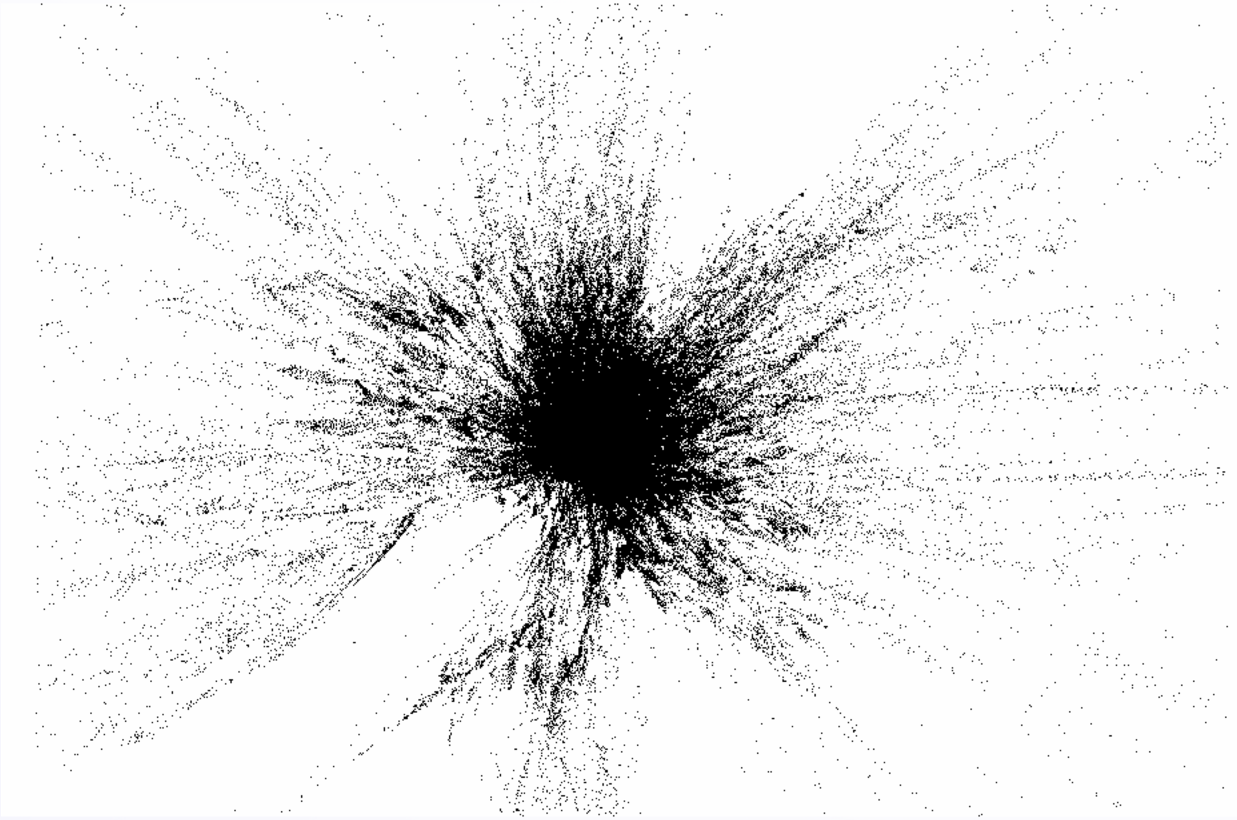
Disparity map after binocular processing

Limitations of Binocular Processing

When matching from many image pairs, candidates are combined:

- Lessens effects of depth discontinuities
  - Occluded surfaces are revealed
- Salient surfaces are reinforced by correct matches from multiple pairs
  - Noise is not

Candidate Matches

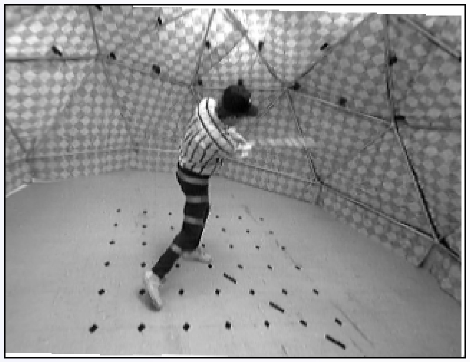


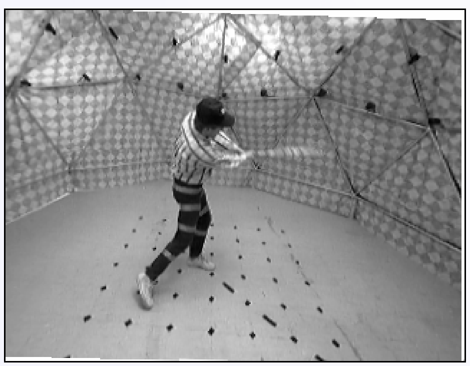
10 image pairs, 1.1 million points
Tensor voting takes 44 min. 30 sec. (March 2004)

Results on “Meditation” Set

Results on “Meditation” Set

Results on "Baseball" Dataset





Overview

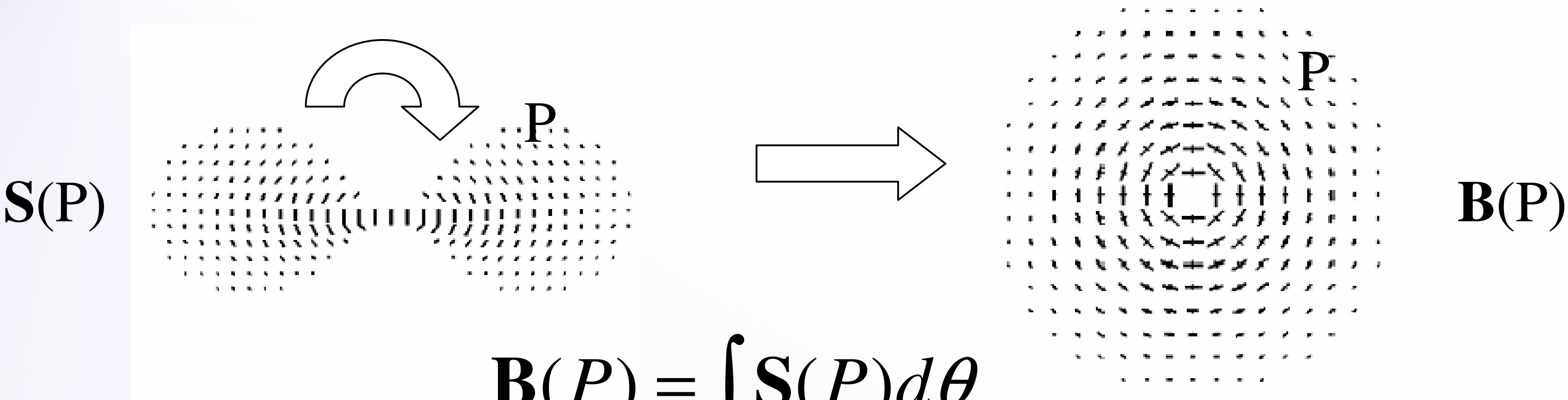
- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

Tensor Representation in N-D

- Non-accidental alignment, proximity, good continuation apply in N-D
  - Robot arm moving from point to point forms 1-D trajectory (manifold) in N-D space
- Noise robustness and ability to represent all structure types also desirable

- Tensor construction:
  - eigenvectors of normal space associated with non-zero eigenvalues
  - eigenvectors of tangent space associated with non-zero eigenvalues

Limitations of Voting Fields

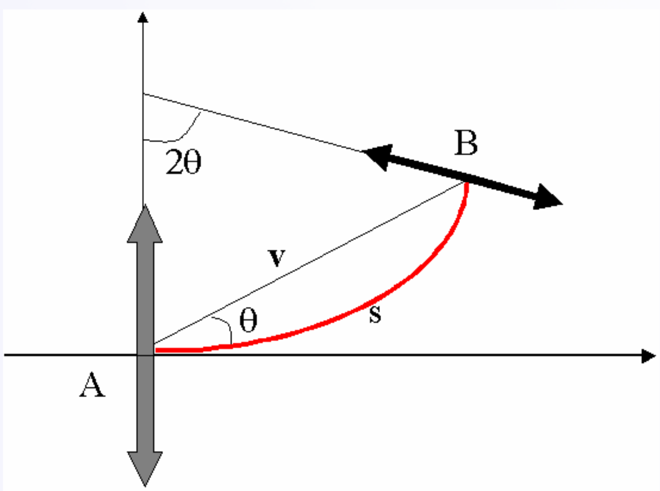


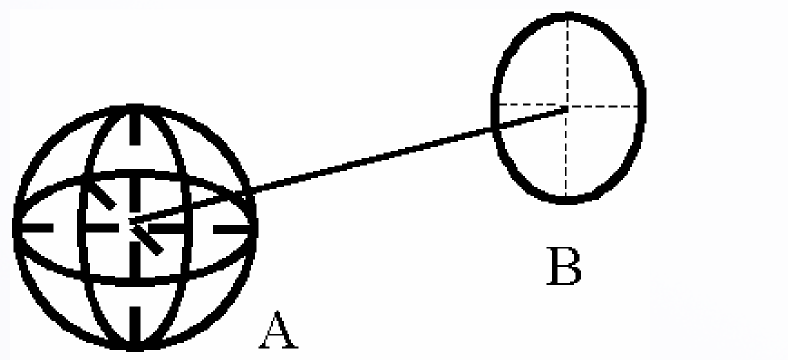
$$\mathbf{B}(P) = \int \mathbf{S}(P)\,d\theta$$

- Hard to generalize to *N* dimensions
- Requirements: *N* *N*-D fields
  - k samples per axis: $O(Nk^N)$ storage requirements
  - N^{th} order integration to compute each sample: $O(k^d)$ computations per sample

Efficient N-D implementation

- Drop uncertainty from vote computation
- Cast votes directly without integration
  - Votes from stick tensors are computed in 2-D subspace regardless of *N*
  - Ball tensors cast votes that support straight lines from voter to receiver



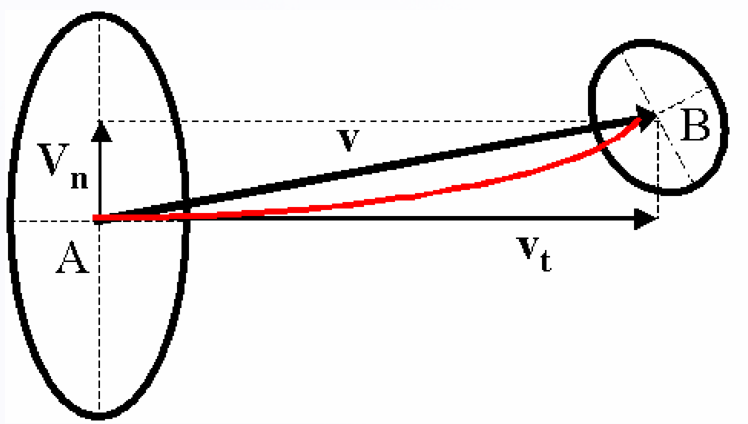


$$\mathbf{T}(\mathrm{s},\theta) = e^{-(\frac{s^2}{\sigma^2})}(\mathcal{I} - \frac{\vec{v}\vec{v}^T}{\|\vec{v}\vec{v}^T\|})$$

Efficient N-D implementation

- Simple geometric solution for arbitrary tensors
- Observation: curvature only needed for saliency computation when θ not zero
- $\mathbf{v_n}$ projection of vector **AB** on normal space of voter
- Define basis for voter that includes $\mathbf{v_n}$
  - 1 vote computation that requires curvature
  - At most *N*-2 vote computations that are scaled stick tensors parallel to voters

$$\mathbf{T} = vote(\vec{b}_1) + \sum_{i \in [2,d]} vote(\vec{b}_i)$$



Vote Analysis

- Tensor decomposition:

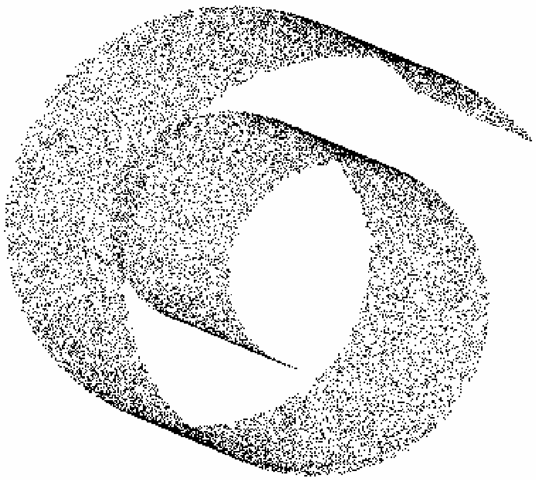
$$\mathbf{T} = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T + \ldots + \lambda_N \cdot e_N e_N^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) + \ldots + \lambda_N \sum_d e_d e_d^T$$

- Dimensionality estimate:
  d with max$\{\lambda_d - \lambda_{d+1}\}$
- Orientation estimate: normal subspace spanned by *d* eigenvectors corresponding to *d* largest eigenvalues

Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

Dimensionality Estimation



"Swiss Roll"

σ^2	Correct Dim. Estimation (%)	Perc. of Dist. Recovered (%)	Time (sec)
50	99.25	93.07	7
100	99.91	93.21	13
200	99.95	93.19	30
300	99.92	93.16	47
500	99.68	93.03	79
700	99.23	92.82	112
1000	97.90	92.29	181

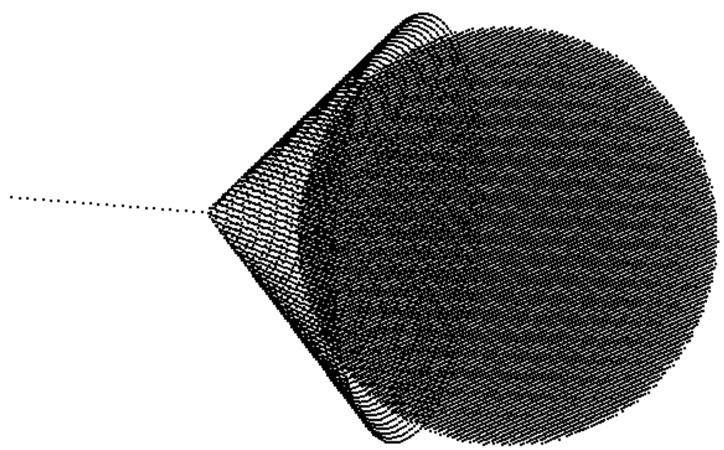
Dimensionality Estimation

- Synthetic data
- Randomly sample input variables (intrinsic dimensionality)
- Map them to higher dimensional vector using linear and quadratic functions
- Add noisy dimensions
- Global rotation

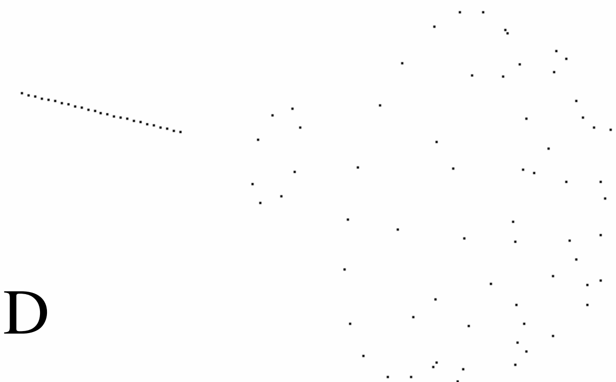
Intrinsic Dim.	Linear Mappings	Quadratic Mappings	Space Dim.	Dim. Est. (%)
4	10	6	50	93.6
3	8	6	100	97.4
4	10	6	100	93.9
3	8	6	150	97.3

Dimensionality Estimation

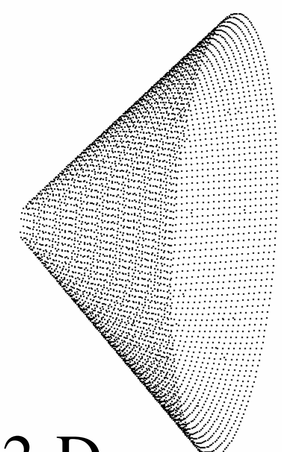
- Point-wise dimensionality estimates
- No global operations



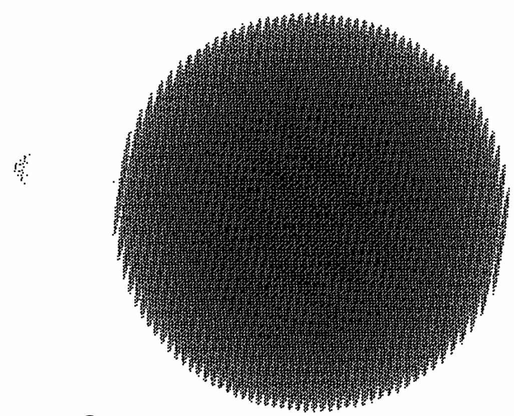
Input in 4-D



1-D



2-D



3-D

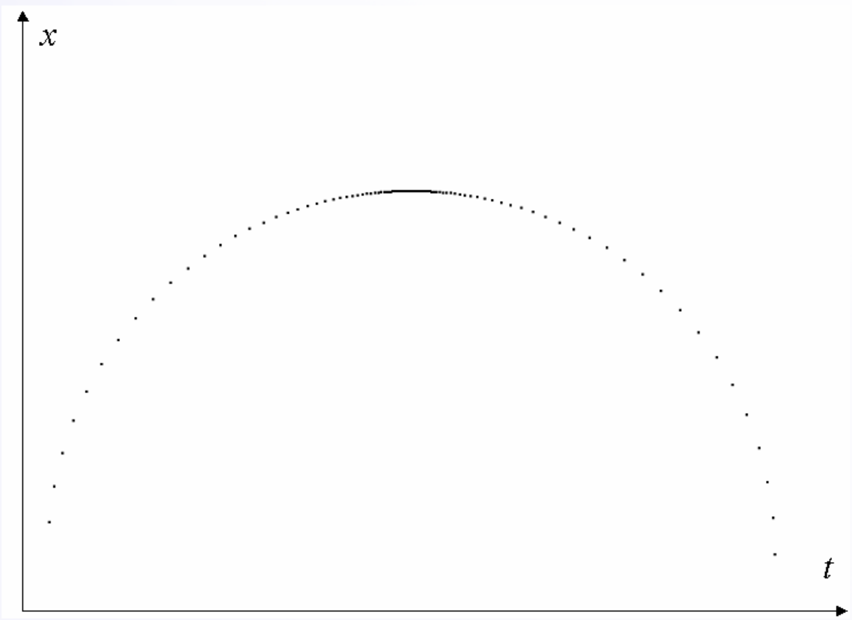
Manifold Learning

- Input: instances (points in *N*-D space)
- Try to infer local structure (manifold) assuming coherence of underlying mechanism that generates instances

- Tensor voting provides:
  - Dimensionality
  - Orientation

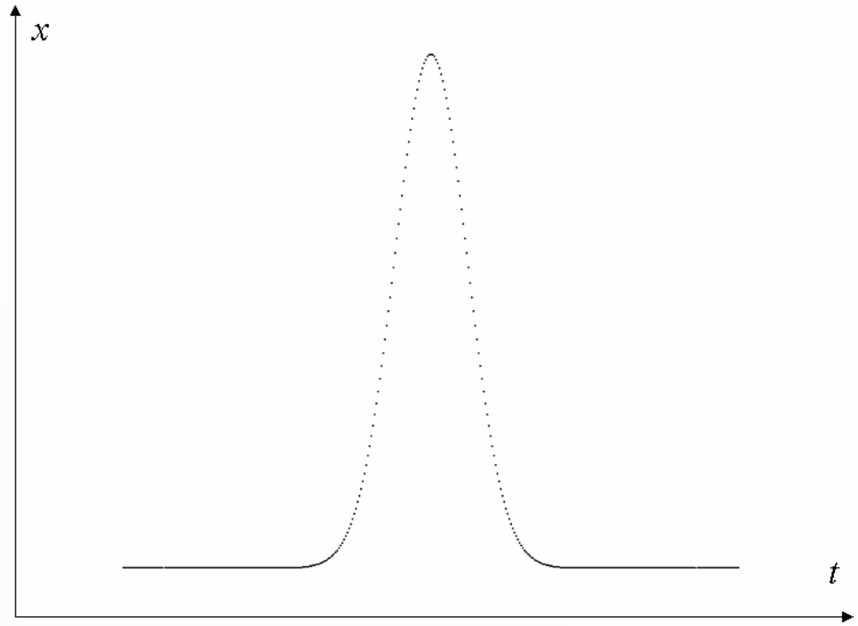
Orientation Estimation

- Functions proposed by [Wang 2004]
- Challenge: non-uniform sampling



$$x_i = [cos(t_i), sin(t_i)]^T$$
$$t_i \in [0, \pi], t_{i+1} - t_i = 0.1(0.001 + |cos(t_i)|$$

Mean error:
0.40 degrees for 152 samples

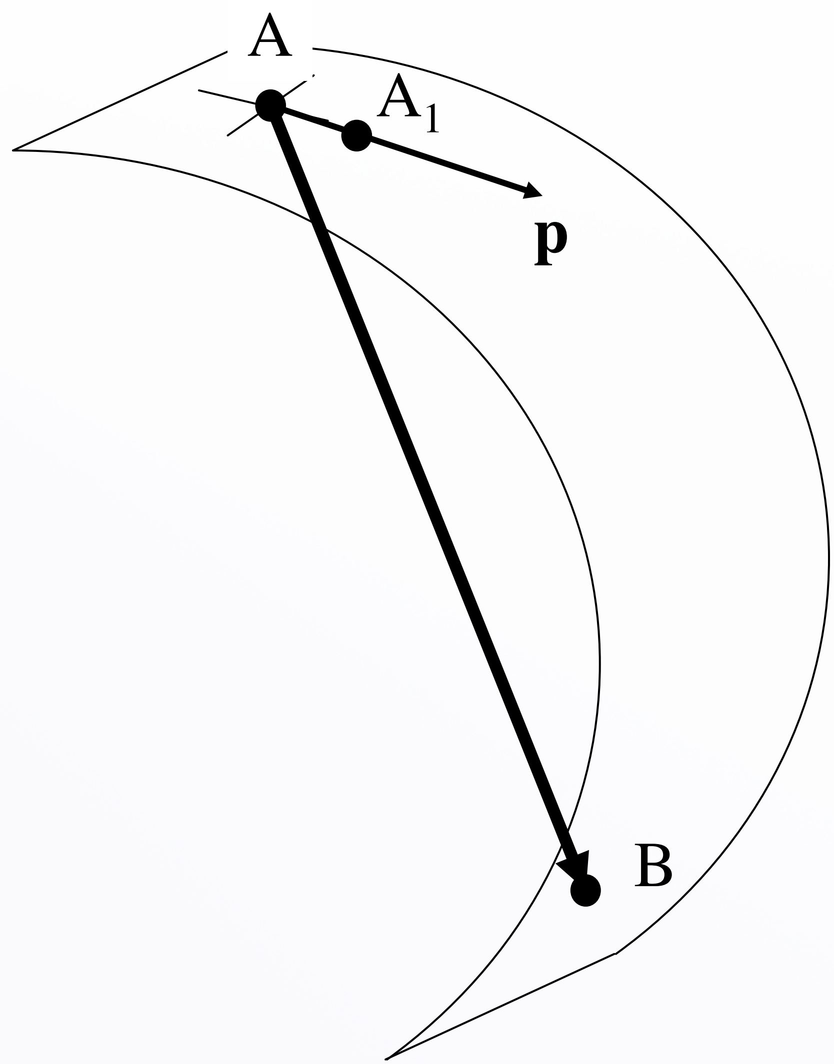


$$x_i = [t_i, \quad 10e^{-t_i^2}]$$

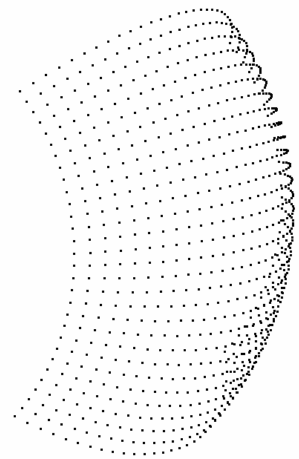
Mean error:
2.02 degrees for 180 samples
0.57 degrees for 360 samples

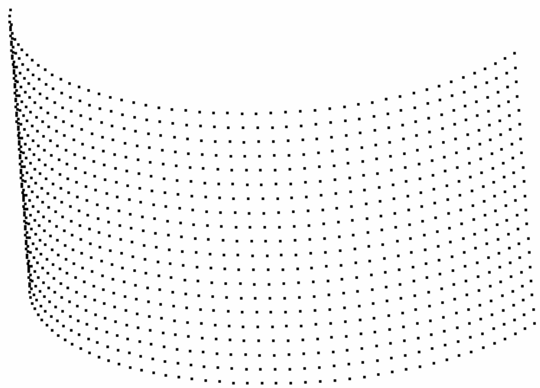
Manifold Distance Measurement

- Do *not* reduce dimensionality
- Start from point on manifold
- Take small step along desired orientation on tangent space
- Generate new point and collect votes
- Repeat until convergence



Distance Measurement: Test Data





Error in orientation estimation: 0.11° 0.26°

- Test data: spherical and cylindrical sections
  - Almost uniformly sampled
  - Ground truth distances between points are known
- Goal: compare against algorithms that preserve pair-wise properties of being far away or close

Experimental Setup

- Comparison of five leading manifold learning algorithms and our approach in distance measurement
- Randomly select pairs of points on the manifold, measure their distance in embedded space and compare with ground truth
  - Apply uniform scaling for algorithms where original distance metric is not preserved

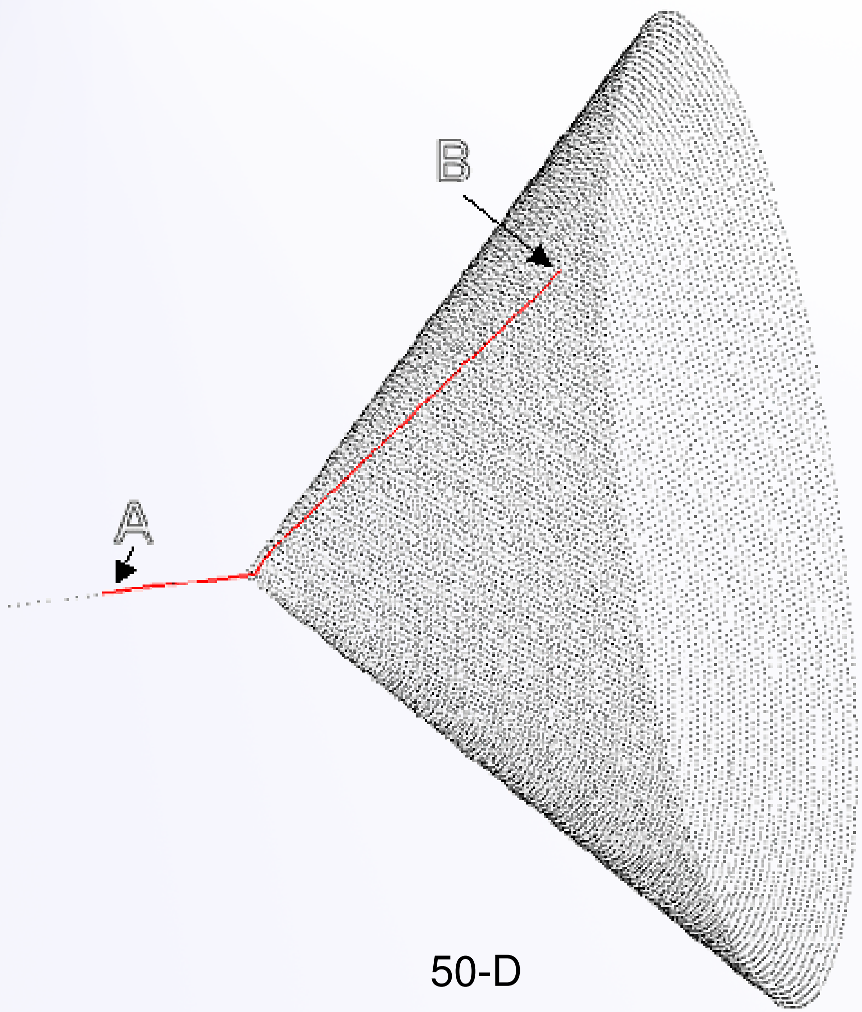
Dataset	Sphere		Cylinder	
	K	Err(%)	K	Err(%)
LLE	18	5.08	6	26.52
Isomap	6	1.98	30	0.35
Laplacian	16	11.03	10	29.36
HLLE	12	3.89	40	26.81
SDE	2	5.14	6	25.57
TV (σ^2)	60	**0.34**	50	**0.62**

Distance Measurement with Outliers

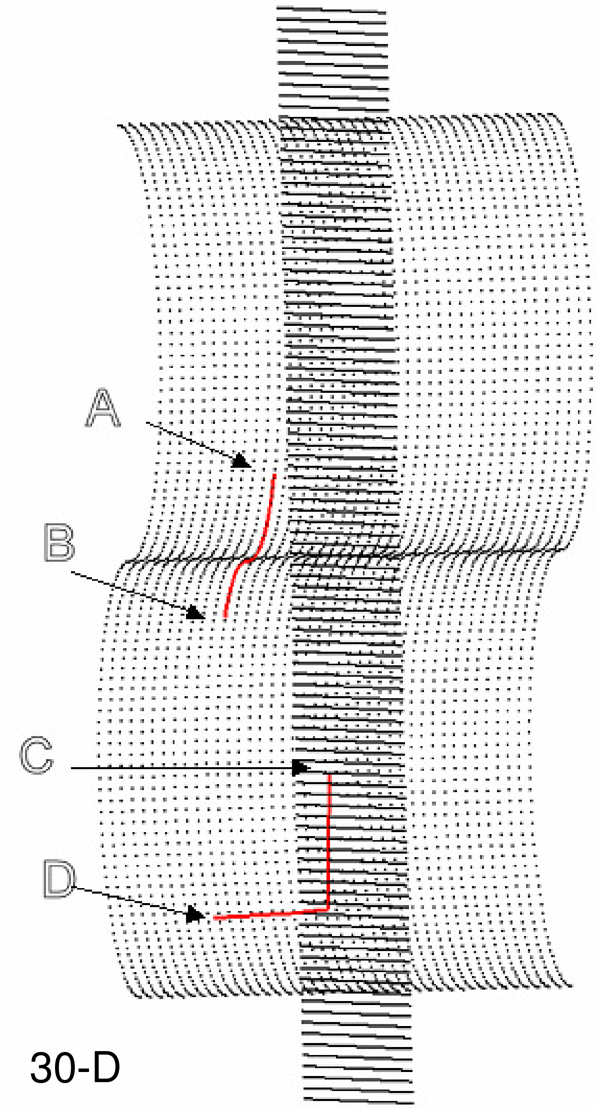
Dataset	Sphere		Cylinder	
	900	outliers	900	outliers
	K	Err(%)	K	Err(%)
LLE	40	60.74	6	15.40
Isomap	18	3.54	14	11.41
Laplacian	6	13.97	14	27.98
HLLE	30	8.73	30	23.67
SDE		N/A		N/A
TV (σ)	70	0.39	100	0.77

Dataset	σ	Error rate
Sphere (3000 outliers)	80	0.47
Sphere (5000 outliers)	100	0.53
Cylinder (3000 outliers)	100	1.17
Cylinder (5000 outliers)	100	1.22

Traveling on Manifolds



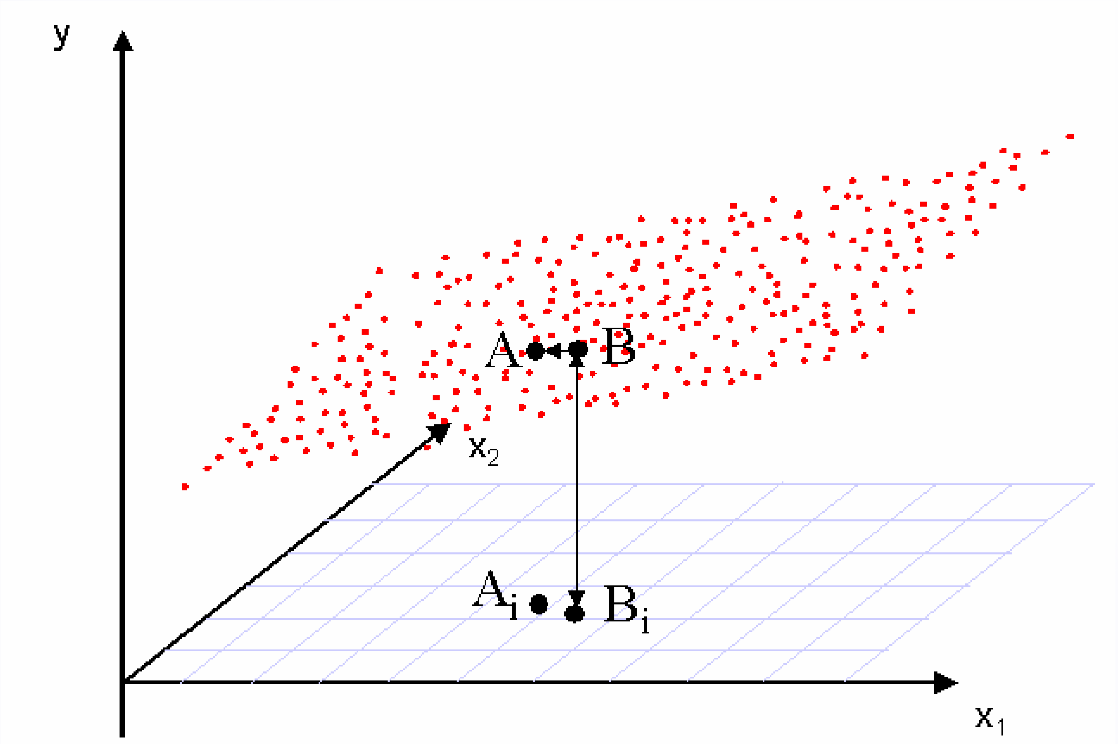
50-D



30-D

Function Approximation

- Problem: given point A_i in input space predict output value(s)
- Find neighbor B_i with known output
- Starting from B in joint input-output space, interpolate until A is reached
  - A projects on input space within ε of A_i

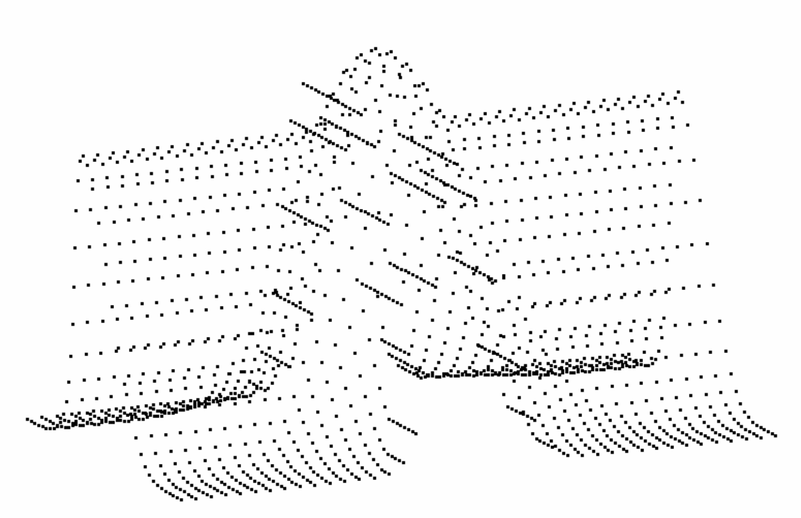


Synthetic Data

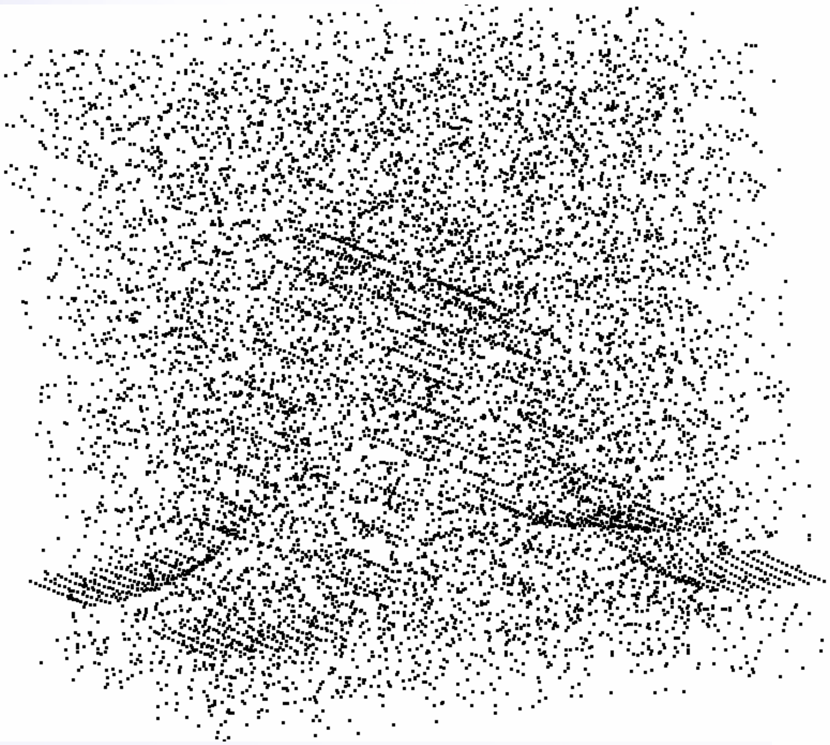
- Sample 1681 points from

$$y = max\{e^{-10x_1^2}, \quad e^{-50x_2^2}, \quad 1.25e^{-5(x_1^2+x_2^2)}\}$$

  proposed by Schall and Atkenson, 1998
- Perform tensor voting and generate new points
  - On original noise-free data
  - On data with 8405 outliers
  - On data with 8405 outliers and Gaussian perturbation of the inliers (σ=0.1)
  - Same data embedded in 60-D



Synthetic Data



Noisy input

New points

Synthetic Data

New points for data with outliers and perturbation in 60-D

Experiment	NMSE
Noise-free	0.0041
Outliers	0.0170
Outliers & N(0, 0.01)	0.0349
Outliers & N(0, 0.01) in 60-D	0.0241

NMSE: MSE normalized by variance of noise free input data

Real Data

Function approximation on datasets from:

- University of California at Irvine Machine Learning Repository
- DELVE archive

- Rescale data (manually) so that dimensions become comparable (variance 1:10 instead of original 1000:1)
- Split randomly into training and test sets according to literature
  - Repeat several times

Results on Real Data

Dataset	Dim.	Training	Test	Mean Error
Abalone	9	3000	1177	1.63
Boston Housing	13	481	25	1.27
Computer Activity	22	2000	6192	1.97

Comparable with recently published results using Bayesian Committee Machine, Gaussian Process Regression, Support Vector Regression etc.

Advantages over State of the Art

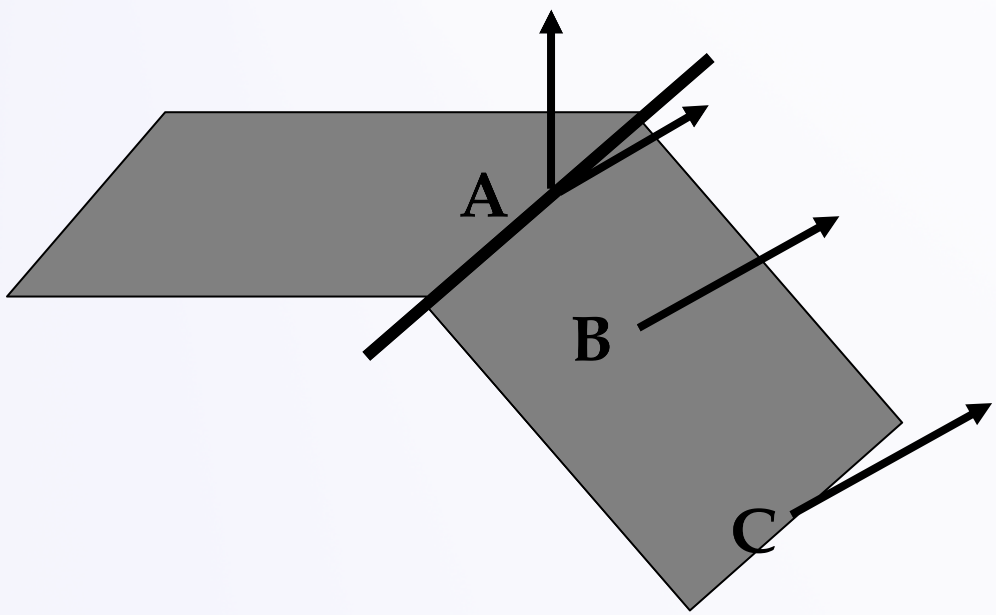
- Broader domain for manifold learning
  - Manifolds with intrinsic curvature (cannot be unfolded)
  - Open and closed manifolds (hyper-spheres)
  - Intersecting manifolds
  - Data with varying dimensionality
- No global computations → O($NM \log M$)
- Noise Robustness

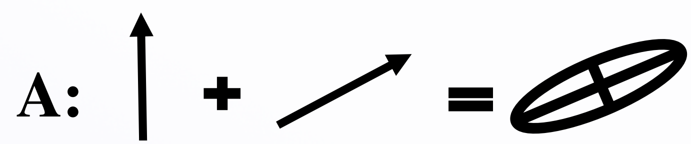
Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

Boundary Inference

- Second-order tensors can represent second order-discontinuities
  - Discontinuous orientation (A)
- But not first-order discontinuities
  - Discontinuous structure (C)





Tensor with dominant plate component (orthogonal to surface intersection)

How to discriminate **B** from **C?**

Boundary Inference: First Order Properties

- Representation augmented with *Polarity Vectors*
- Sensitive to direction from which votes are received
- Boundaries have all their neighbors on the same side of the half-space

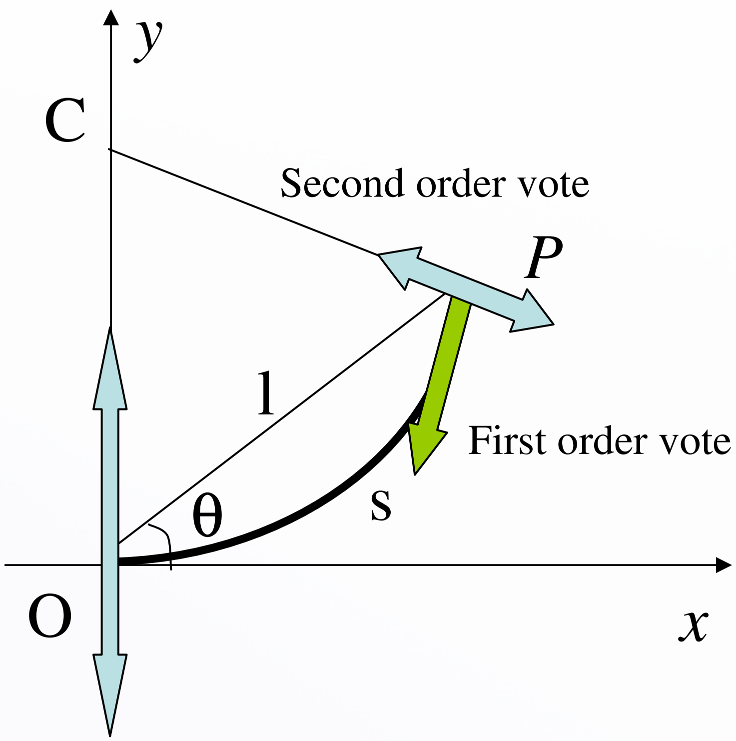
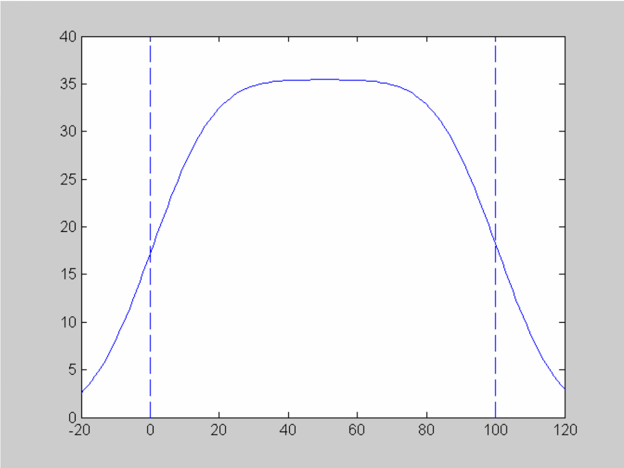


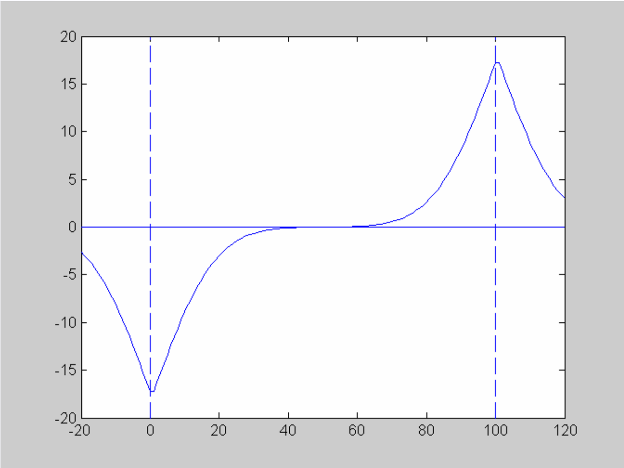
Illustration of Polarity

....

Input

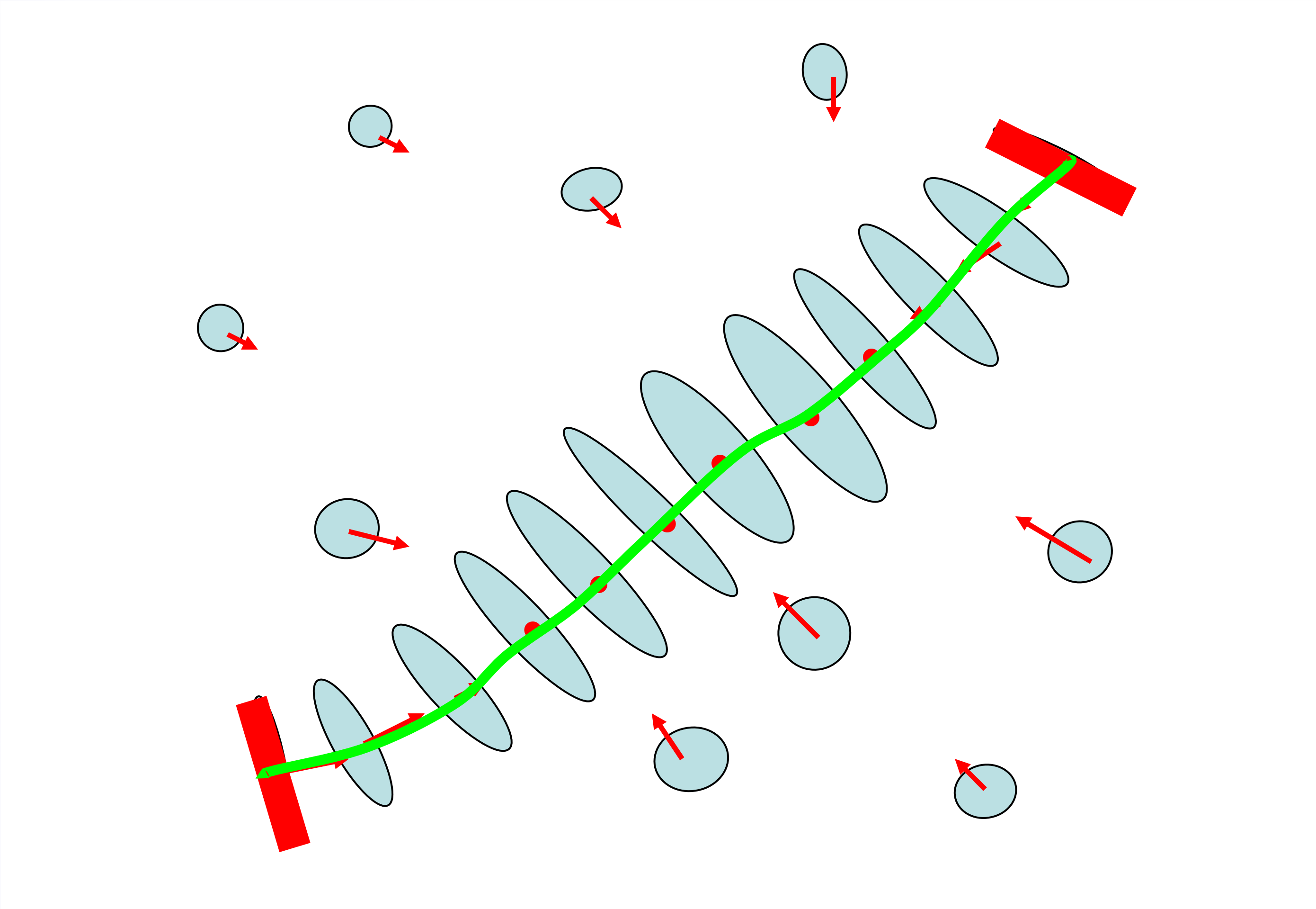


Curve Saliency



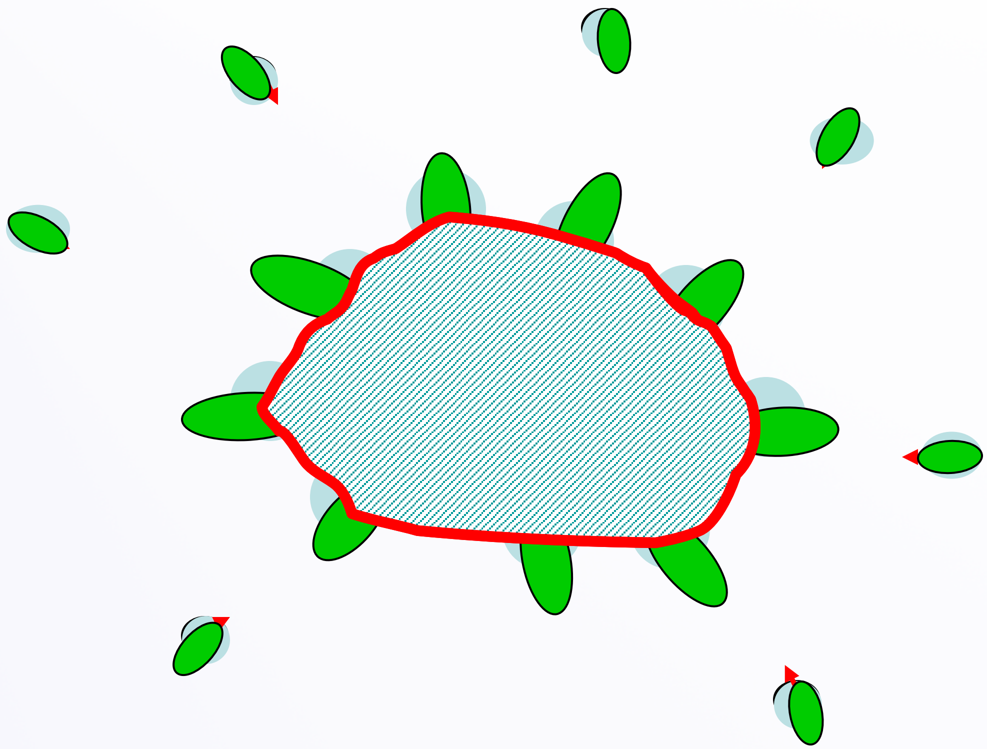
Polarity

Illustration of First Order Voting

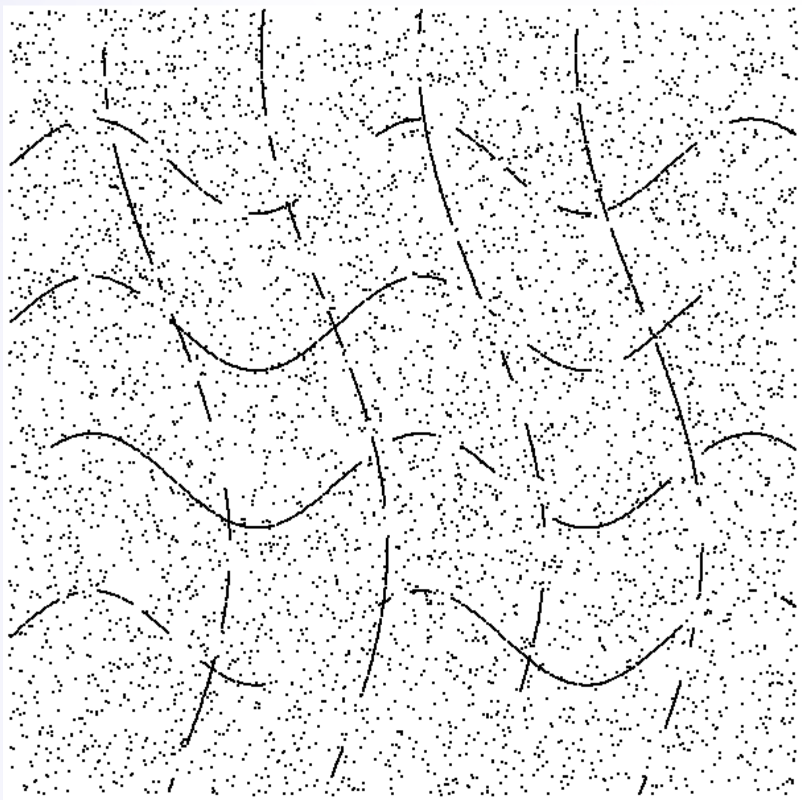


Tensor Voting with first order properties

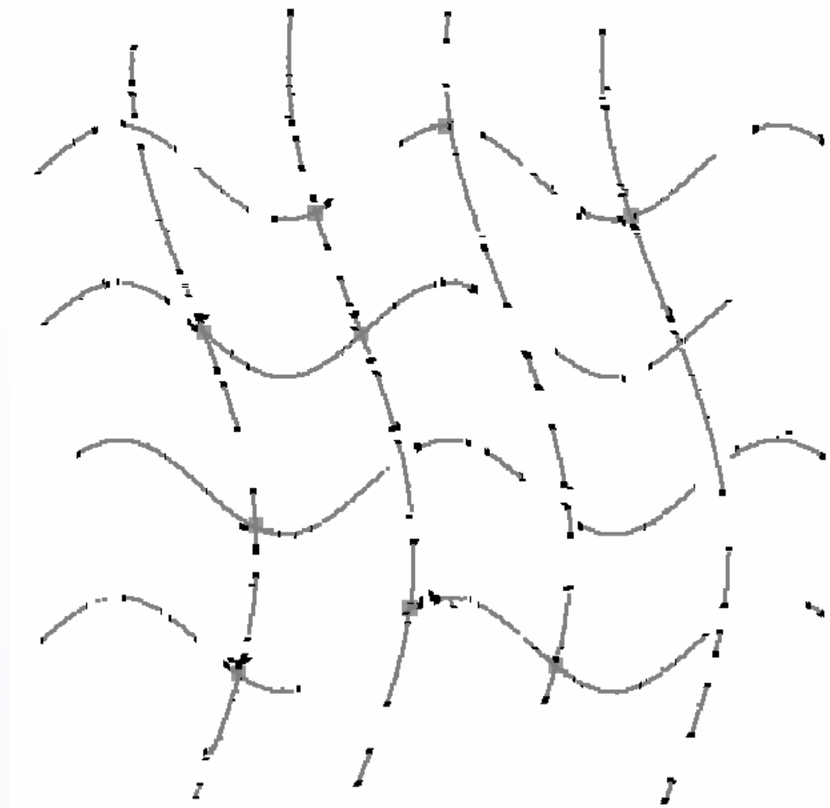
Illustration of Region Inference



Results in 2-D

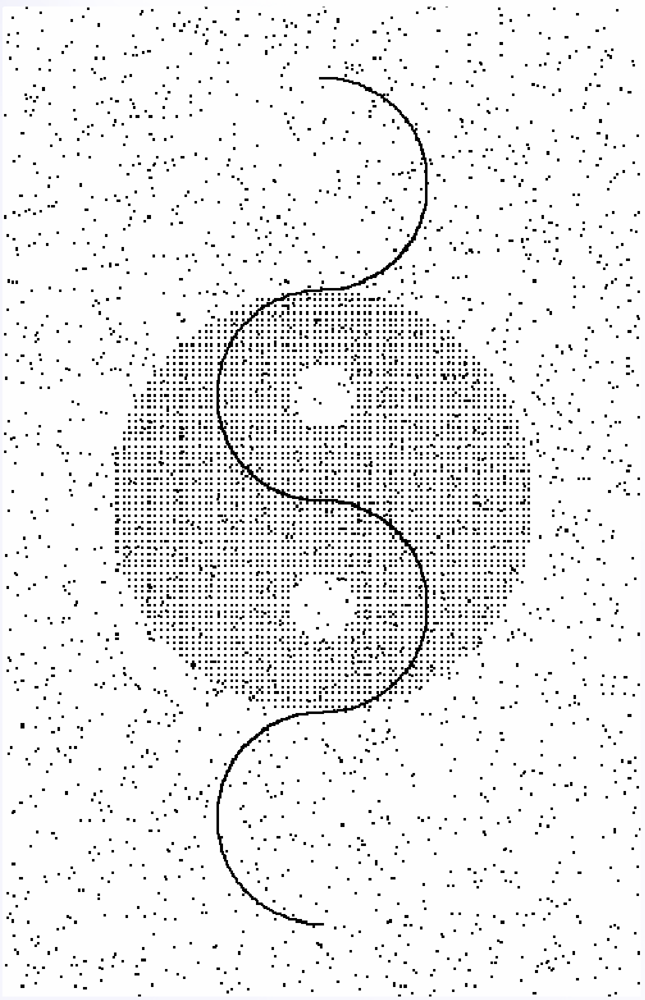


Input

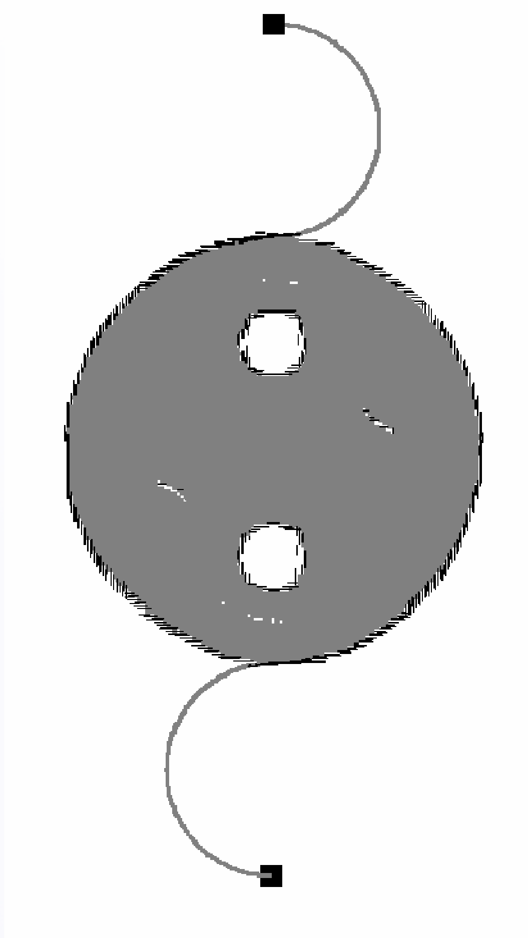


Gray: curve inliers
Black: curve endpoints
Squares: junctions

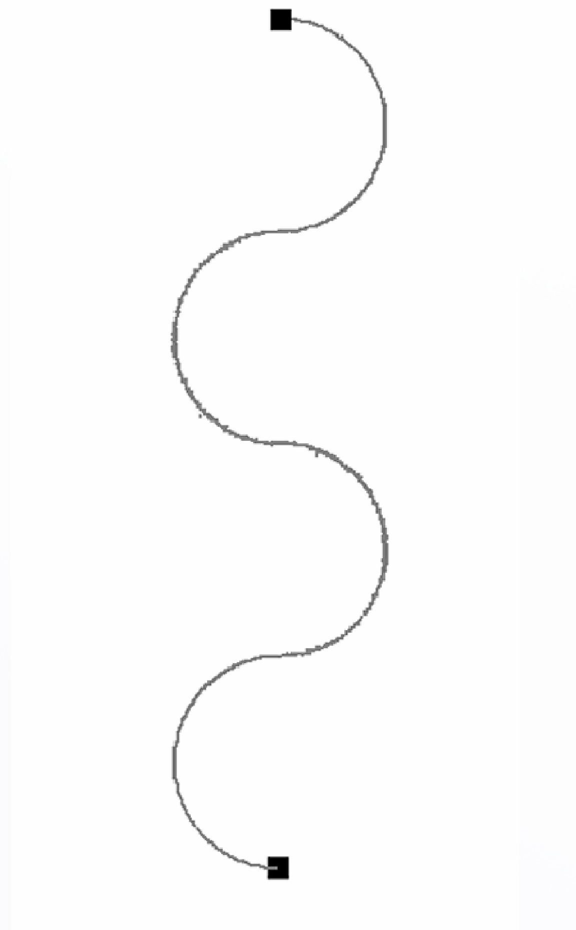
Results in 2-D



Input

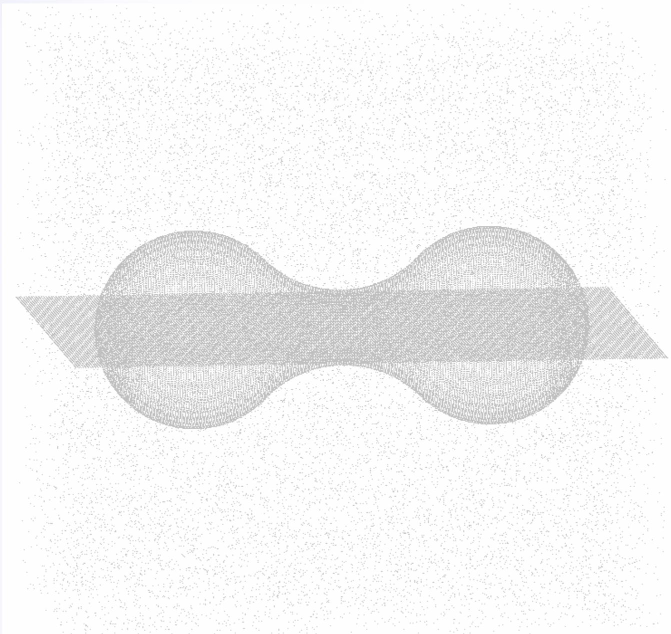


Curves, endpoints,
regions and region boundaries

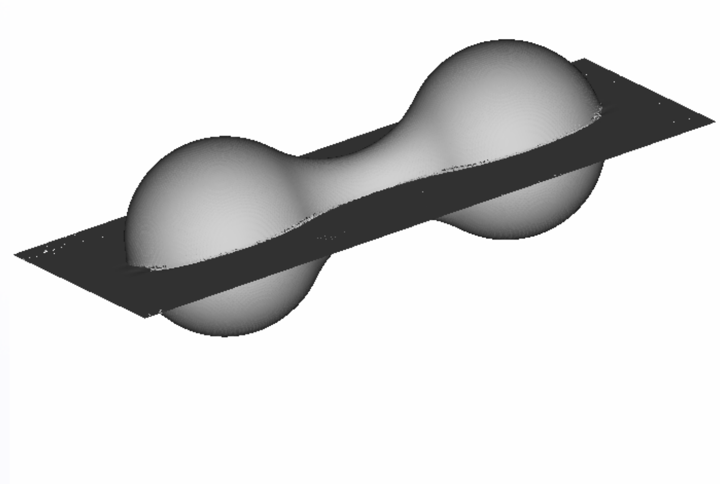


Curves and endpoints only

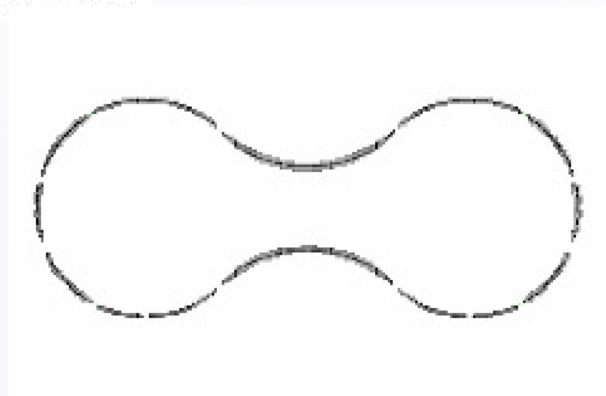
Results in 3-D



Input

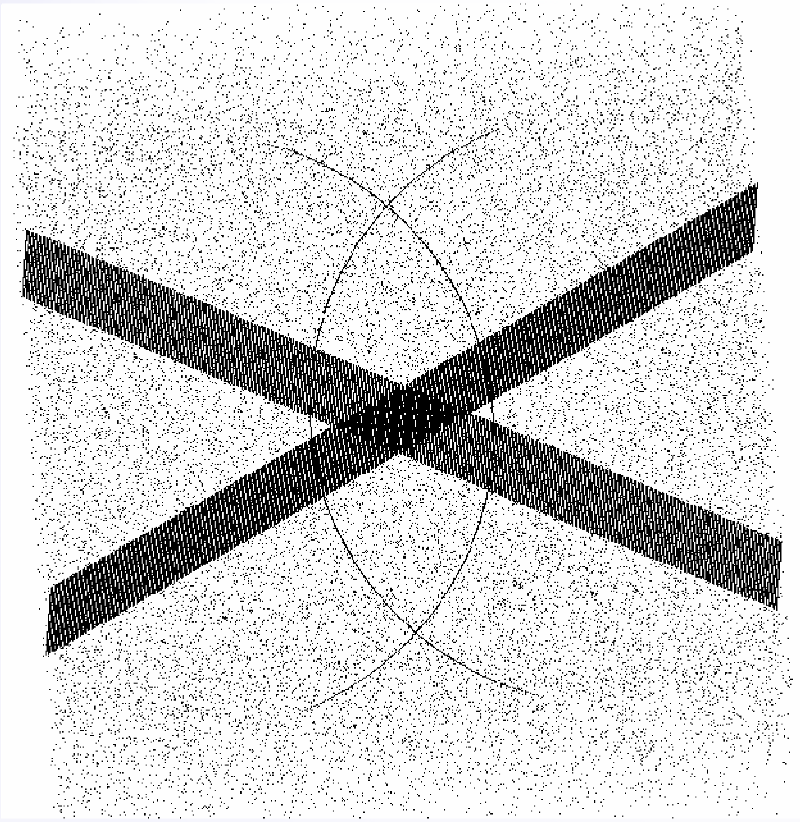


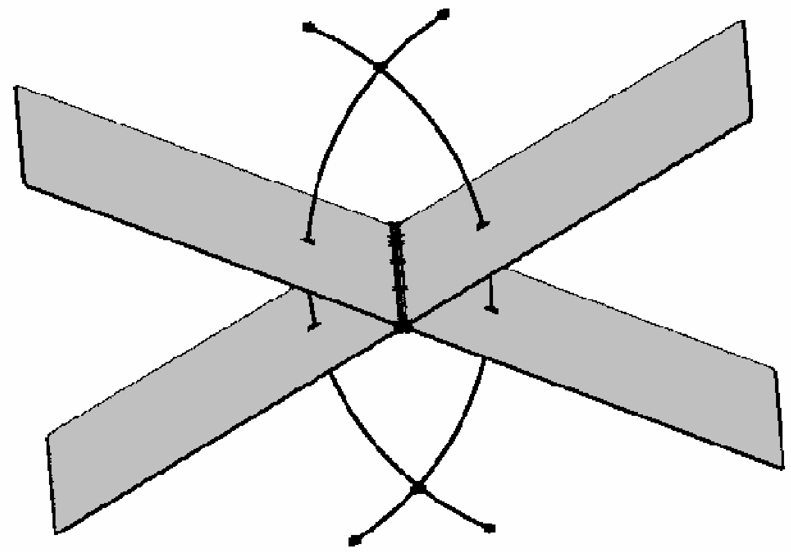
Surfaces



Surface Intersections

Results in 3-D





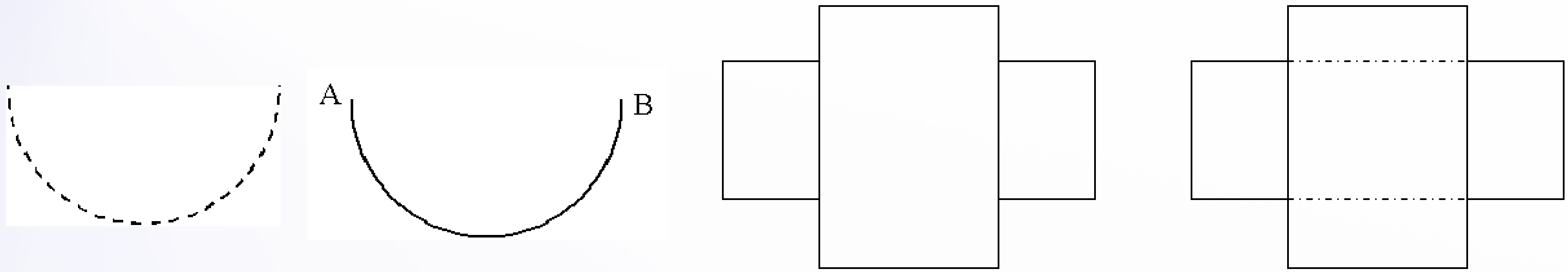
Input

Surfaces - Surface Boundaries – Surface Intersections
Curves – Endpoints - Junctions

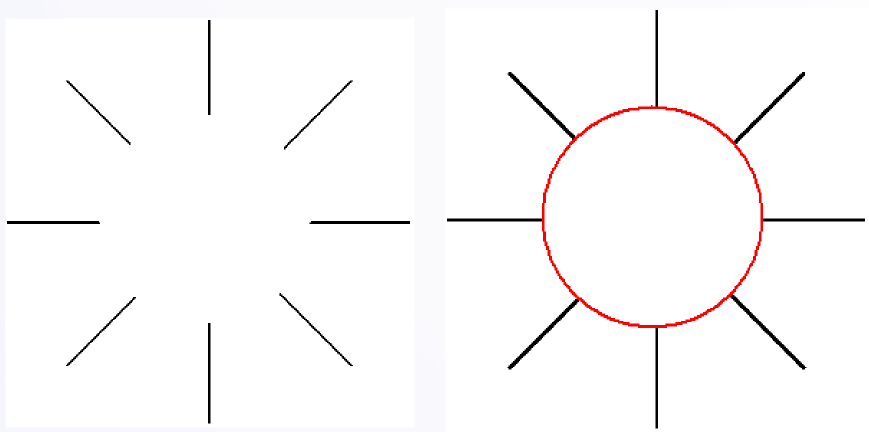
Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

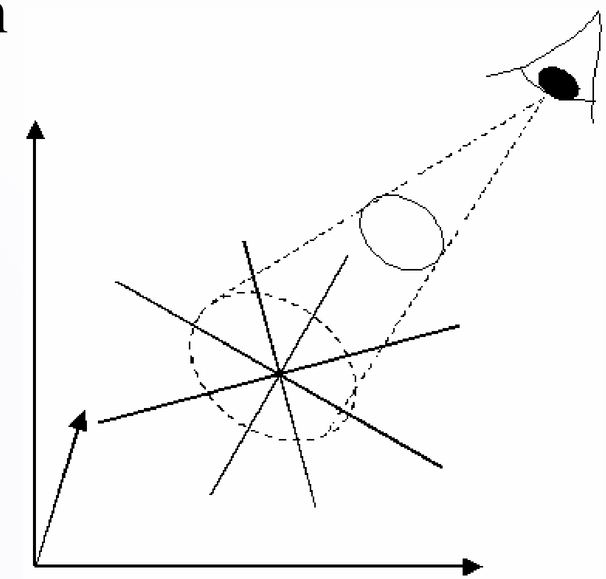
Figure Completion



Amodal completion



Modal completion



Layered interpretation

Motivation

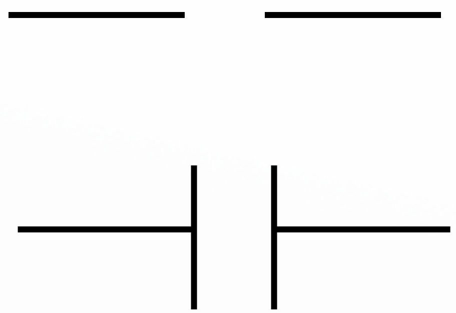
- Approach for modal and amodal completion
- Automatic selection between them
- Explanation of challenging visual stimuli consistent with human visual system

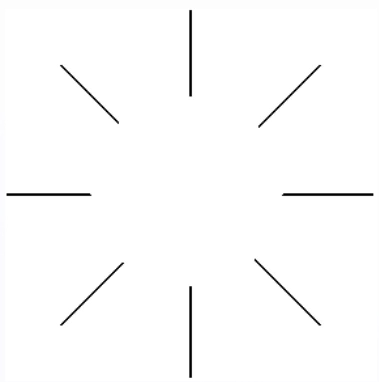
Keypoint Detection

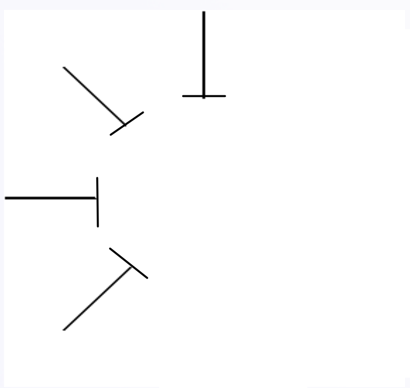
- Input binary images
- Infer junctions, curves, endpoints, regions and boundaries via Tensor Voting
- Look for completions supported by endpoints, L and T-junctions
- W, X and Y-junctions do not support completion by themselves

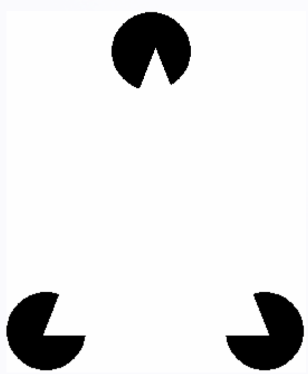
Support for Figure Completion

- Amodal:
  - Along the tangent of endpoints
  - Along the stem of T-junctions
- Modal:
  - Orthogonal to endpoints
  - Along the bar of T-junctions
  - Along either edge of L-junctions



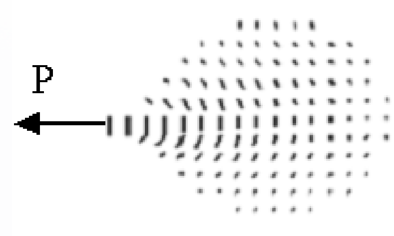


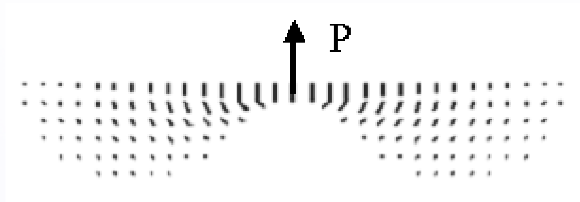


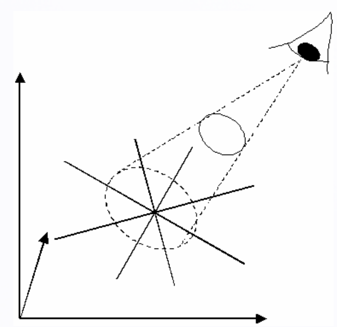


Voting for Completion

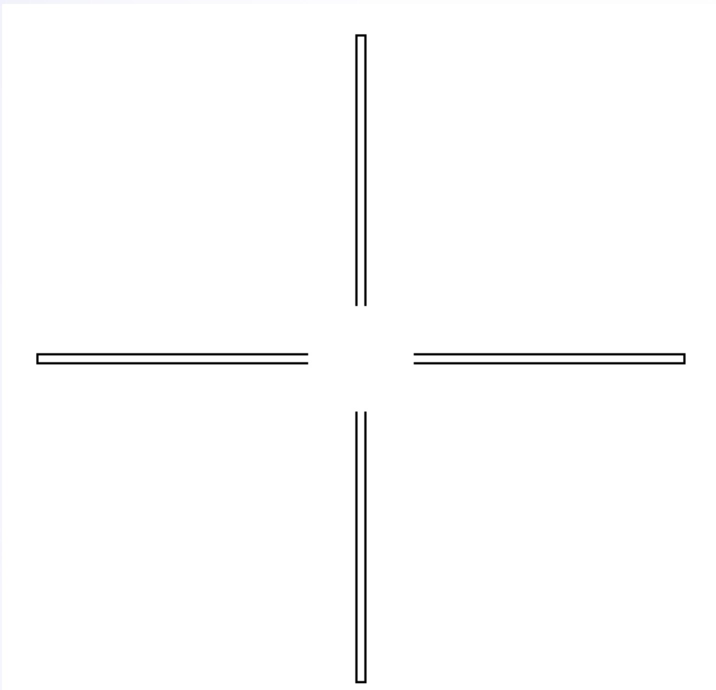
- At least two keypoints of appropriate type needed
- Possible cases:
  - No possible continuation
  - Possible amodal completion (parallel field)
  - Possible modal completion (orthogonal field)
  - If both possibilities available, modal completion is perceived as occluding amodal one



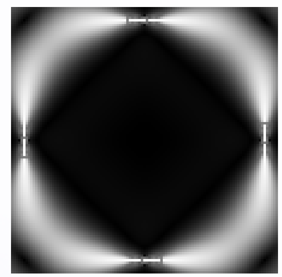




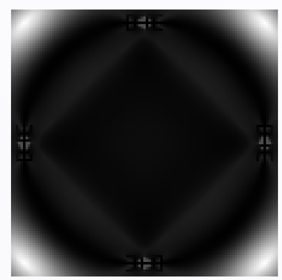
The Koffka Cross



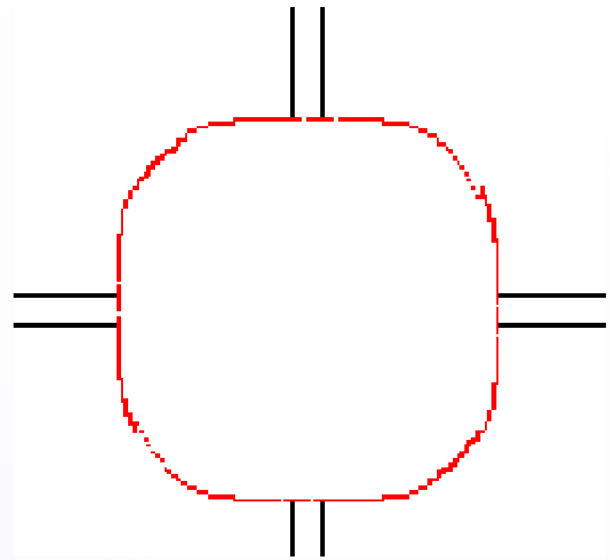
Input



Curve saliency

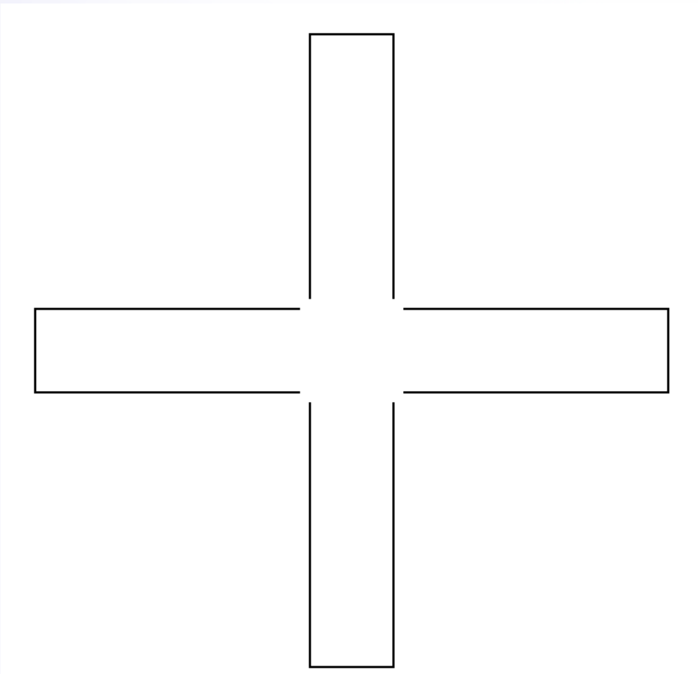


Junction saliency

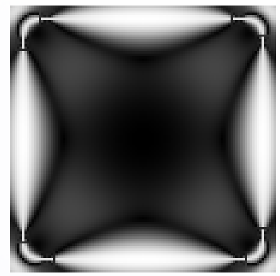


Output

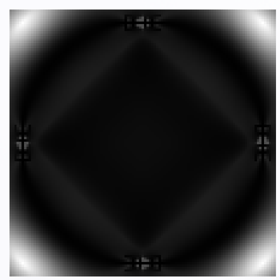
The Koffka Cross



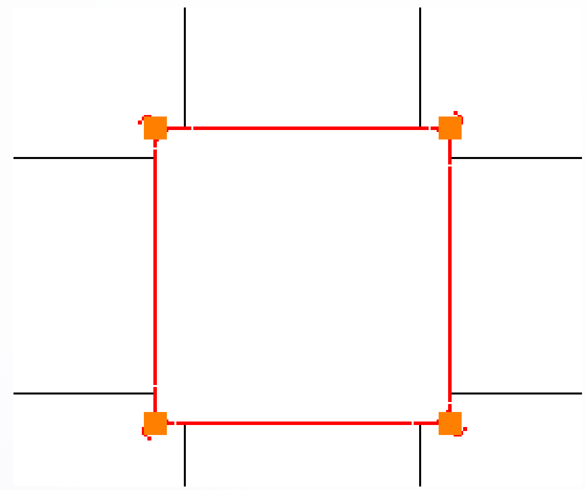
Input



Curve saliency



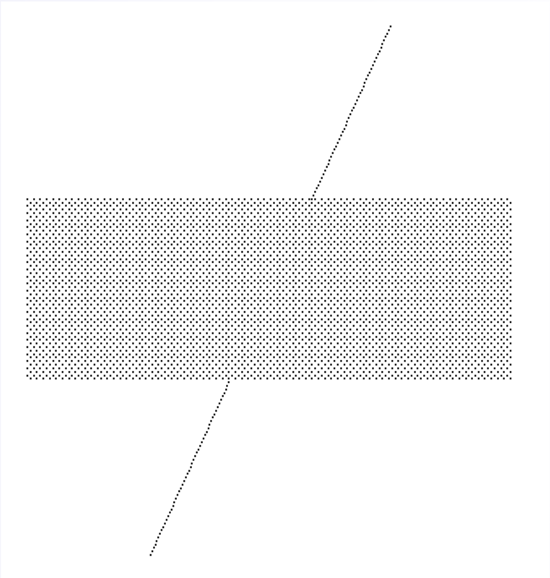
Junction saliency



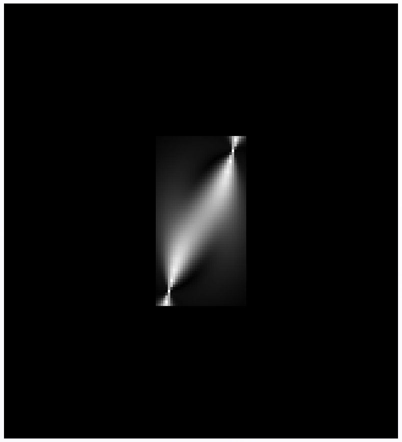
Output

Note: maximum junction saliency here is 90% of maximum curve saliency, but only 10% in the previous case

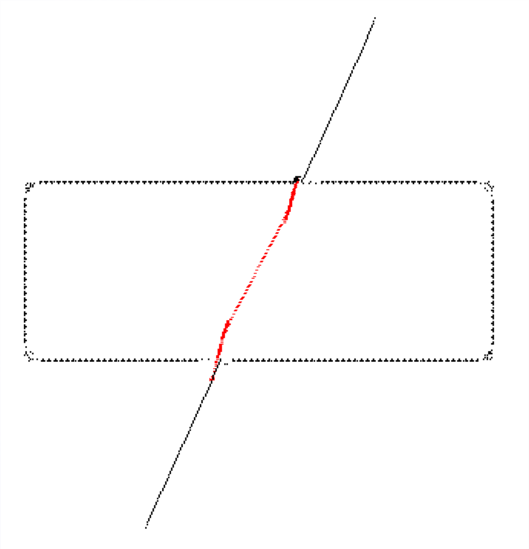
The Poggendorf Illusion



Input

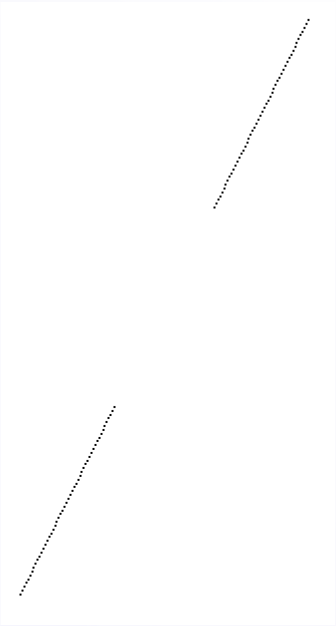


Curve saliency

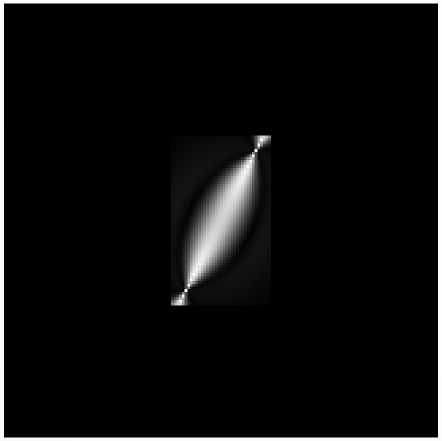


Output

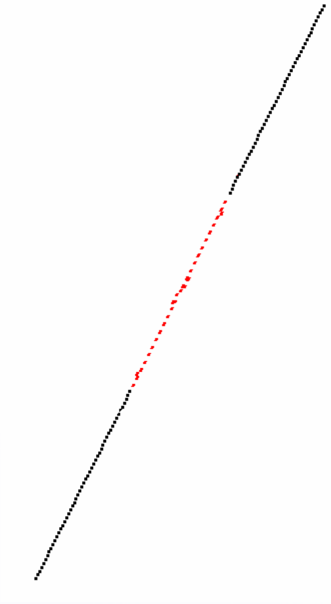
The Poggendorf Illusion



Input



Curve saliency



Output

Overview

- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- Conclusions

Conclusions

- General framework for perceptual organization
- Unified and rich representation for all types of structure, boundaries and intersections
- Model-free

- Applications in several domains