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

Short Biography

- Ph.D. from University of Southern California with Gérard Medioni (2000-2005)
  - Perceptual organization
  - Binocular and multiple-view stereo
  - Feature inference from images
  - Figure completion
  - Manifold learning
  - Dimensionality estimation
  - Function approximation

- Postdoctoral researcher at University of North Carolina with Marc Pollefeys (2005-present)
  - Real-time video-based reconstruction of urban environments
  - Multiple-view reconstruction
  - Temporally consistent video-based reconstruction

Audience

- Academia or industry?
- Background:
  - Perceptual organization
  - Image processing
  - Image segmentation
  - Human perception
  - 3-D computer vision
  - Machine learning
- Have you had exposure to tensor voting before?

Objectives

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

- Applications:
  - Computer vision problems
  - Instance-based learning

Overview

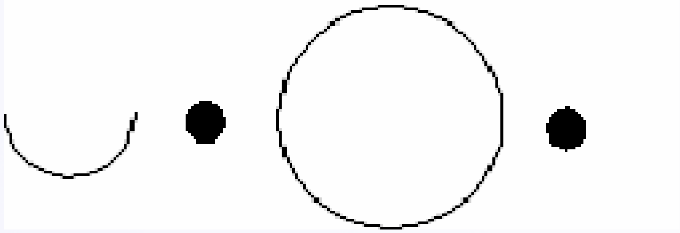
- Introduction
- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- More Tensor Voting Research
- Conclusions

Motivation

- Computer vision problems are often inverse
  - Ill-posed
  - Computationally expensive
  - Severely corrupted by noise
- Many of them can be posed as perceptual grouping of primitives
  - Solutions form *perceptually salient* non-accidental structures (e.g. surfaces in stereo)
  - Only input/output modules need to be adjusted in most cases

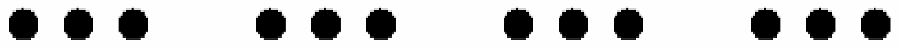
Motivation

- 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

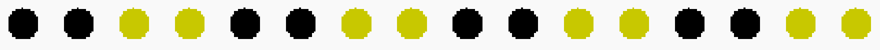


Gestalt Principles

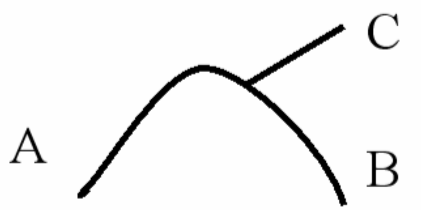
- Proximity



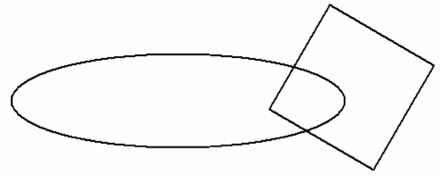
- Similarity



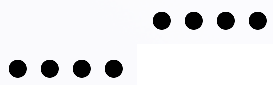
- Good continuation



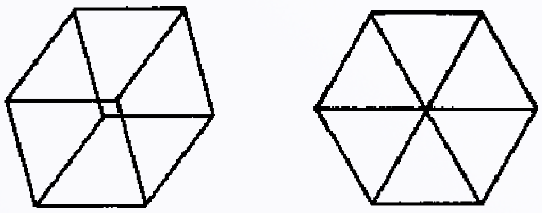
- Closure



- Common fate

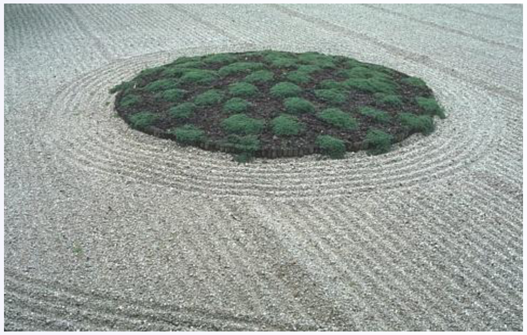


- Simplicity

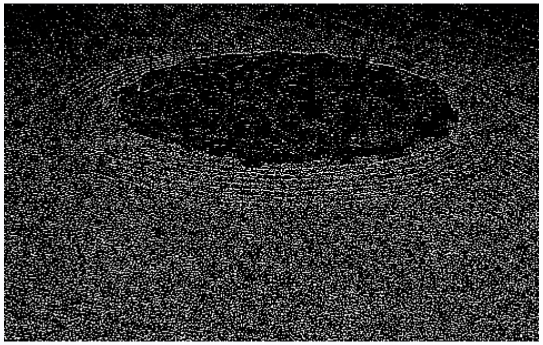


Structural Saliency

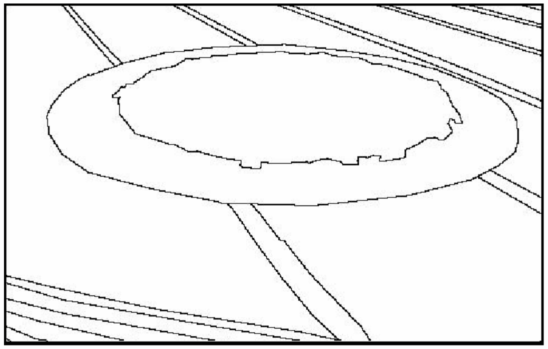
- Property of structures to stand out due to *proximity* and *good continuation*



Input



Edge detector



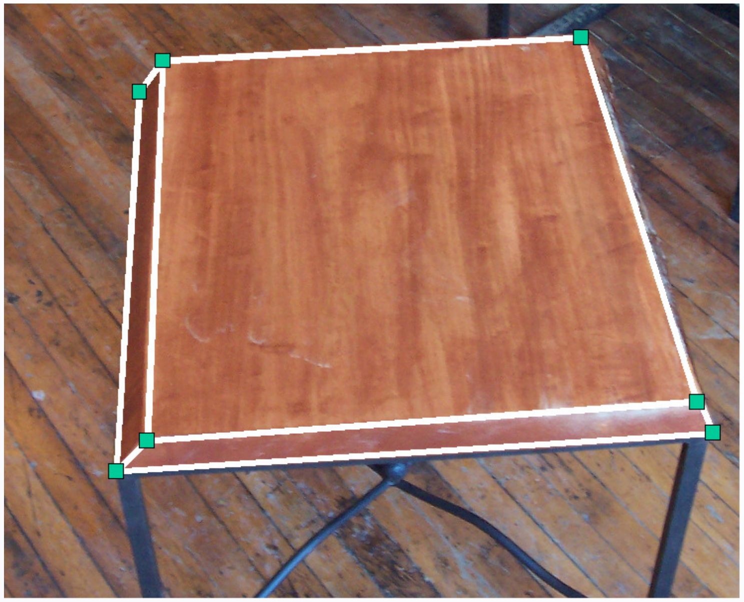
Human observer

- 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

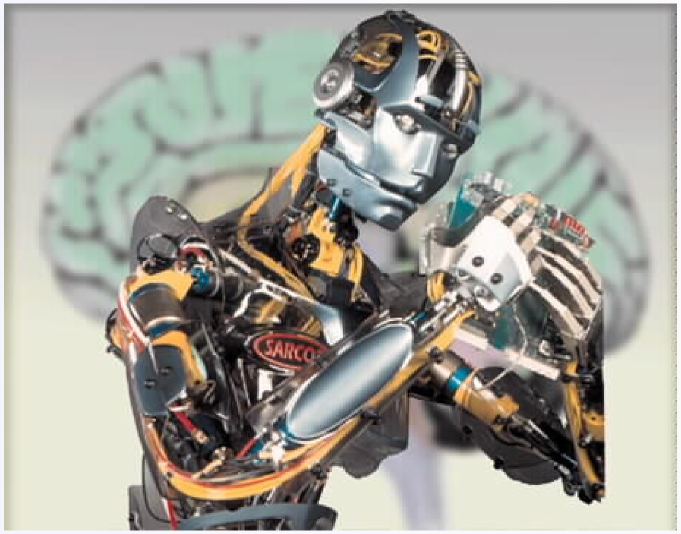


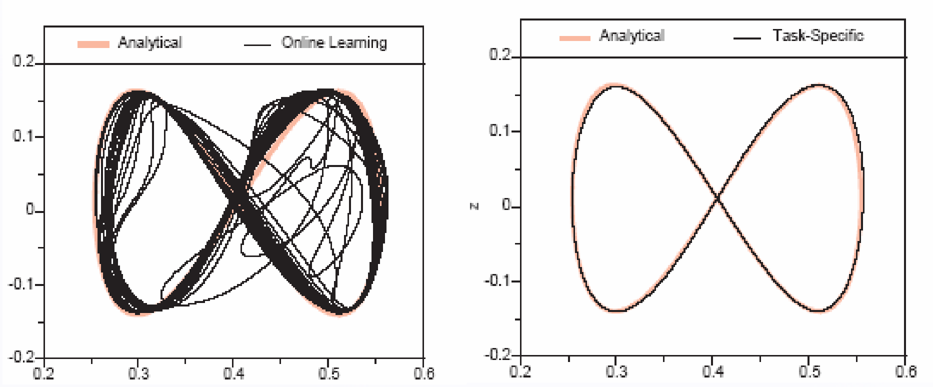
Desired Properties

- Local, data-driven descriptions
  - More general, model-free solutions
  - Local changes affect descriptions locally
  - Global optimization often requires simplifying assumptions (NP-complete problems)
- 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

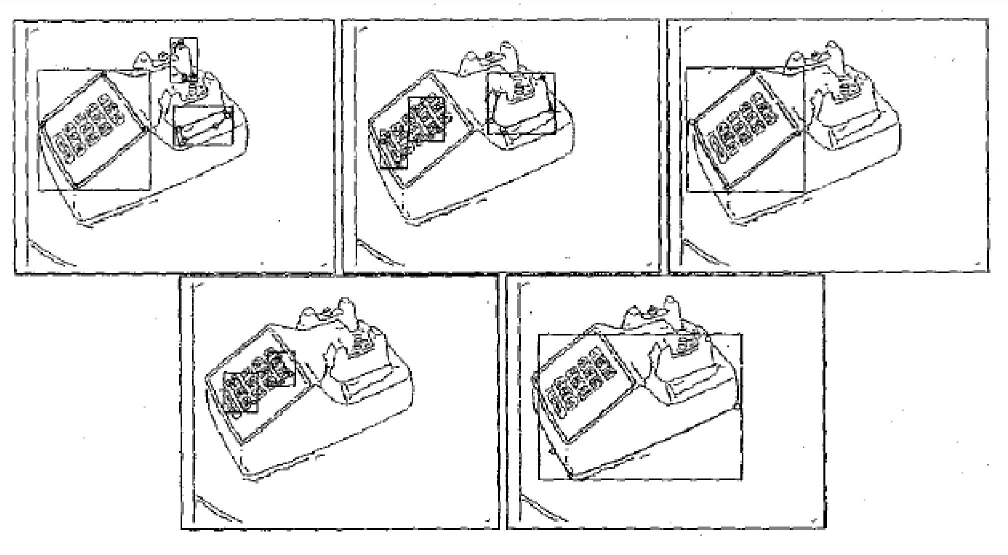
- Symbolic methods
- Clustering
- Local interactions
- Inspired by human visual system

Symbolic Methods

- Operate on *symbols* not *signals*
- Marr (1982): hierarchical grouping of symbols
  - Primal sketch
  - 2 ½-D sketch
  - 3-D model
- Lowe (1985): 3-D object recognition based on grouping of edgels
  - Gestalt principles
  - Viewpoint invariance
  - Low likelihood of accidental alignment
- Saund (2003): perceptually closed paths in sketches and drawings
  - Loosely convex, mostly closed

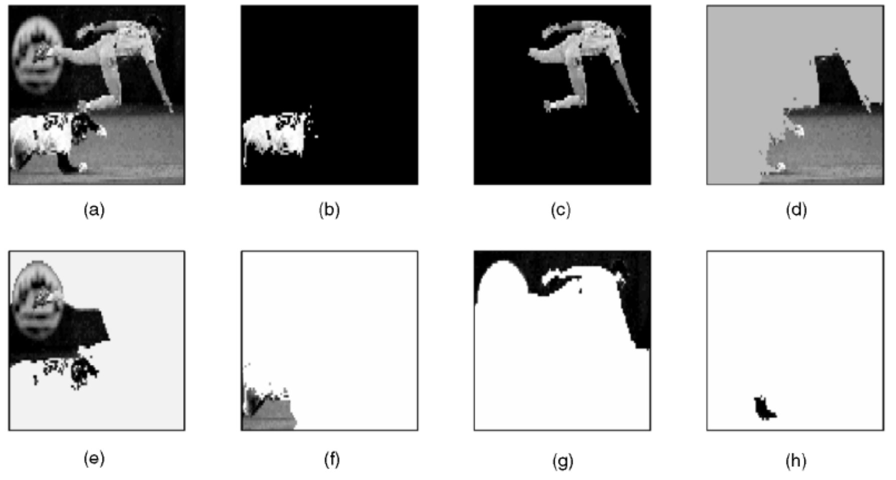
Symbolic Methods

- Mohan and Nevatia (1992): bottom-up hierarchical with increasing levels of abstraction
  - 3-D scene descriptions from collations of features
- Dolan and Riseman (1992): hierarchical curvilinear structure inference
- Jacobs (1996): salient convex groups as potential object outlines
  - Convexity, proximity, contrast



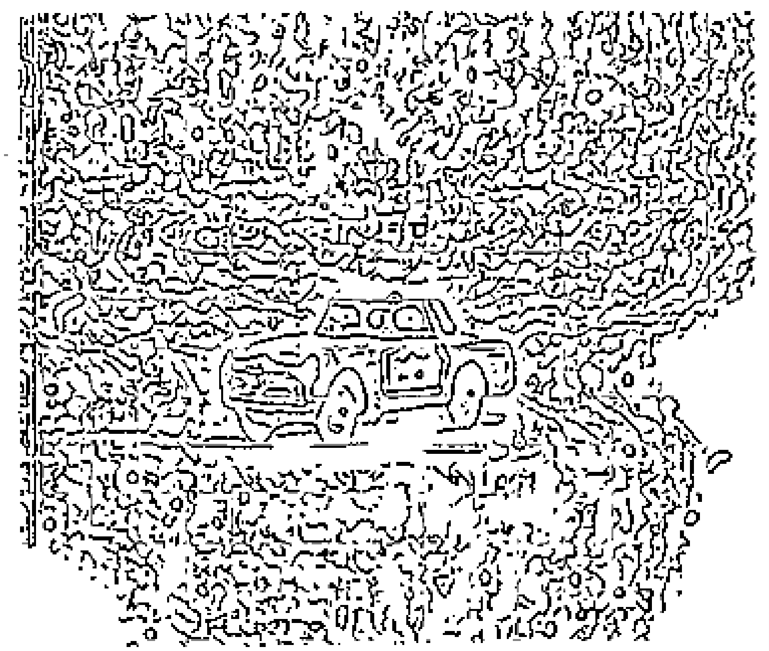
Clustering

- Jain and Dubes (1988): textbook
- Shi and Malik (2000): normalized cuts on graph
  - Edges encode affinity between nodes
- Boykov et al (2001): α-expansion algorithm on labeling graph to minimize objective function
  - Single-node data terms
  - Pair-wise regularization terms



Methods based on Local Interactions

- Shashua and Ullman (1988): structural saliency due to length and smoothness of curvature of curves going through each token
- Parent and Zucker (1989): trace points, tangents and curvature from noisy data
- Sander and Zucker (1990): 3-D extension
- Guy and Medioni (1996, 1997): predecessor of tensor voting
  - Voting fields
  - Tensor analysis for feature inference
  - Unified detection of surfaces, curves and junctions

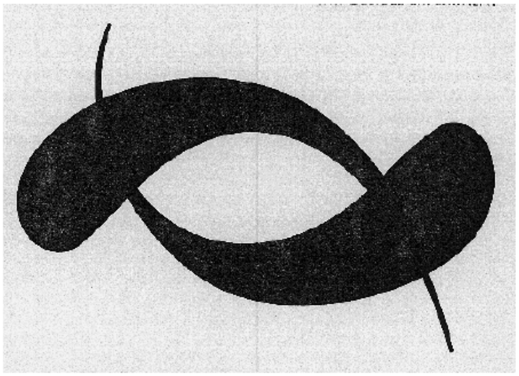


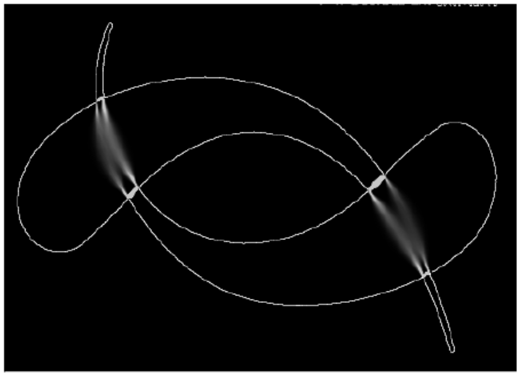
Inspired by Human Visual System

- Grossberg and Mingolla (1985), Grossberg and Todorovic (1988): Boundary Contour System and Feature Contour System
  - BCS: boundary detection, competition and cooperation, includes cells that respond to “end-stopping”
  - FCS: surface diffusion mechanism limited by BCS boundaries
- Heitger and von der Heydt (1993): curvels grouped into contours via convolution with orientation-selective kernels
  - Responses decay with distance, difference in orientation
  - Detectors for endpoints, T-junctions and corners
  - Orthogonal and parallel grouping
  - Explain occlusion, illusory contours consistently with perception

Inspired by Human Visual System

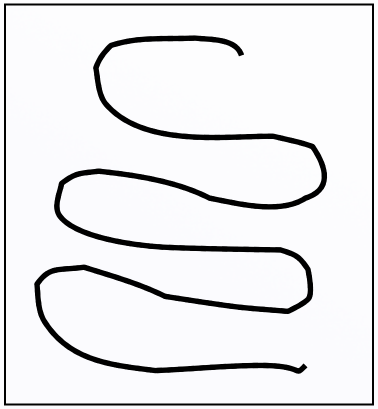
- Williams and Jacobs (1997): stochastic completion fields
  - Probabilistic model based on random walks in image lattice
- Li (1998): contour integration with excitatory and inhibitory fields
- Yen and Finkel (1998): voting-based approach
  - Votes along tangent of osculating circle at receiver, attenuate with distance





Differences with our Approach

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



Overview

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

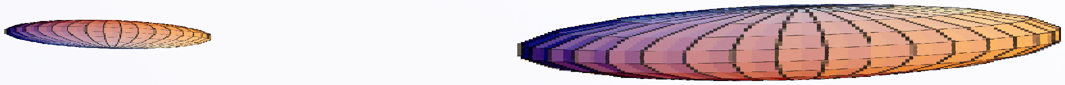
The Original Tensor Voting Framework

- 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

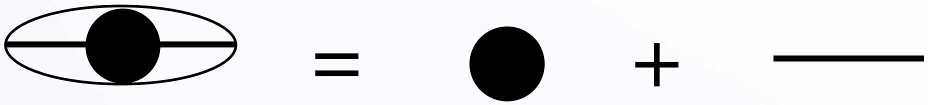
- *Symmetric, non-negative definite*
- Equivalent to:
  - Ellipse in 2-D or ellipsoid in 3-D
  - 2x2 or 3x3 matrix
- Properties that can be encoded:
  - shape: orientation certainty

  - size: feature saliency



Second Order Tensors in 2-D

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

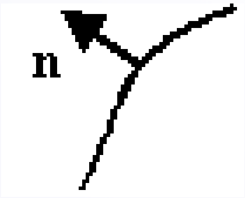
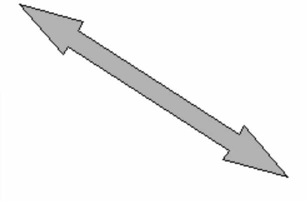
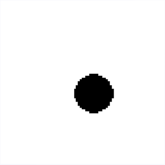
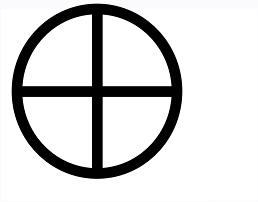


$$\begin{bmatrix} a^2+b^2 & a^2 \\ a^2 & a^2 \end{bmatrix} = \begin{bmatrix} a^2 & a^2 \\ a^2 & a^2 \end{bmatrix} + \begin{bmatrix} b^2 & 0 \\ 0 & 0 \end{bmatrix}$$

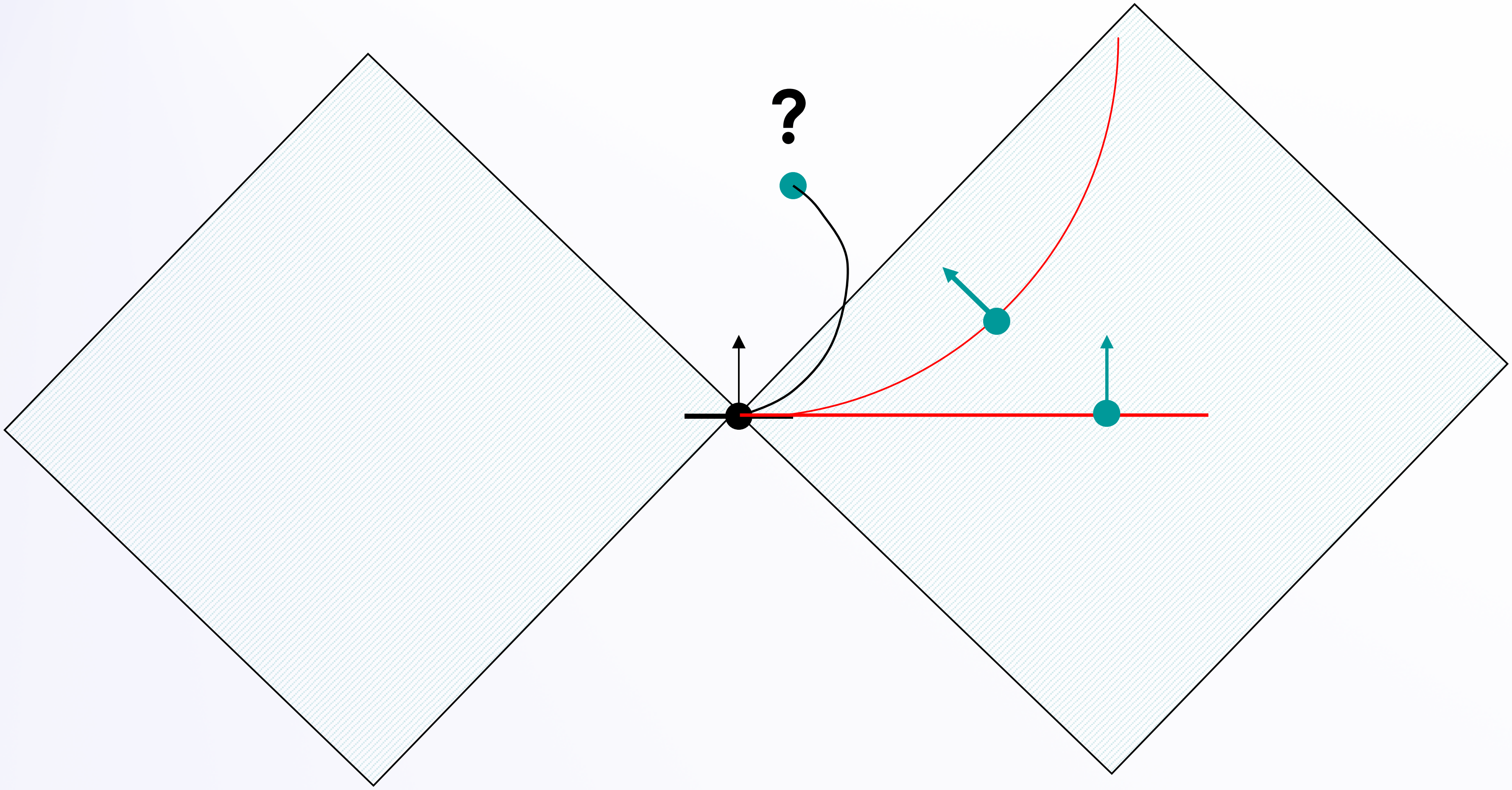
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T =$$

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

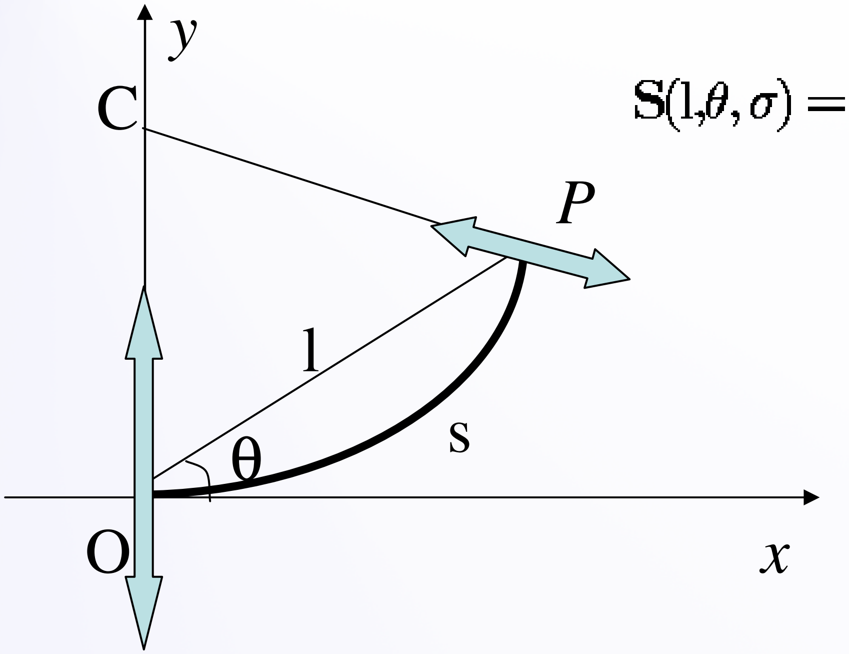
Representation with Tensors

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



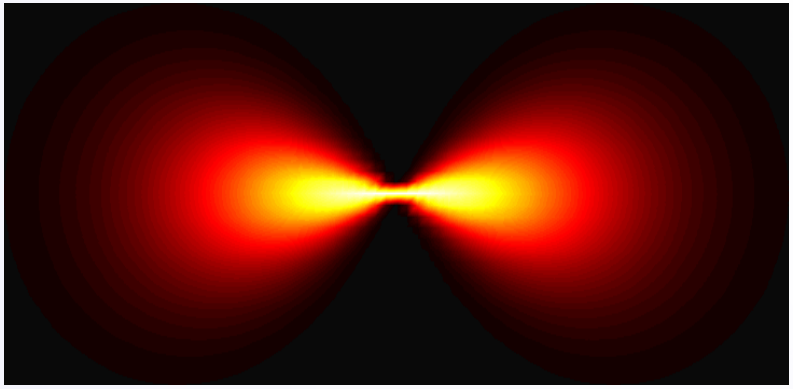
$$\mathbf{S}(l,\theta,\sigma) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)} \begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix} \begin{bmatrix} -sin(2\theta) & cos(2\theta) \end{bmatrix}$$

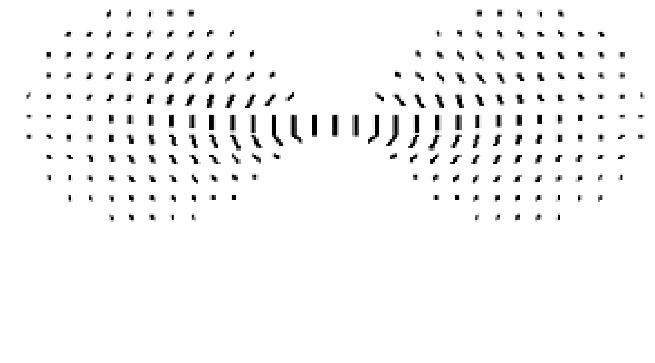
$$s = \frac{\theta l}{\sin\theta}$$

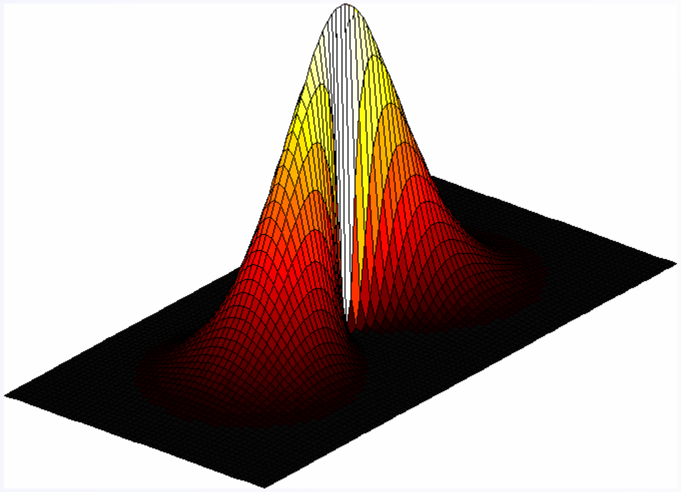
$$\kappa = \frac{2\sin\theta}{l}$$

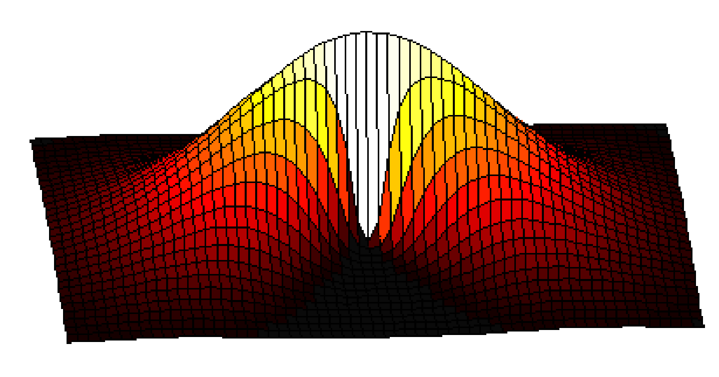
- Votes attenuate with length of smooth path and curvature
- Stored in pre-computed *voting fields*

Fundamental Stick Voting Field







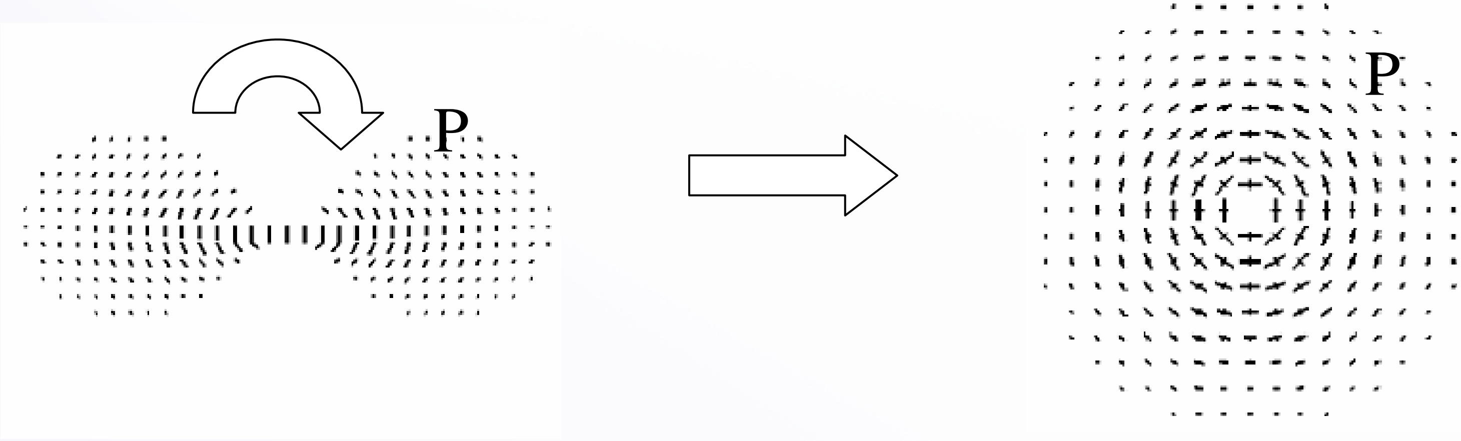


Fundamental Stick Voting Field

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)

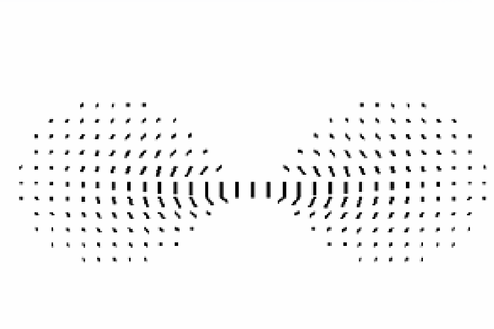
Ball field computed by integrating the contributions of rotating stick

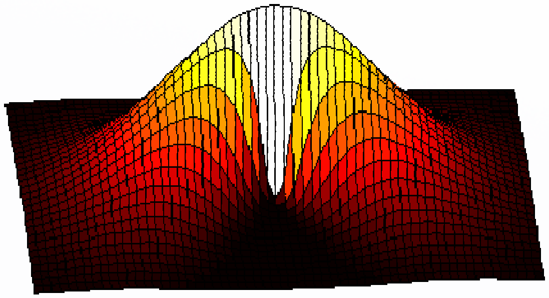
$$\mathbf{B}(P) = \int \mathbf{S}(P)\,d\theta$$

2-D Voting Fields

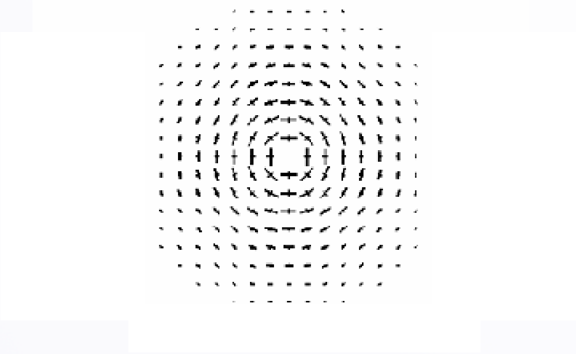
Each input site *propagates* its information in a *neighborhood*

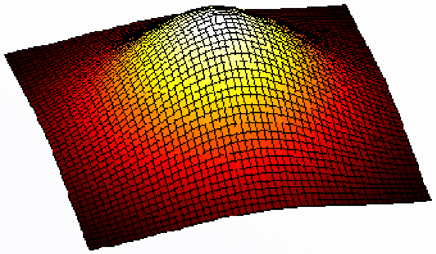
— votes with

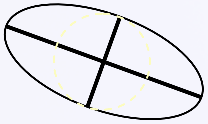




● votes with



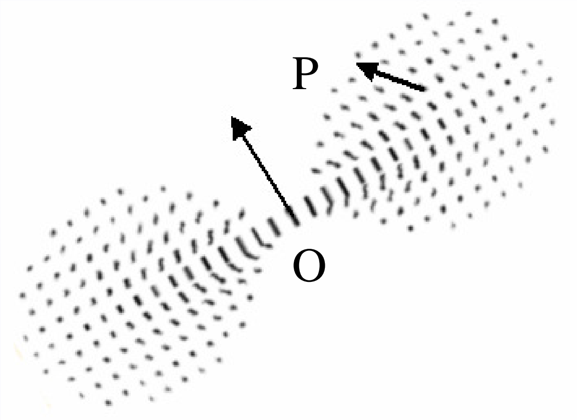




votes with ╲ + ●

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

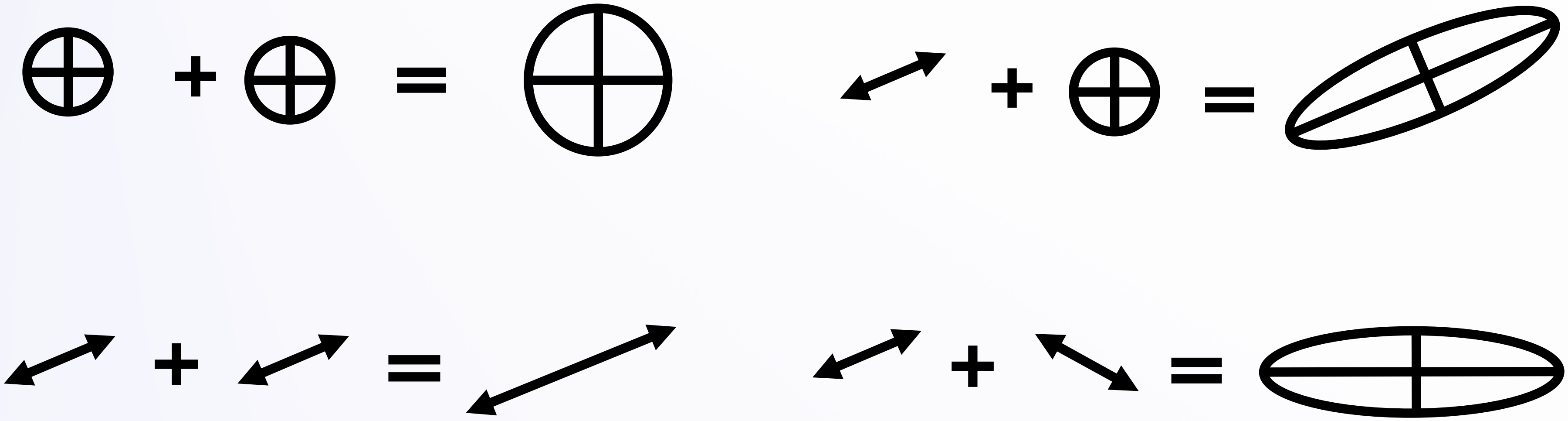
- The Scale of Voting is the single critical parameter in the framework
- Essentially defines size of voting neighborhood
  - Gaussian decay has infinite extend, but it is cropped to where votes remain meaningful (e.g. 1% of voter saliency)

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*

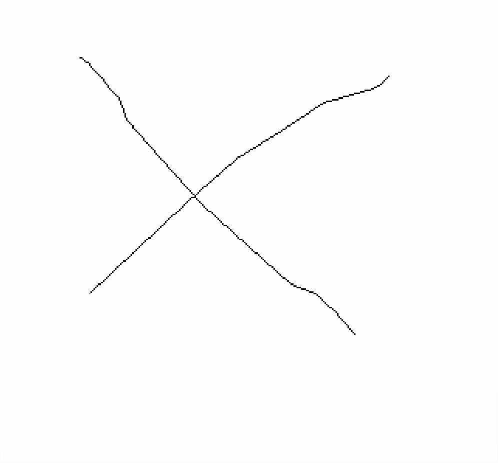
Vote Analysis

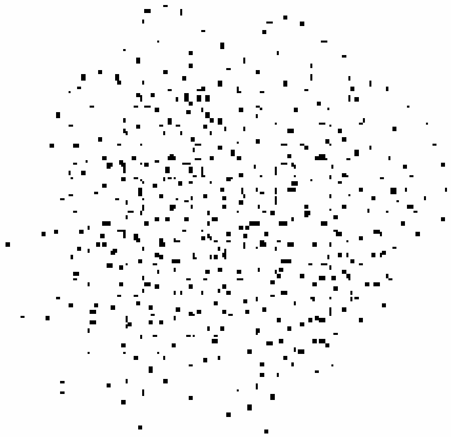
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + \lambda_2 (e_1 e_1^T + e_2 e_2^T)$$

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

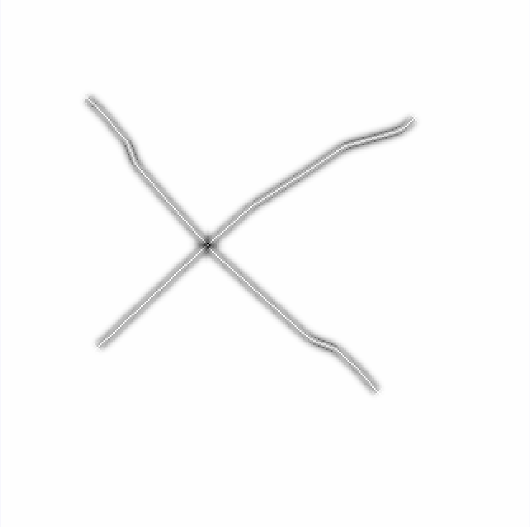
Junction or Region Inlier?

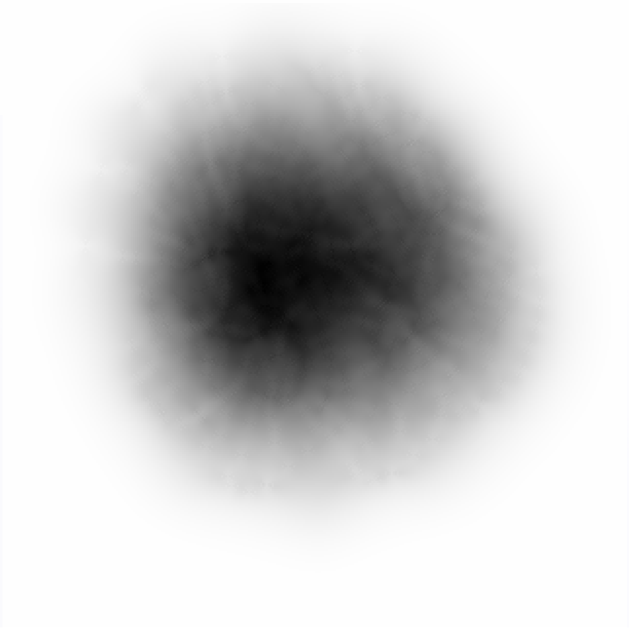
Input



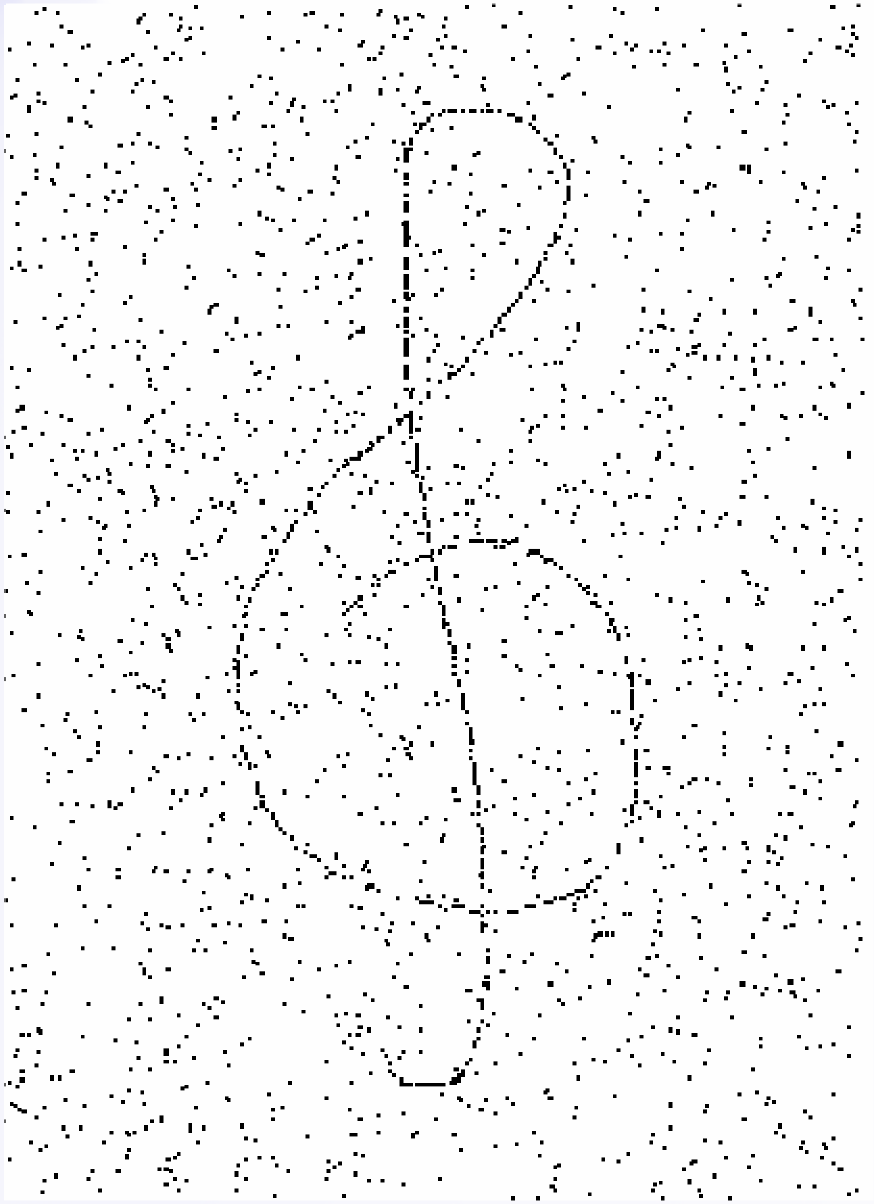


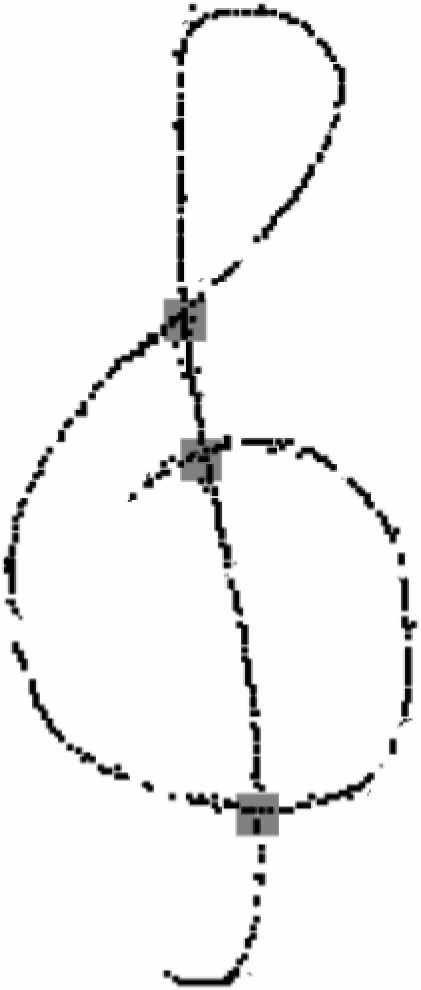
Ball saliency map





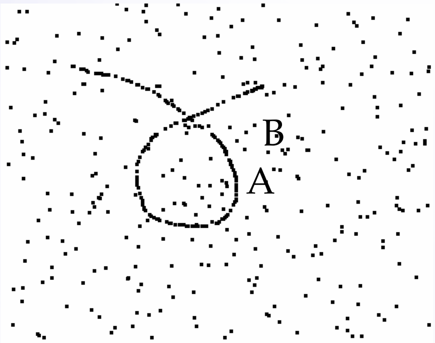
Results in 2-D



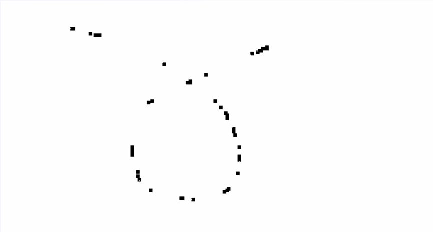


Sensitivity to Scale

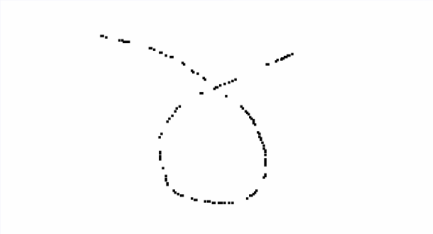
Input: 166 un-oriented inliers, 300 outliers
Dimensions: 960x720
Scale ∈ [50, 5000]
Voting neighborhood ∈ [12, 114]



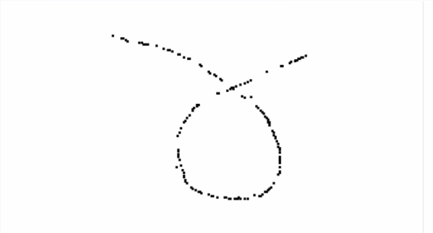
Input



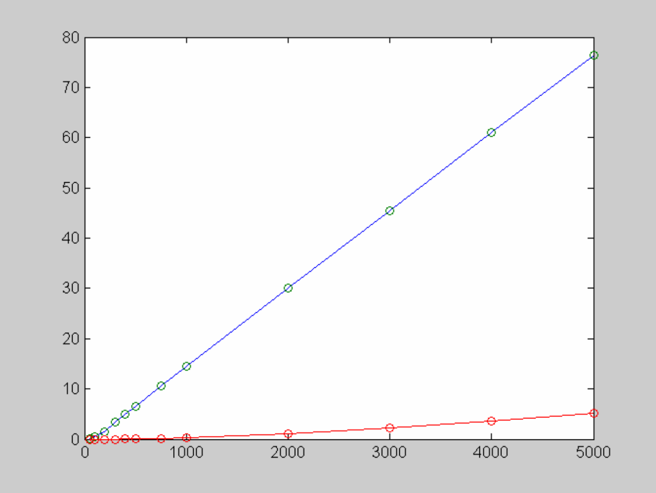
σ = 50



σ = 500



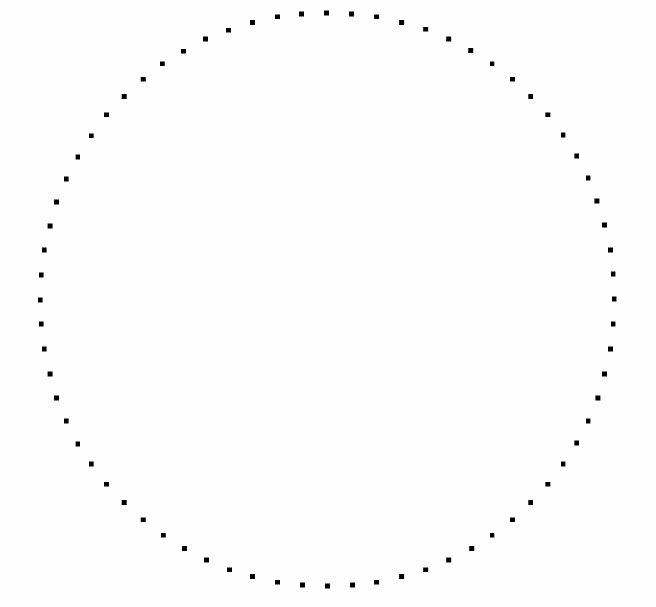
σ = 5000



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

Sensitivity to Scale

Circle with radius 100 (unoriented tokens)

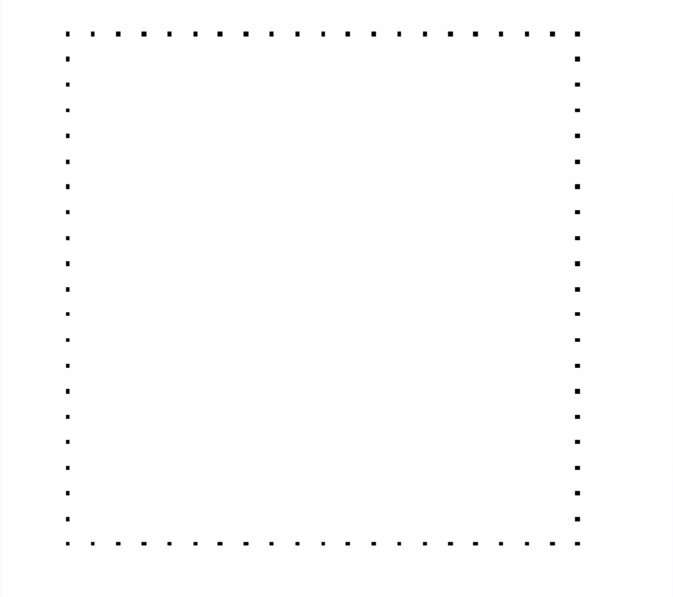


Scale	Average angular error (degrees)
50	1.01453
100	1.14193
200	1.11666
300	1.04043
400	0.974826
500	0.915529
750	0.813692
1000	0.742419
2000	0.611834
3000	0.550823
4000	0.510098
5000	0.480286

As more information is accumulated,
the tokens better approximate the circle

Sensitivity to Scale

Square 20x20 (unoriented tokens)



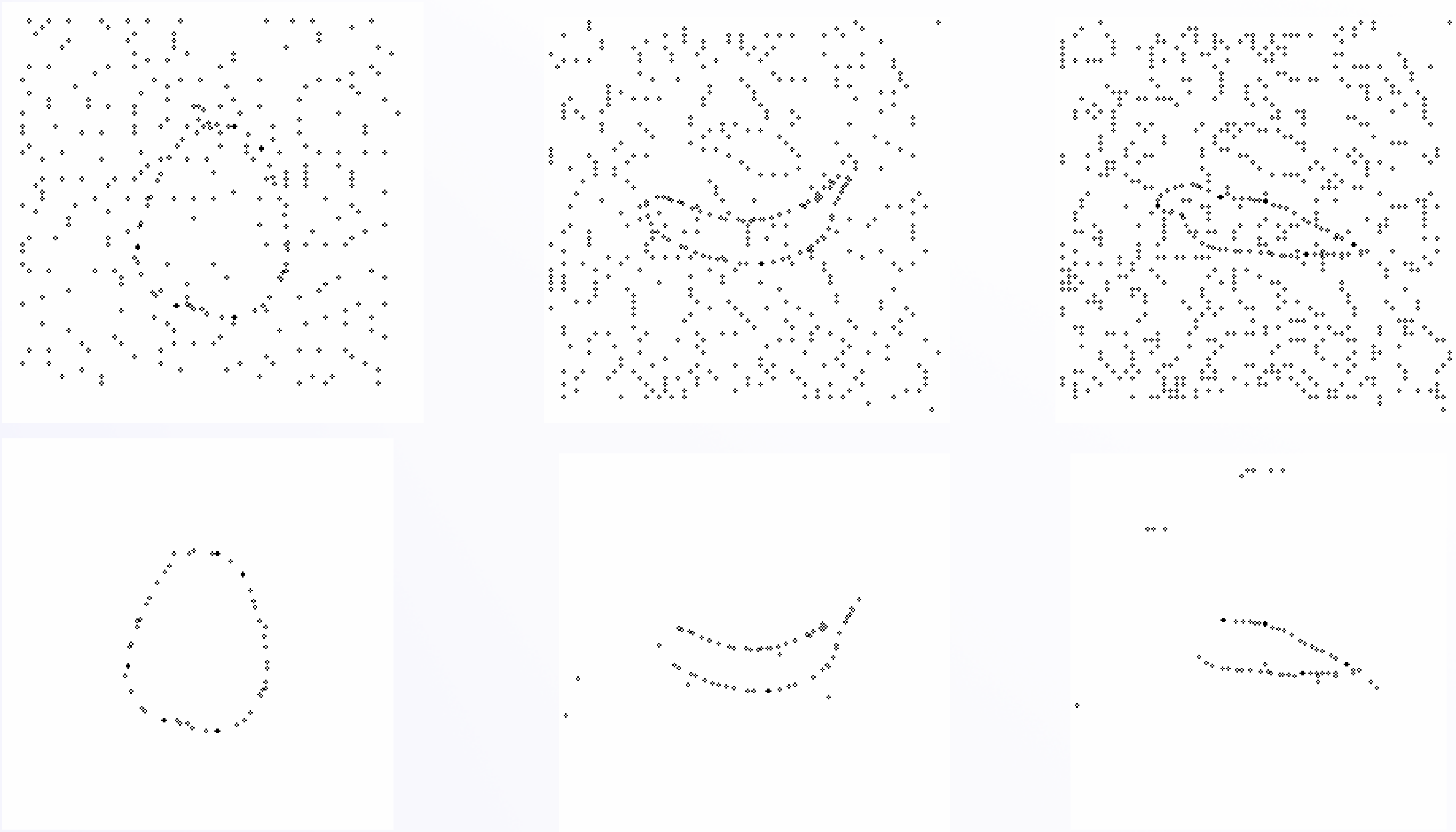
Scale	Average angular error (degrees)
50	1.11601e-007
100	0.138981
200	0.381272
300	0.548581
400	0.646754
500	0.722238
750	0.8893
1000	1.0408
2000	1.75827
3000	2.3231
4000	2.7244
5000	2.98635

Junctions are detected and excluded

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

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

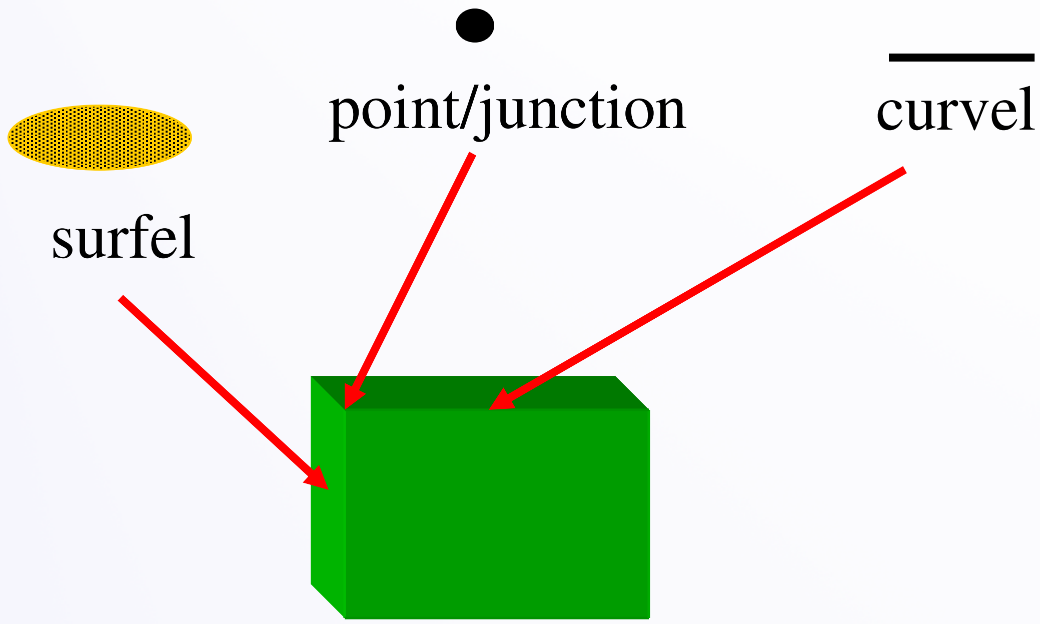
- SNR: ratio of foreground edgels to background edgels
- FPR: false positive rate for foreground detection

- 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



point/junction

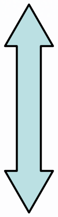
curvel

surfel

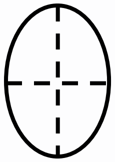
3-D Second Order Tensors

- Encode **normal** orientation in tensor

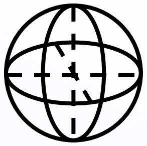
- Surfel: 1 normal → "stick" tensor



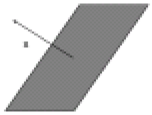
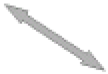
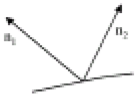
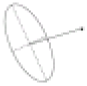
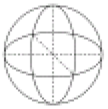
- Curvel: 2 normals → "plate" tensor



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

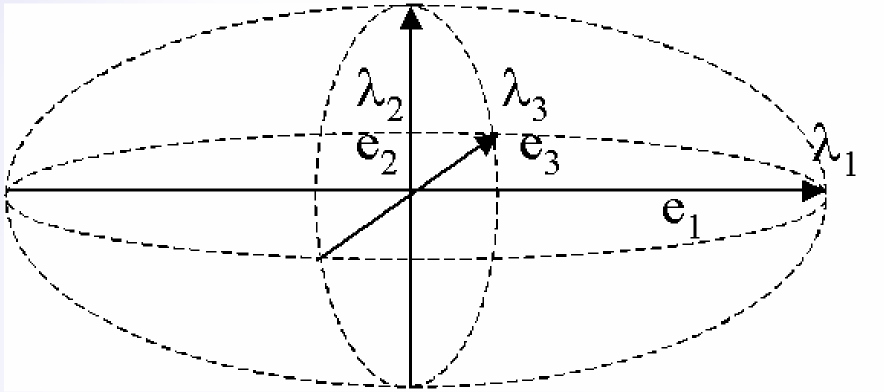


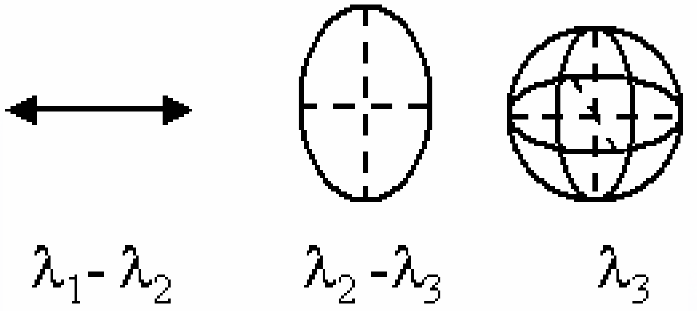
Representation in 3-D

Input	Tensor	Eigenvalues	Quadratic form
 surfel	 stick tensor	$\lambda_1 = 1,\ \lambda_2 = \lambda_3 = 0$	$\begin{bmatrix} n_1^2 & n_1n_2 & n_1n_3 \\ n_1n_2 & n_2^2 & n_2n_3 \\ n_1n_3 & n_2n_3 & n_3^2 \end{bmatrix}$
 curvel	 plate tensor	$\lambda_1 = \lambda_2 = 1,\ \lambda_3 = 0$	$\mathbf{P}$ (see below)
 unoriented	 ball tensor	$\lambda_1 = \lambda_2 = \lambda_3 = 1$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\mathbf{P}= \begin{bmatrix} n_{11}^2 + n_{21}^2 & n_{11}n_{12} + n_{21}n_{22} & n_{11}n_{13} + n_{21}n_{23} \\ n_{11}n_{12} + n_{21}n_{22} & n_{12}^2 + n_{22}^2 & n_{12}n_{13} + n_{22}n_{23} \\ n_{11}n_{13} + n_{21}n_{23} & n_{12}n_{13} + n_{22}n_{23} & n_{13}^2 + n_{23}^2 \end{bmatrix}$$

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: λ_1- λ_2 normal: $\mathbf{e}_1$
- Curve saliency: λ_2- λ_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

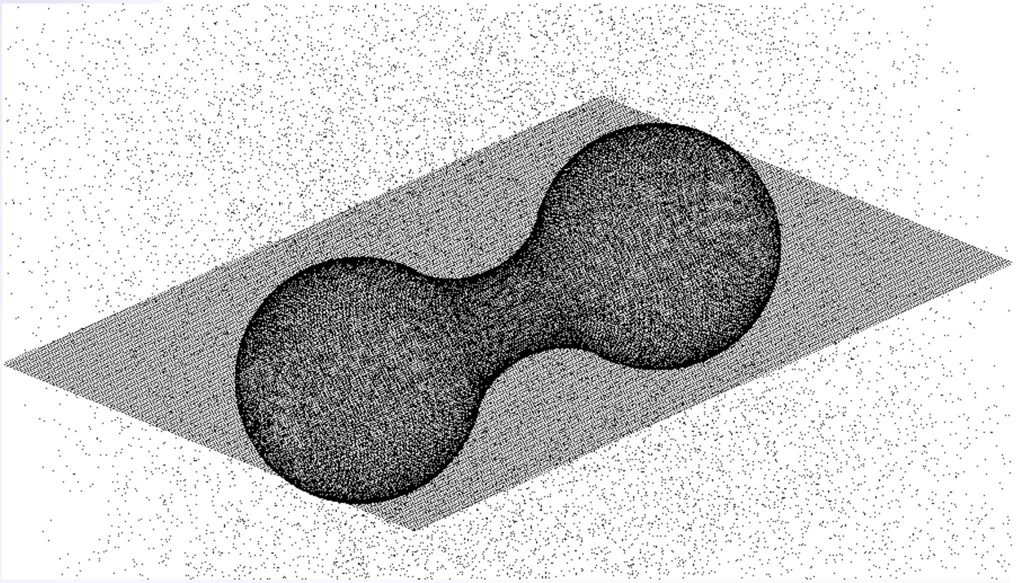
Vote Analysis in 3-D

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

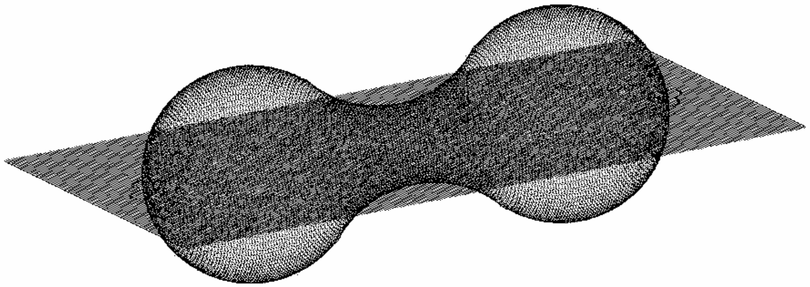
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) + \lambda_3 (e_1 e_1^T + e_2 e_2^T + e_3 e_3^T)$$

- λ_1- λ_2>λ_2- λ_3 and λ_1- λ_2>λ_3: stick saliency is maximum. Likely surface.
- λ_2- λ_3>λ_1- λ_2 and λ_2- λ_3>λ_3 : plate saliency is maximum. Likely curve or surface intersection
- λ_3>λ_1- λ_2 and λ_3> λ_2- λ_3 : ball saliency is maximum. Likely junction
- $\lambda_1 \approx \lambda_2 \approx \lambda_3 \approx 0$: Low saliency. Outlier.

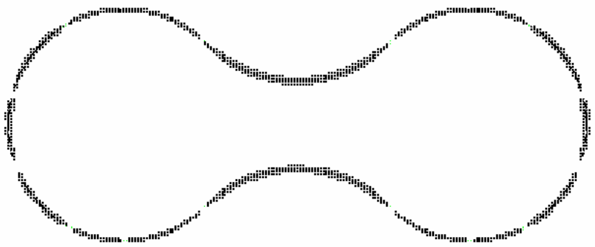
Results in 3-D



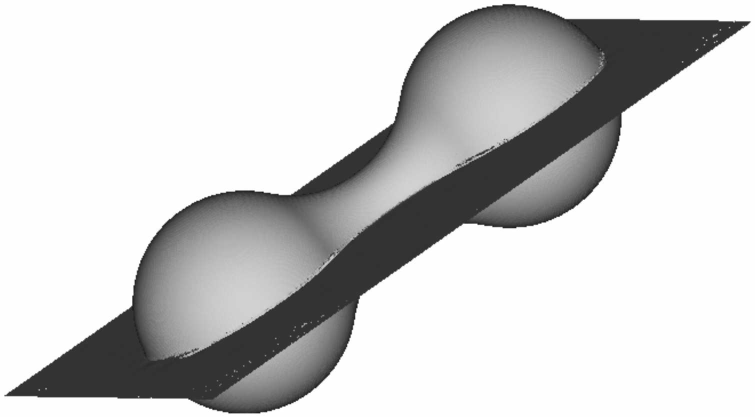
Noisy input



Surface inliers



Surface intersections



Dense surfaces

Overview

- Introduction
- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- More Tensor Voting Research
- 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

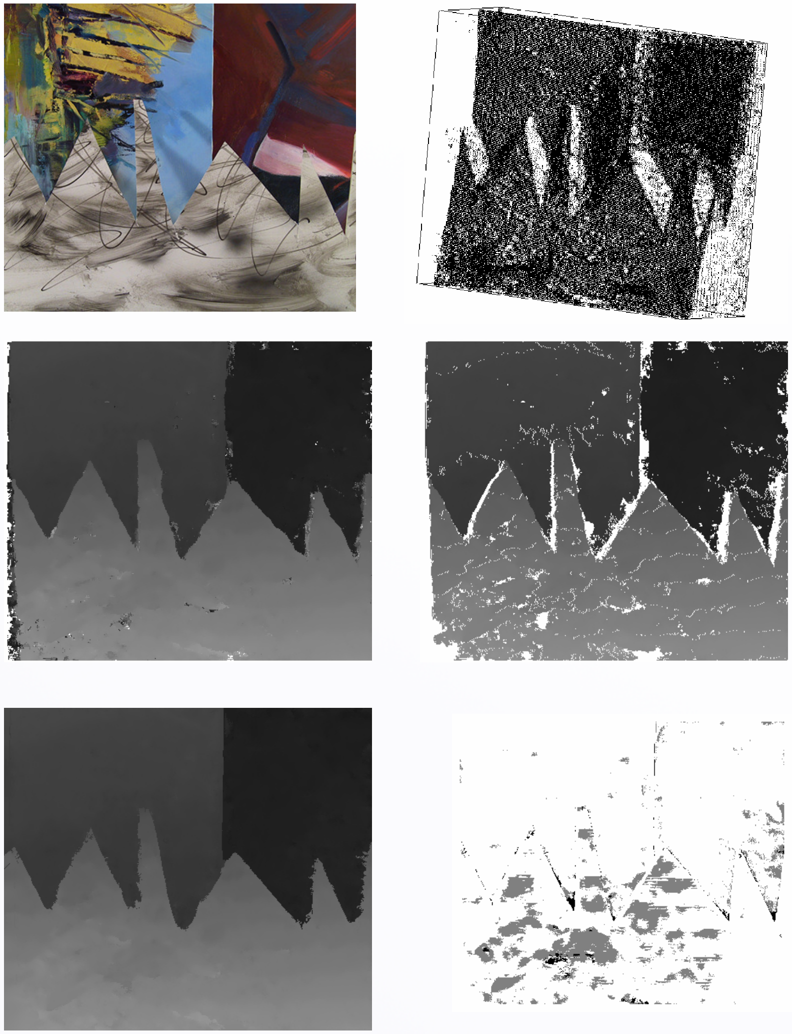
[Mordohai and Medioni, ECCV 2004 and PAMI 2006]

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



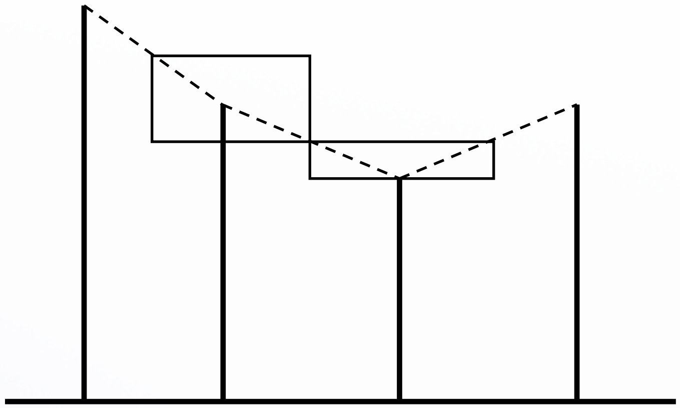
Initial Matching

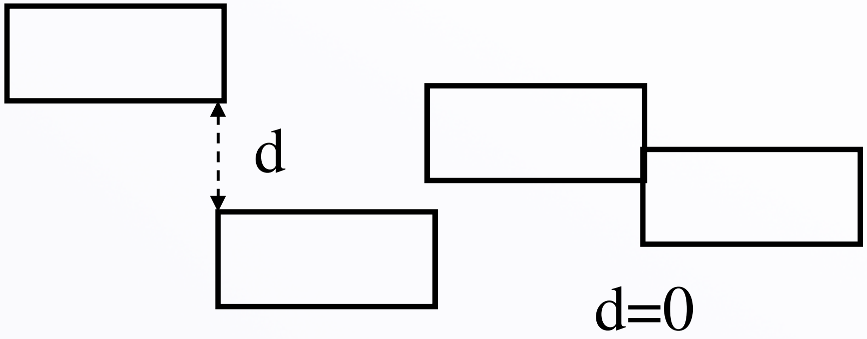
- Each matching technique has different strengths
- Use multiple techniques and both images as reference:
  - 5×5 normalized cross correlation (NCC) window
  - 5×5 shiftable NCC window
  - 25×25 NCC window for pixels with very low color variance
  - 7×7 symmetric interval matching window with truncated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

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

Symmetric Interval Matching

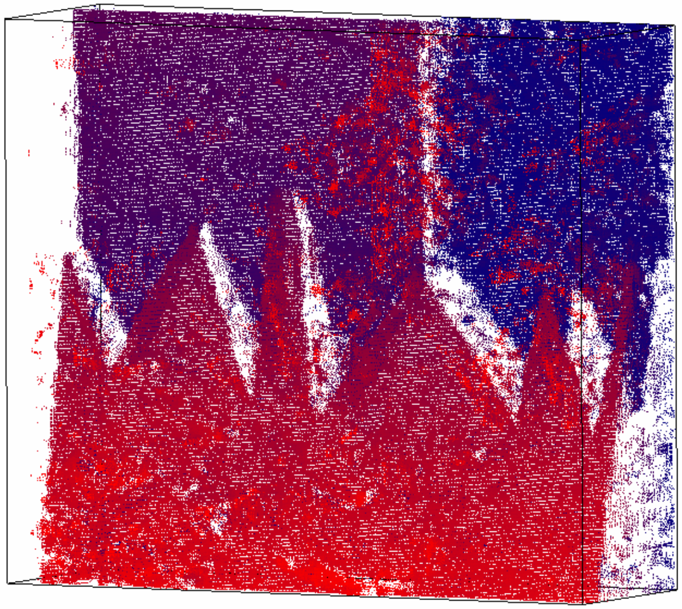
- Upsample scanlines
- Represented the color of pixel (x,y) as the interval (x-½, y) to (x+½, y)
- Dissimilarity measure: distance between intervals
- Truncated to increase performance at discontinuities



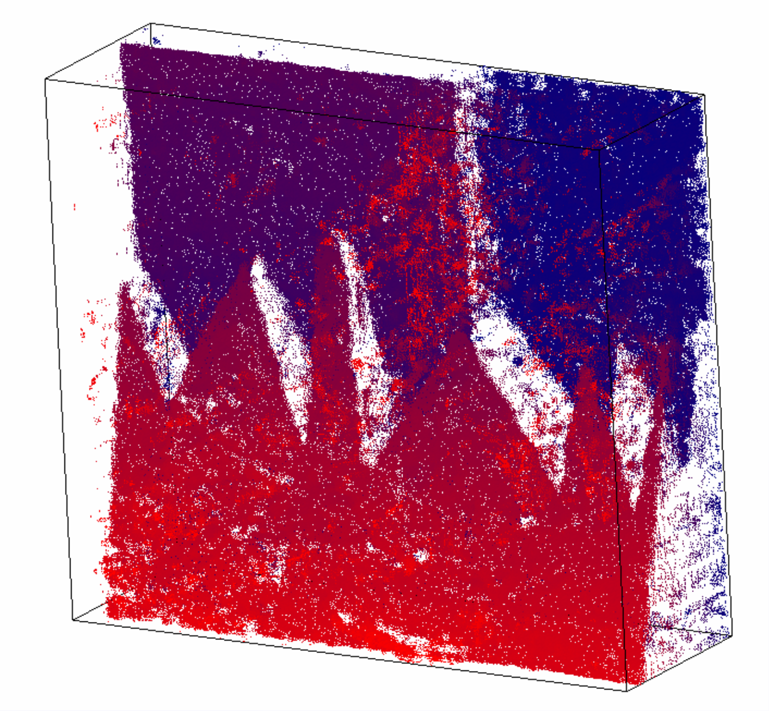


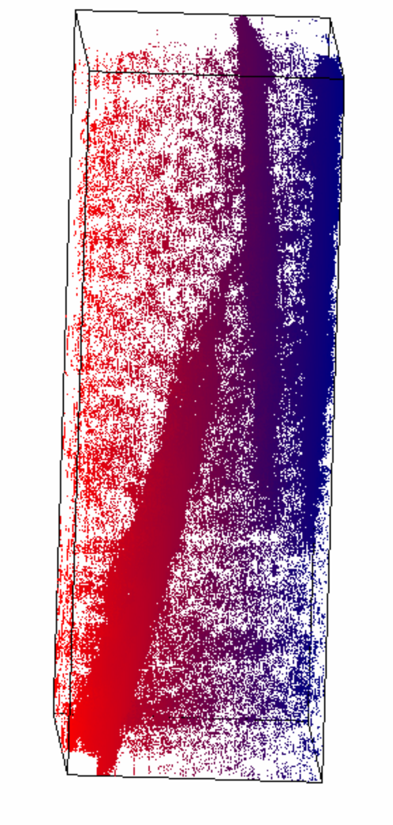
Candidate Matches

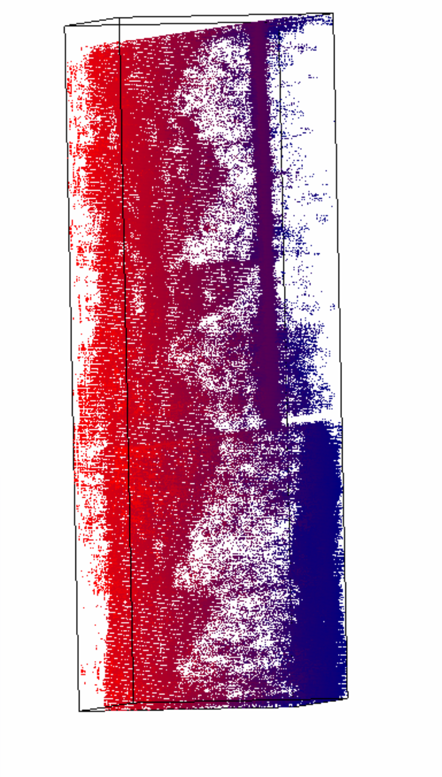
- Compute sub-pixel estimates (parabolic fit)
- Keep all good matches (peaks of NCC)
- Drop scores
  - Depend on texture properties
  - Hard to combine across methods



Candidate Matches

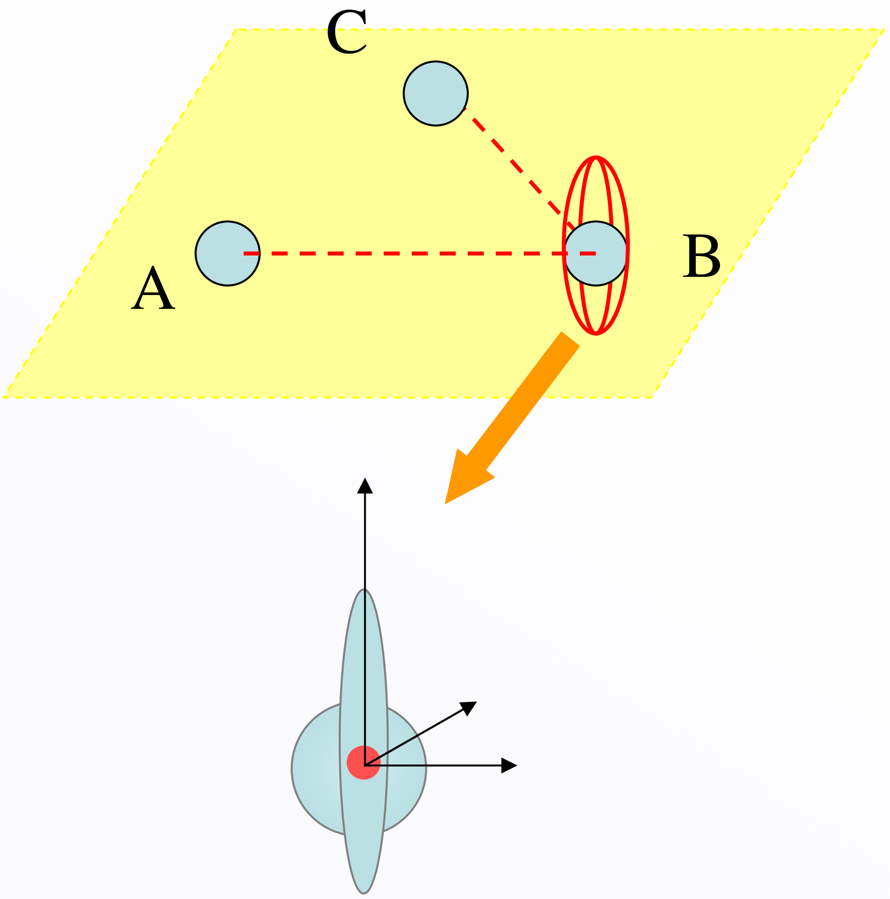






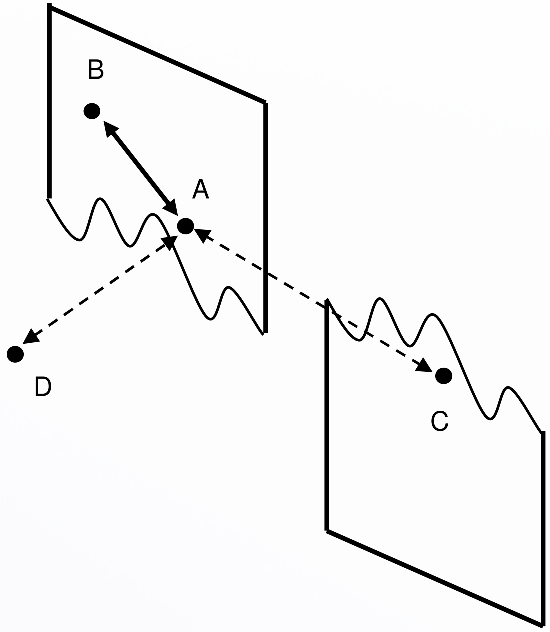
Surfaces from Unoriented Data

- 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



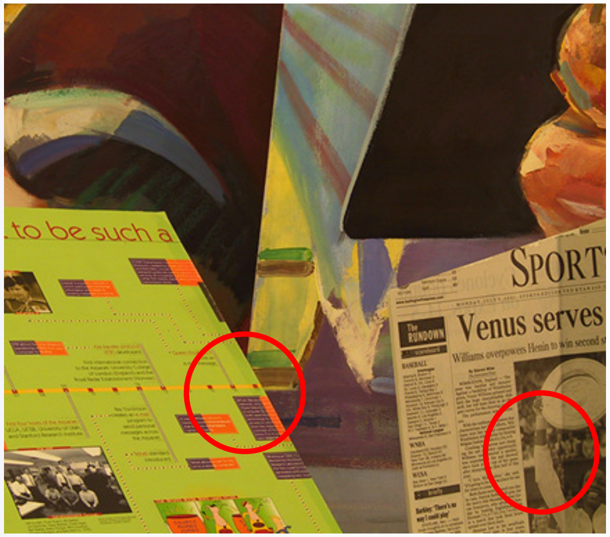
Uniqueness vs. Visibility

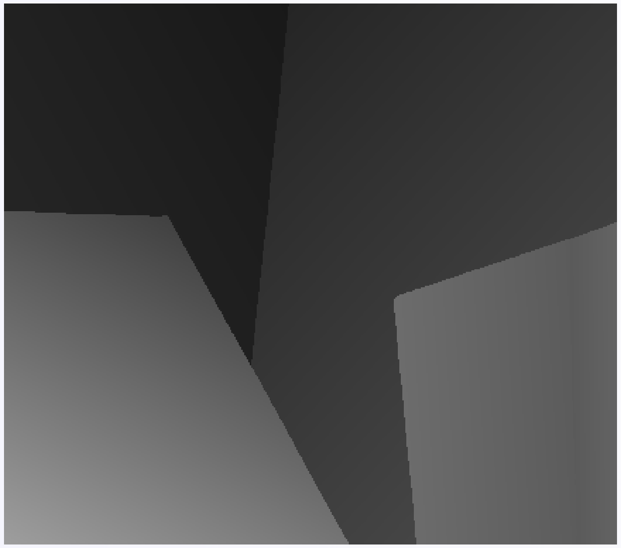
- Uniqueness constraint: One-to-one pixel correspondence
  - Exact only for fronto-parallel surfaces
- Visibility constraint : *M*-to-*N* pixel correspondences
  - [Ogale and Aloimonos 2004][Sun et al. 2005]
  - One match per ray of each camera

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 local collections of colors

Nonparametric Color Model





- Each match has been assigned to a surface now
  - Pixel on match's ray takes same label
- No difficulties caused by:
  - Adjacent image regions with similar color properties
  - Surfaces with varying color distribution
- A GMM would be more complicated and not exact

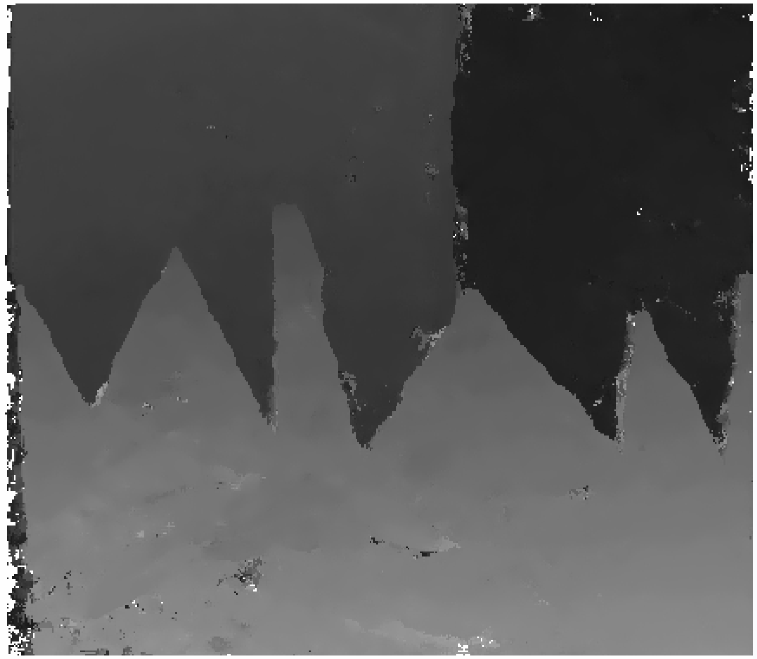
Surface Refinement

- 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

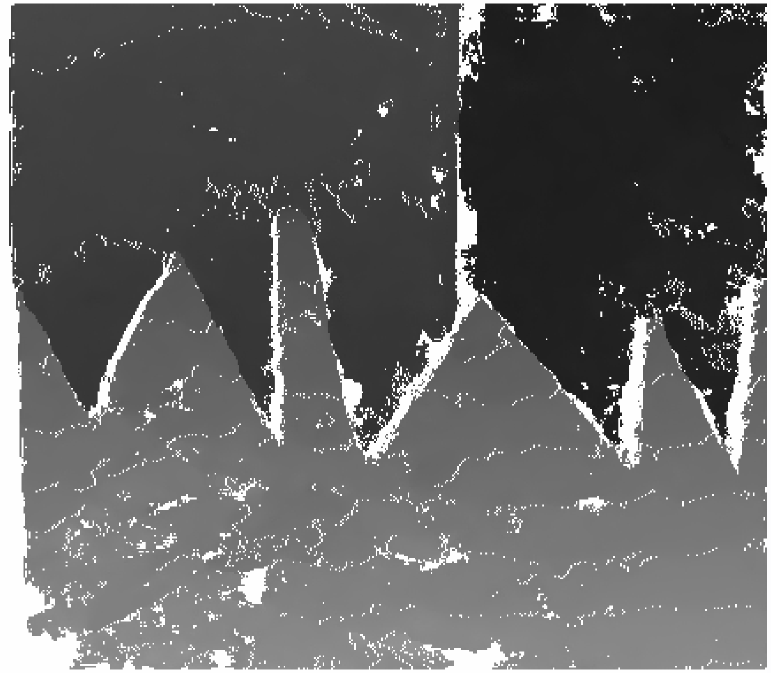
➢ *Set of reliable matches* (reduced foreground fattening)

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

Surface Refinement Results

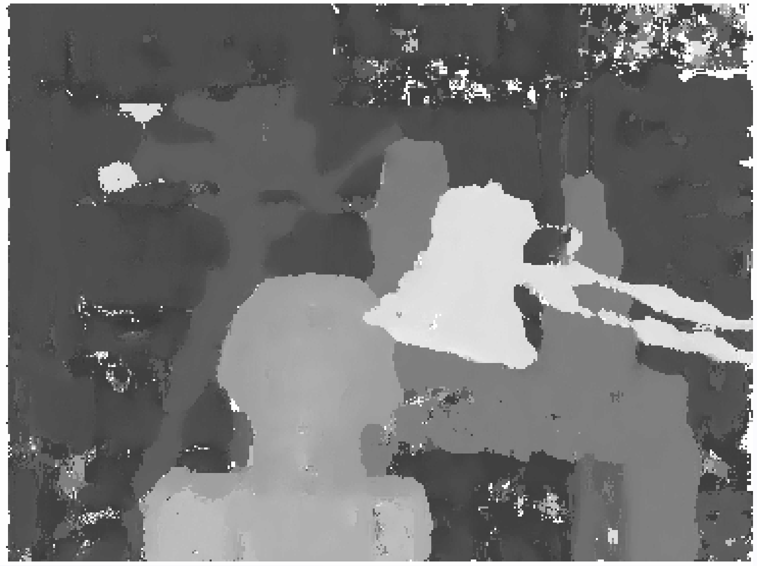


144808 matches
4278 errors

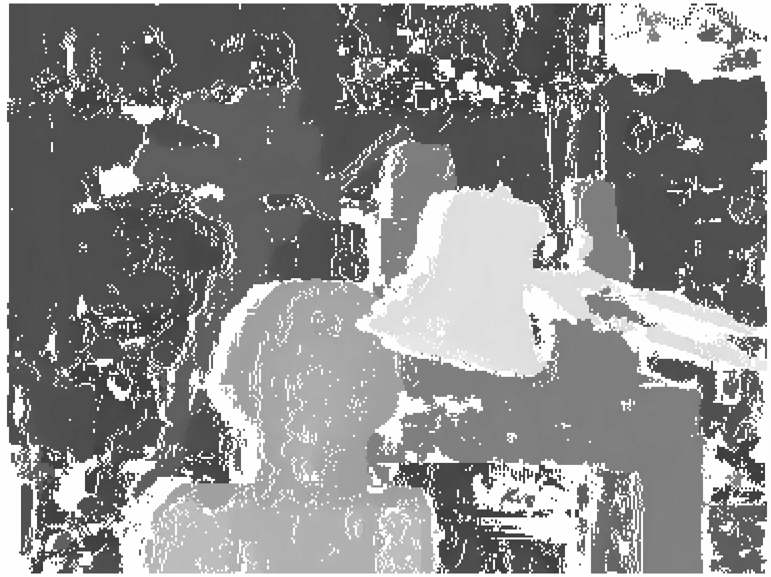


136894 matches
1481 errors

Surface Refinement Results

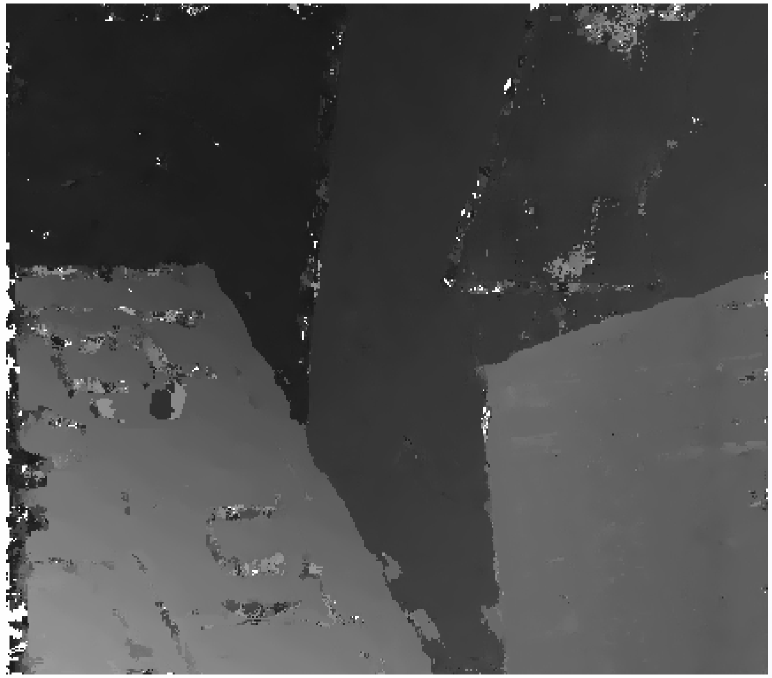


84810 matches
4502 errors

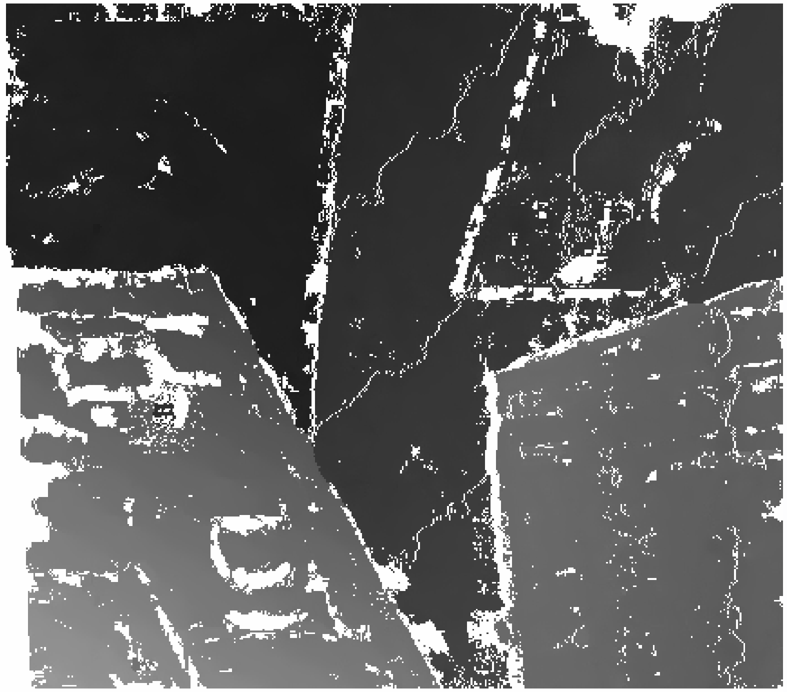


69666 matches
928 errors

Surface Refinement Results



147320 matches
9075 errors



132480 matches
1643 errors

Surface Refinement Results

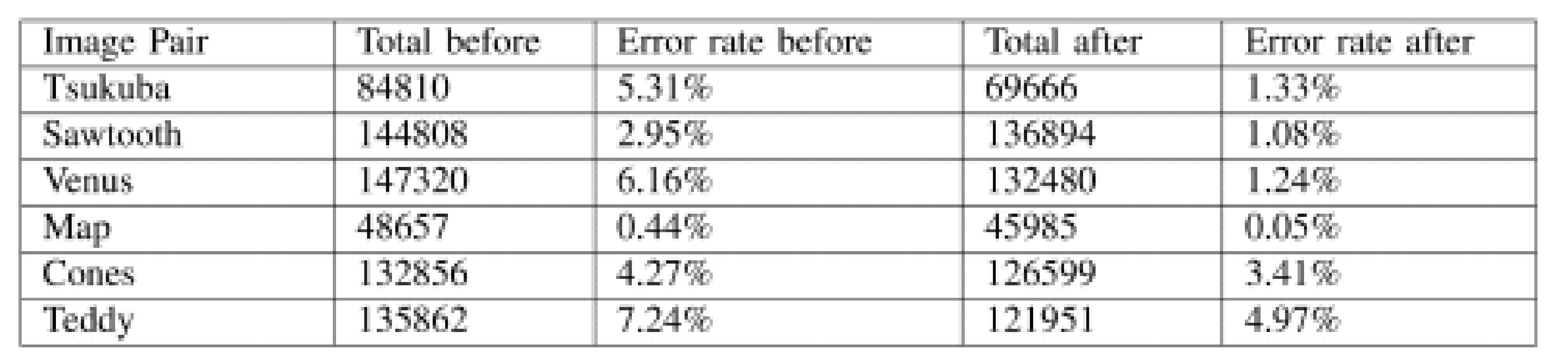
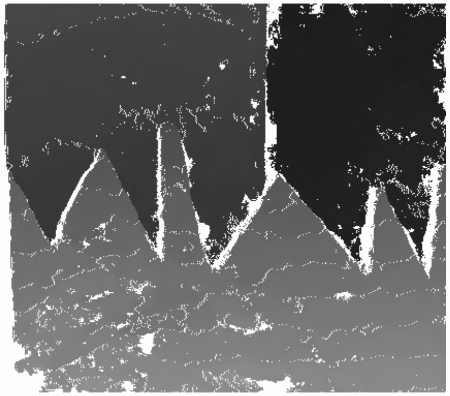


Image Pair	Total before	Error rate before	Total after	Error rate after
Tsukuba	84810	5.31%	69666	1.33%
Sawtooth	144808	2.95%	136894	1.08%
Venus	147320	6.16%	132480	1.24%
Map	48657	0.44%	45985	0.05%
Cones	132856	4.27%	126599	3.41%
Teddy	135862	7.24%	121951	4.97%

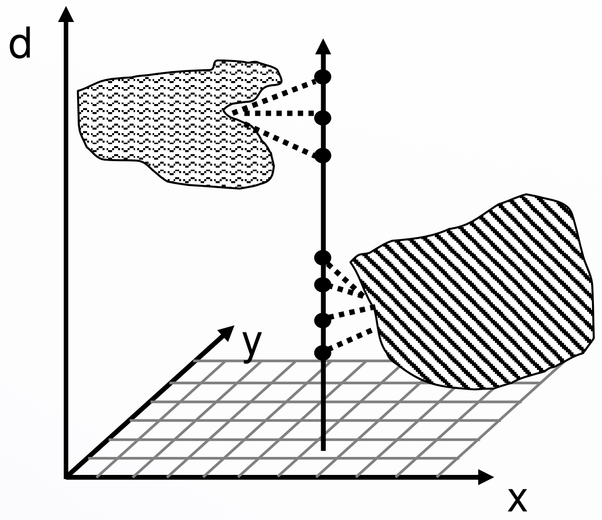
Disparity Hypotheses for Unmatched Pixels

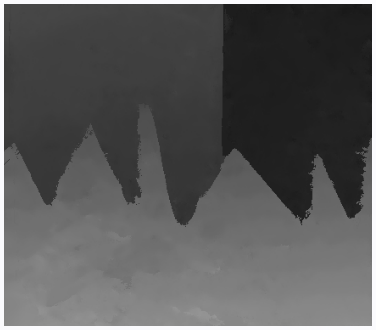
- Check color consistency with nearby layers (on both images if not occluded)
- Generate hypotheses for membership in layers with similar color properties
  - Find disparity range from neighbors (and extend)
  - Allow occluded hypotheses
  - Do not allow occlusion of reliable matches
- Progressively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

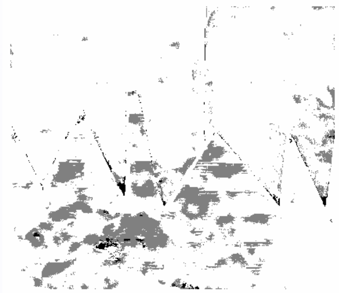


Disparities for Unmatched Pixels

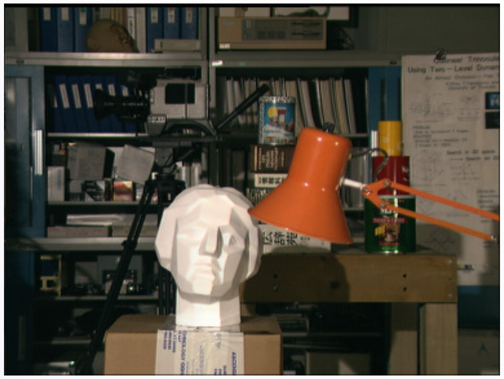
- Vote from neighbors of same surface
- Keep most salient
  - Update occlusion information



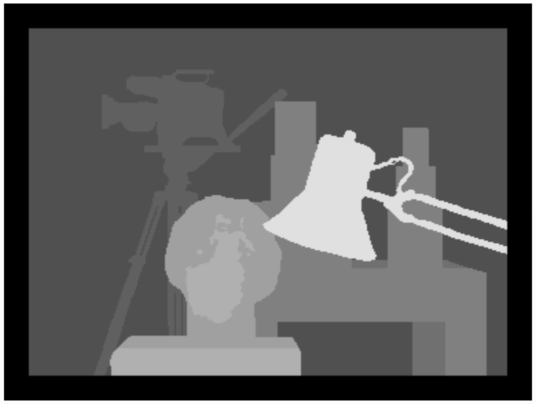

Disparity Map


Error Map

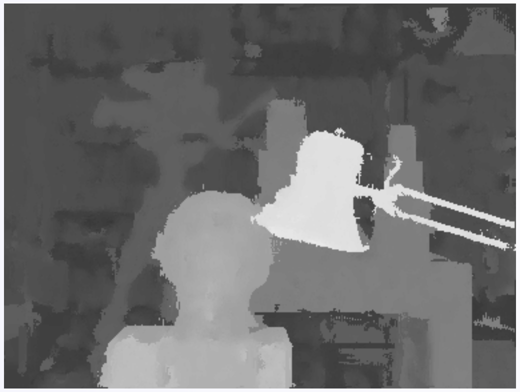
Results: Tsukuba



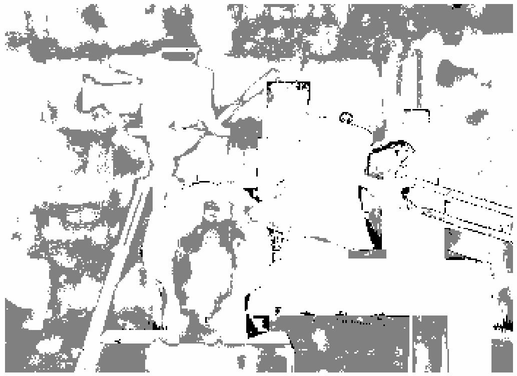
Left image



Ground truth

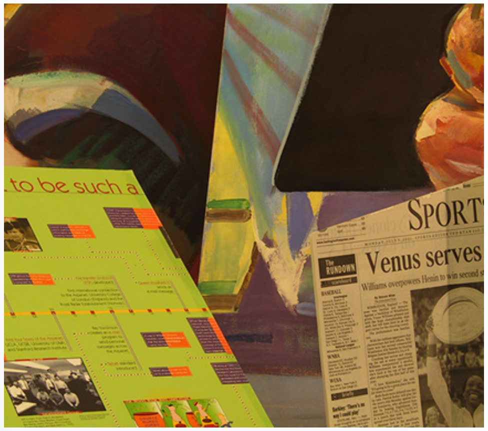


Disparity Map

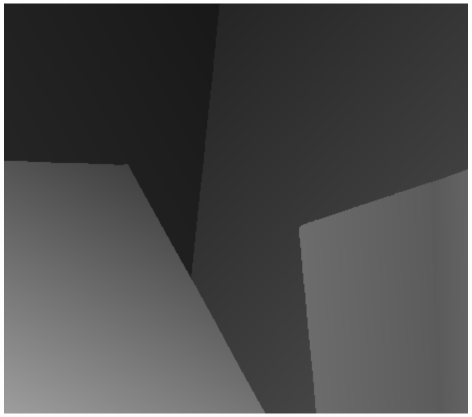


Error Map

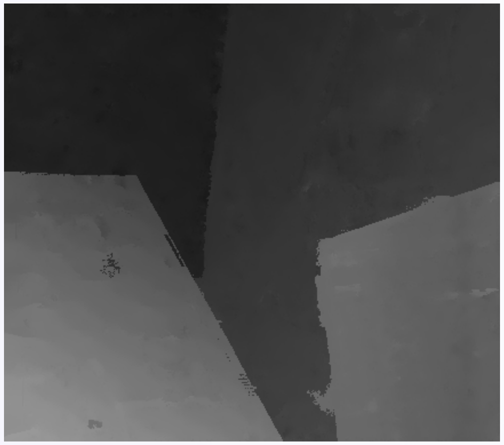
Results: Venus



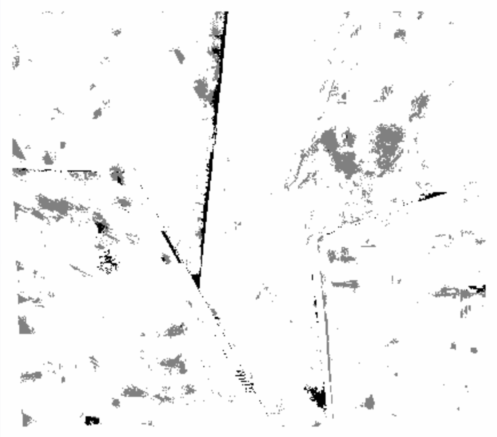
Left image



Ground truth

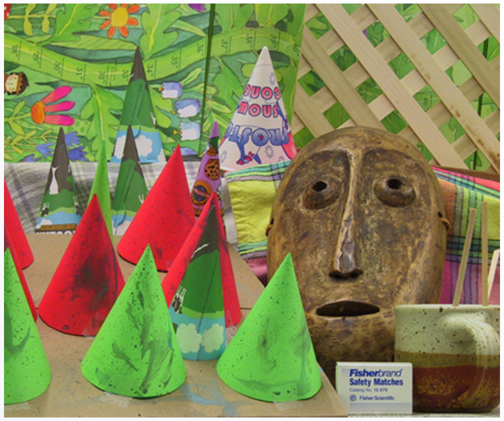


Disparity Map

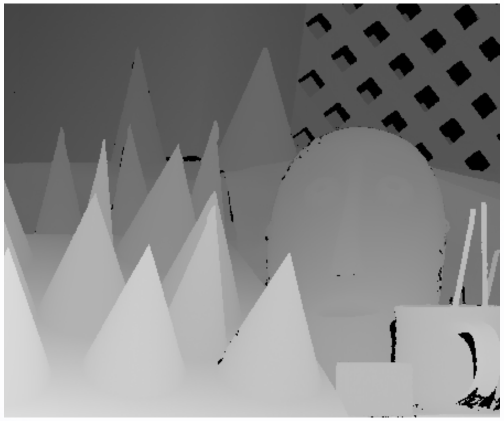


Error Map

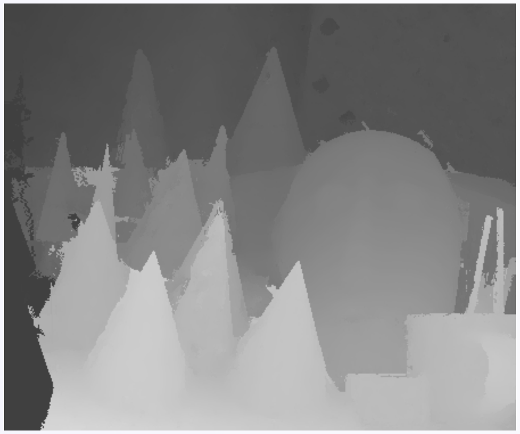
Results: Cones



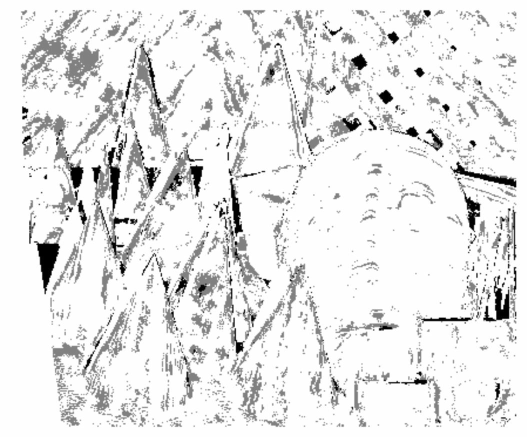
Left image



Ground truth

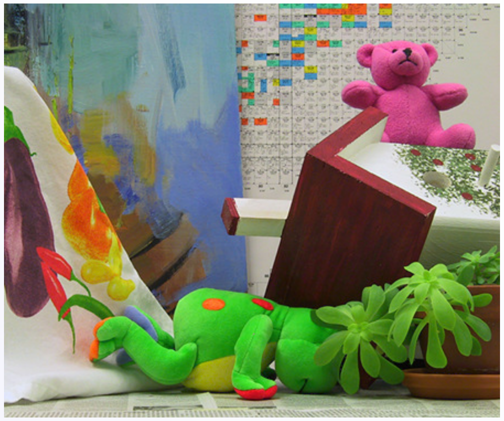


Disparity Map

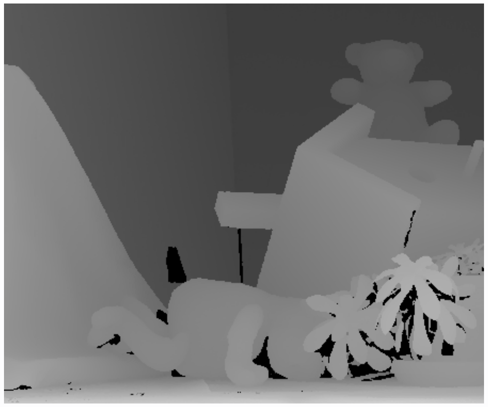


Error Map

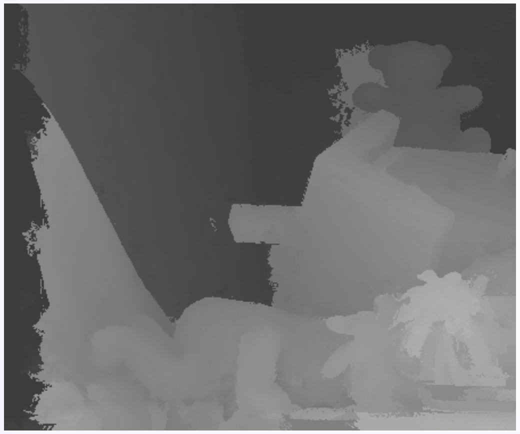
Results: Teddy



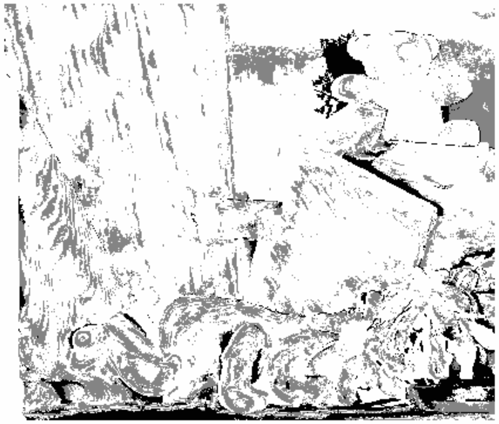
Left image



Ground truth

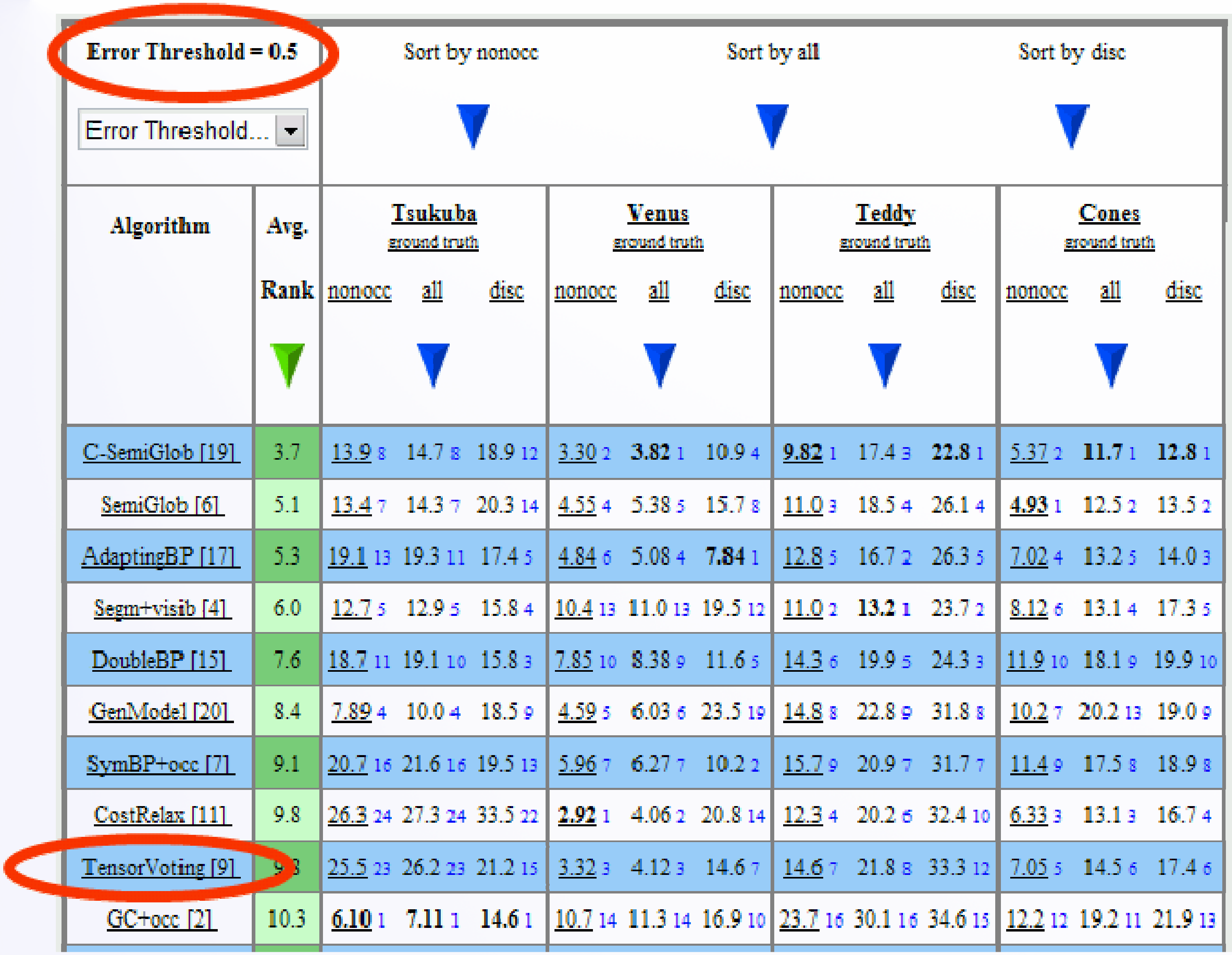


Disparity Map



Error Map

Quantitative Results



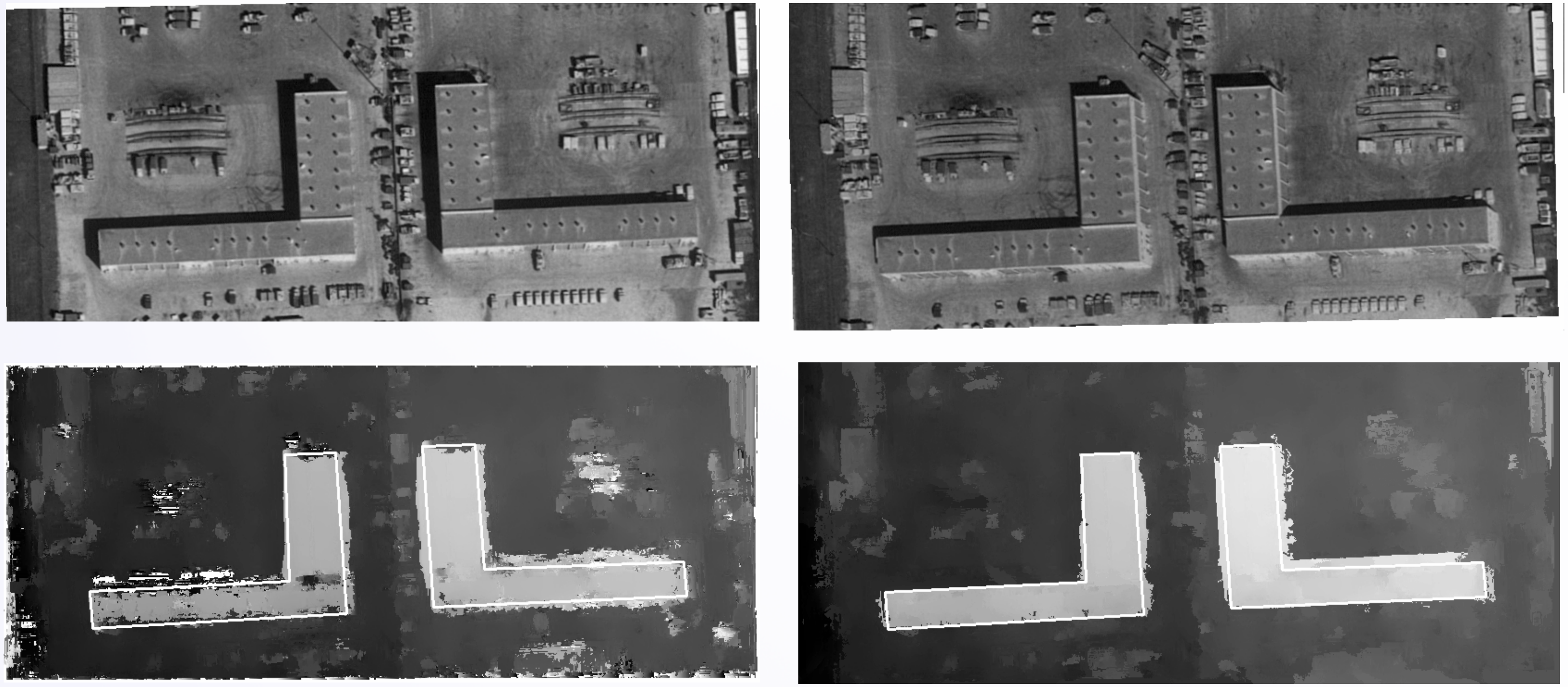
Error Threshold = 0.5

Error Threshold...

Sort by nonocc | Sort by all | Sort by disc

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

Aerial Images



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

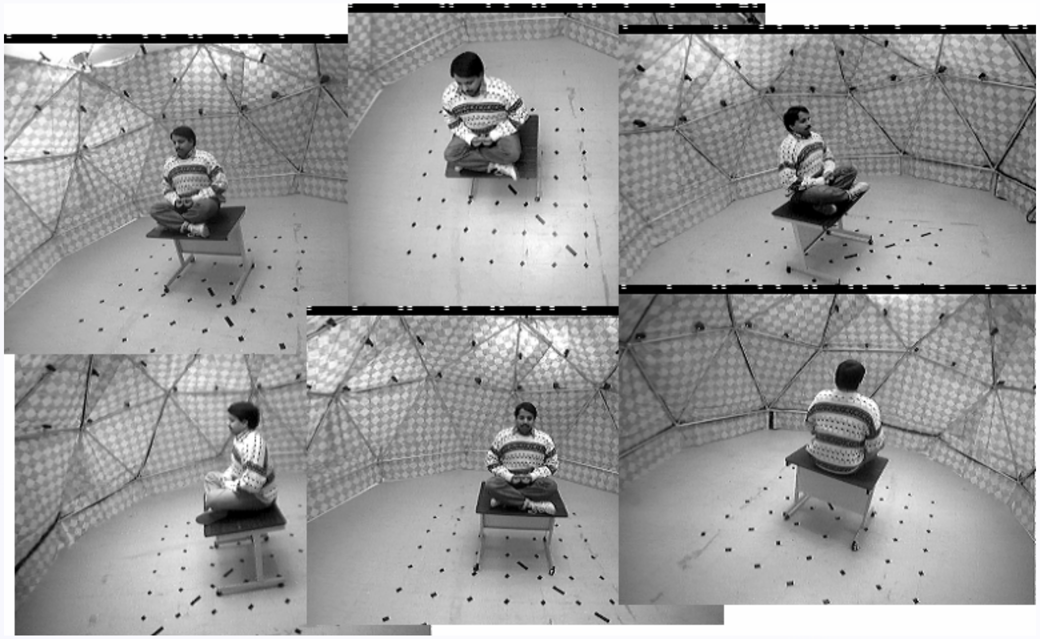
[Mordohai and Medioni, 3DPVT 2004]

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

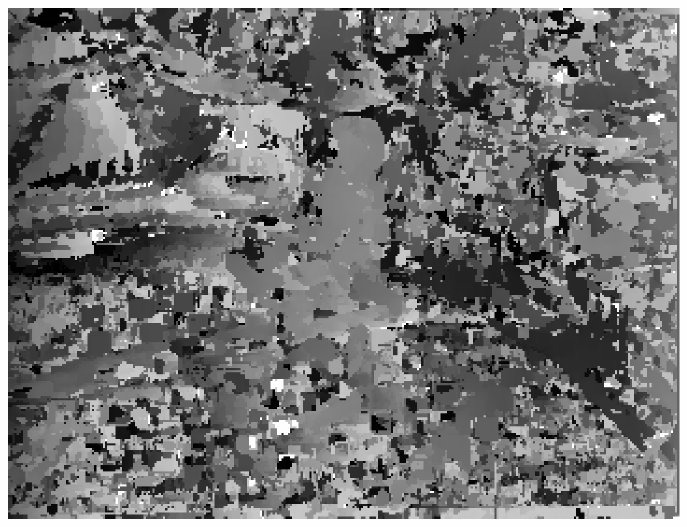
Captured at the CMU dome for the *Virtualized Reality* project



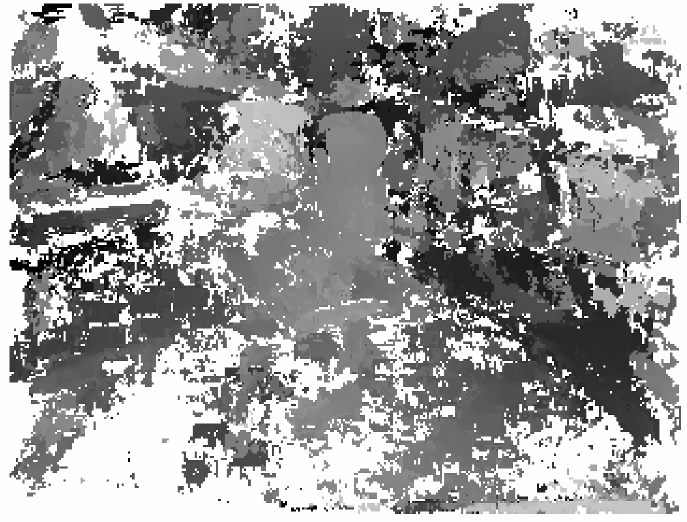
Limitations of Binocular Processing

Errors due to:

- Occlusion
- Lack of texture



Matching candidates in disparity space



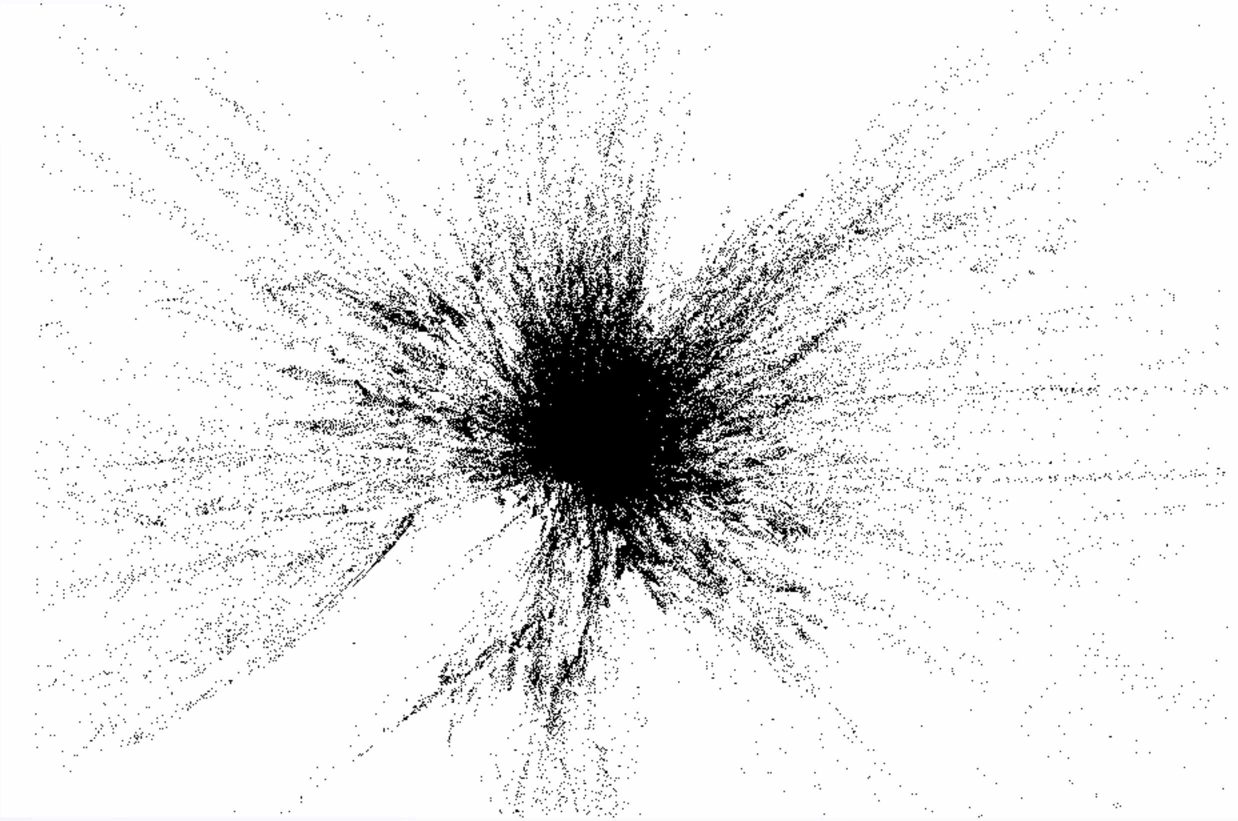
Disparity map after binocular processing

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

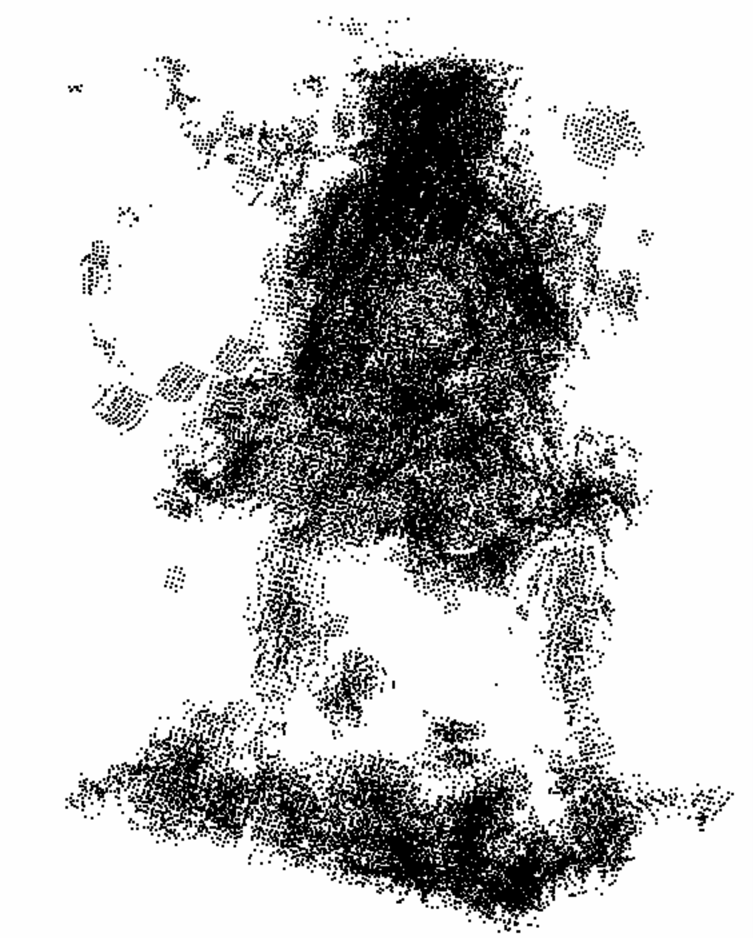


10 image pairs, 1.1 million points

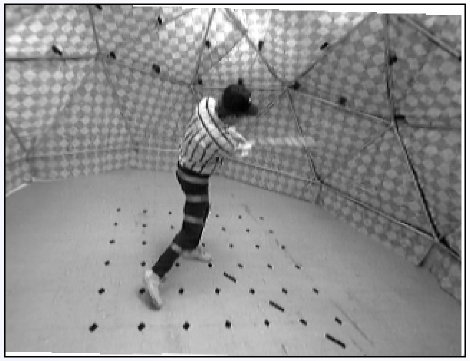
Tensor voting takes 44 min. 30 sec. (March 2004)

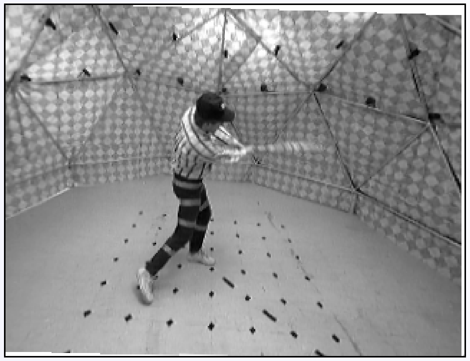
Results on “Meditation” Set

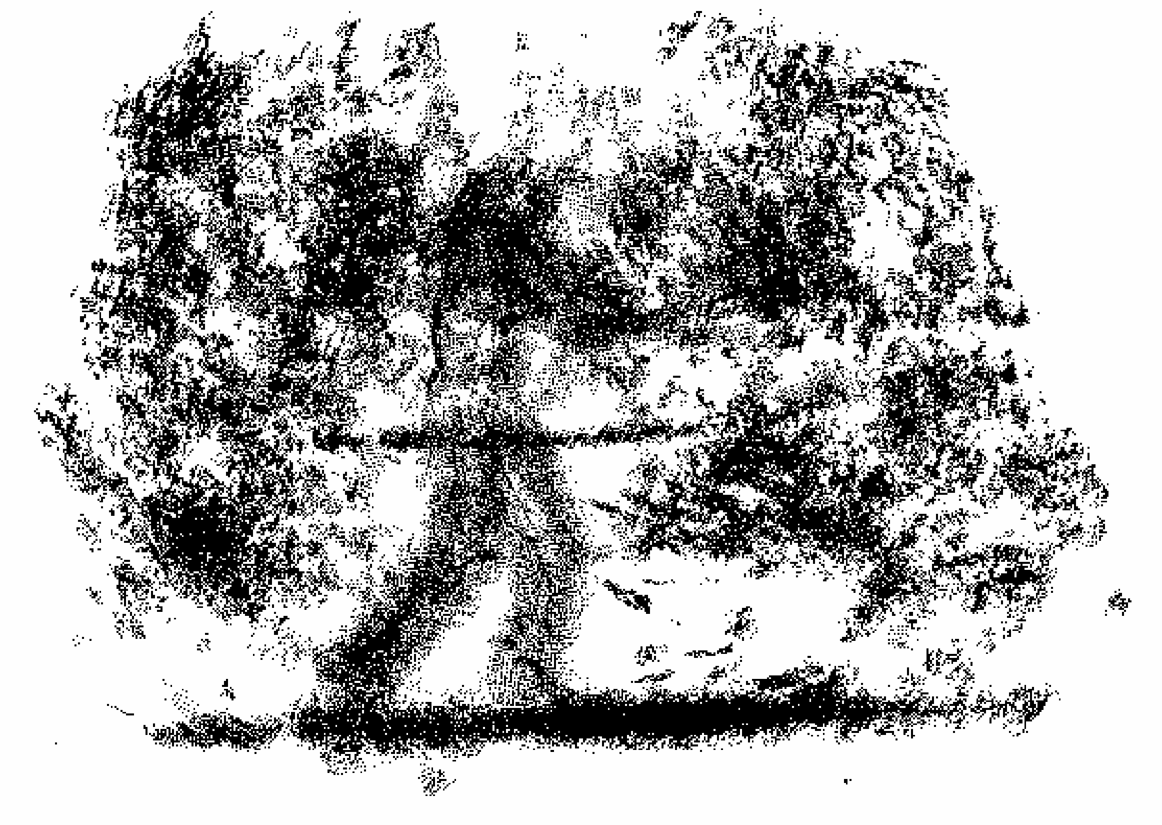
Results on “Meditation” Set



Results on “Baseball” Dataset







Overview

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

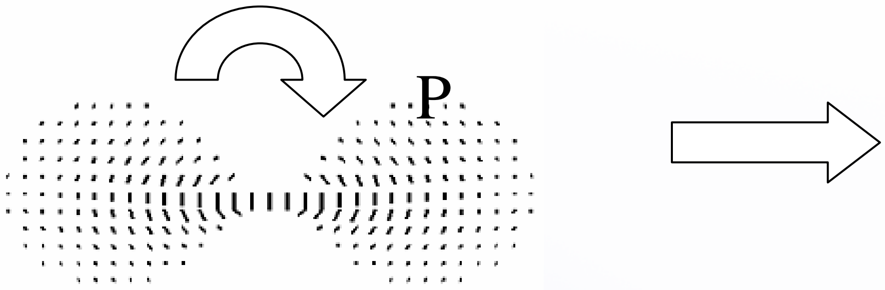
Tensor Representation in N-D

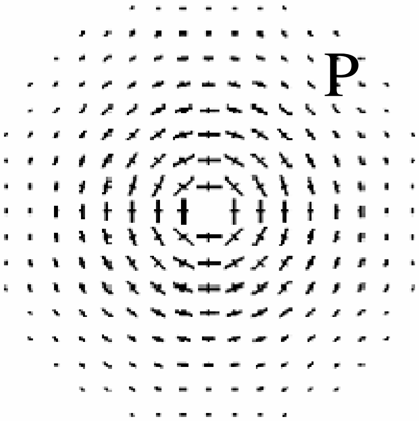
- Non-accidental alignment, proximity, good continuation apply in N-D
  - Robot arm moving from point to point forms 1-D trajectory (manifold) in N-D space
- Noise robustness and ability to represent all structure types also desirable
- Tensor construction:
  - eigenvectors of normal space associated with non-zero eigenvalues
  - eigenvectors of tangent space associated with non-zero eigenvalues

[Mordohai, PhD Thesis 2005]

Limitations of Voting Fields

S(P)





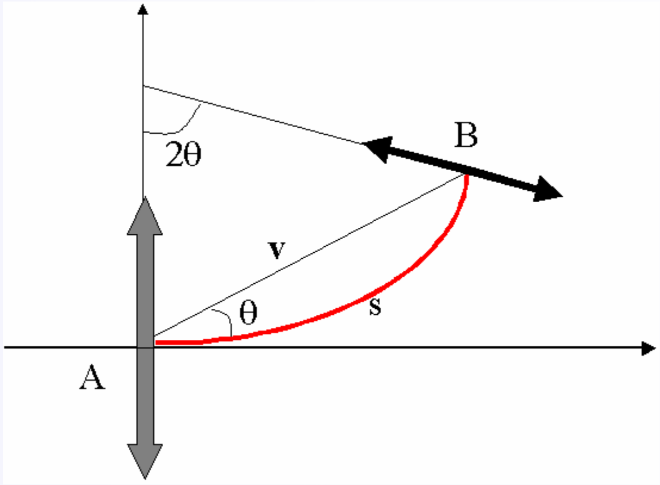
B(P)

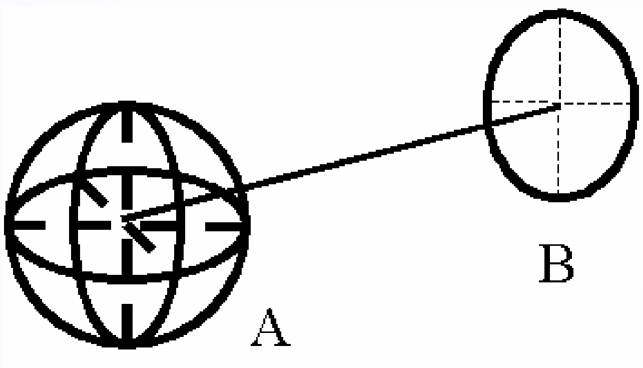
$$\mathbf{B}(P) = \int \mathbf{S}(P)\,d\theta$$

- 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

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



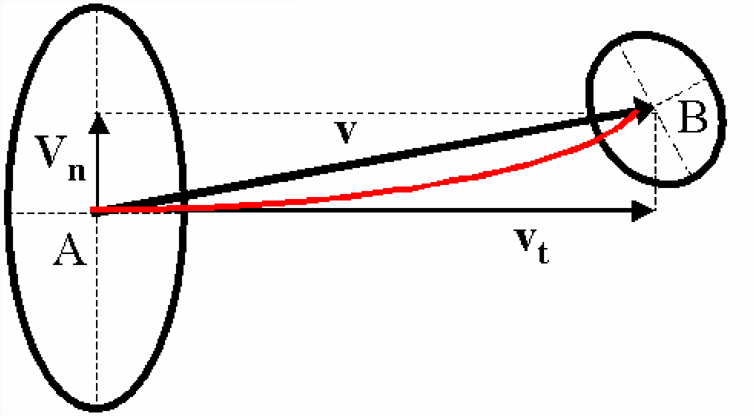


$$\mathbf{T}(\mathrm{s},\theta) = e^{-\left(\frac{s^2}{\sigma^2}\right)}\left(\mathcal{I} - \frac{\vec{v}\vec{v}^T}{\|\vec{v}\vec{v}^T\|}\right)$$

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

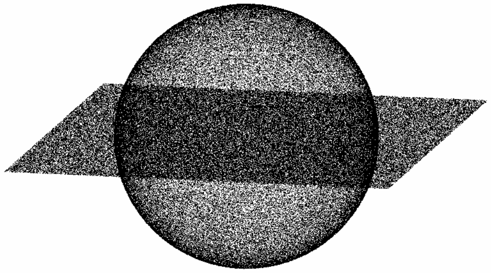
- Tensor decomposition:

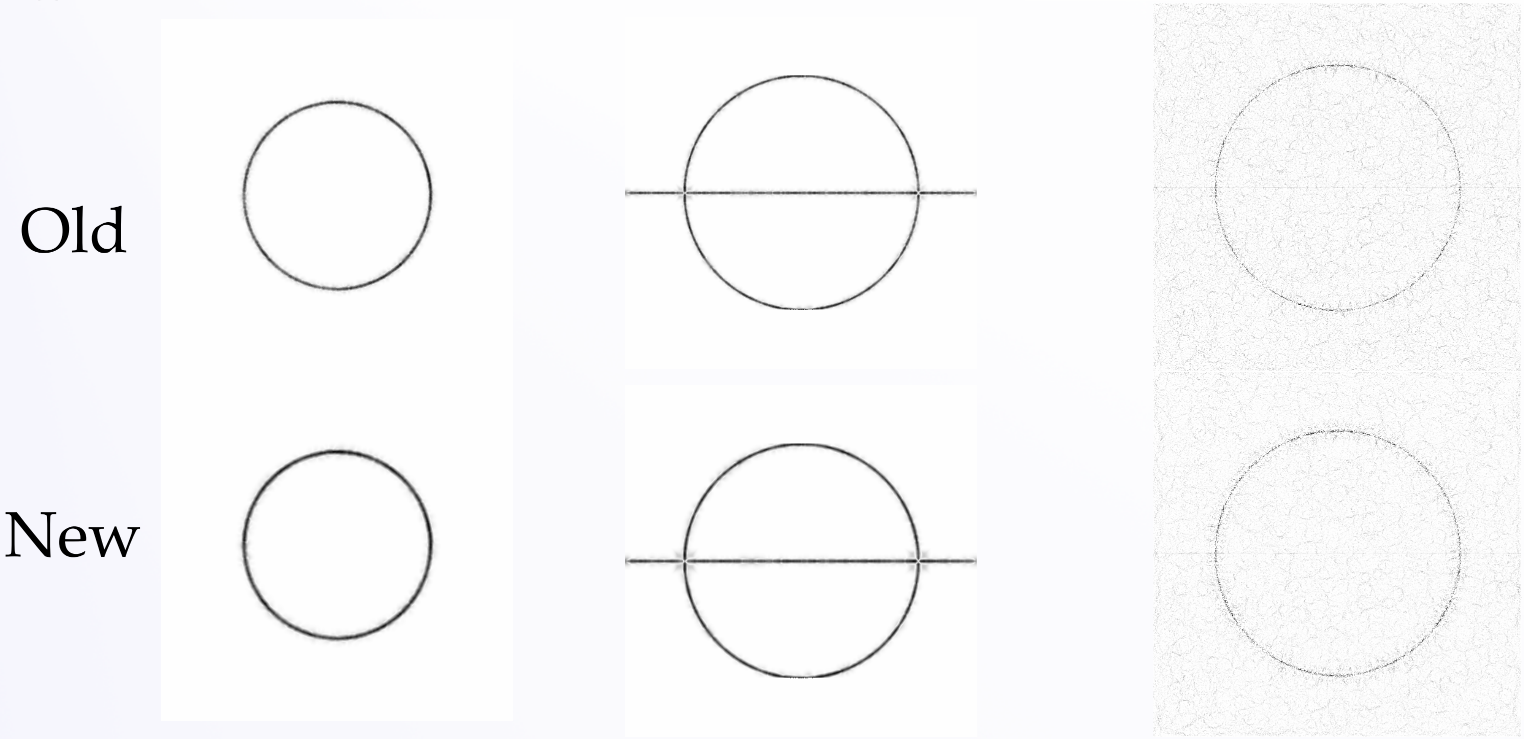
$$\mathbf{T} = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T + \ldots + \lambda_N \cdot e_N e_N^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) + \ldots + \lambda_N \sum_d e_d e_d^T$$

- Dimensionality estimate: *d* with max{λ_d- λ_{d+1}}
- Orientation estimate: normal subspace spanned by *d* eigenvectors corresponding to *d* largest eigenvalues

Comparison with Original Implementation

- Tested on 3-D data
- Saliency maps qualitatively equivalent





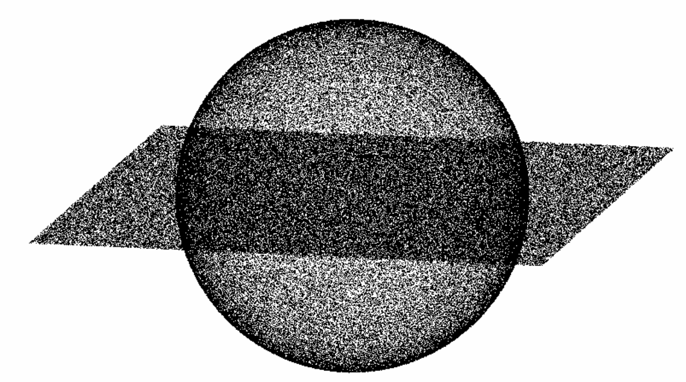
Surface saliency
z=120

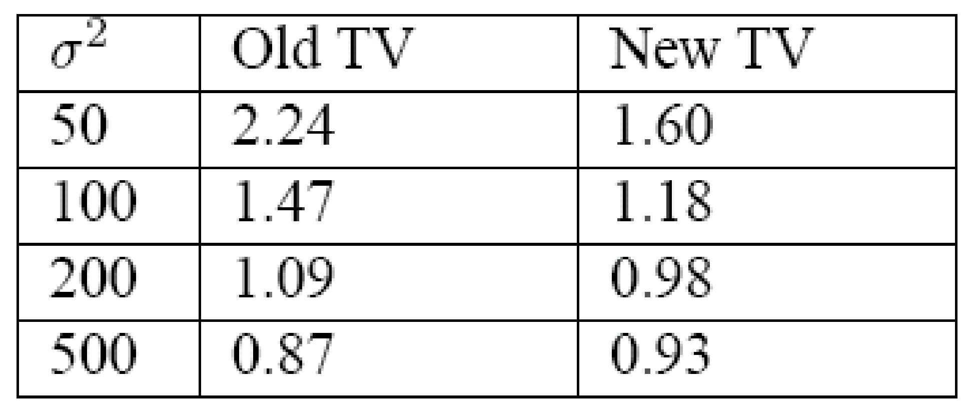
Surface saliency
z=120

Curve saliency
z=0

Comparison with Original Implementation

- Surface orientation estimation
  - Inputs encoded as ball tensors
- Slightly in favor of new implementation
  - Pre-computed voting fields use interpolation

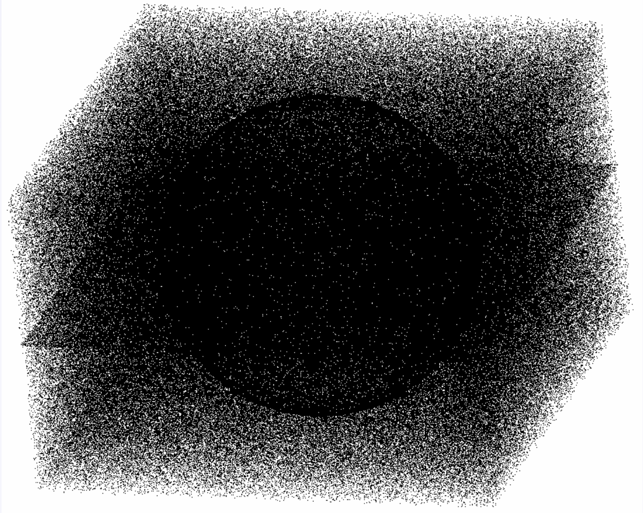




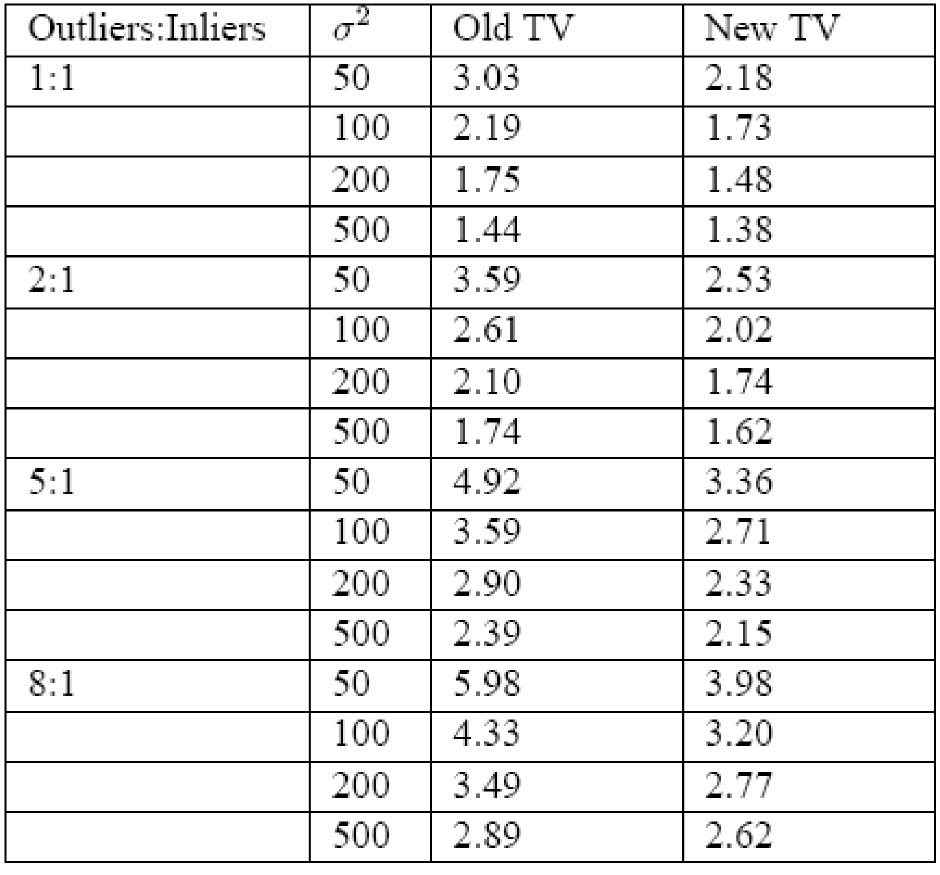
σ^2	Old TV	New TV
50	2.24	1.60
100	1.47	1.18
200	1.09	0.98
500	0.87	0.93

Comparison with Original Implementation

- Noisy data



5:1 outlier to inlier ratio



Outliers:Inliers	σ^2	Old TV	New TV
1:1	50	3.03	2.18
	100	2.19	1.73
	200	1.75	1.48
	500	1.44	1.38
2:1	50	3.59	2.53
	100	2.61	2.02
	200	2.10	1.74
	500	1.74	1.62
5:1	50	4.92	3.36
	100	3.59	2.71
	200	2.90	2.33
	500	2.39	2.15
8:1	50	5.98	3.98
	100	4.33	3.20
	200	3.49	2.77
	500	2.89	2.62

Overview

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

Instance-based Learning

- Learn from observations in continuous domain
- Observations are *N*-D vectors
- Estimate:
  - Intrinsic dimensionality
  - Orientation of manifold through each observation
  - Generate new samples on manifold

Approach

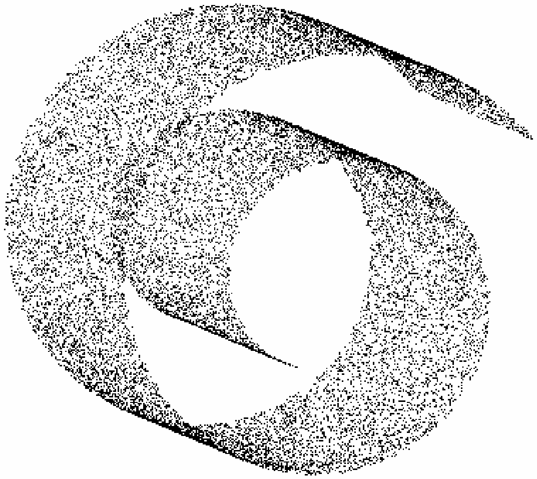
- Vote as before in *N*-D space
- Intrinsic dimensionality found as maximum gap of eigenvalues
- Do not perform dimensionality reduction
- Applicable to:
  - Data of varying dimensionality
  - Manifolds with intrinsic curvature (distorted during unfolding)
  - “Unfoldable” manifolds (spheres, hyper-spheres)
  - Intersecting manifolds

Approach

- Manifold orientation from eigenvectors
- “Eager learning” since all inputs are processed and queries do not affect estimates
  - “Lazy” alternative: collect votes on query points
  - Requires stationary data distribution

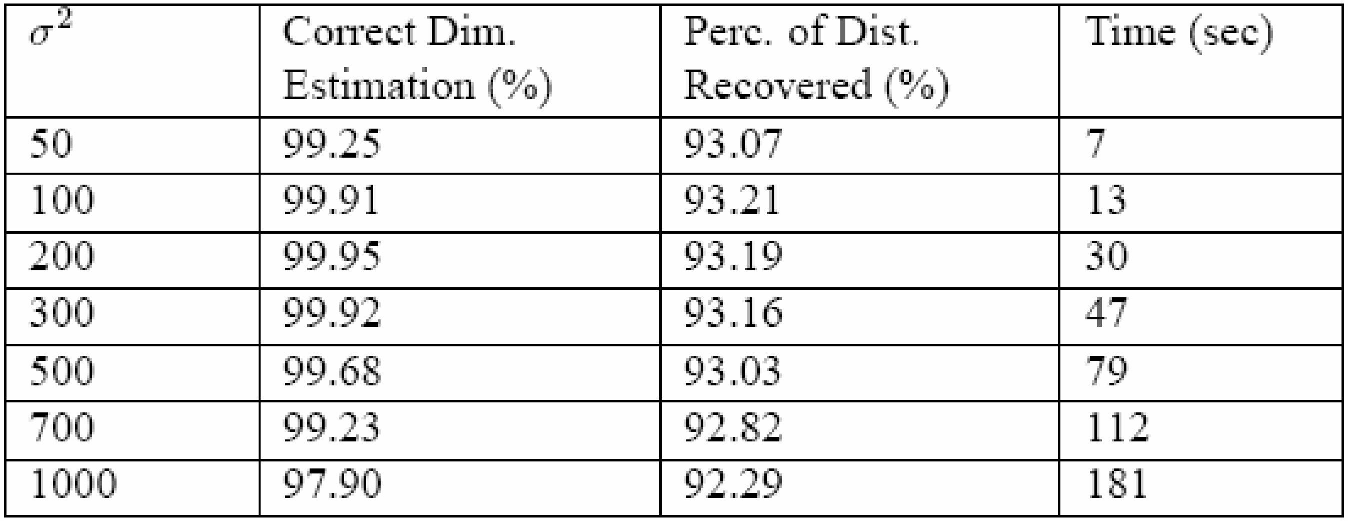
Dimensionality Estimation

[Mordohai and Medioni, IJCAI 2005]



20,000 points on 2-D manifold in 3-D

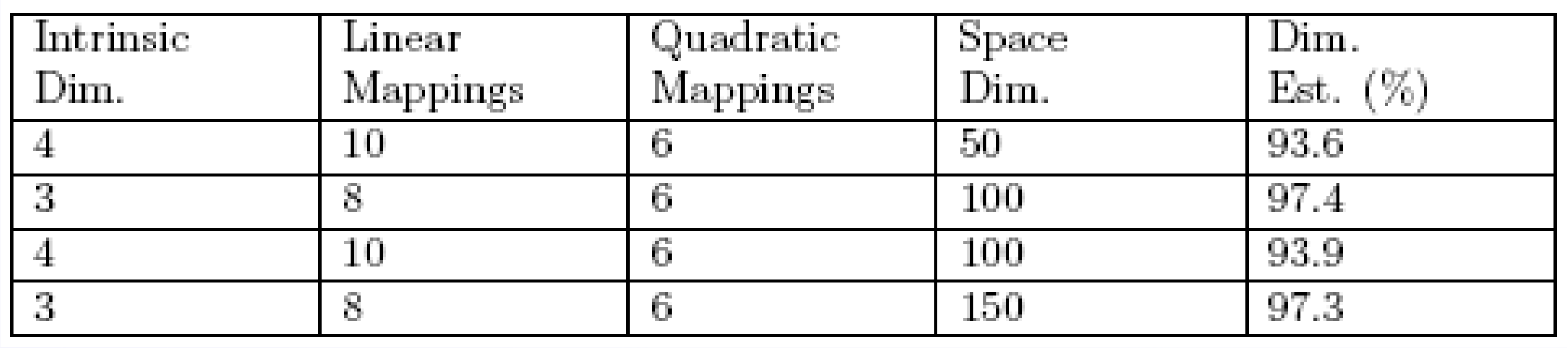
"Swiss Roll"



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

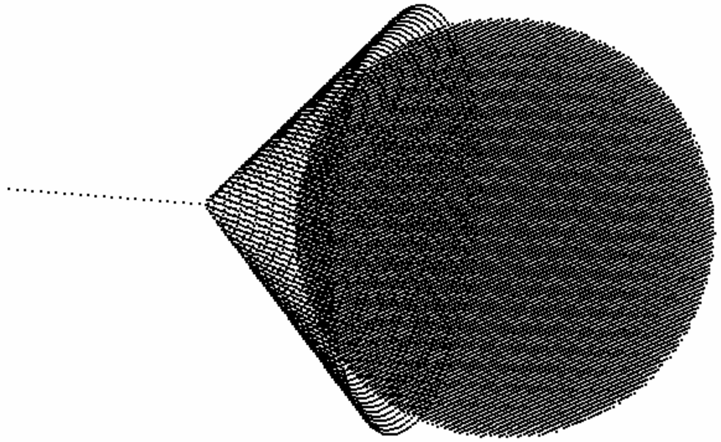
- Synthetic data
- Randomly sample input variables (intrinsic dimensionality)
- Map them to higher dimensional vector using linear and quadratic functions
- Add noisy dimensions
- Global rotation



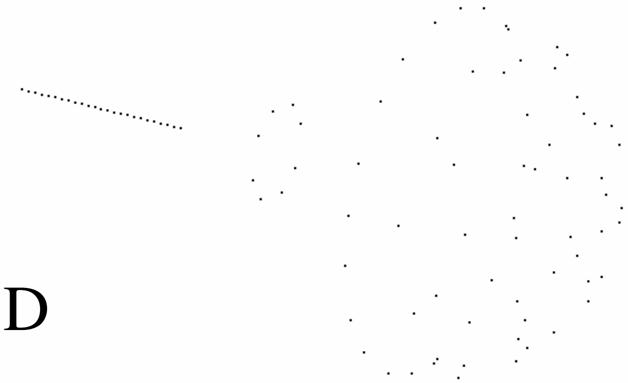
Intrinsic Dim.	Linear Mappings	Quadratic Mappings	Space Dim.	Dim. Est. (%)
4	10	6	50	93.6
3	8	6	100	97.4
4	10	6	100	93.9
3	8	6	150	97.3

Dimensionality Estimation

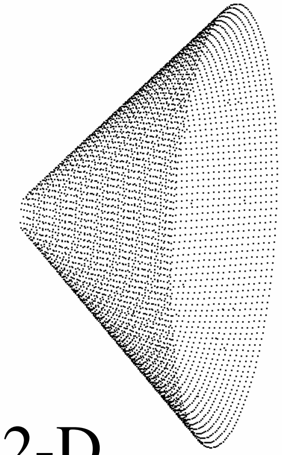
- Point-wise dimensionality estimates
- No global operations



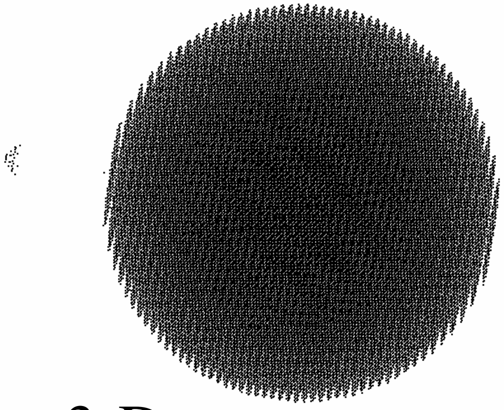
Input in 4-D



1-D



2-D



3-D

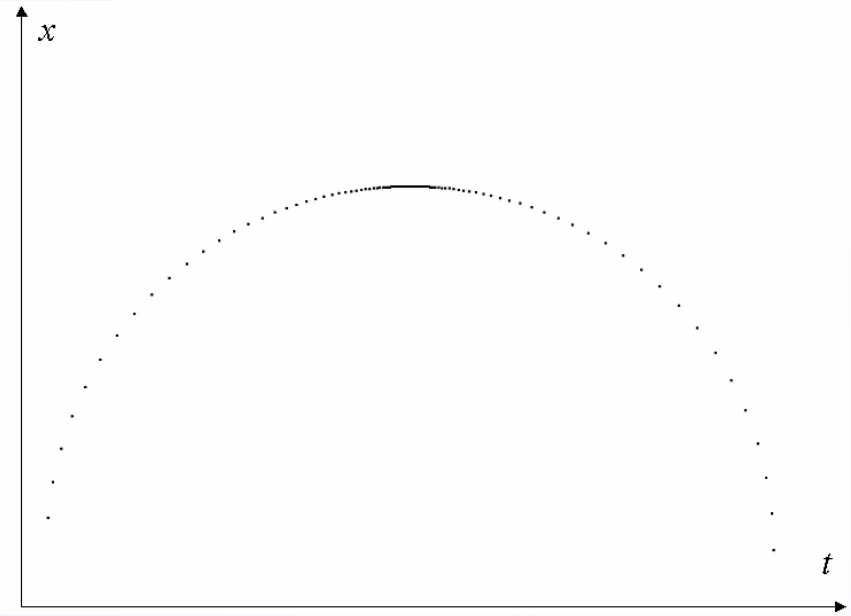
Manifold Learning

- 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

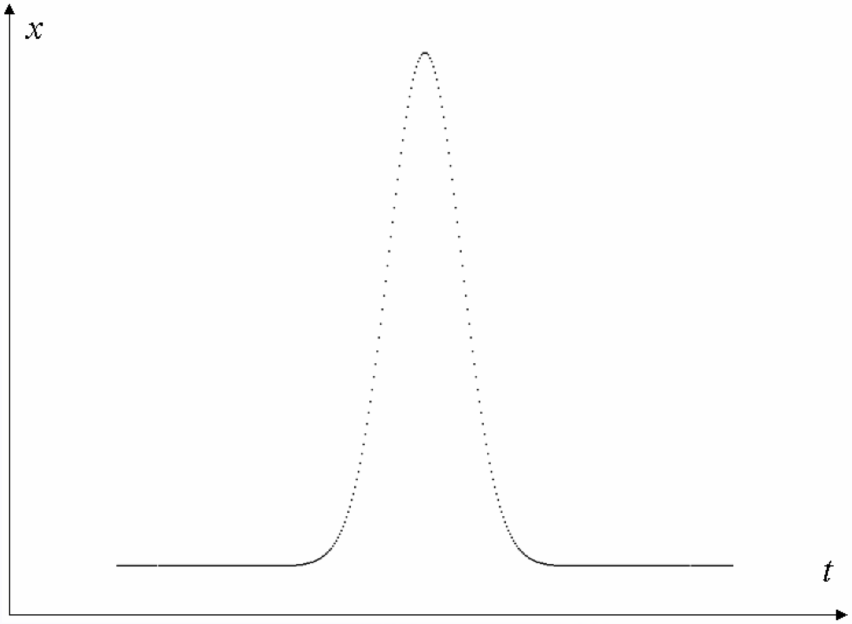
Orientation Estimation

- Functions proposed by [Wang et al. 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

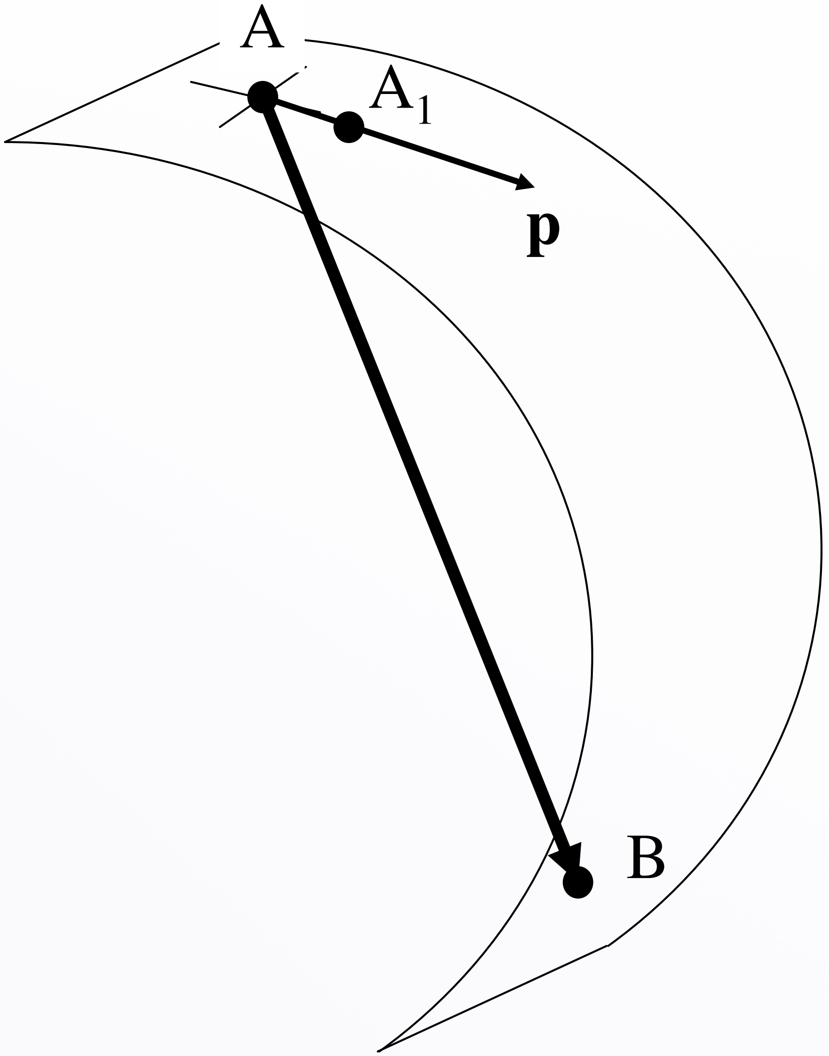


$$x_i = [t_i, \quad 10e^{-t_i^2}]$$

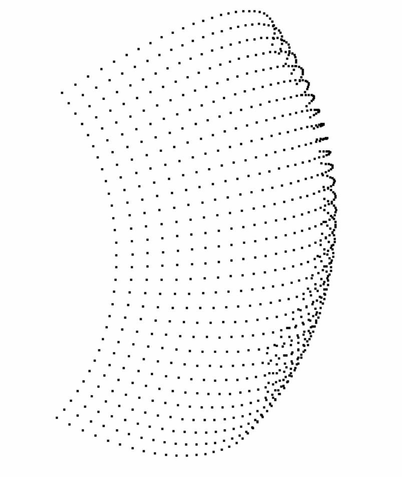
Mean error:
2.02 degrees for 180 samples
0.57 degrees for 360 samples

Manifold Distance Measurement

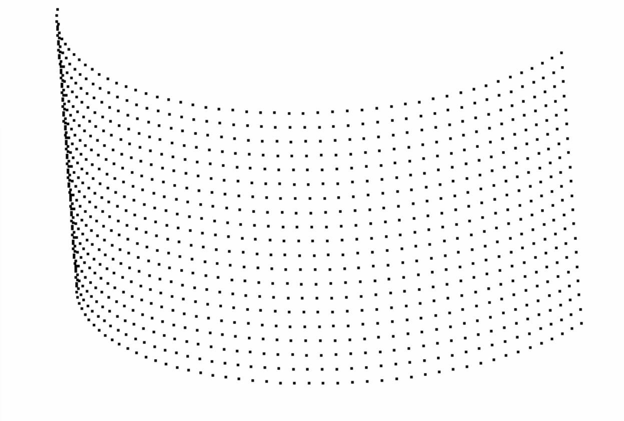
- Do *not* reduce dimensionality
- Start from point on manifold
- Take small step along desired orientation on tangent space
- Generate new point and collect votes
- Repeat until convergence



Distance Measurement: Test Data



Error in orientation estimation: 0.11°

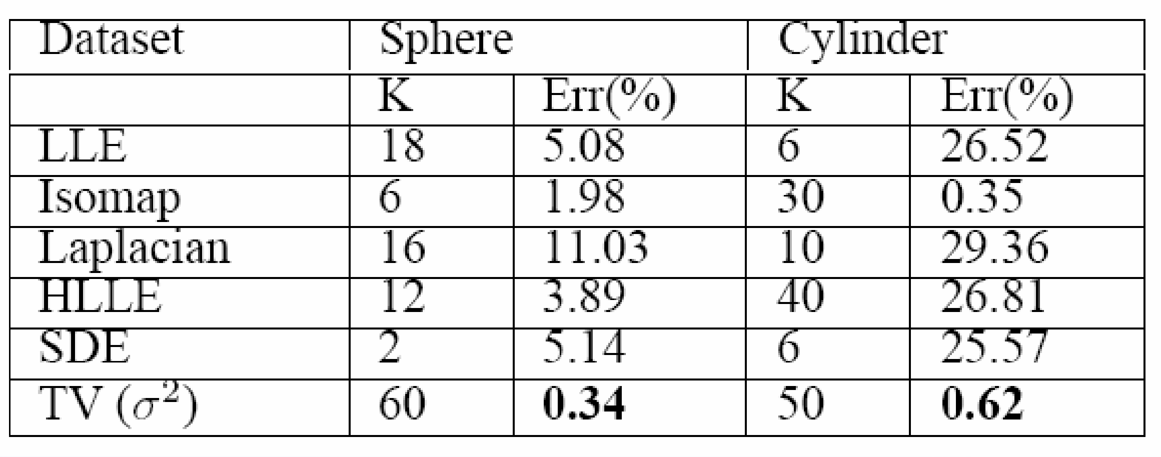


0.26°

- 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

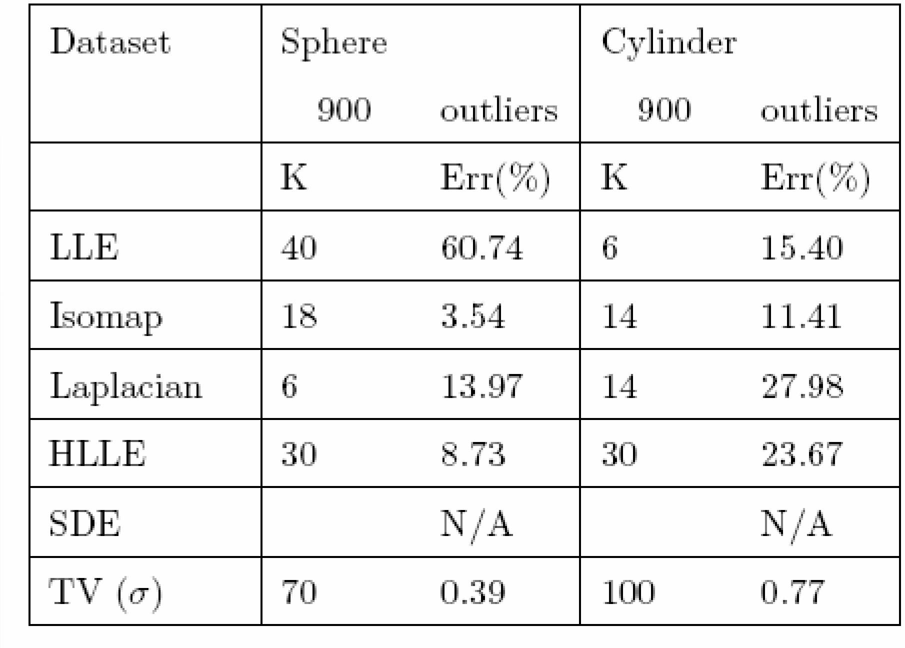
Experimental Setup

- 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

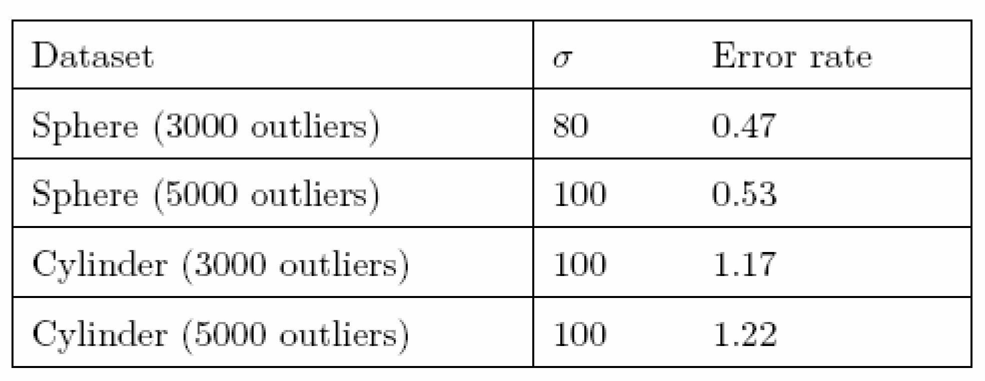


Dataset	Sphere		Cylinder	
	K	Err(%)	K	Err(%)
LLE	18	5.08	6	26.52
Isomap	6	1.98	30	0.35
Laplacian	16	11.03	10	29.36
HLLE	12	3.89	40	26.81
SDE	2	5.14	6	25.57
TV (σ^2)	60	**0.34**	50	**0.62**

Distance Measurement with Outliers

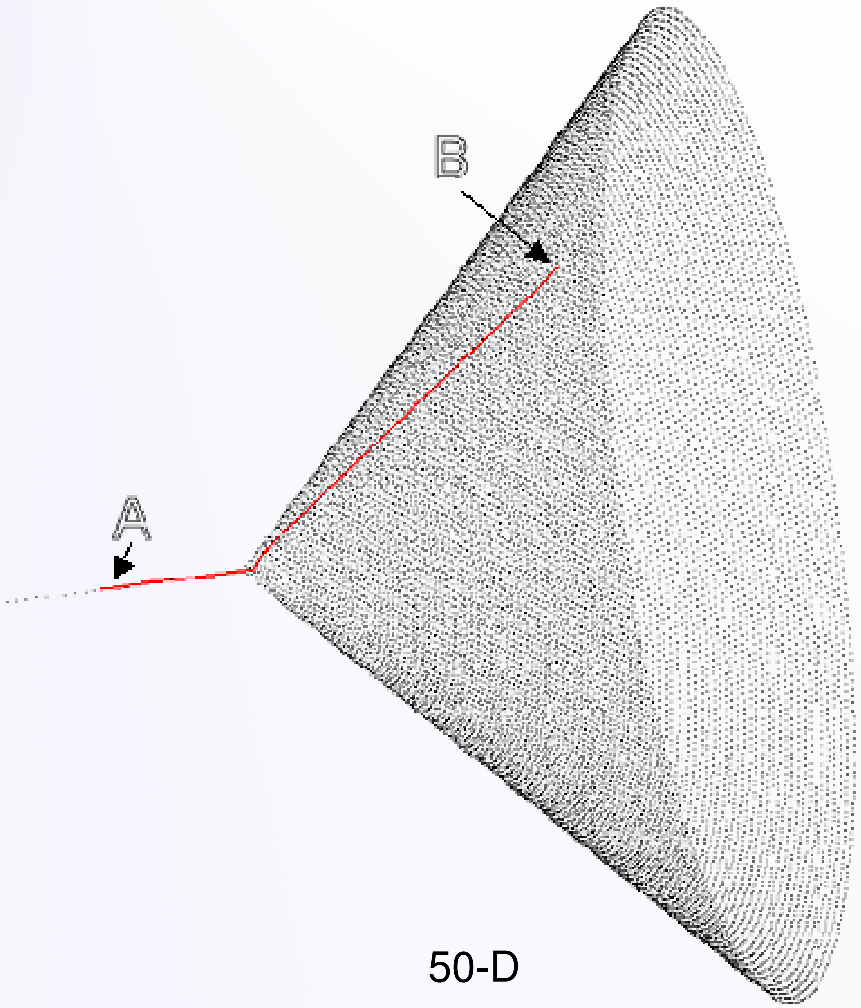


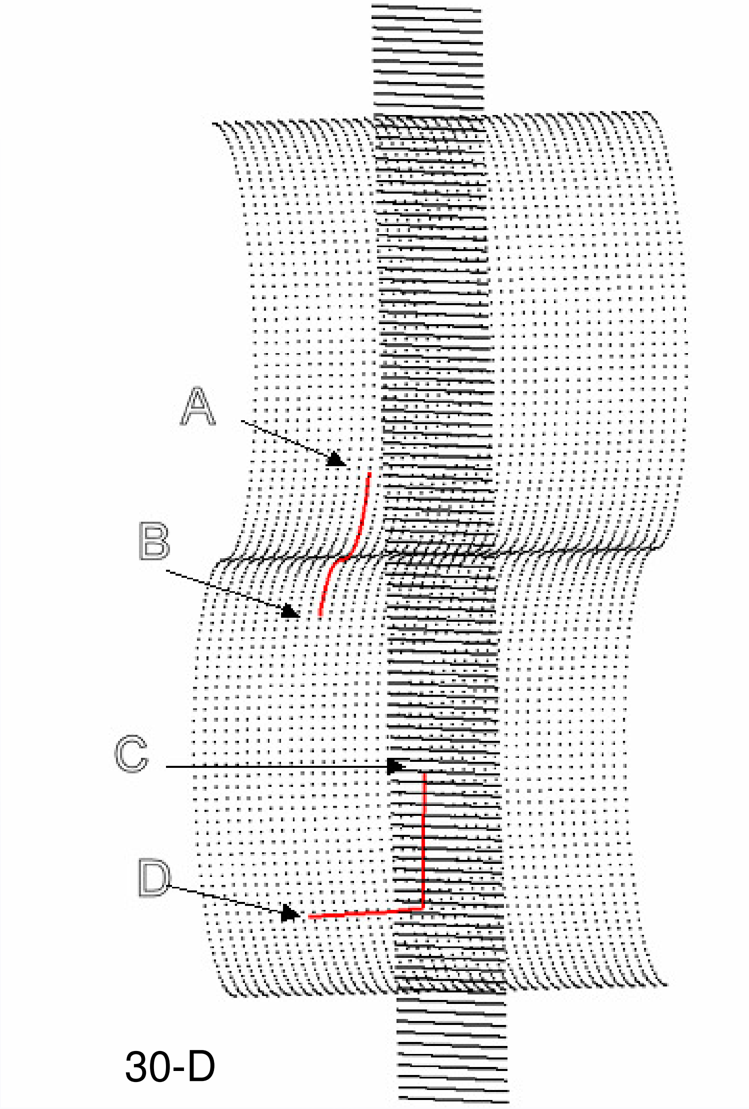
Dataset	Sphere		Cylinder	
	900	outliers	900	outliers
	K	Err(%)	K	Err(%)
LLE	40	60.74	6	15.40
Isomap	18	3.54	14	11.41
Laplacian	6	13.97	14	27.98
HLLE	30	8.73	30	23.67
SDE		N/A		N/A
TV (σ)	70	0.39	100	0.77



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

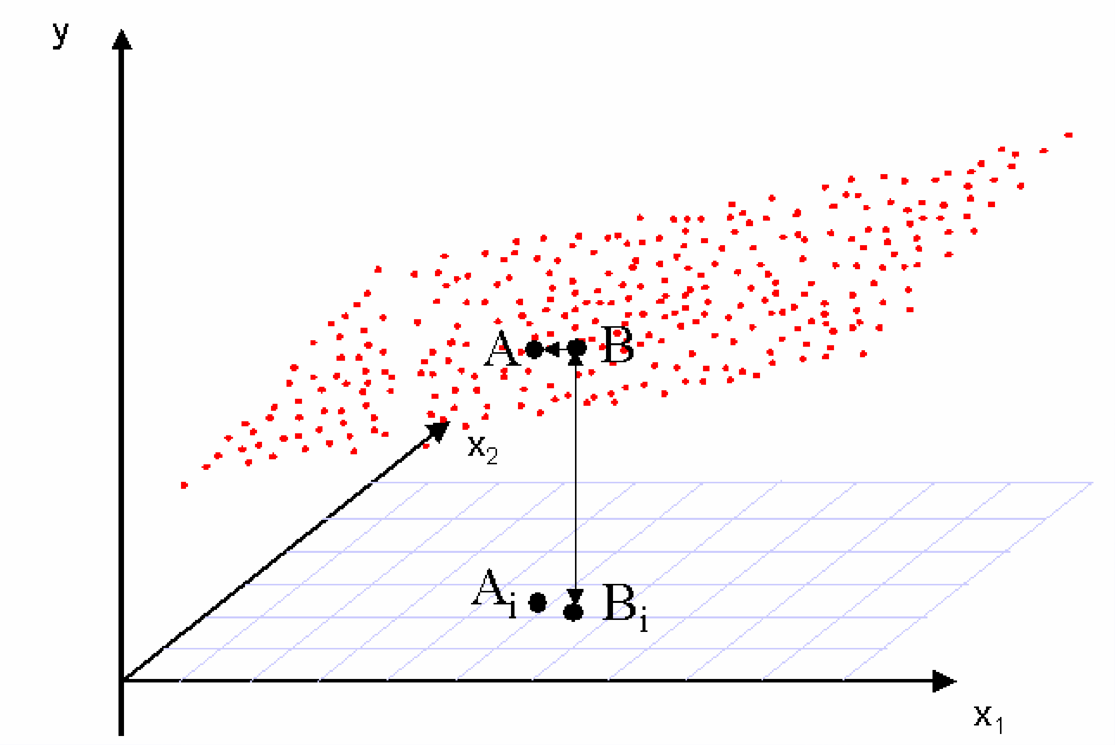
Traveling on Manifolds





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

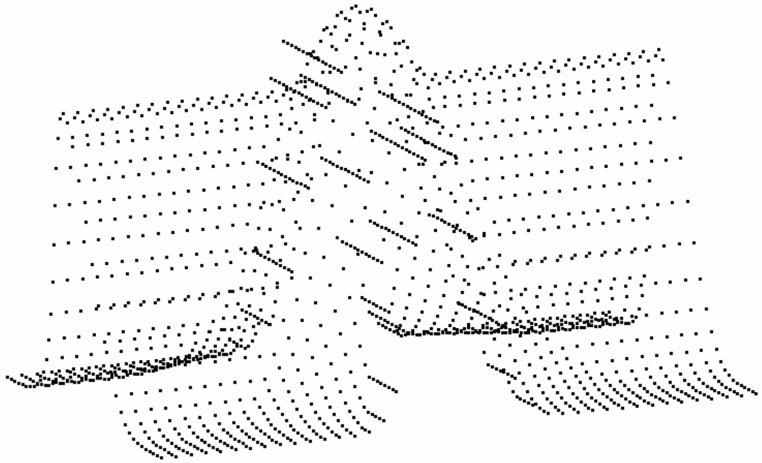


Synthetic Data

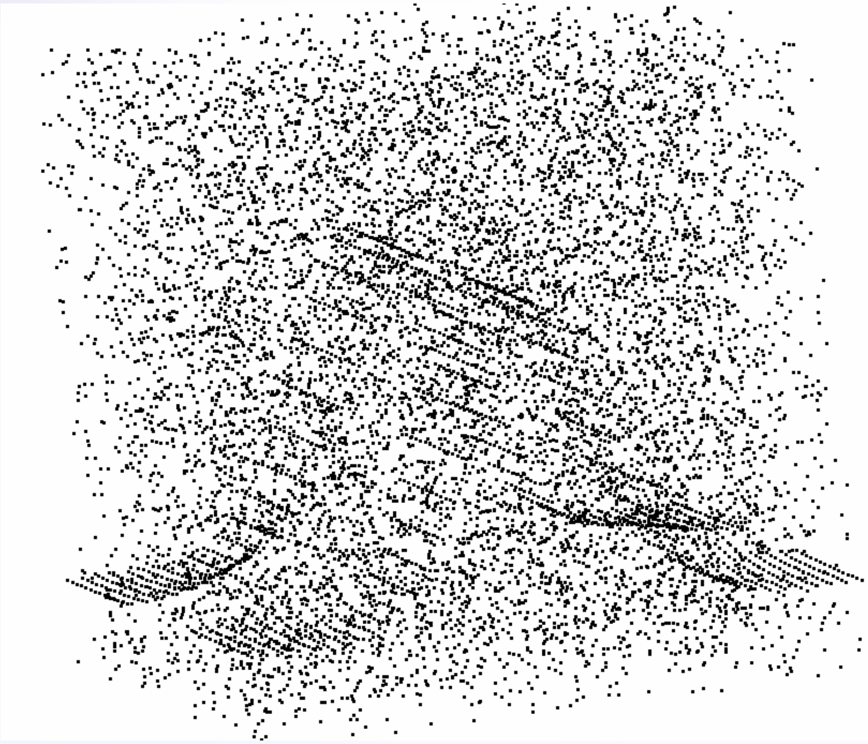
- Sample 1681 points from

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

  proposed by Schaal 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

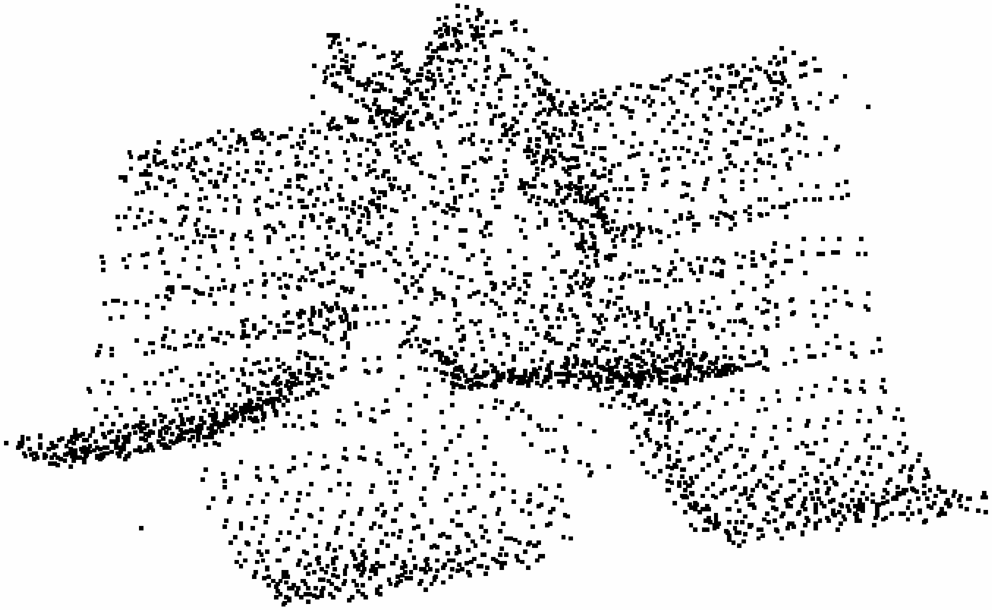


Synthetic Data

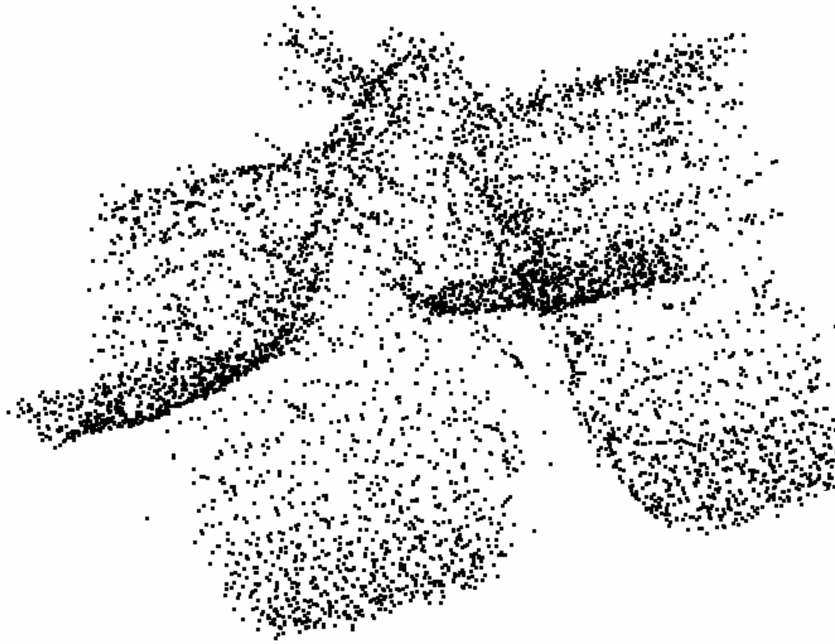


Noisy input

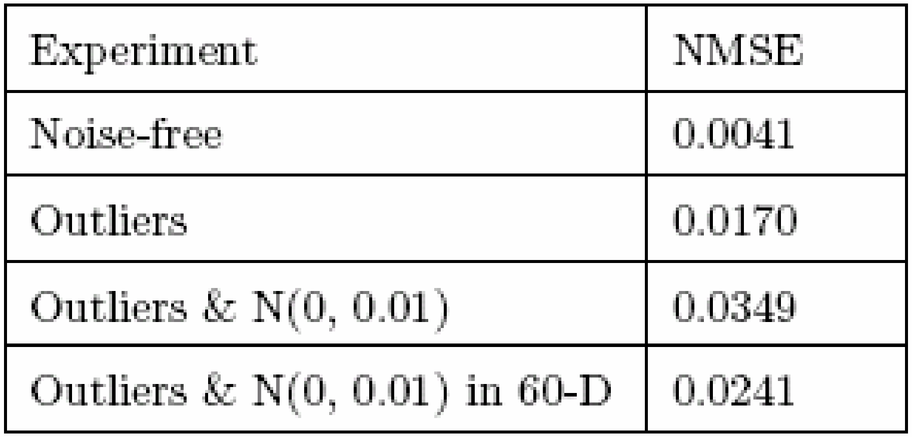
New points



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

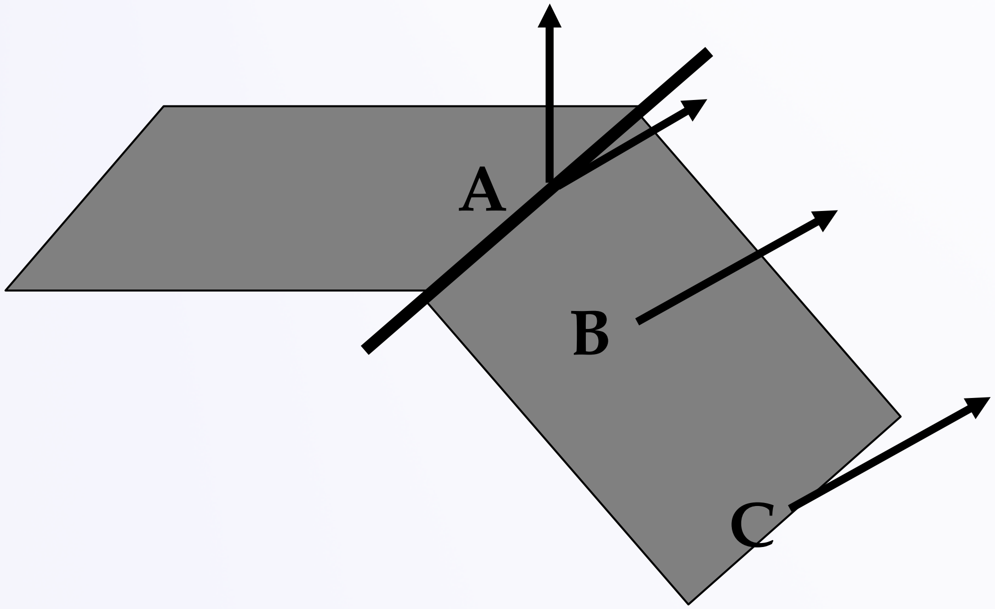
- Limitations:
  - Need more data
  - "Regular" distribution

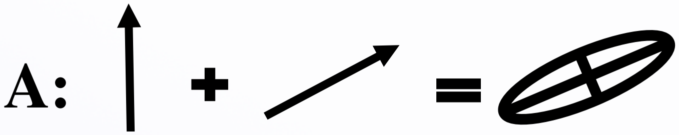
Overview

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

Boundary Inference

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



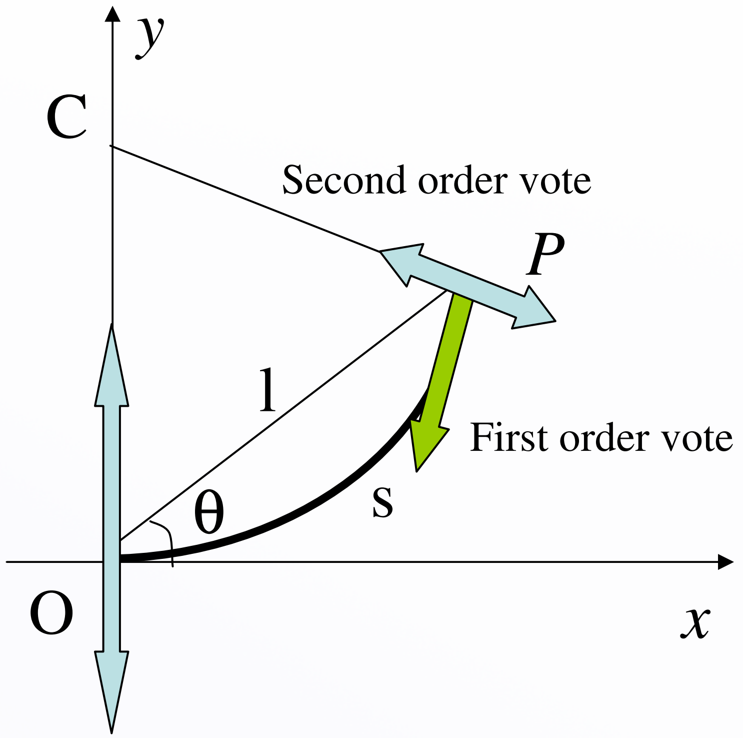


Tensor with dominant plate component (orthogonal to surface intersection)

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



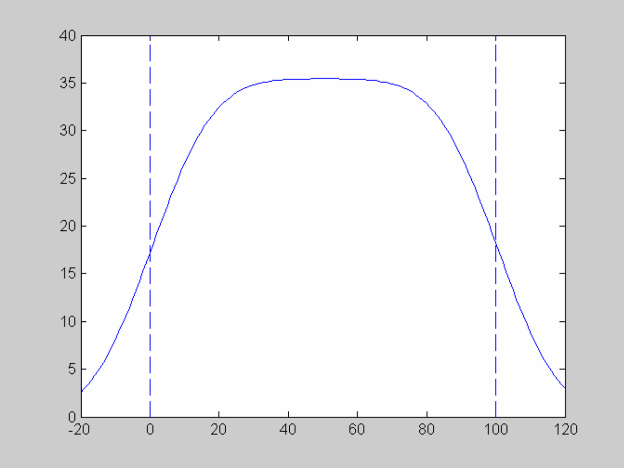
$$\mathbf{S}_{fo}(l,\theta,\sigma) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)} \begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix}$$

[Tong, Tang, Mordohai and Medioni, PAMI 2004]

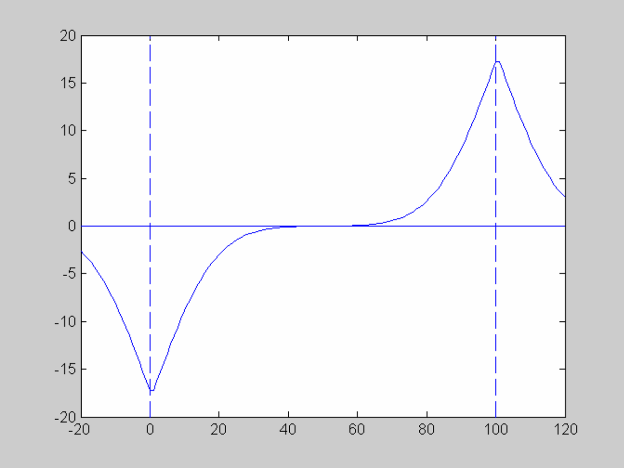
Illustration of Polarity

....

Input

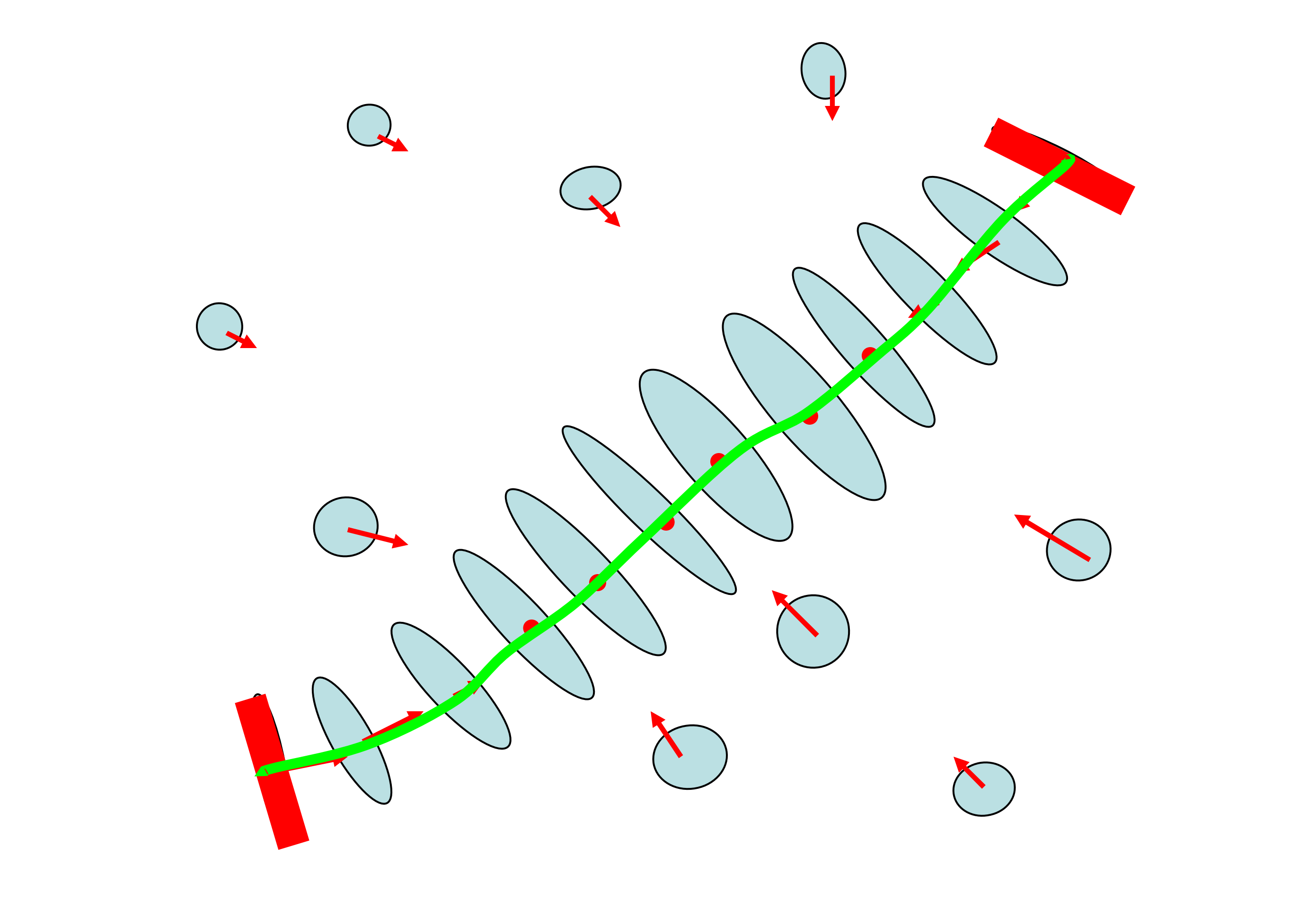


Curve Saliency



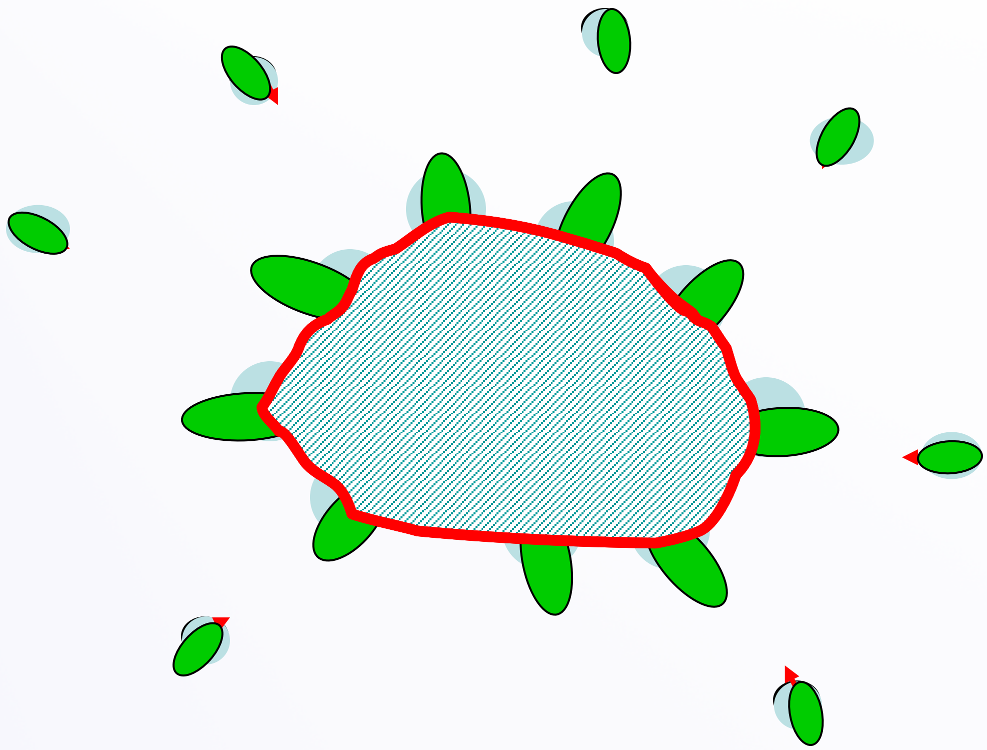
Polarity

Illustration of First Order Voting

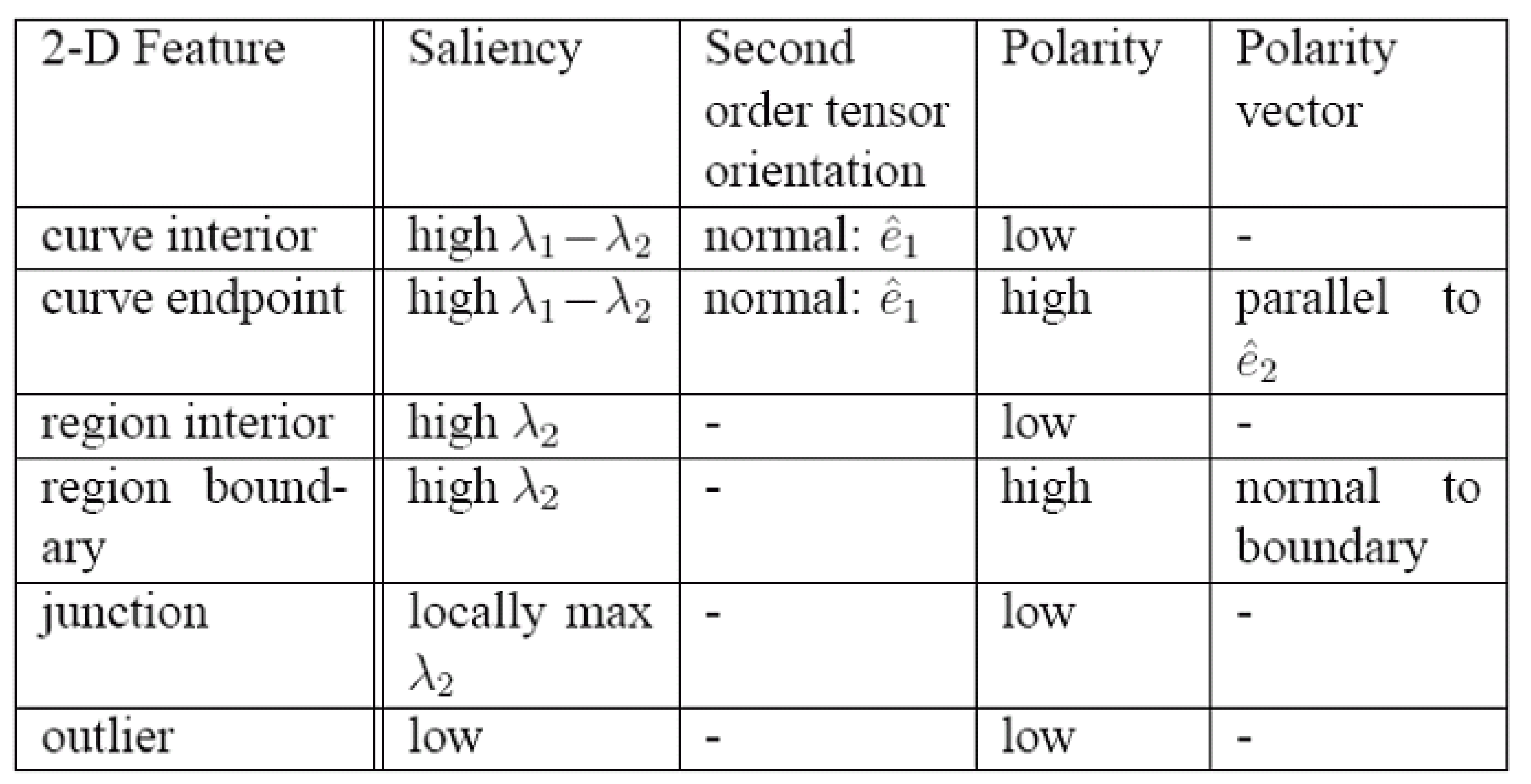


Tensor Voting with first order properties

Illustration of Region Inference

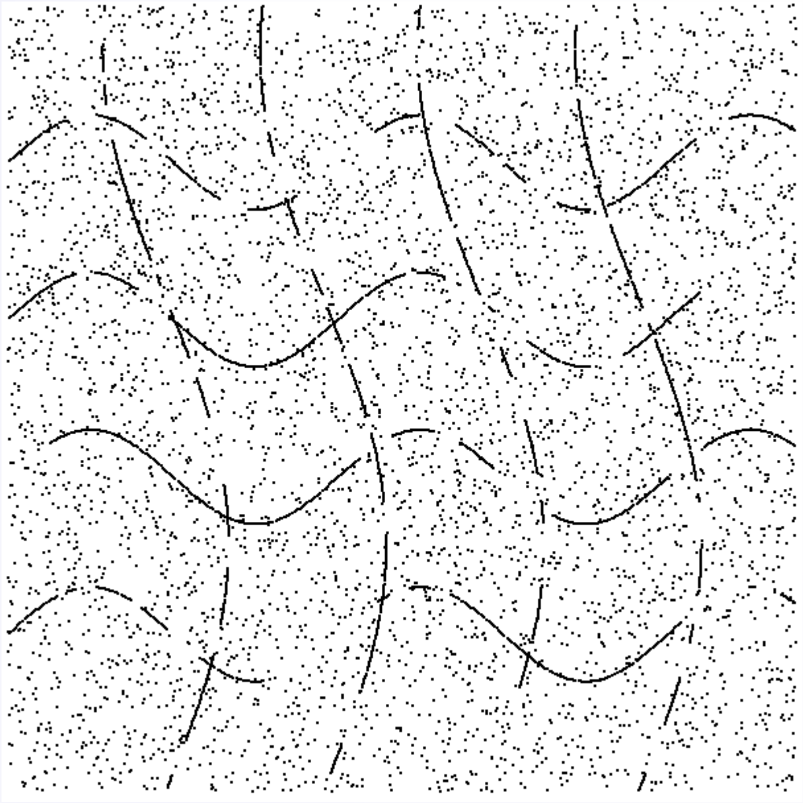


Vote Analysis in 2-D

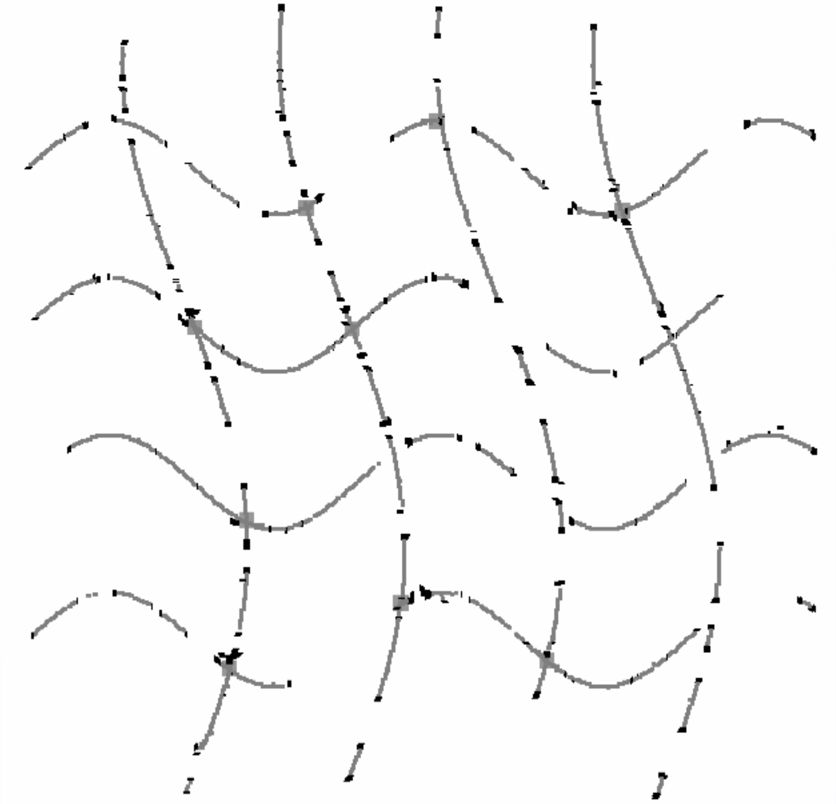


2-D Feature	Saliency	Second order tensor orientation	Polarity	Polarity vector
curve interior	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	low	-
curve endpoint	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	high	parallel to $\hat{e}_2$
region interior	high λ_2	-	low	-
region boundary	high λ_2	-	high	normal to boundary
junction	locally max λ_2	-	low	-
outlier	low	-	low	-

Results in 2-D

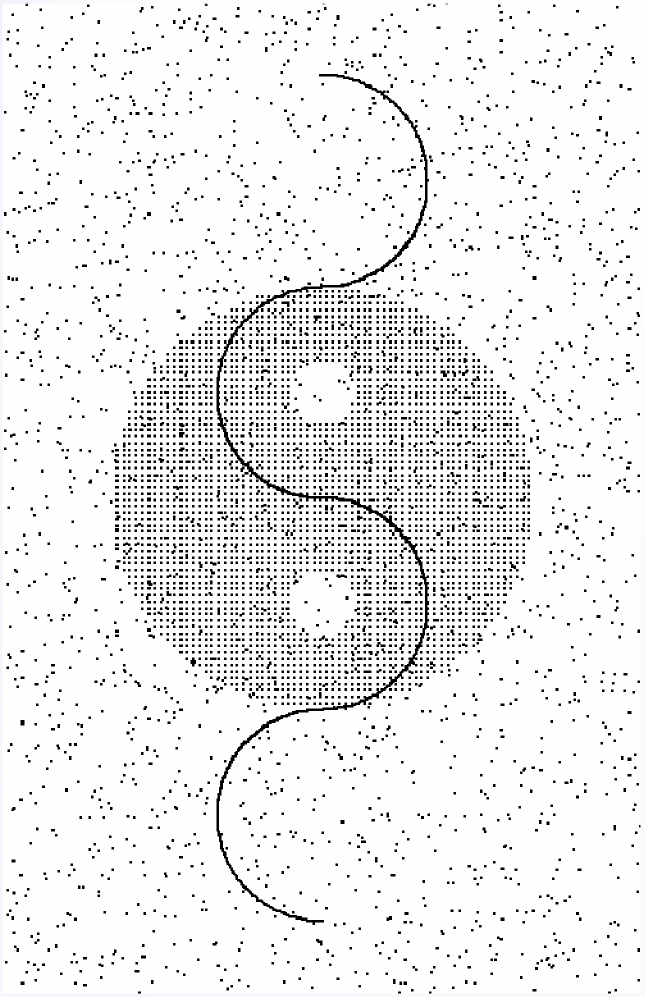


Input

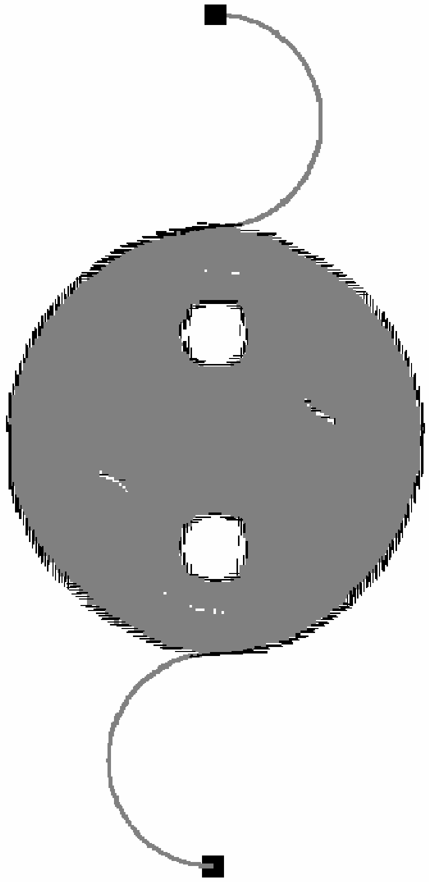


Gray: curve inliers
Black: curve endpoints
Squares: junctions

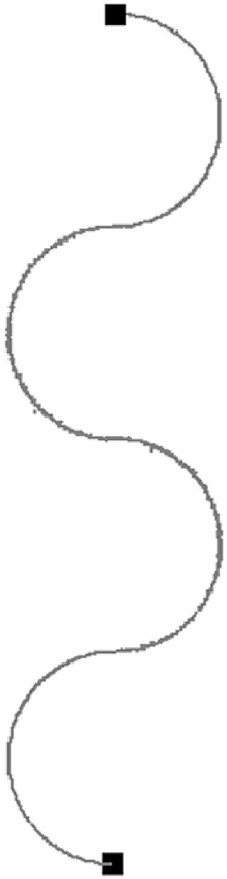
Results in 2-D



Input

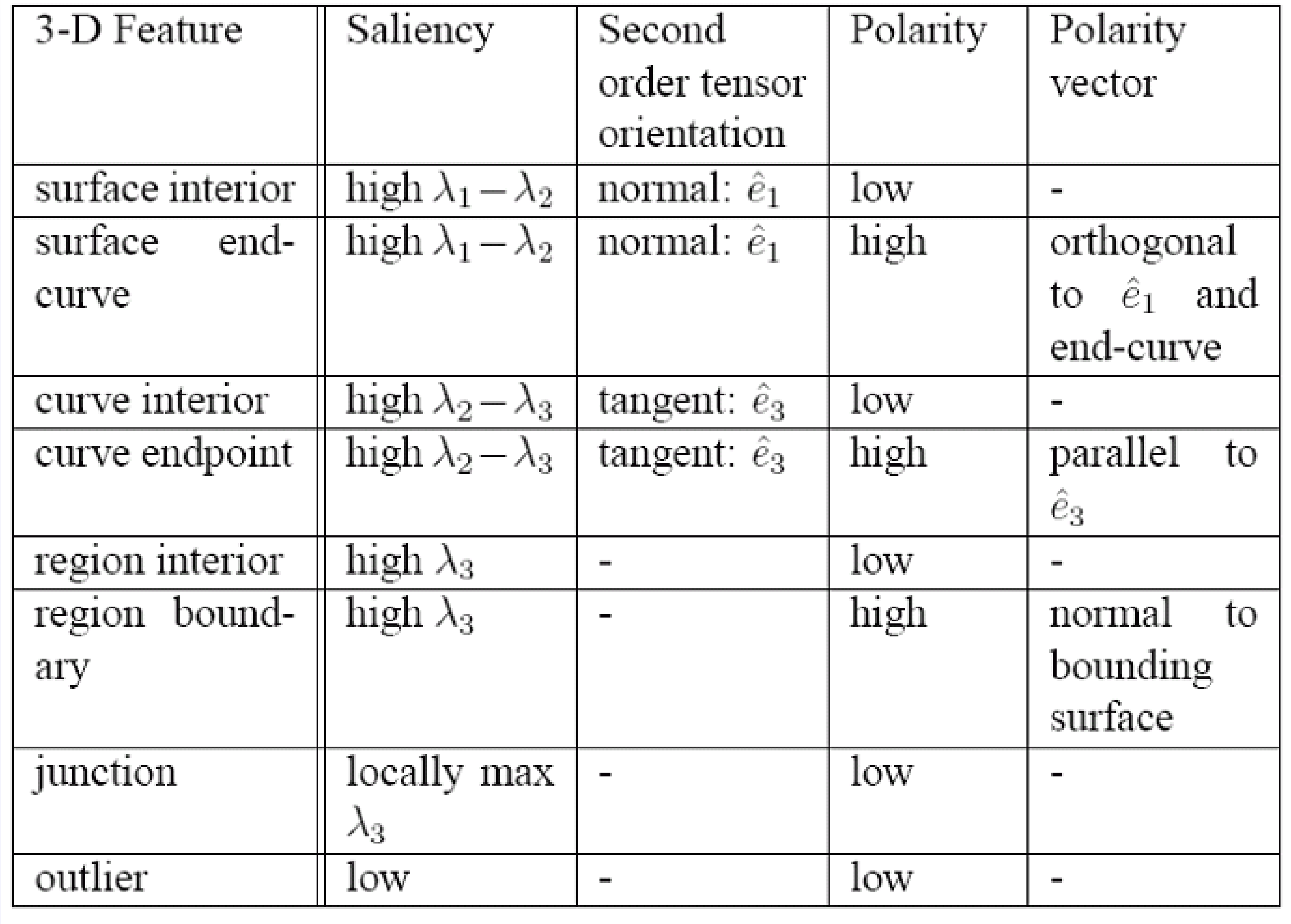


Curves, endpoints, regions and region boundaries



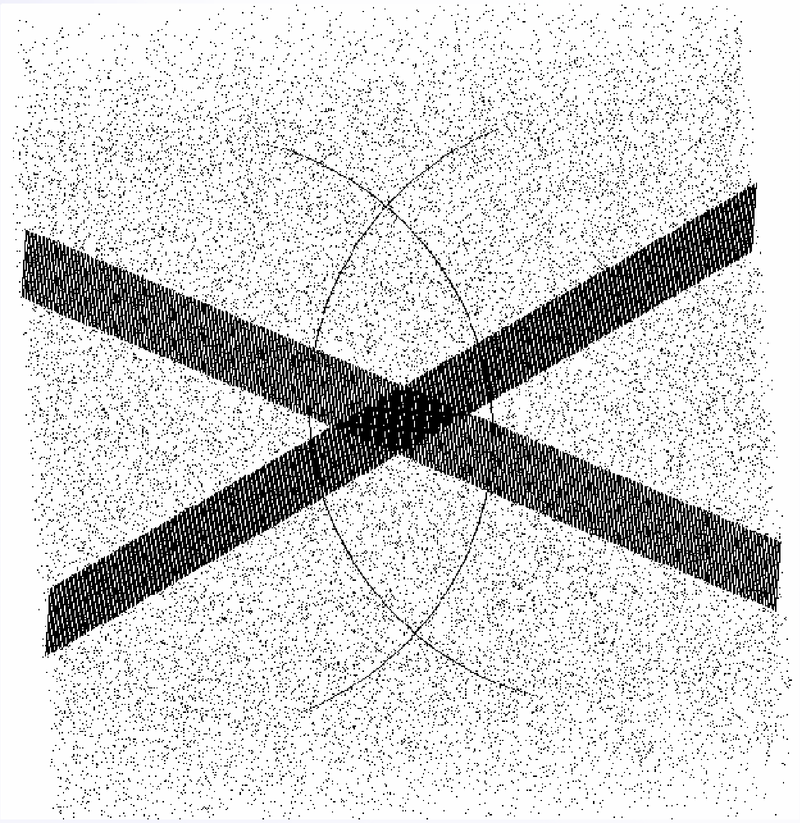
Curves and endpoints only

Vote Analysis in 3-D

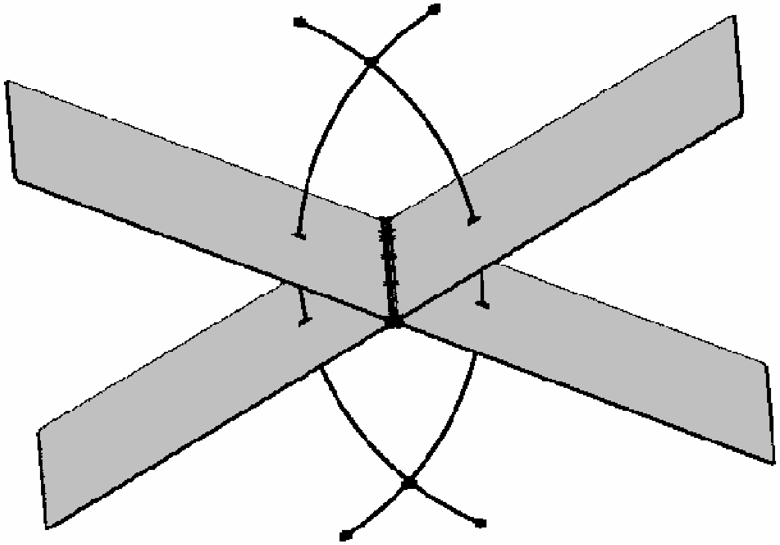


3-D Feature	Saliency	Second order tensor orientation	Polarity	Polarity vector
surface interior	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	low	-
surface end-curve	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	high	orthogonal to $\hat{e}_1$ and end-curve
curve interior	high $\lambda_2 - \lambda_3$	tangent: $\hat{e}_3$	low	-
curve endpoint	high $\lambda_2 - \lambda_3$	tangent: $\hat{e}_3$	high	parallel to $\hat{e}_3$
region interior	high λ_3	-	low	-
region boundary	high λ_3	-	high	normal to bounding surface
junction	locally max λ_3	-	low	-
outlier	low	-	low	-

Results in 3-D

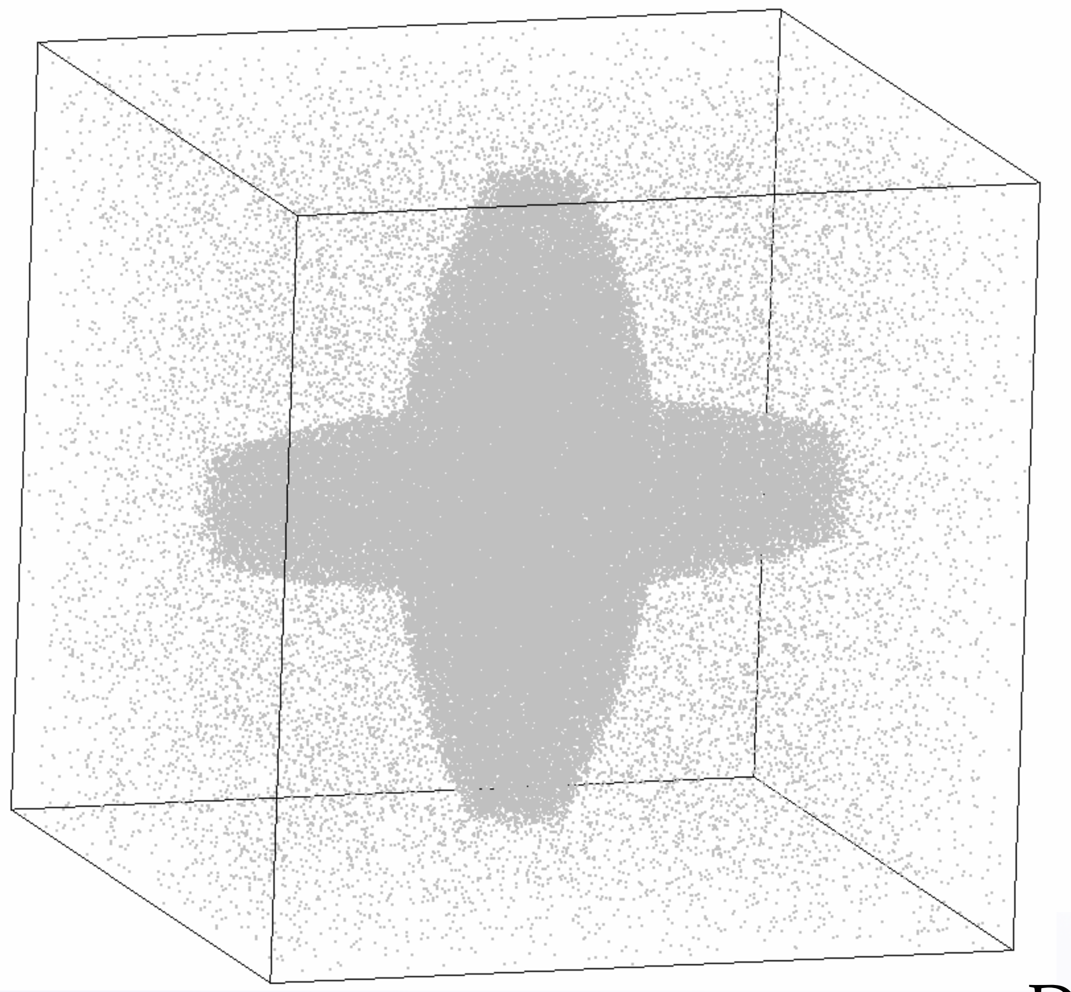


Input

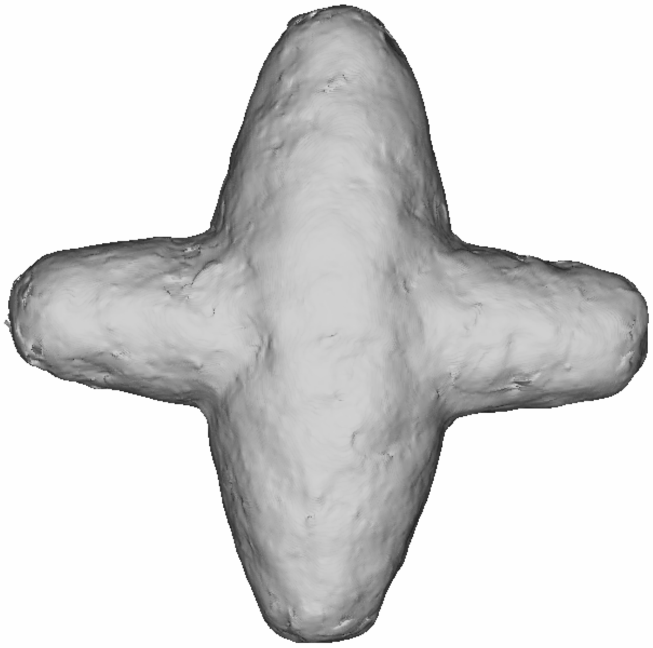


Surfaces - Surface Boundaries – Surface Intersections
Curves – Endpoints - Junctions

Results in 3-D

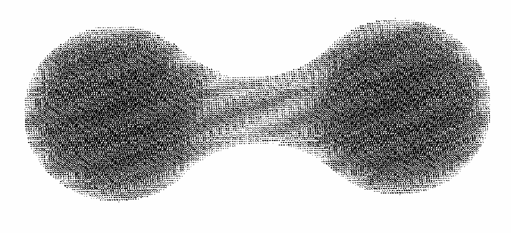


Noisy input

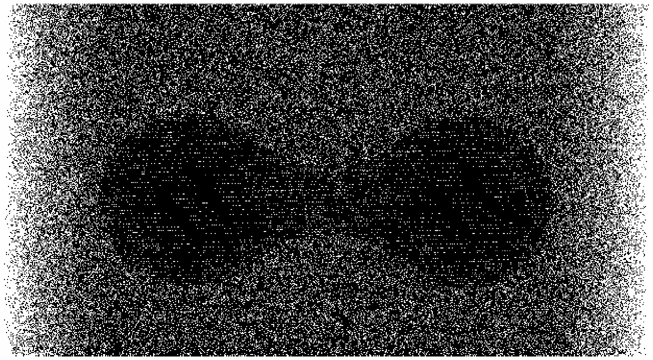


Dense surface boundary (after marching cubes)

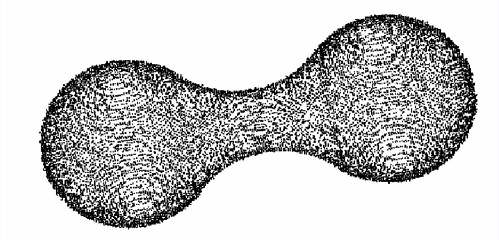
Results in 3-D



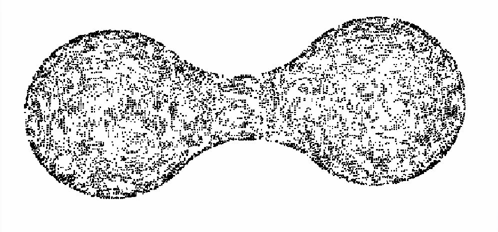
Input (600k unoriented points)



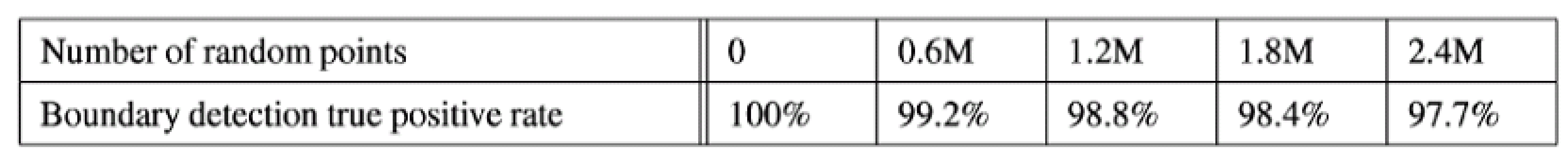
Input with 600k outliers



Output with 1.2M outliers



Output with 2.4M outliers

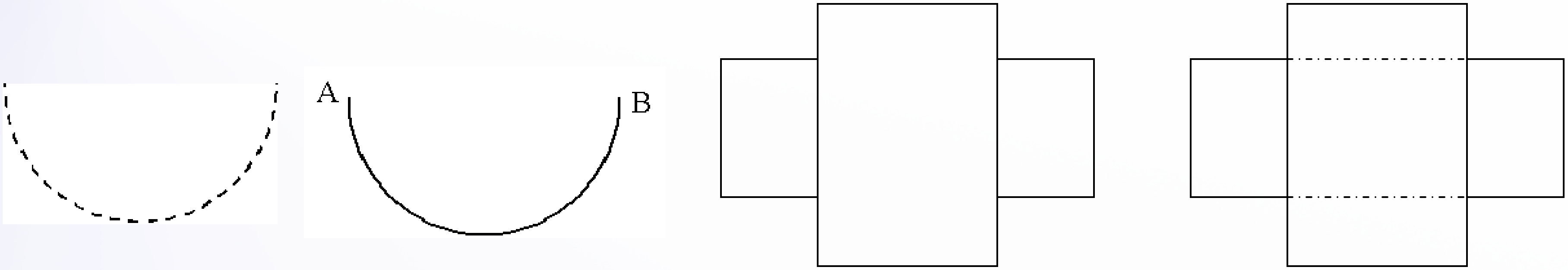


Number of random points	0	0.6M	1.2M	1.8M	2.4M
Boundary detection true positive rate	100%	99.2%	98.8%	98.4%	97.7%

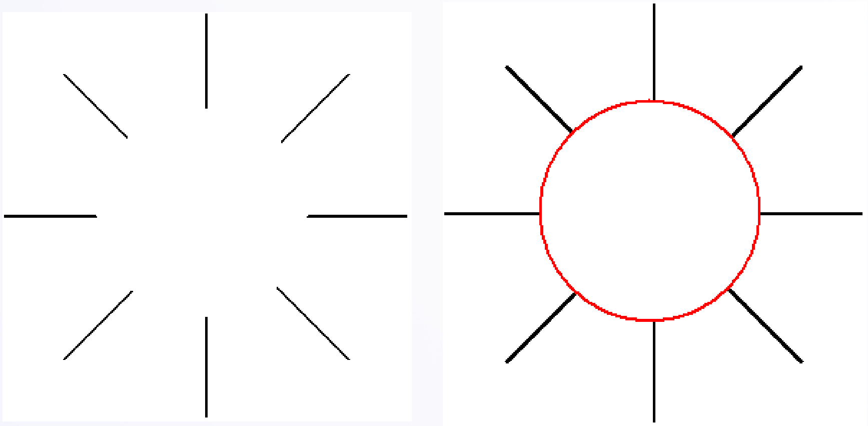
Overview

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

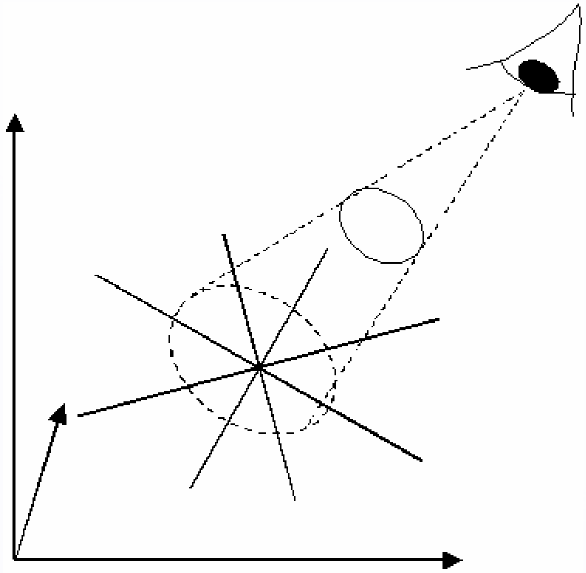
Figure Completion



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

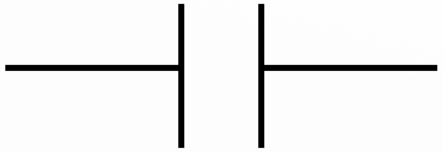
[Mordohai and Medioni, POCV 2004]

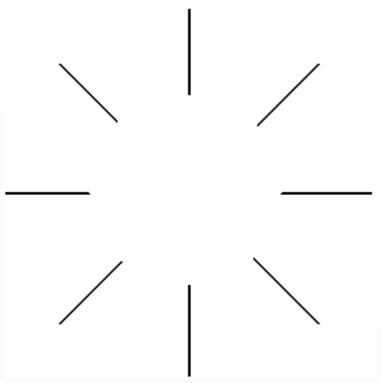
Keypoint Detection

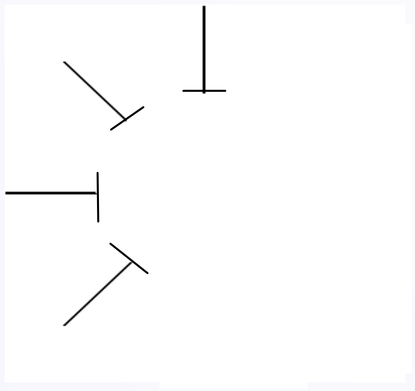
- Input binary images
- Infer junctions, curves, endpoints, regions and boundaries
- Look for completions supported by endpoints, L and T-junctions
- W, X and Y-junctions do not support completion by themselves

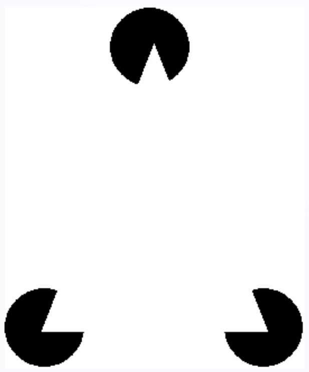
Support for Figure Completion

- 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



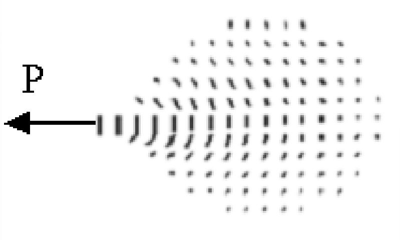


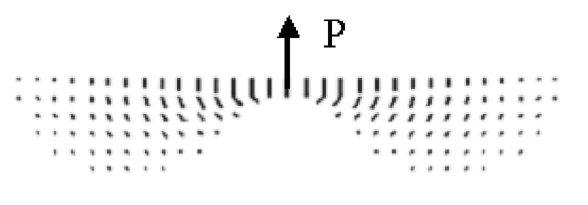


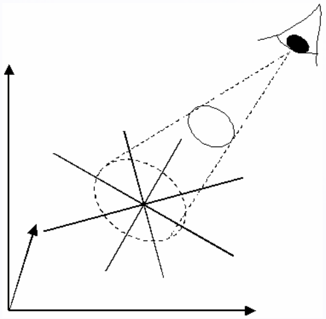


Voting for Completion

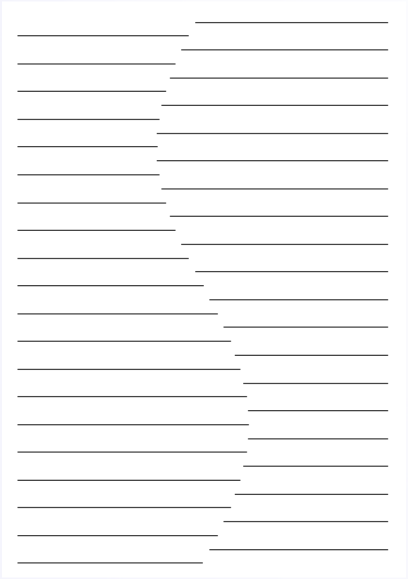
- At least two keypoints of appropriate type needed
- Possible cases:
  - No possible continuation
  - Possible amodal completion (parallel field)
  - Possible modal completion (orthogonal field)
  - If both possibilities available, modal completion is perceived as occluding amodal one



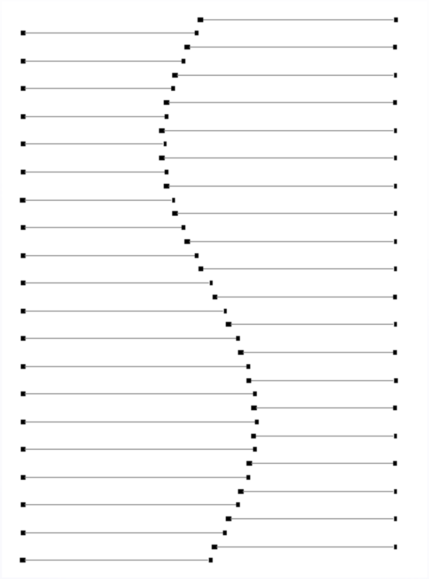




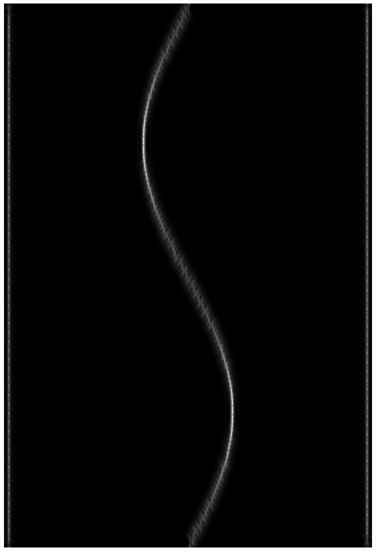
Results: Modal Completion



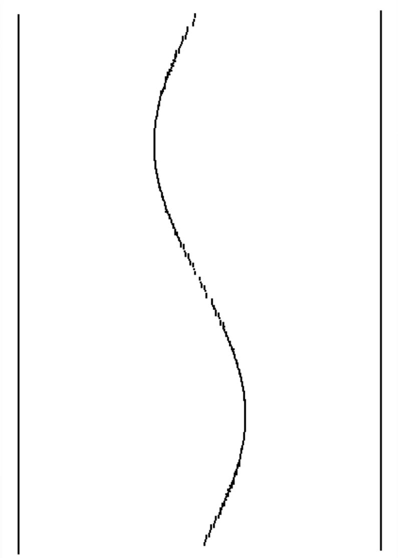
Input



Curves and endpoints

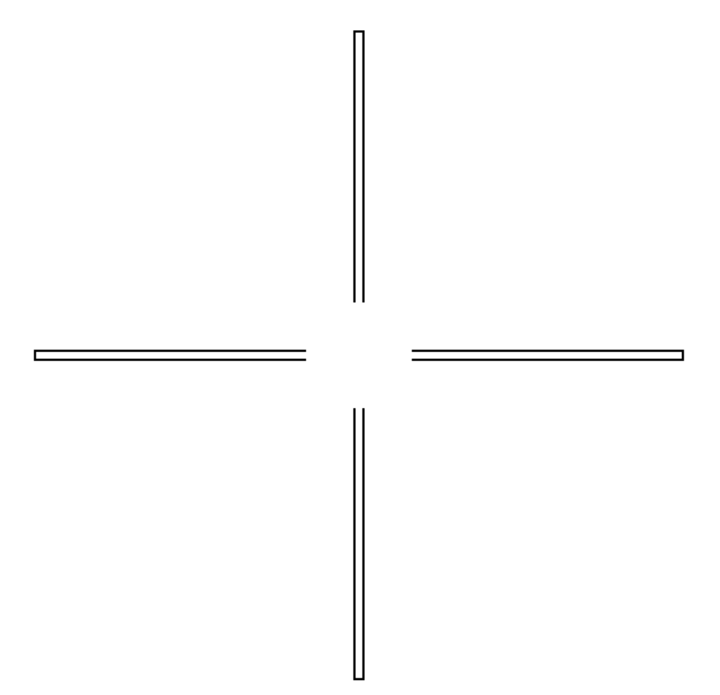


Curve saliency

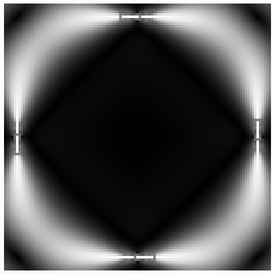


Output

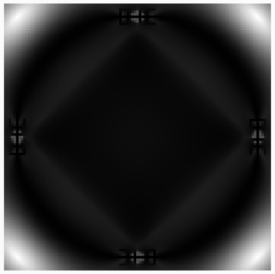
The Koffka Cross



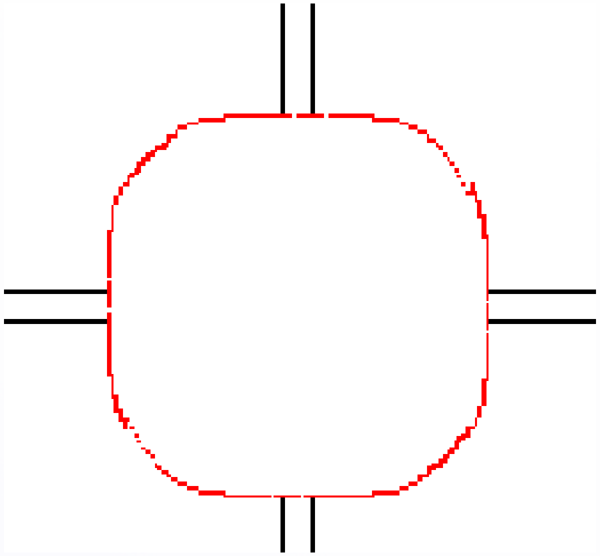
Input



Curve saliency

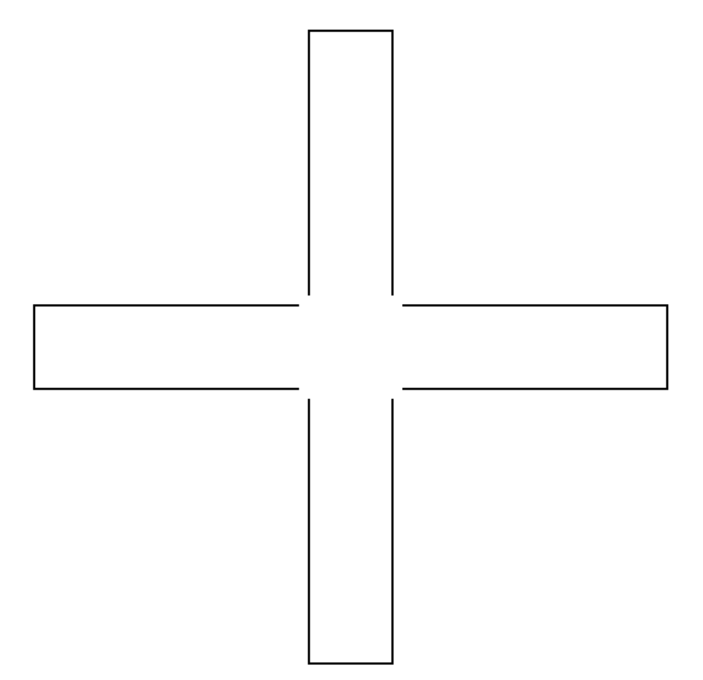


Junction saliency

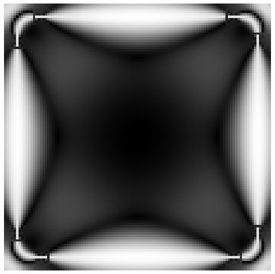


Output

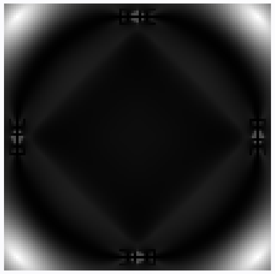
The Koffka Cross



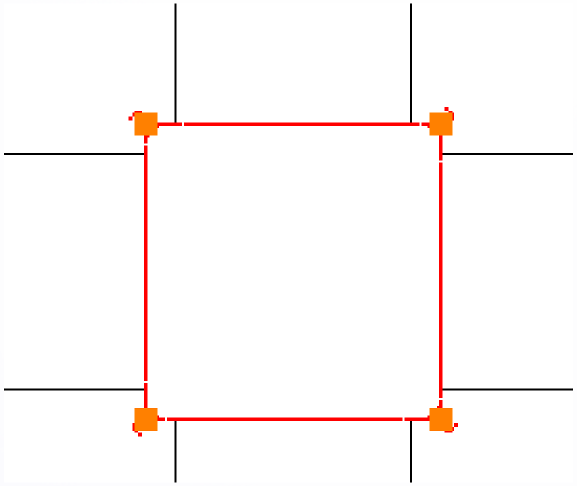
Input



Curve saliency



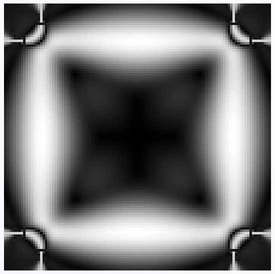
Junction saliency



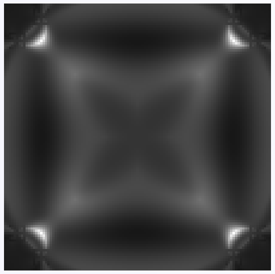
Output

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

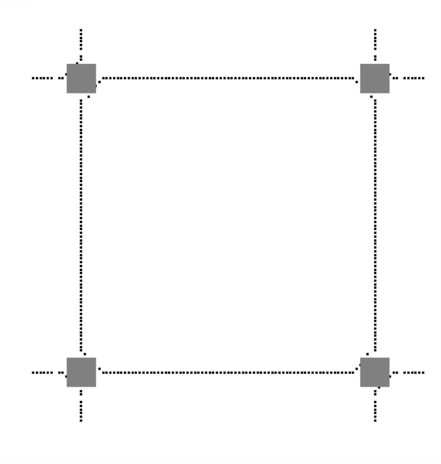
Koffka Cross: Amodal Completion



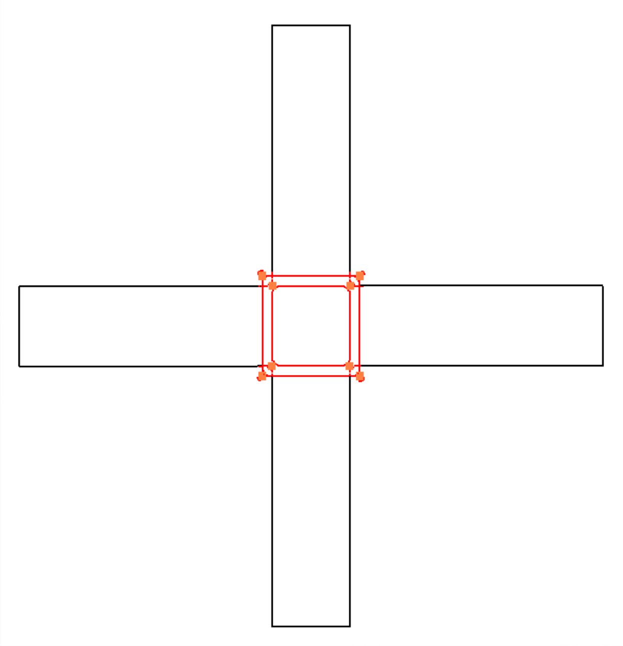
Curve saliency



Junction saliency

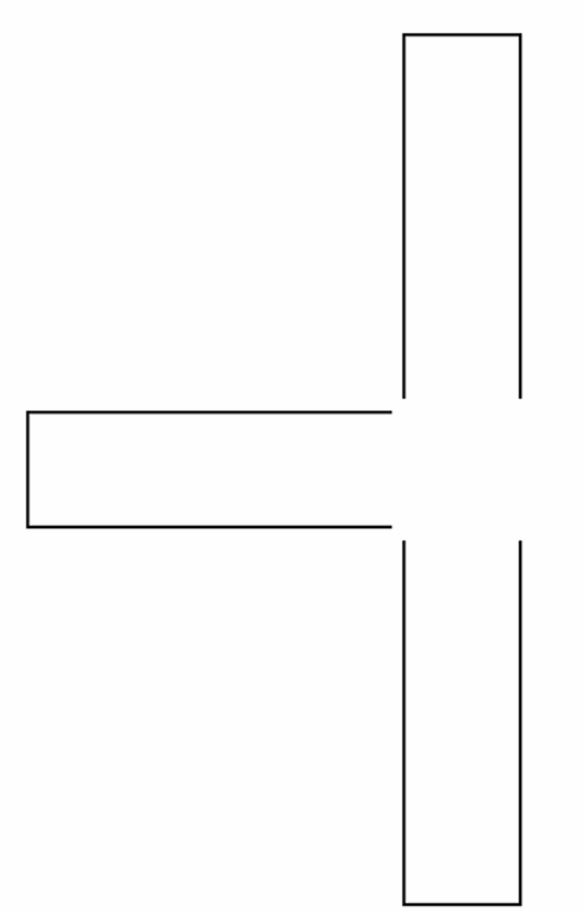


Amodal completion (occluded)

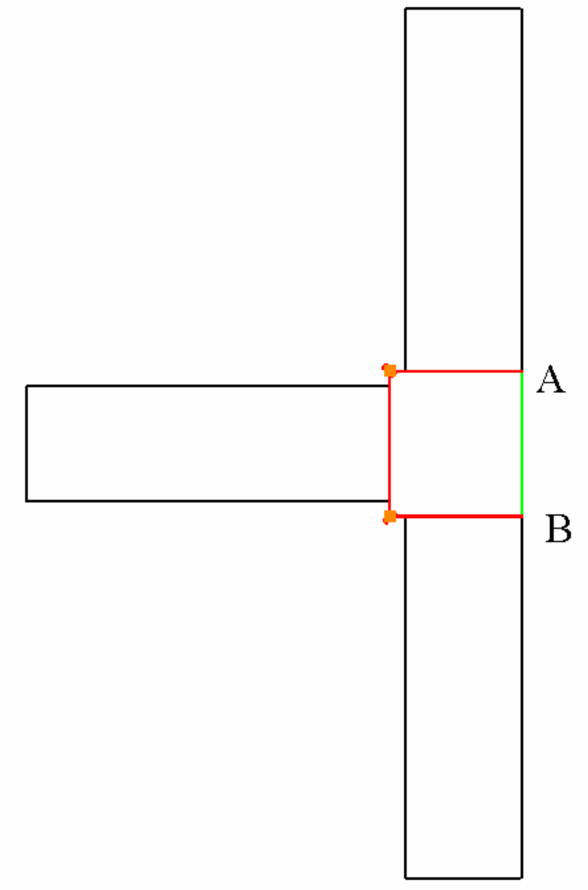


Output

Koffka Cross: Both Types of Completion

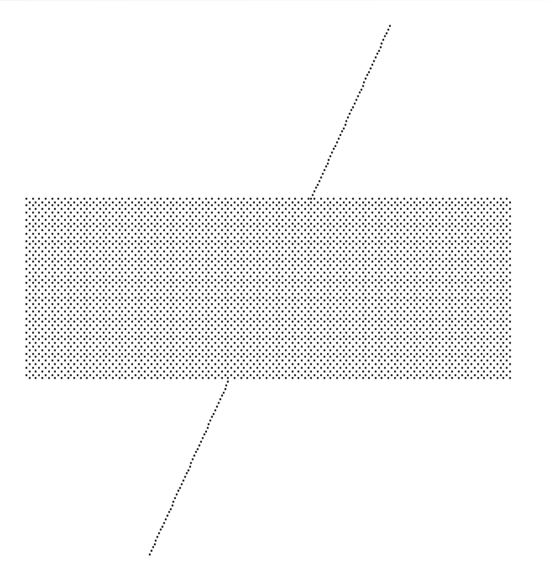


Input

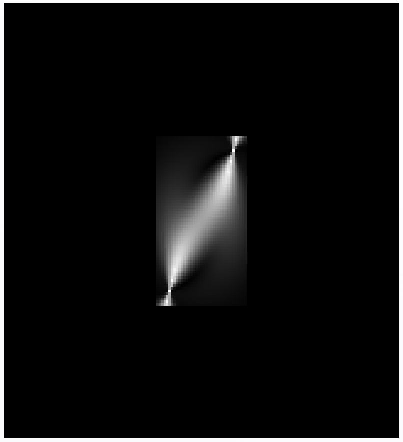


Output

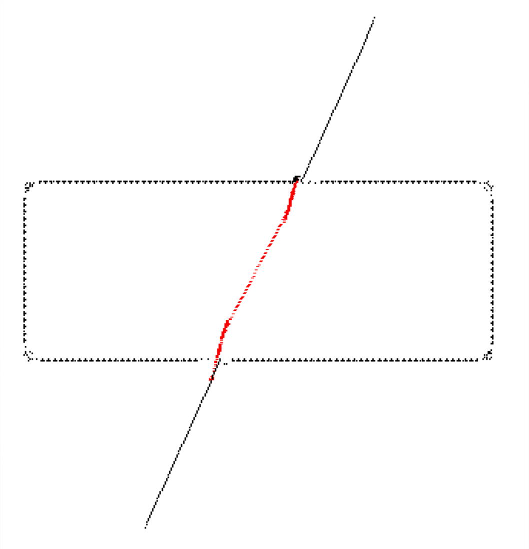
The Poggendorf Illusion



Input

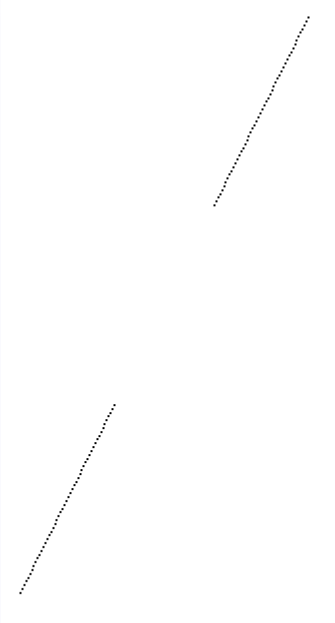


Curve saliency

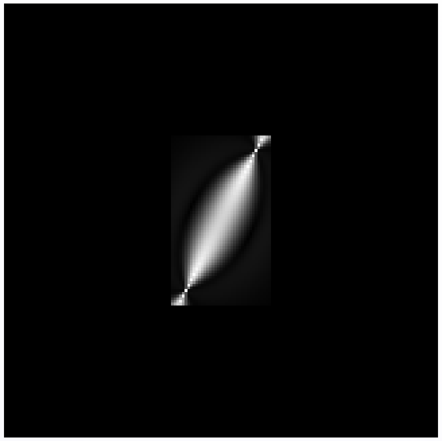


Output

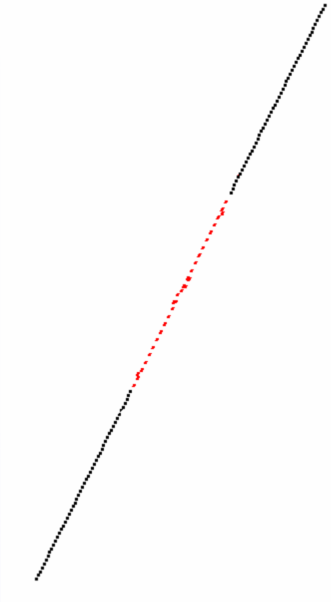
The Poggendorf Illusion



Input



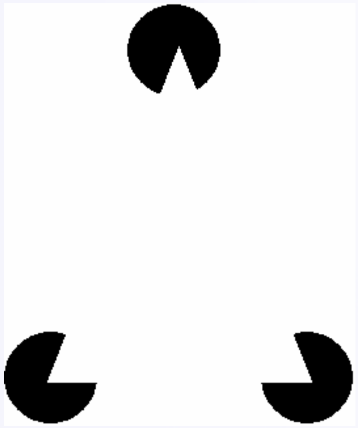
Curve saliency

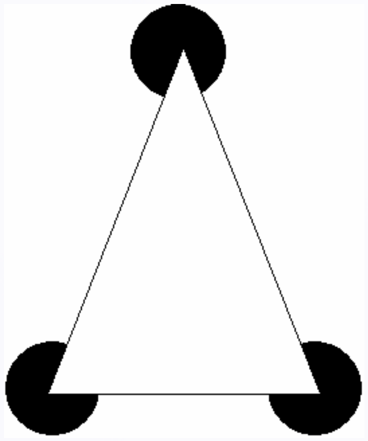


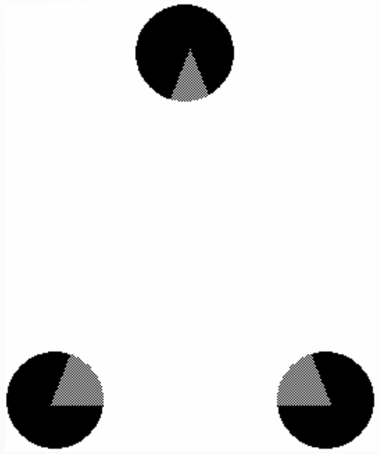
Output

Discussion

- Current approach:
  - Implements modal and amodal completion and automatically selects appropriate type
  - Interprets correctly complex perceptual phenomena
- More work needed on:
  - L-junctions which offer two alternatives
  - Inference of hierarchical descriptions







Overview

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

More Tensor Voting Research

- Curvature estimation
- Visual motion analysis
- Epipolar geometry estimation for non-static scenes
- Texture synthesis

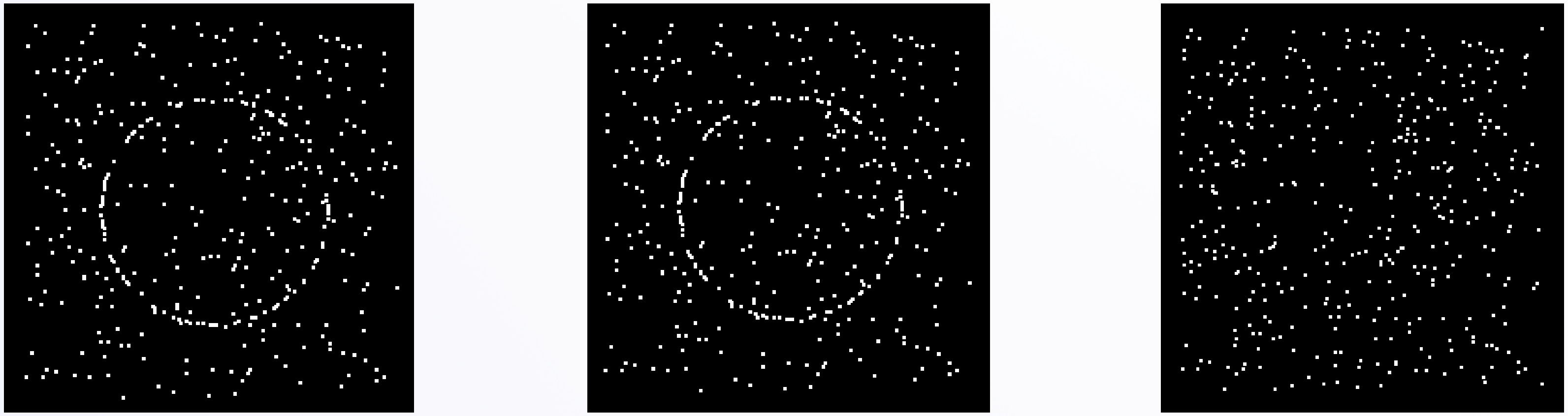
Curvature Estimation

- Challenging for noisy, irregular point cloud
- Three passes
  - Estimate surface normals
  - Compute subvoxel updates for positions of points
  - Compute curvature by collecting votes from 8 directions (45^{o} apart)
- Infer dense surface

[Tong and Tang, PAMI 2005]

Visual Motion Analysis

- Grouping exclusively based on motion cues
  [Nicolescu and Medioni, ICPR 2002]
- Motion-segmentation on real images
  - Accurate object boundaries

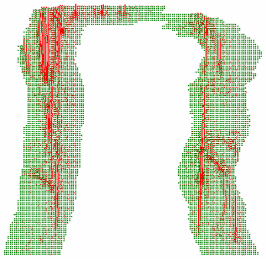


Visual Motion Analysis

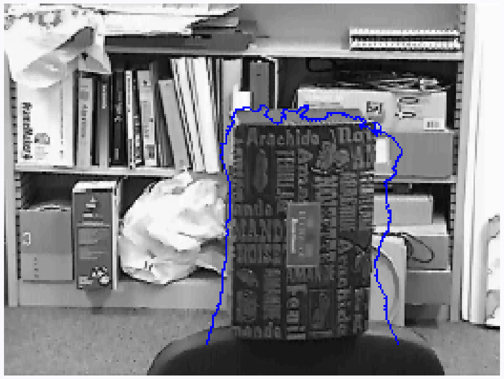
- Potential matches represented by 4-D tensors (x, y, v_x, v_y)
- Desired motion layers have maximum λ_2-λ_3
- Results on challenging, non-rigid datasets

Motion Segmentation

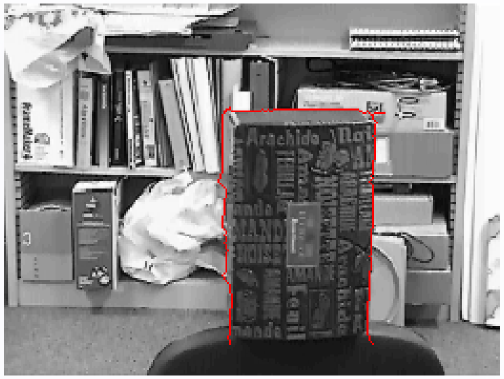
- Group candidate matches in “surfaces” as in stereo
- Boundaries may be inaccurate
  - Use intensity cues (edges) from original images
  - Infer most salient curves in vicinity or object boundaries
    - Parallel to initial boundaries



[Nicolescu and Medioni, PAMI 2005]



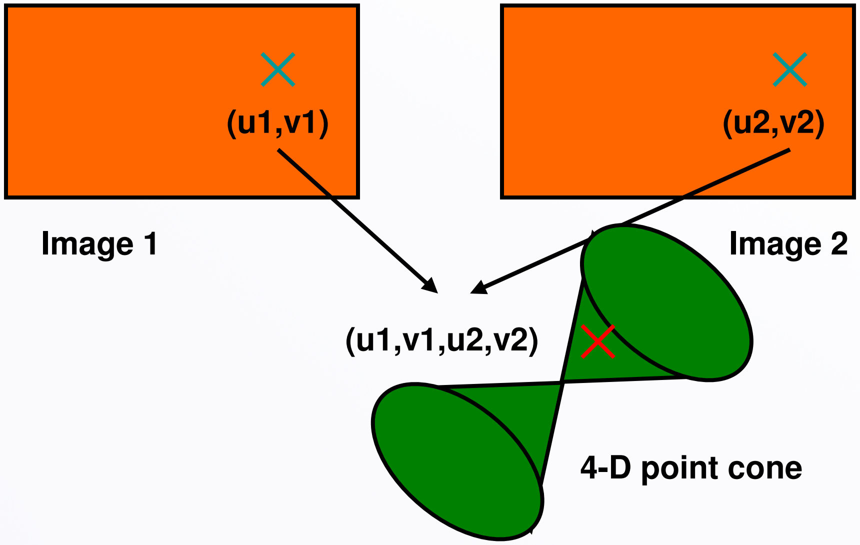
Blue: initial boundaries



Red: refined boundaries

Epipolar Geometry Estimation

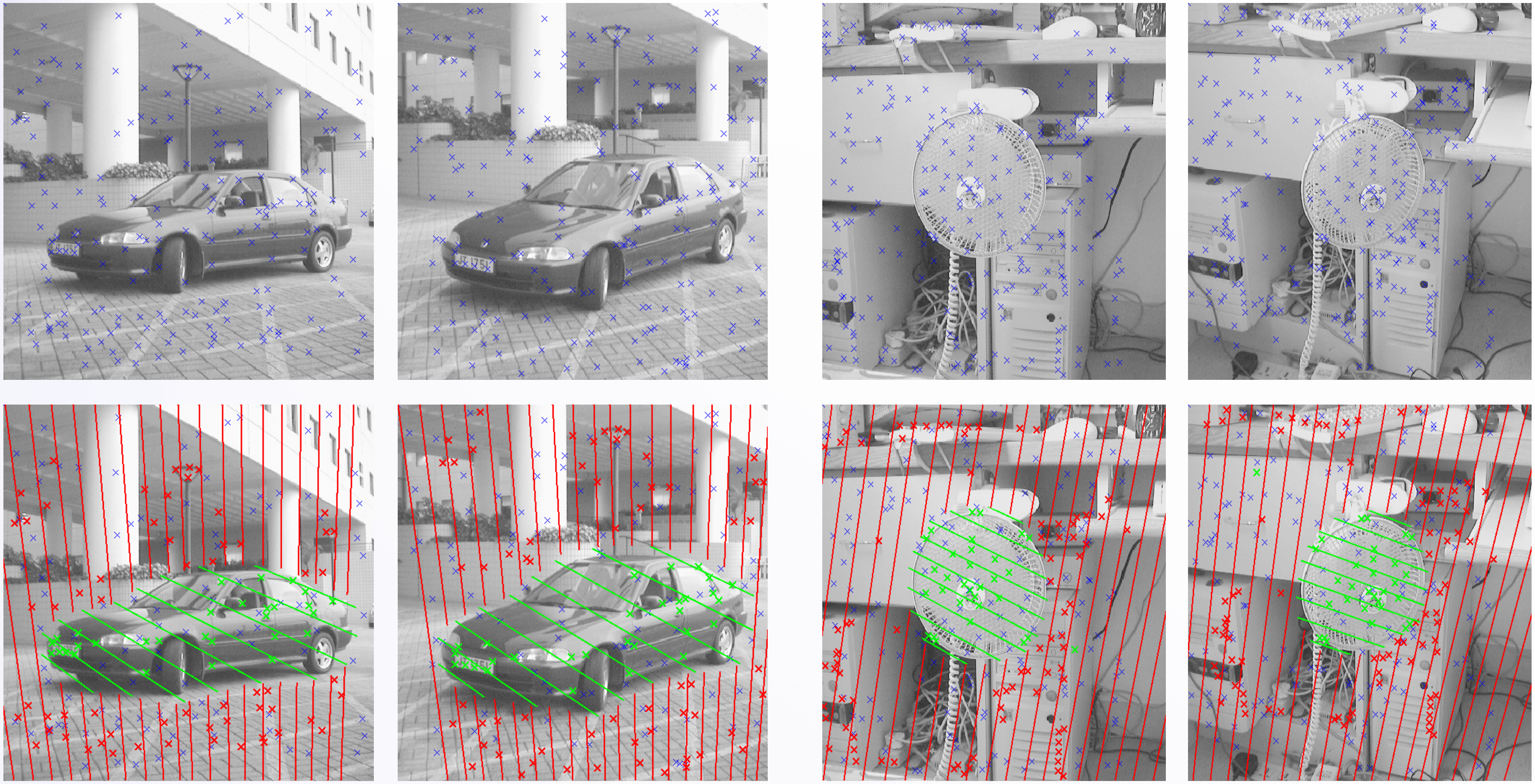
- Epipolar geometry defines a 4-D point cone in the joint image space [Anandan, ECCV 2000]
- Vote in 4-D to detect points on cone or cones
  - Each cone corresponds to an epipolar geometry



Joint image space

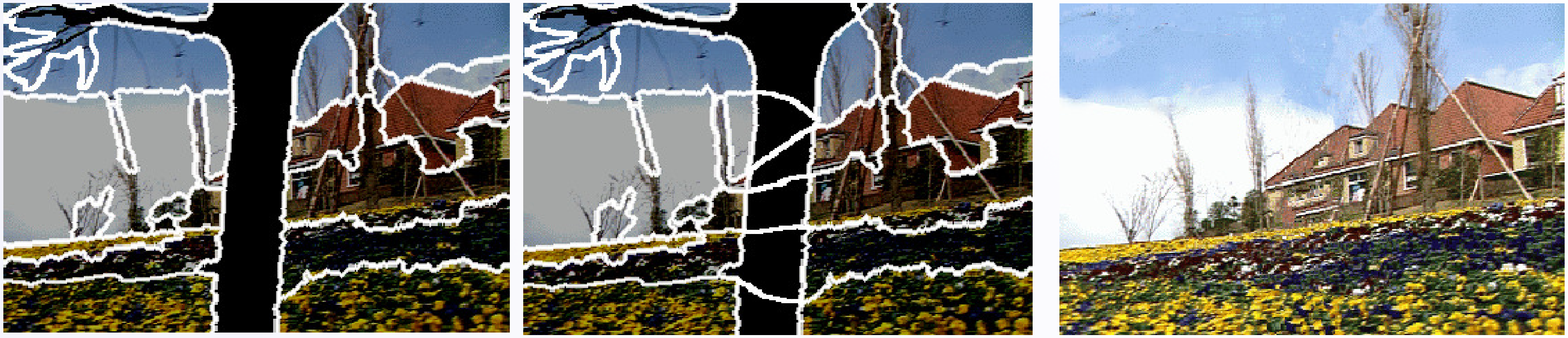
[Tong, Tang and Medioni, PAMI 2004]

Epipolar Geometry Estimation



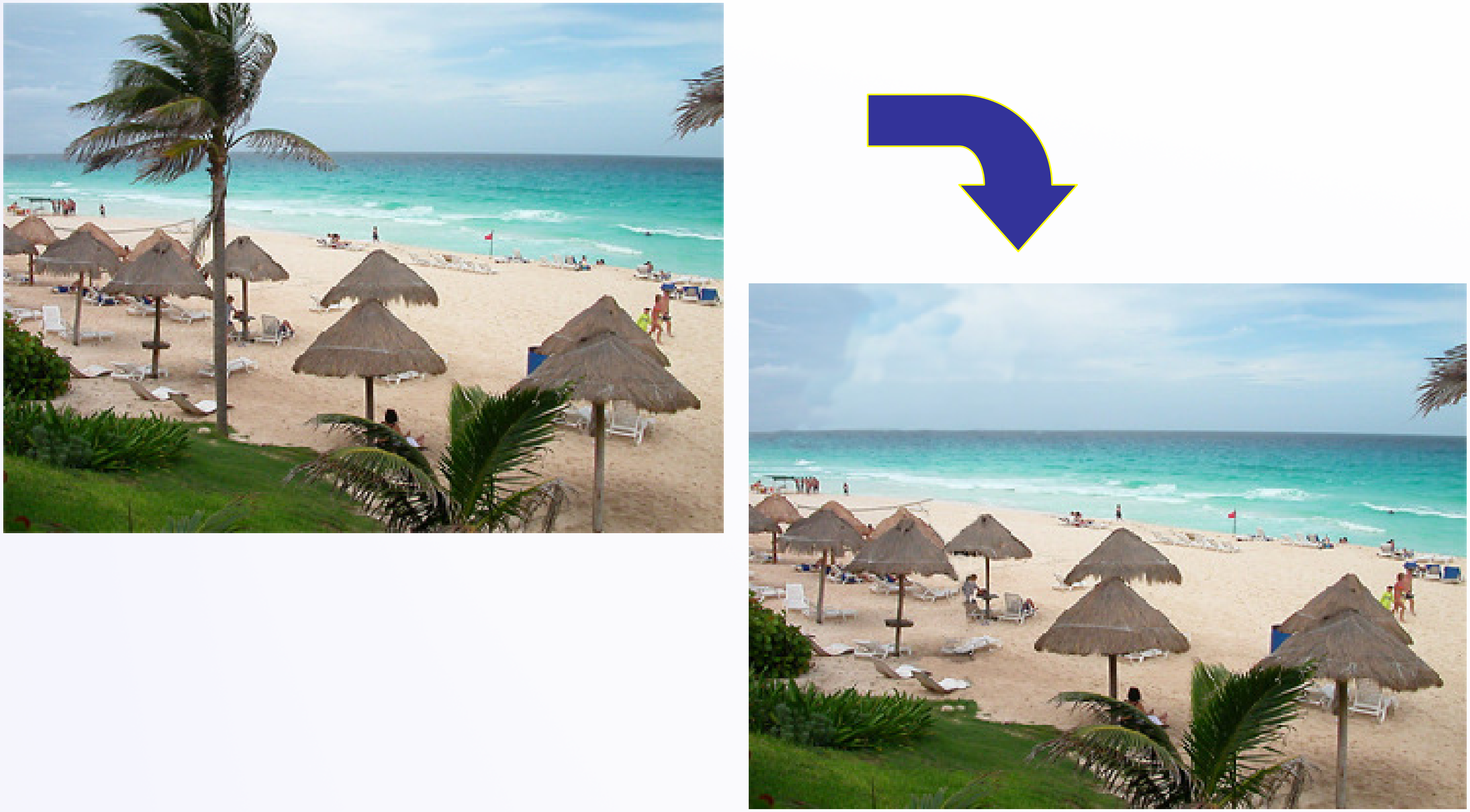
Texture Synthesis

- Given image with user-specified target region
- Segment
- Connect curves across target region
- Synthesize texture via *N*-D tensor voting

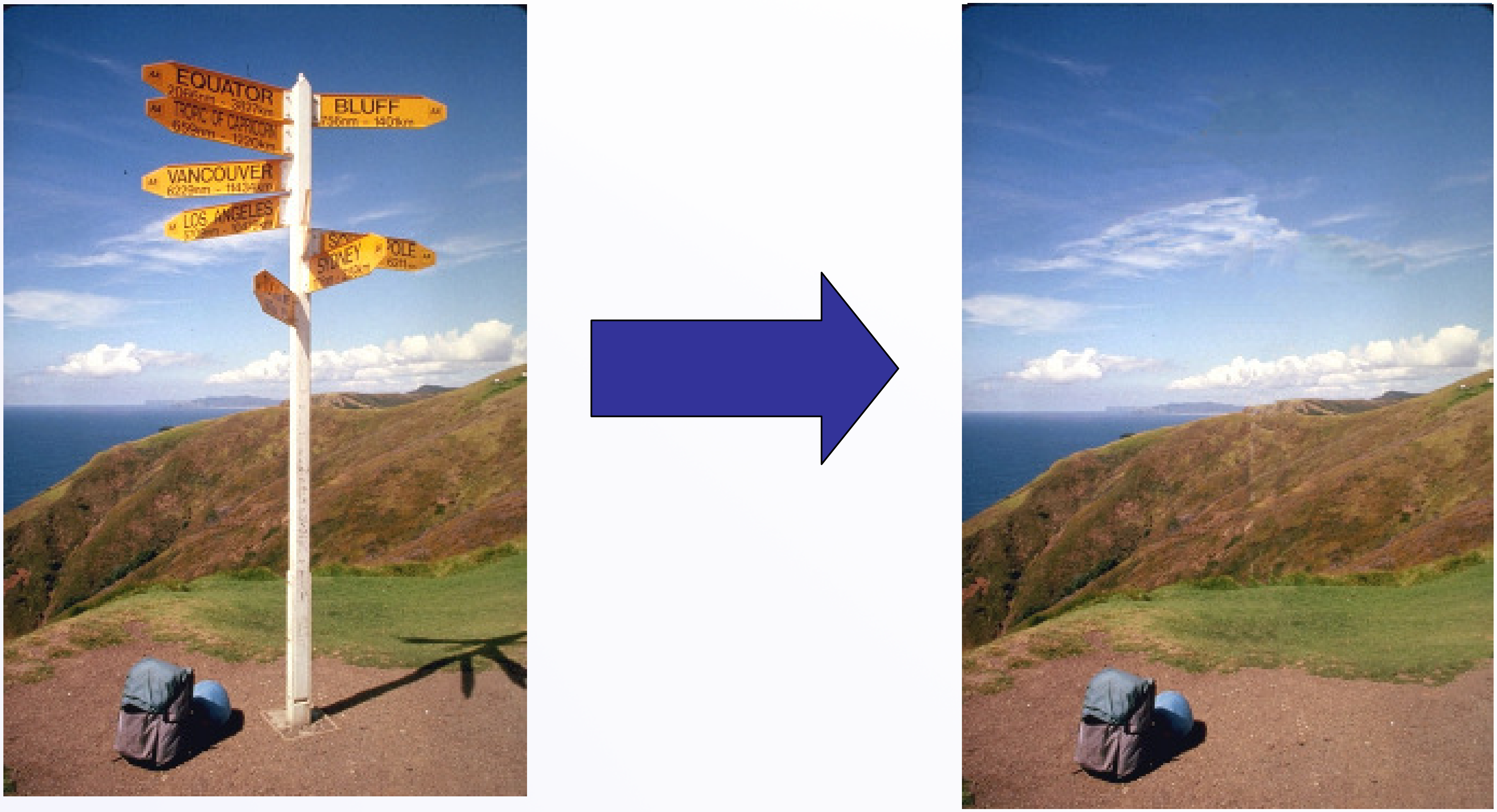


[Jia and Tang, PAMI 2004]

Texture Synthesis Results



Texture Synthesis Results



Overview

- Introduction
- Tensor Voting
- Stereo Reconstruction
- Tensor Voting in *N*-D
- Machine Learning
- Boundary Inference
- Figure Completion
- More Tensor Voting Research
- 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

Future Work

- Integrated feature detection in real images [Förstner94, Köthe03]
  - Integrated inference of all feature types (step and roof edges, all junction types)
- Decision making strategies for interpretation as in [Saund 2003] for binary images and sketches
  - Symmetry, length, parallelism
  - Good continuation vs. maximally turning paths
- Hierarchical descriptions

Future Work

- Rich monocular descriptions in terms of:
  - Shape (invariant as possible)
  - Occlusion, depth ordering, completion
- Applications
  - Reconstruction
  - Recognition

Future Work

- Extend manifold learning work
  - Classification
  - Complex systems such as forward and inverse kinematics
  - Data mining
- Learn from imperfect descriptions
  - Symbolic not signal-based
  - Overcome limitations of image-as-vector-of-pixels representation