The Tensor Voting Framework

Gérard Medioni

University of Southern California

Chi-Keung Tang

Hong Kong University of Science and Technology

Motivation

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

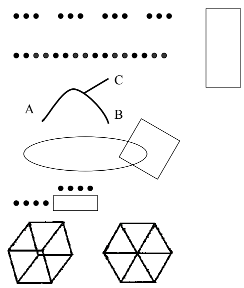
Need for Constraints

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

Perceptual Organization

Gestalt principles:

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



The Smoothness Constraint

Matter is cohesive → Smoothness

Difficult to implement, as true "almost everywhere" only

Overview

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

Regularization

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

➢ Selection of objective function is not trivial
➢ Iterative

Relaxation Labeling

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

➢ Iterative

Robust Methods

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

➢ Can operate only with limited and known a priori models

Level Set Methods

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

➢ Sensitive to initialization
➢ Iterative

Clustering

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

Structural Saliency

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

➢ Saliency defined as scalar

Grouping with Cooperative and Inhibitive Fields

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

Grouping with Cooperative and Inhibitive Fields

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

Overview

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

Tensor Voting in 2-D

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

The Tensor Voting Framework

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

Desirable Properties of the Representation

- Local
- Layered
- Object-centered

Our Approach in a Nutshell

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

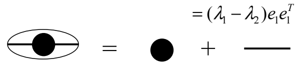
Properties of Tensor Voting

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

Second Order Symmetric Tensors

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

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

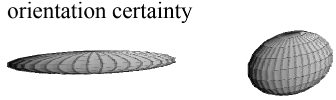


$$\begin{bmatrix} a^2 + b^2 & a^2 \\ a^2 & a^2 \end{bmatrix} = \begin{bmatrix} a^2 & a^2 \\ a^2 & a^2 \end{bmatrix} + \begin{bmatrix} b^2 & 0 \\ 0 & 0 \end{bmatrix}$$

Second Order Symmetric Tensors

Properties captured by second order symmetric Tensor

– shape: orientation certainty

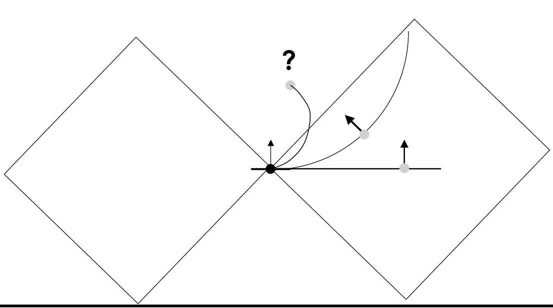


– size: feature saliency

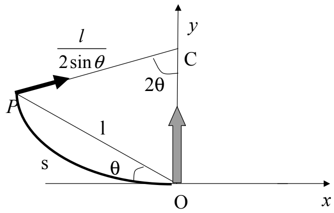
Representation with Second Order Symmetric Tensors

Input	Second Order Tensor	Eigenvalues	Quadratic Form
		λ_1=1 λ_2=0	$\begin{bmatrix} n_x^2 & n_x n_y \\ n_x n_y & n_y^2 \end{bmatrix}$
		λ_1=λ_2=1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Design of the Voting Field



Saliency Decay Function



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

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

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

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

Scale of Voting

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

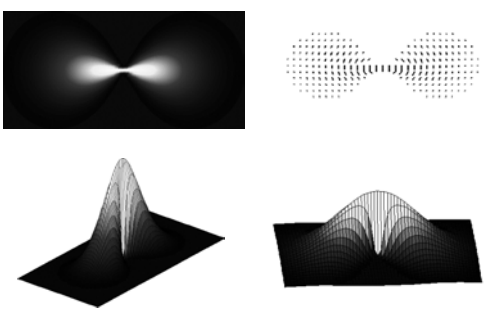
Scale of Voting

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

Scale of Voting

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

Fundamental Stick Voting Field

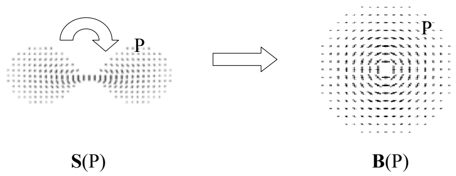


Fundamental Stick Voting Field

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

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

2-D Ball Field

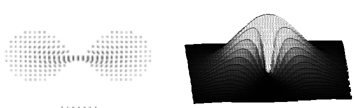


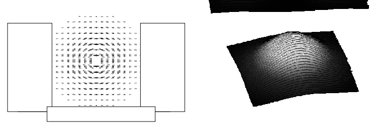
Ball field computed by integrating the contributions of rotating stick

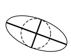
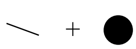
$$\mathbf{B}(P) = \int \mathbf{S}(P) d\theta$$

2-D Voting Fields

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

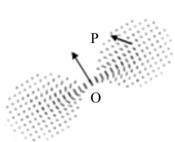
— votes with 

● votes with 

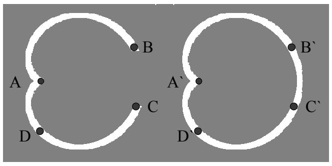
 votes with 

Voting

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



Need for First Order Information



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

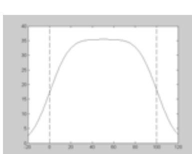
- Second order tensors are insensitive to *signed* orientation
- They cannot discriminate between interior points and terminations of perceptual structures

Polarity Vectors

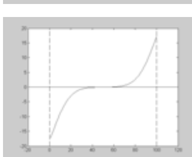
- Representation augmented with Polarity Vectors (first order tensors)
- Sensitive to direction from which votes are received
- Exploit property of boundaries to have all their neighbors on the same side of the half-space

Polarity

Input



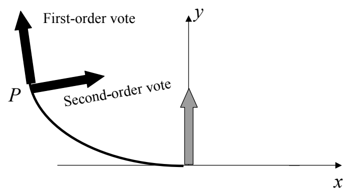
Saliency



Polarity

First Order Voting

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



First Order Voting Fields

- Magnitude is the same as in the second order case

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

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

First and Second Order Voting

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

Vote Collection

Each site **collects** the information cast there

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

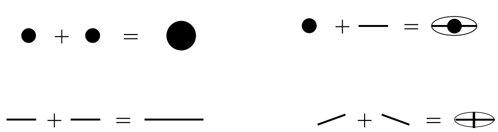
$$V_{SO} = \sum \mathrm{V}_i$$

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

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

Tensor Addition

Each site accumulates second order votes by tensor addition:



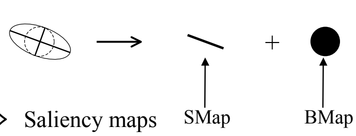
Results of accumulation are usually *generic tensors*

Second Order Vote Interpretation

Salient features correspond to local extrema

At each site

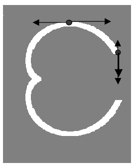
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + \lambda_2 (e_1 e_1^T + e_2 e_2^T)$$



$\Rightarrow$ Saliency maps

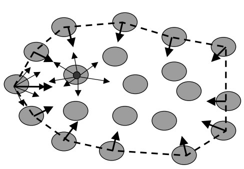
First Order Vote Interpretation (curves)

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



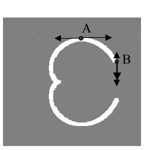
First Order Vote Interpretation (regions)

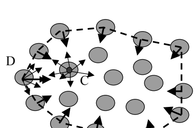
- Tokens near discontinuities accumulate first order votes from consistent direction
- Tokens in the interior of smooth structures receive contradicting votes that cancel out

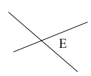


Structure Inference in 2-D

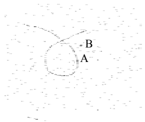
Structure Type	**Saliency**	**Tensor Orientation**	**Polarity**	**Polarity orientation**
Curve inlier **A**	High $\lambda_1 - \lambda_2$	Normal: $\mathbf{e}_1$	Low	-
Curve endpoint **B**	High $\lambda_1 - \lambda_2$	Normal: $\mathbf{e}_1$	High	Normal to $\mathbf{e}_1$
Region inlier **C**	High λ_2	-	Low	-
Region boundary **D**	High λ_2	-	High	Normal to boundary
Junction **E**	Distinct locally max λ_2	-	Low	-
Outlier	Low	-	Indifferent	-







Sensitivity to Scale



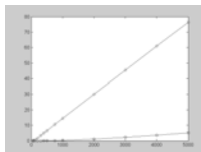
Input

$\sigma = 50$

$\sigma = 500$

$\sigma = 5000$

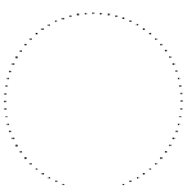
Input: 166 un-oriented inliers, 300 outliers
Dimensions: 960x720
Scale $\in$ [50, 5000]
Voting neighborhood $\in$ [12, 114]



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

Sensitivity of Orientation to Scale

Circle with radius 100 (unoriented tokens)

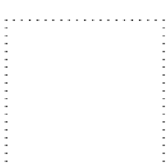


Scale	Average angular error (degrees)
50	1.01453
100	1.14193
200	1.11666
300	1.04043
400	0.974826
500	0.915529
750	0.813692
1000	0.742419
2000	0.611834
3000	0.550823
4000	0.510098
5000	0.480286

As more information is accumulated,
the tokens better approximate circle

Sensitivity of Orientation to Scale

Square 200x200 (unoriented tokens)

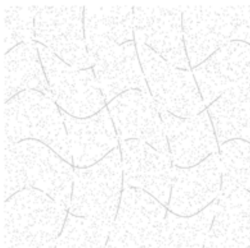


Scale	Average angular error (degrees)
50	1.11601e-007
100	0.138981
200	0.381272
300	0.548581
400	0.646754
500	0.722238
750	0.8893
1000	1.0408
2000	1.75827
3000	2.3231
4000	2.7244
5000	2.98635

Junctions are detected and excluded

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

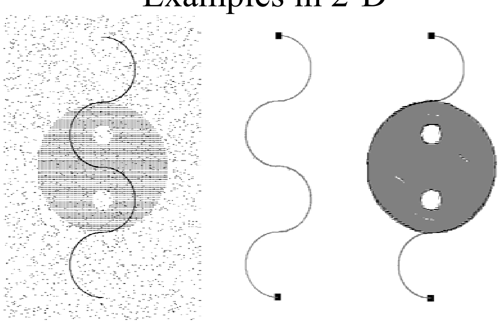
Examples in 2-D



Input

Gray: curve inliers
Black: curve endpoints
Red: junctions

Examples in 2-D



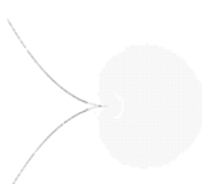
Input

Curves and endpoints only

Curves, endpoints and regions

Examples in 2-D

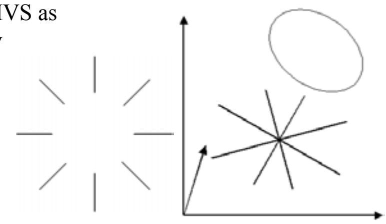
Input



Curve and region inliers

Illusory Contours

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



Illusory Contours in the Tensor Voting Framework

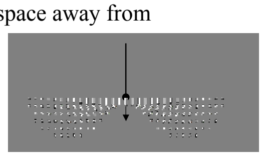
- Endpoint detection
- Used as inputs for illusory contour inference
- Use polarity vector (parallel curve’s tangent) as curve normal

Illusory Contours and Voting Fields

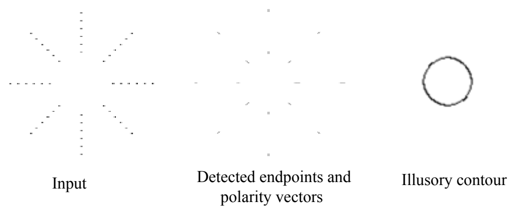
- Since polarity vector (tangent of detected curve segment) serves curve normal

=> fields orthogonal to regular ones

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



Illusory Contour Example



Overview

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

Tensor Voting in 3-D

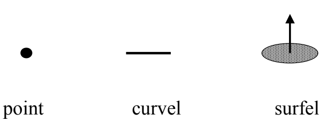
- Representation with tensors
- Tensor voting and voting fields
- First order voting
- Vote analysis and structure inference
- Examples
- Curvature

3-D Tensor Voting

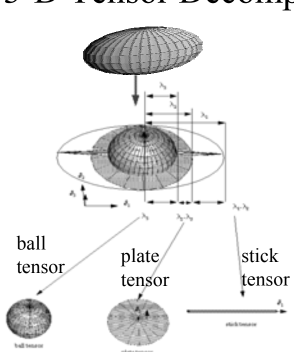
- Representation: **3-D Tensors**
- Constraints: **3-D Voting Fields**
- Data communication: **Voting**

3-D Tensors

The input may consist of



3-D Tensor Decomposition



3 eigenvalues ($\lambda_{max}\lambda_{mid}\lambda_{min}$)

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

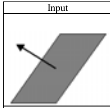
3-D second order Tensors

Equivalent to:

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

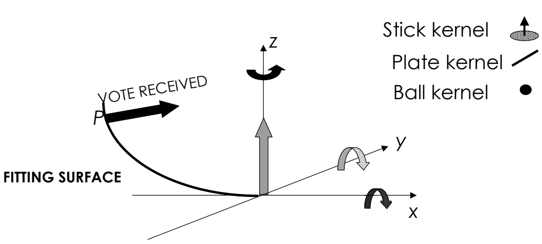
$$T = \lambda_1 \cdot e_1 e_1^T + \lambda_2 \cdot e_2 e_2^T + \lambda_3 \cdot e_3 e_3^T =$$
$$= (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3)(e_1 e_1^T + e_2 e_2^T) + \lambda_3 (e_1 e_1^T + e_2 e_2^T + e_3 e_3^T)$$

Representation

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

3-D Voting Fields

Derived from the Fundamental 2-D Stick Field



Voting Fields in 3-D

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

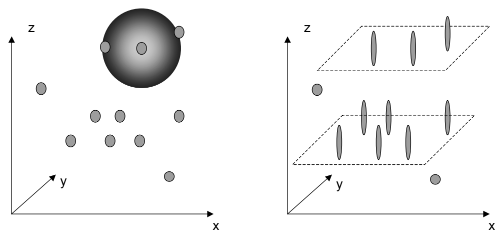
Pre-computed Voting Fields

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

Tensor Voting in 3-D

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

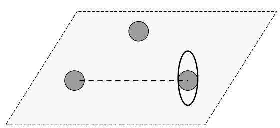
Second Order Voting



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

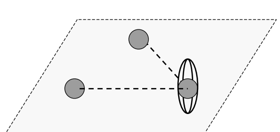
Surface Normal Inference from Unoriented Tokens

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



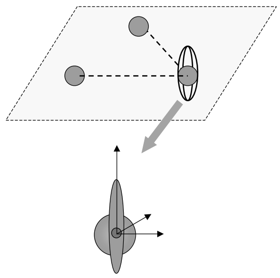
Surface Normal Inference from Unoriented Tokens

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

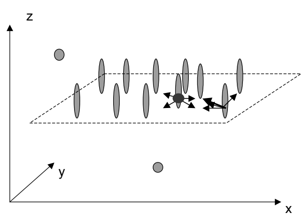


Surface Normal Inference from Unoriented Tokens

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



First Order Voting



- Tokens in the interior of a structure receive first order votes from all directions
- Tokens at boundaries receive first order votes from one side of a half-space

Interpretation of Resulting Tensors

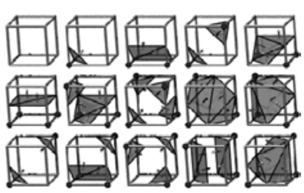
Structure Type	**Saliency**	**Tensor Orientation**	**Polarity**	**Polarity orientation**
Surface inlier	High λ_1- λ_2	Normal: $\mathbf{e_1}$	Low	-
Surface boundary	High λ_1- λ_2	Normal: $\mathbf{e_1}$	High	Normal to $\mathbf{e_1}$ and boundary
Curve inlier	High λ_2- λ_3	Tangent: $\mathbf{e_3}$	Low	-
Curve endpoint	High λ_2- λ_3	Tangent: $\mathbf{e_3}$	High	Parallel to $\mathbf{e_3}$
Volume inlier	High λ_3	-	Low	-
Volume boundary	High λ_3	-	High	Normal to bounding surface
Junction	Distinct locally max λ_3	-	Low	-
Outlier	Low	-	Indifferent	-

Structure Inference in 3-D

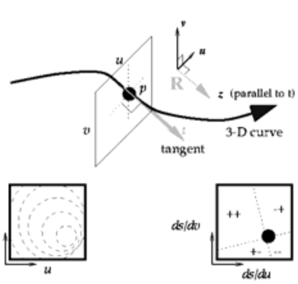
- Surfaces and curves extracted as local maxima of surface and curve saliency
- Perform **Dense Vote**, where votes are collected at all locations
- Detect *zero-crossings* of first derivative of saliency
- Extract surfaces using *Marching Cubes*
- Extract curves similarly

Surface Extraction

Surface Patch — Surface Saliency along normal direction — First derivative of Surface Saliency

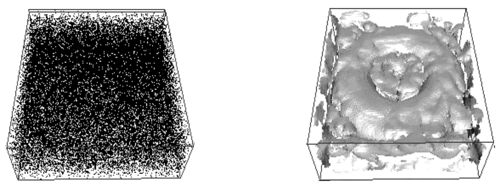


Curve Extraction

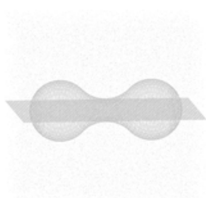


Graceful Degradation with Noise

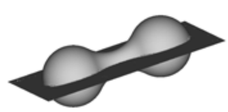
[redacted]



Examples in 3-D



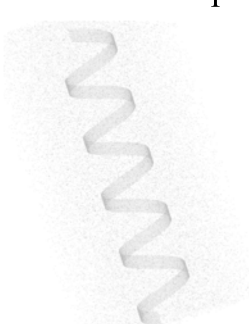
Input



Surfaces

Surface Intersections

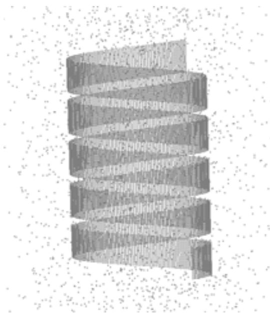
Examples in 3-D



Input

Surface Boundaries

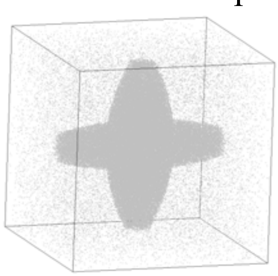
Examples in 3-D



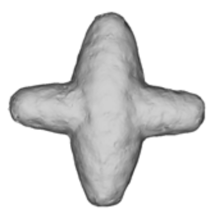
Input

Surface Boundaries

Examples in 3-D



Input



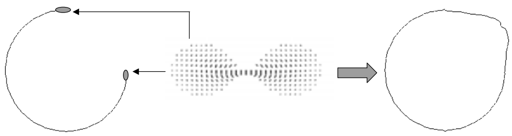
Volume Boundaries

Curvature

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

Why Curvature?

voting with regular field



a circle will not be produced

Our approach on Curvature Estimation

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

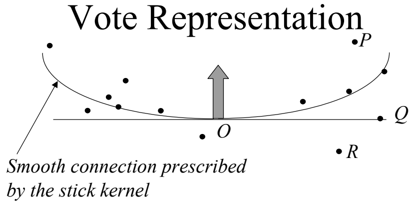
Two Estimations

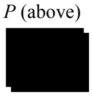
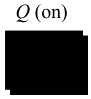
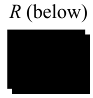
- Sign of principal curvature
- Principal direction

Sign of Principal Curvature

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

Vote Representation



	P (above)	*Q* (on)	*R* (below)
vote **direction**			
vote **strength**	inversely proportional to distance from *O*		

Vote collection at *O*

- compute **mean** μ
  - preferred side

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

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

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

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

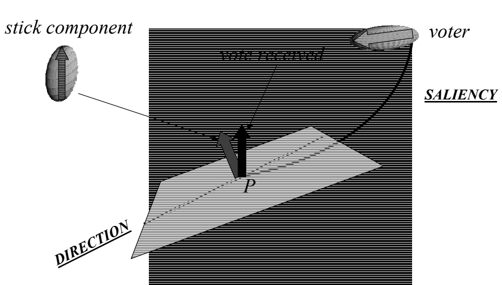
Geometric Interpretation

$\|\mu\| \approx 0\,?$	$\Sigma \approx 0\,?$	
✓	✓	**planar**
✗	✓	**elliptic**
✓	✗	**hyperbolic**
✗	✗	**parabolic**

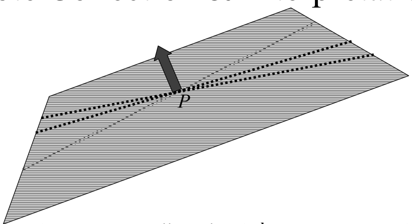
Two Estimations

- Sign of principal curvature
- Principal direction

Principal Direction



Vote Collection & Interpretation



2D votes are collected as 2^{nd} **oreder symmetric tensors**

V_{max} = **maximum** direction

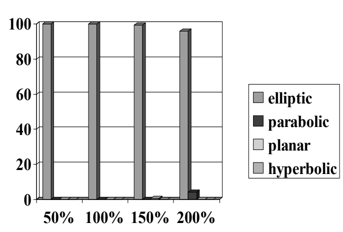
V_{min} = **minimum** direction

Curvature-Based Stick Kernel

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

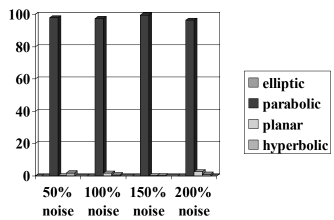
Accuracy of Labeling

Sphere (489 points)



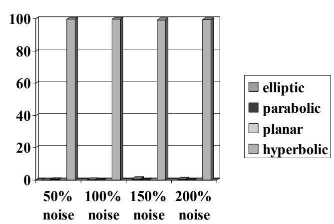
Accuracy of Labeling

Cylinder (3844 points)

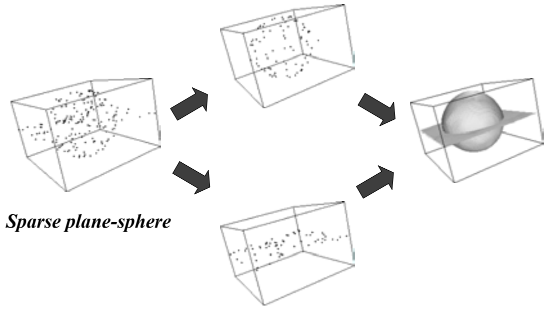


Accuracy of Labeling

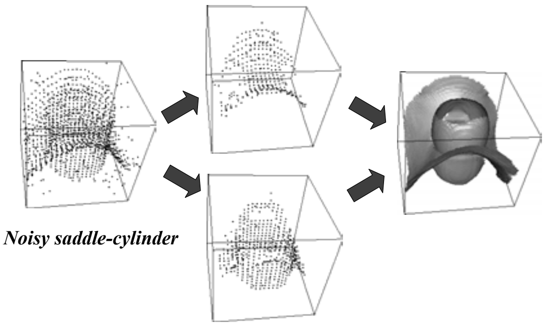
Saddle (605 points)



Grouping by Curvature

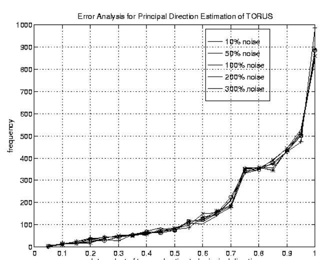


Grouping by Curvature



Noisy saddle-cylinder

Robustness of Principal Curvature Estimation



Estimated principal direction does not adversely affected by noise

Overview

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

Tensor Voting in N-D

Direct generalization from 2-D and 3-D cases

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

Voting Fields in N-D

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

Applications in N-D

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

Applications in N-D

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

Issues in N-D

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

Overview

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

Real Computer Vision Problems

- Vision problems can be posed as perceptual organization
  - Eg smooth surfaces in stereo, smooth motion layers in motion analysis
- Need means to generate tokens in each case

Token Generation

- So far, perceptual organization of tokens
- In real problems tokens represent image primitives:
  - Intensity
  - Color
  - Contrast
  - Disparity
  - Optical flow

Token Generation

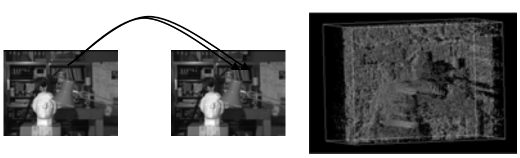
- So far, perceptual organization of tokens
- In real problems tokens represent image primitives:
  - Intensity
  - Color
  - Contrast
  - Disparity
  - Optical flow

Token Generation from Pixel Correspondences

- For stereo and motion estimation
- Tokens initialized in appropriate space if potential pixel correspondence is detected
  - 3-D space for Stereo (x, y, d)
  - 4-D space for Visual Motion (x, y, v_x, v_y)
- Usually initializes as ball tensors (no prior information) of unit size (matching score discarded)

Initial Matching

- Normalized cross-correlation
  - Use multiple square windows
  - Retain all peaks as potential matches
- Delay decisions until saliency is available

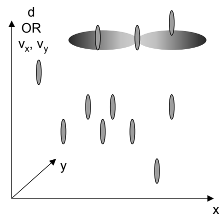


Initial Matching

- Goal is detection of as many of the correct matches as possible
  - Tensor Voting can survive large false positive rate
- Can incorporate other matching methods since matching score is discarded (not done so far)
  - Interval matching
  - Rank transforms
- Increase saliency of candidate matches confirmed by multiple windows or methods

Tensor Voting

- Tokens initialized at locations of initial matches
  - As *balls* since no prior information is available
- Cast first and second order votes to neighbors



Uniqueness

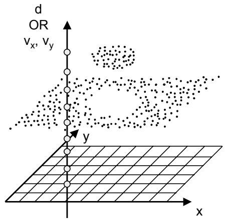
- Tokens are classified according to *saliency* and *polarity*
- Most salient token along each Line of Sight is retained
  - Disambiguation of initial matches
- Outliers rejected based on low saliency
- Other constraints can also be added here

Estimates at every Position

- Disparity or velocity estimates required at every position
  - But salient matches might not exist for some pixels
- One alternative is dense structure extraction
  - Computationally expensive
  - Not generalizable beyond 3-D
- Instead compute discrete estimates for missing disparities or velocities

Discrete Densification

- At each pixel (x y) generate discrete d or (v_x v_y) candidates
- Collect votes at each candidate
- Use surface saliency as affinity measure
- Choose most salient candidate



Overview

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

Stereo

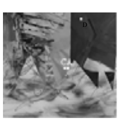
- Binocular stereo
- Multiple view stereo

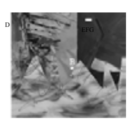
Stereo as Perceptual Organization

- Smooth surfaces in the scene appear as smooth surfaces in 3-D disparity space
- Group neighboring points in 3-D with compatible normals
- Overcomes limitations of 1-D or 2-D neighborhoods
- Delay matching decisions until saliency information is available

Challenges

- Occluded pixels
  - Sometimes can generate higher matching scores than correct correspondences
- Textureless pixels
  - Ambiguous matching





A: occluded pixel, has better matching score to B than correct match C
D: textureless pixel, can be matched to E, F or G

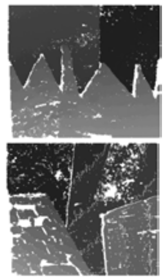
Input Data: Middlebury Stereo Evaluation Dataset

Grayscale versions used in all experiments

Results after Sparse Voting

- Initial matching using 3x3 and 5x5 windows
- Uniqueness wrt to left image
- Candidate matches with low saliency removed
- Results comparable to R. Sara (ECCV 2002)
  - Higher map coverage
  - Higher error rates

Dataset	Map density	Error Rate
Tsukuba	50.8%	1.94%
Sawtooth	94.8%	1.51%
Venus	87.5%	1.23%
Map	93.8%	0.55%



Sawtooth and Venus after sparse voting (white: no salient match)

Discrete Densification

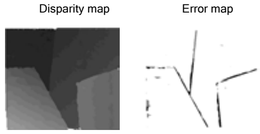
For each pixel without disparity estimate:

- Find disparity range from neighboring pixels
  - extend by a few disparity levels
- Generate candidate tokens for each potential disparity
- Collect votes at candidates
- Select most salient

Discrete Densification Resutls

- Textureless pixels have been treated using smoothness of surfaces, since intensity is ambiguous
- Most errors in occluded pixels
  - Black: errors >1 disparity level
  - Gray: errors between 0.5 and 1 disparity level

Disparity map Error map



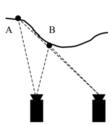
Disparity map Error map

Use of Monocular Information: Depth Discontinuities

- Correct errors due to occlusion
  - These occur near depth discontinuities
- Intensity edges may appear on the image at depth discontinuities
  - Since adjacent pixels at depth discontinuities may come from different scene surfaces

Use of Monocular Information: Depth Discontinuities

- Intensity edges do not always occur at depth discontinuities in both images
- Detect edges corresponding to *left occluded regions* (visible in left image) in the right image and vice versa



Occluded region visible in:	Left image	Right image
Edge better localized in:	Right image	Left image

Uncertainty Zones

- Use depth discontinuities of dense disparity map to define “uncertainty zones” on both images
- Transition from high to low disparity defines an uncertainty zone in the left image
- Transition from low to high disparity defines an uncertainty zone in the right image
- Width proportional to disparity jump and matching window size

Edge Detection

- Compute right disparity map and mark zones on both images
- Compute gradient responses within marked zones
- Multiply them by prior: $Pr(x, y) = e^{-\frac{(x-x_o)^2}{\sigma_p^2}}$
  - Where x_o is initial discontinuity

Edge detection

- Perform 2-D Tensor Voting
- Extract edges starting from seeds
- Update disparities and project right edges to left image
- Discontinuities along y-axis are processed in left image

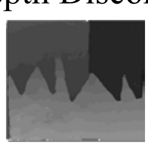
Left Image

Right Image

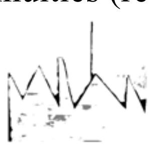
White: uncertainty zones

Black: no disparity estimate

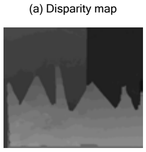
Use of Monocular Information: Depth Discontinuities (results)



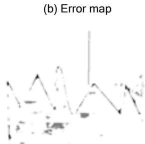
(a) Disparity map



(b) Error map

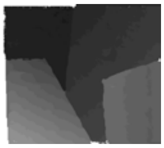


(c) Final disparity map

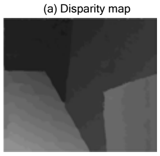


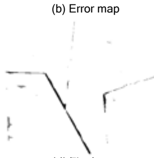
(d) Final error map

Venus

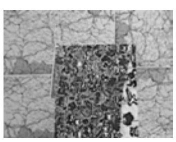

(a) Disparity map


(b) Error map

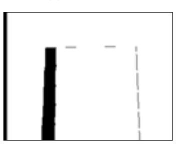

(c) Final disparity map


(d) Final error map

Map


(a) Left image

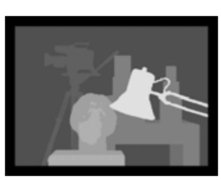

(b) Disparity map with ground truth discontinuities

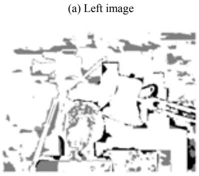

(c) True occlusion map

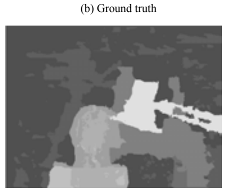

(d) Final disparity map

Tsukuba


(a) Left image


(b) Ground truth


(c) Error Map


(d) Disparity Map

Improvement after Discontinuity Detection

	Before discontinuity detection		After discontinuity detection	
	Un-occluded	All	Un-occluded	All
Tsukuba	4.97%	6.03%	4.68%	5.62%
Sawtooth	1.96%	4.00%	0.98%	2.09%
Venus	1.39%	2.18%	1.12%	1.38%
Map	1.08%	1.61%	1.09%	1.38%

Quantitative Evaluation

Results on graylevel images from the Middlebury Stereo Vision page evaluation
Rank among both graylevel and color images (for un-occluded pixels only)

	Error Rates		Rank in Middlebury evaluation
	Un-occluded	All	
Tsukuba	4.68%	5.62%	15
Sawtooth	0.98%	2.09%	5
Venus	1.12%	1.38%	4
Map	1.09%	1.38%	15

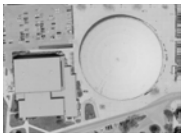
Sensitivity to Scale

- Errors in Sawtooth after sparse vote
  - Better evaluation than after full algorithm
- Error rate and coverage are insensitive to scale
  - Classification of specific pixels as inliers varies

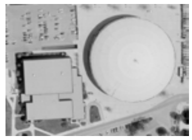
Scale	Error Rate
10	1.32%
20	1.27%
50	1.09%
100	0.97%
200	0.92%
500	0.93%
1000	1.06%
2000	1.10%

Arena

(non-parametric surfaces)



Left image



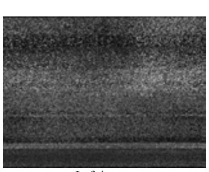
Right image

Views of reconstructed model in projective space

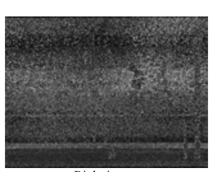
Random Texture

- Intensities randomly re-arranged in left image
- Ground truth disparities used to create right image
- Now, every pixel is textured

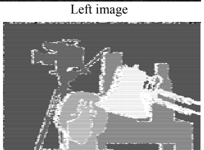
Random Texture



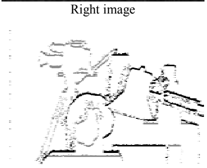
Left image



Right image



Results of sparse vote



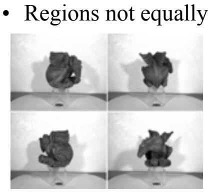
Error map

Random Texture

- Coverage after sparse vote: 92.4%
  - 50.4% on regular images
- Error rate: 3.9%
  - 4.3% after discontinuity correction on regular images
- Failures in initial matching in regular images cause most of the errors

Multiple View Stereo:
Input and challenges

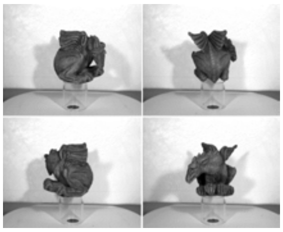
- Multiple images of complex scenes
- 360º views
- No features visible in all views
- Non-planar surfaces
- Regions not equally rich in texture



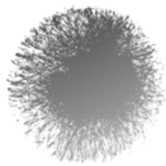
Approach

- Process all inputs simultaneously
  - Instead of merging binocular pairs
- Look for coherent surfaces
  - As opposed to individual pixel color consistency
- The Tensor Voting Framework fits the needs of the problem
  - Processing of $\sim 10^6$ tokens
  - Object-centered representation
  - Multiple overlapping layers

Dragon Input

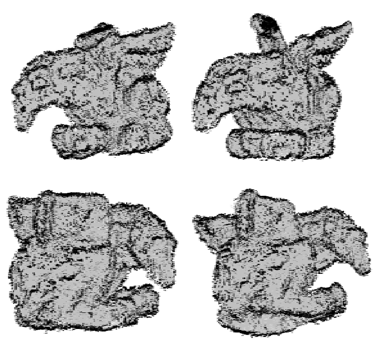


Four of the 36 input images



Initial matches (viewed from above)

Dragon Output



Gray: surface inliers
Black: discontinuities

Lighthouse Input

Four of the 36 input images

Initial matches

Lighthouse Output

Black: surface inliers
Red: discontinuities

Overview

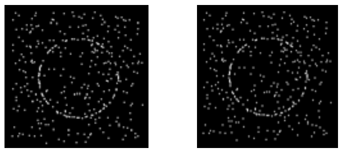
- Related Work
- Tensor Voting in 2-D
- Tensor Voting in 3-D
- Tensor Voting in N-D
- Application to Vision Problems
- Stereo
- Visual Motion
- Binary-Space-Partitioned Images
- 3-D Surface Extraction from Medical Data
- Epipolar Geometry Estimation for Non-static Scenes
- Image Repairing
- Range and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

Visual Motion Analysis

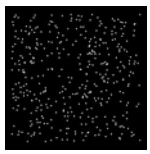
- From motion cues only
- From real images

Monocular vs. Motion Cues

Structure inference possible from one image only…?



…or from motion only ?

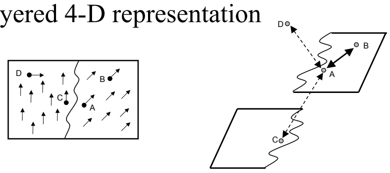


Computational Processes

- Matching
  - Establish token correspondences across images
  - Recover a (possibly sparse and noisy) velocity field
- Motion capture
  - Obtain a dense representation :
  - Dense velocity field
  - Boundaries
  - Regions

4-D Voting Approach

- Layered 4-D representation



- Match: $(x\ y) \rightarrow (x+v_x\ y+v_y)$
- Represent each candidate match as a $(x\ y\ v_x\ v_y)$ point in 4-D
- Motion layers $\leftrightarrow$ smooth surfaces in the 4-D space

Second Order Tensors in 4-D

Elementary tensors

Feature	λ_1	λ_2	λ_3	λ_4	e_1	e_2	e_3	e_4	Tensor
Point	1	1	1	1	Any orthonormal basis				Ball
Curve	1	1	1	0	n_1	n_2	n_3	t	C-Plate
Surface	1	1	0	0	n_1	n_2	t_1	t_2	S-Plate
Volume	1	0	0	0	n	t_1	t_2	t_3	Stick

A generic tensor

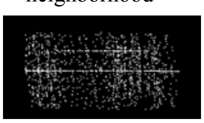
Feature	Saliency	Normals	Tangents
Point	λ_4	none	none
Curve	$\lambda_3 - \lambda_4$	e_1 e_2 e_3	e_4
Surface	$\lambda_2 - \lambda_3$	e_1 e_2	e_3 e_4
Volume	$\lambda_1 - \lambda_2$	e_1	e_2 e_3 e_4

Generating Candidate Matches

Input images:

- sparse identical point tokens
- motion cues only

- Establish a potential match with all tokens in a neighborhood

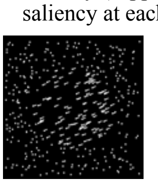


Candidate matches:

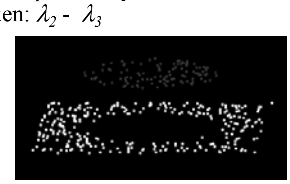
- $(x\ y\ v_x\ v_y)$ points in 4-D

Selection

- Wrong matches appear as outliers, receiving little or no support
- Affinity (support) is expressed by the surface saliency at each token: $\lambda_2 - \lambda_3$

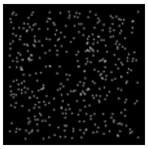


Sparse velocity field



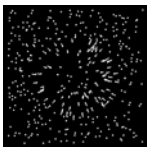
Recovered v_x velocities

Expanding Disk

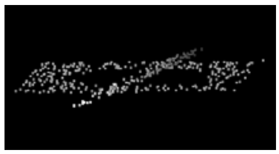


Input

Non-rigid motion

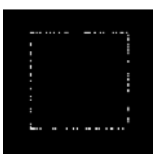


Sparse velocity field



3-D view of recovered v_x velocities

Rotating Square



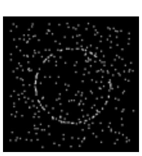
Input

Non-smooth curve

Sparse velocity field

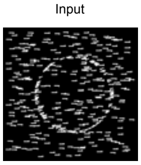
3-D view of recovered v_x velocities

Translating Circle



Input

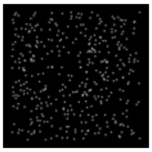
Handling both curves and surfaces



Sparse velocity field

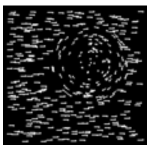
3-D view of recovered v_x velocities

Rotating Disk-Translating Background



Input

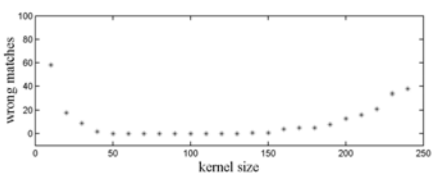
No separation even in 4-D



Sparse velocity field

3-D view of recovered v_x velocities

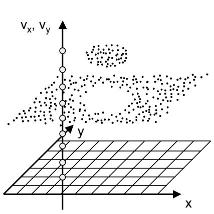
Scale Sensitivity

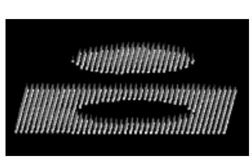


- Tested on the translating disk example
- Number of input points = 400
- Image size = 200 x 200

Densification

- At each pixel (*x y*):
  - generate discrete (v_x v_y) candidates
  - at each candidate $\rightarrow$ collect votes from the input tokens
  - use surface saliency (λ_2 - λ_3) as an affinity measure
  - choose most salient candidate





Dense velocity field and layer orientations

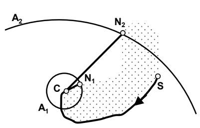
Region Grouping

- Propagate region labels
- Criterion → smoothness of both:
  - pixel velocities ↔ distance in the $(v_x\ v_y)$ space
  - layer orientations ↔ normal vectors e_1 and e_2

Regions

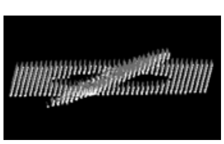
Boundary Extraction

- 2-D process, that extracts a "locally convex" hull
- Irregularity – function of the scale factor
- At large scale → convex hull



Motion boundaries

Expanding Disk

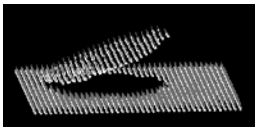


Dense velocity field

Regions

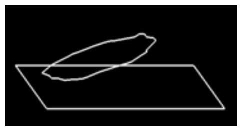
Boundaries

Rotating Disk-Translating Background



Dense velocity field

Regions

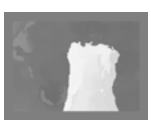


Boundaries

Incorporating Intensity Information

- Why not use monocular cues first?

- Augment motion with monocular cues:



Approach

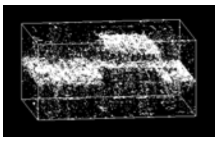
- The general framework: } remains the same
  - 4-D layered representation
  - Token affinity communication through voting
- Issues:
  - Generation of initial candidate matches
  - Accurate boundary inference, in the presence of occlusion

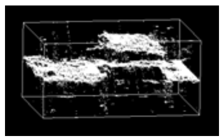
Generating Candidate Matches

- Use an intensity-based, cross-correlation procedure
- All peaks of correlation are retained as candidates
- Repeat for multiple scale values (correlation window sizes)

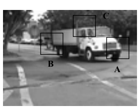
Small scale → fine detail, effective next to boundaries, noisy

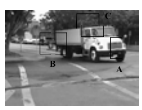
Large scale → smoother, more affected by occlusion, less noisy





Uncertainty at Motion Boundaries



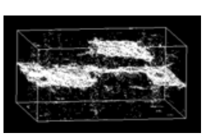


- Correlation is inherently unreliable at motion boundaries
- Non-similarity between regions
- Wrong matches may be actually consistent with the correct ones → cannot be rejected as noise

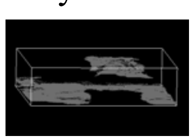
- Formulate motion analysis as a two-component process:
  - Enforce smoothness of motion, except at its discontinuities
  - Enforce smoothness of such discontinuities, aided by monocular cues

Extraction of Motion Layers

Input



Candidate matches



Dense layers

Layer boundaries

– Layers can still be over or under-extended, mainly due to occlusion

– Approach → incorporate intensity cues (edges) from original images

Boundary Saliency Map

- Define a boundary saliency map in the uncertainty zones along layer boundaries
- Encode 2-D stick tensors:
  - Orientation ← gradient orientation

$$G_x(x,y) = I(x,y) - I(x-1,y)$$
$$G_y(x,y) = I(x,y) - I(x,y-1)$$

  - Saliency ← gradient magnitude

$$sal = W \cdot \sqrt{G_x^2 + G_y^2}$$

$$W = e^{-\frac{(x-x_c)^2}{\sigma_W^2}}$$


Boundary saliency map

Detecting the Boundary

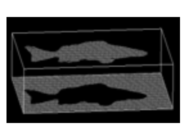
- Enforce smoothness of motion discontinuities → 2-D voting process within zones of boundary uncertainty
- After voting, grow boundary in the uncertainty zones, according to maximal curve saliency, given by (λ_1 - λ_2)

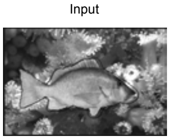

Refined boundaries

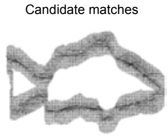
Fish Sequence (synthetic)

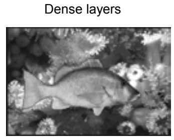

Input


Candidate matches


Dense layers

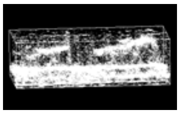

Layer boundaries


Boundary saliency map

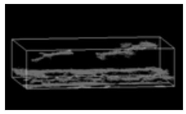

Refined boundaries

Barrier Sequence

Input



Candidate matches



Dense layers

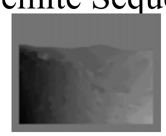
Layer boundaries

Boundary saliency map

Refined boundaries

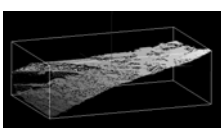
Yosemite Sequence

Input



x-velocities

y-velocities



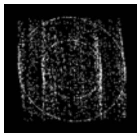
Motion layer (x-velocities)

Yosemite Sequence

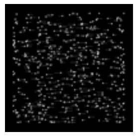
Technique	Average error	Standard deviation	Coverage
Nicolescu and Medioni	3.74°	4.3°	100%
Anandan	15.54°	13.46°	100%
Uras et al. (unthresholded)	16.45°	21.02°	100%
Horn and Schunck	22.58°	19.73°	100%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	3.55°	7.11°	8.8%
Uras et al. ($det(H) \geq 2.0$)	3.75°	3.44°	6.1%
Fleet and Jepson ($\tau = 2.5$)	4.29°	11.24°	34.1%
Fleet and Jepson ($\tau = 1.25$)	4.95°	12.39°	30.6%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	5.20°	9.45°	35.1%
Uras et al. ($det(H) \geq 1.0$)	5.97°	11.74°	23.4%
Heeger	11.74°	19.0°	44.8%

Cylinders Sequence

Input

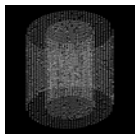


Candidate matches



Velocities

Dense layers



3-D structure

Overview

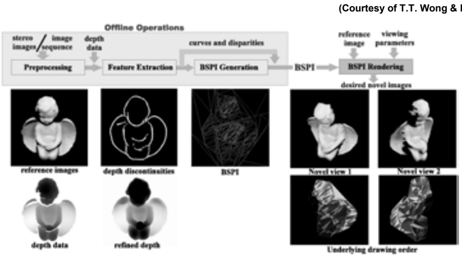
- Related Work
- Tensor Voting in 2-D
- Tensor Voting in 3-D
- Tensor Voting in N-D
- Application to Vision Problems
- Stereo
- Visual Motion
- Binary-Space-Partitioned Images
- 3-D Surface Extraction from Medical Data
- Epipolar Geometry Estimation for Non-static Scenes
- Image Repairing
- Range and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

Overview of System

Application: BSPI Rendering system

BSPI (Binary-Space-Partitioned Images)

(Courtesy of T.T. Wong & Philip Fu)



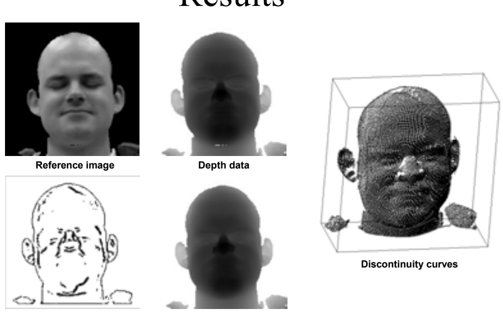
Criteria for partitioning

- Occlusion boundaries
- Depth and orientation discontinuity curves
- Give strong parallax effect when viewpoint changes
- Only depth data is considered
- Not reference image

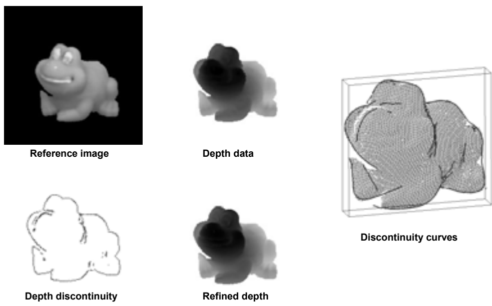
Feature Curves Extraction for BSPI

- Steps
  1. Refinement of the noisy depth map
  2. Estimate sign of curvature (simplified version)
     - since normal direction is estimated in step 1
     - sign of curvature will just have 2 choices
  3. Vote for endcurves
  4. Feature points extraction by simple thresholding of saliency

Feature Curves Extraction Results



Feature Curves Extraction Results



Overview

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

Motivation

- Surface models extraction is important
  - visualization
  - surgery planning
  - medical analysis
- Two major issues
  - segmentation with relevant tokens
  - surface fitting

Motivation

- Most methods require a knowledge model
  - easily break down to slightly abnormal case
- Propose to use tensor voting approach
  - enforcement of continuity constraint only
  - extract detail features with scale detection algorithm (CVPR’01)

Related Work

- Marching cubes algorithm
  - efficient, simple
  - good result only with accurate and dense data
  - Lorensen & Cline (Computer Graphics’87)
- Deformable model approach
  - initialization required
  - iterative
  - prior knowledge is often required
  - Kass, Witkin and Terzopoulous (IJCV’87)

Related Work

- Model or atlas based approach
  - a sub-class of deformable model approach
  - a patient specific model is required for deformation
  - model or training set must be obtained first
  - Leemput et al. (MICCAI’98)

Related Work

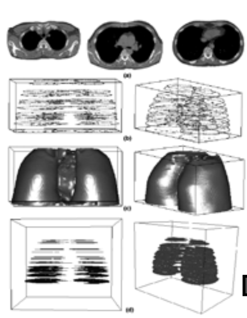
- Differential geometry approach
  - surface can be characterized if second-order differential properties are known
  - good result can be achieved for accurate data
  - Thirion & Gourdon (CVIU'95)
- Level set approach
  - zero crossings of higher dimensional space
  - allow topological changes, non-manifolds
  - methodology is iterative
  - require careful initialization
  - Duncan et al. (MICCAI'98)

Related Work

- Non-iterative approach
  - tensor voting approach
  - no model or other specific assumption is needed
  - Tang, Medioni, Duret (MICCAI'98)

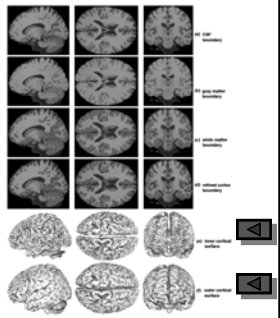
Results (Thorax Dataset)

- 12 real set of CT scans of the thorax is used
- Extract thorax by simple thresholding
- Region boundary inference
- Second order tensor voting is used to refine normal
- Sign of curvatures is detected
- Extraction of surface model is done with scale detection algorithm (CVPR'01)



Results (McGill Brain Dataset)

- MRI dataset of a brain is used
- The CSF, gray matter and white matter are extracted with simple thresholding
- Region boundary inference
- Inference of CSF/GM boundary and GM/WM boundary is done by intersection
- Surface model is generated with similar methods as the thorax
- Ground truth comparison shows our method is comparable with others (CVPR'01)



Overview

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

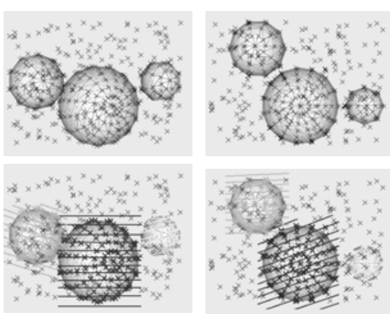
Motivation

- Epipolar geometry is an important constraint
  - stereo registration
  - stereo reconstruction
- Matching points are used for estimation
  - corresponding matching program are never accurate enough
  - false matches are unavoidable
  - non-static scene is even more complicated

Motivation

- Estimate epipolar constraint corresponding to:
  - not only background
  - but also salient motion
  - in the presence of large amount of false matches
  - quite difficult for most of the previous method

Example



Previous Work

- Eight Point Algorithm
  - earliest, simplest, most efficient
  - become inaccurate with large amount of noise contamination
  - Hartley suggest: normalization of data set (PAMI'97)
  - more accurate and comparable to non-linear approach - but still sensitive to outlier noise

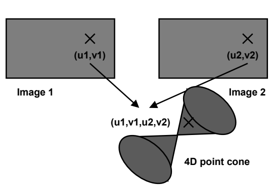
Previous Work

- Robust Algorithms
  - non-linear iterative optimization methods
    - LMedS by Zhang et al. (IJCV'98)
  - random sampling approach
    - RANSAC by Torr and Murray (IJCV'97)
  - good with static scene but unstable with non-static scenes and false matches contamination
- Local Homography Assumption
  - by Pritchett and Zisserman (ICCV'98)
  - point matches are generated by homography
  - but does not apply to curved surfaces

Previous Work

- Noise Removal Techniques
  - ROR by Adam and Rivlin (PAMI'01)
    - with proper rotation of image points
    - correct matching pairs will create line segments pointing in approx same direction
    - require intensive search of the correct rotation
  - 8D Tensor Voting by Tang and Medioni (PAMI'01)
    - voting for the most salient hyperplane in the 8D parameter space
    - the 8D space is not isotopic
    - high dimension require multi-passes of voting

Joint Image Space

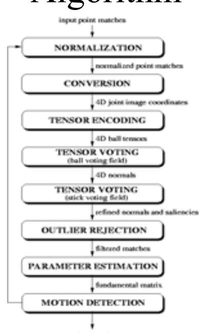


- Suggested by Anandan that the epipolar geometry defines a 4D point cone in the joint image space (ECCV'00)

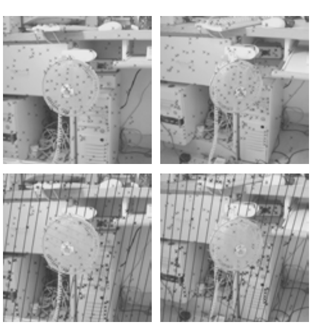
Our Approach

- Use 4D tensor voting to filter off matches not likely on a smooth structure
- Use robust methods: such as RANSAC to estimate epipolar geometry correspond to different motions
- Epipolar geometry of motion object is just the combination of camera and their motions

Algorithm



Results



Results

Results

	Three Spheres			Fan		Umbrella		Car	
	most salient	2nd salient	3rd salient	most salient	2nd salient	most salient	2nd salient	most salient	2nd salient
No. of correct data points S_i	192	83	33	111	30	84	21	93	34
No. of incorrect data points	192	192	191	151	150	101	99	101	100
noise/signal ratio	1.000	2.313	5.788	1.360	5.000	1.202	4.714	1.086	2.941
(A) Results on 4D tensor voting									
No. of correct inliers T_i	150	72	33	80	27	61	19	71	20
No. of incorrect inliers	2	1	0	3	9	3	7	7	2
(B) Results on parameter estimation									
No. of correct inliers R_i	109	50	33	81	30	63	21	59	30
No. of incorrect inliers	0	1	0	1	1	2	2	1	0
Scale used in 4D analysis σ	400			400		400		400	
No. of random trials	1000			10000		10000		10000	
No. of points in each subset	10			15		15		15	

Overview

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

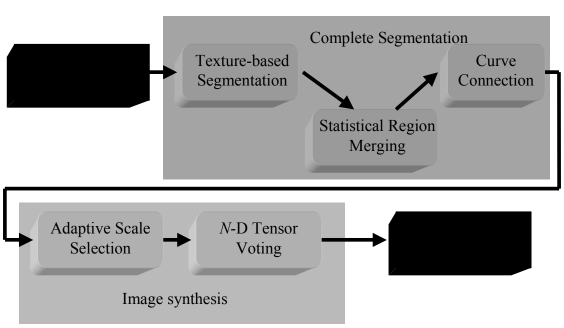
Motivation

- Main difficulties to repair a severely damaged image of natural scene
  - Mixture of texture and colors
  - Inhomogeneity of patterns
  - Regular object shapes

Motivation

- Given as few as one image without additional knowledge, we address:
  - How much color and shape information in the existing part is needed to seamlessly fill the hole?
  - How good can we achieve in order to reduce possible visual artifact when the information available is not sufficient?

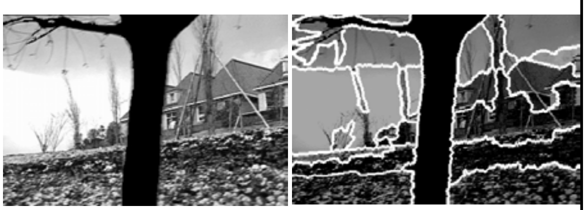
Image repairing system



Segmentation

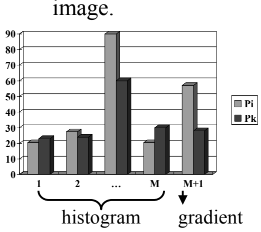
- JSEG [Deng and Manjunath 2001]
  - color quantization
  - spatial segmentation
- Mean shift [Comanicu and Meer 2002]
- Deterministic Annealing Framework [Hofmann et al 1998]

Texture-based Segmentation



Statistical Region Merge

- $(M+1)$D intensity vector V_i^{M+1} for each region P_i, where M is the maximum color depth in the whole image.



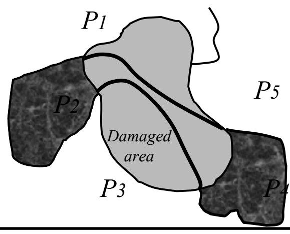
$$V_i\ (M+1) = \frac{\alpha}{N_i}\sum_{j \in P_i} \| \nabla j \|$$

$$P = P_i \cup P_k \qquad \text{if}$$

$$s_{i,k} = \| V_i - V_k \| < Threshold$$

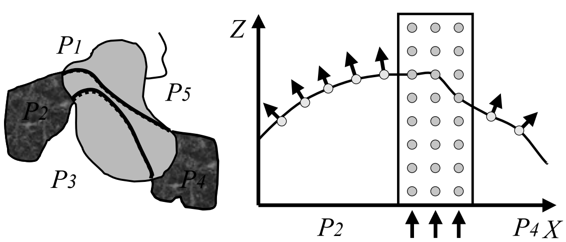
Why Region Merge?

- Decrease the complexity of region topology
- Relate separate regions



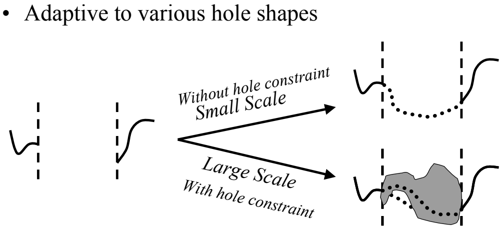
Curve Connection

- 2-D tensor voting method



Why Tensor Voting?

- The parameter of the voting field can be used to control the smoothness of the resulting curve.
- Adaptive to various hole shapes



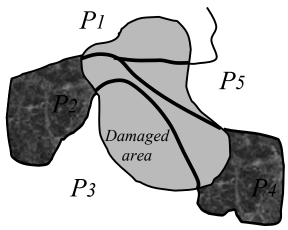
Connection Sequence

- Greedy algorithm
  - Always connect the most similar regions

P2 **and** *P4*

P3 **and** *P5*

P1



Complete Segmentation

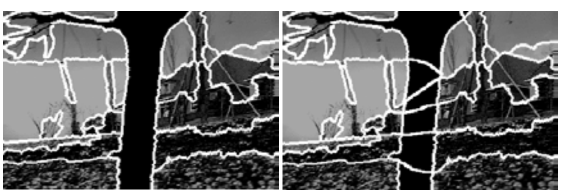
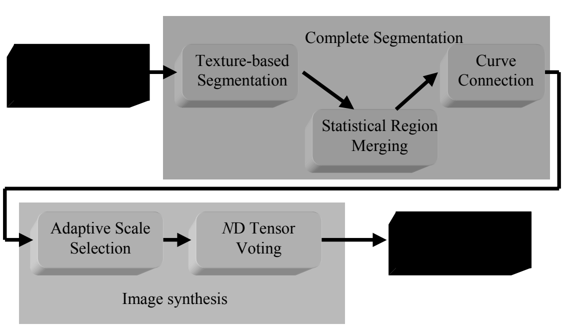
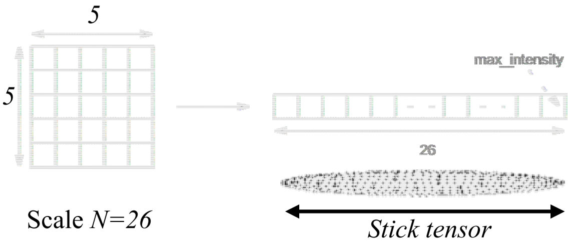


Image repairing system



N-D Tensor Voting

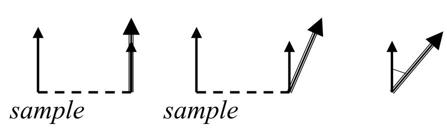
- Tensor encoding
  - Each pixel is encoded as an N-D stick tensor



Scale $N=26$

N-D Tensor Voting

- Voting process in N-D space
  - An osculating circle becomes an osculating hypersphere.
  - N-D stick voting field is uniform sampling of normal directions in the N-D space.



Adaptive Scaling

- Texture inhomogeneity in images gives difficulty to assign only one global scale N [Lindeberg et al 1996].
- For each pixel i in images, we calculate:

$$M_{N_i} = AVG_{N_i}\{(\nabla I)(\nabla I)^T\}$$

- *trace*(M) measures the average strength of the square of the gradient magnitude in the window of size Ni

Adaptive Scaling

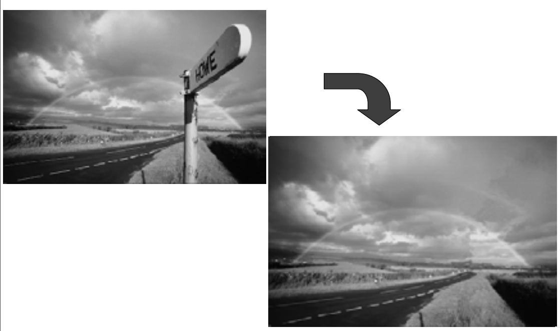
- For each sample seed:
  - Increase its scale *Ni* from the lower bound to the upper bound
  - If *trace*(M_{N_i}) < *trace*(M_{N_i-1}) - α where α is a threshold to avoid small perturbation or noise interference, set *Ni* - 1 → *Ni* and return
  - Otherwise, continue the loop until maxima or upper bound is reached

Results

Results

Results

Results

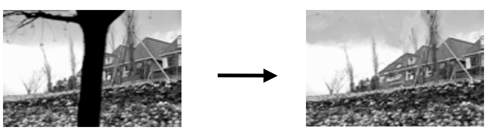


Results

Limitations

- Lack of samples.
- Meaningful and semi-regular objects.

Conclusion



- An automatic image repairing system.
- Region partition and merging.
- Curve connection by 2D tensor voting.
- *N*D tensor voting based image synthesis.
- Adaptive scale.

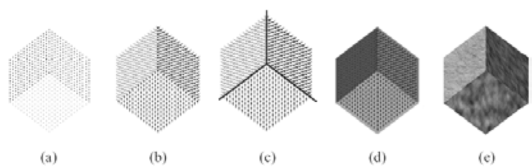
Overview

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

Introduction

- Geometric hole filling
- Range data: one depth value for a pixel
- Extension of 2-D curve connection

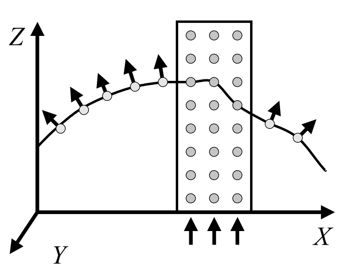
Procedure



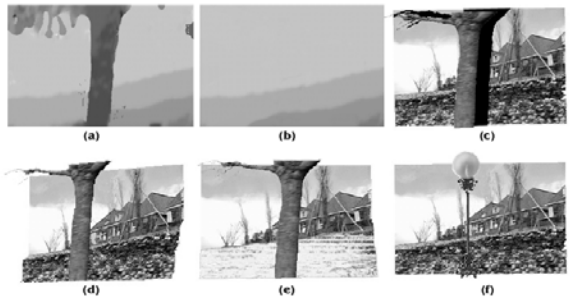
- Data inference
- Normal estimation
- Curve junctions
- Surface mesh
- Texture synthesis

Range data

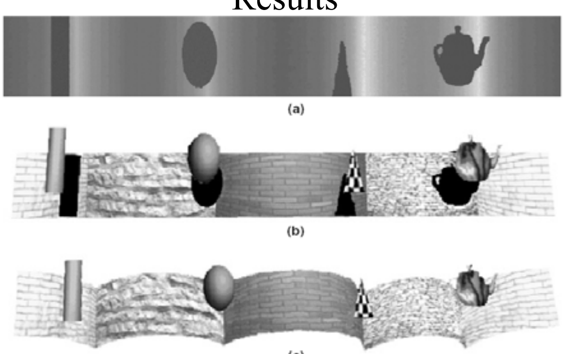
- 3-D data synthesis with additional Y axis



Results



Results



Results

Overview

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

Motivation

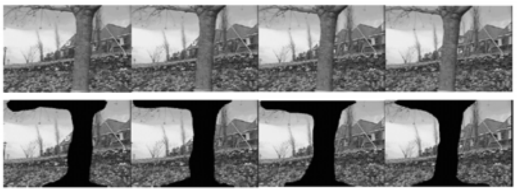
- Video Retouching
- Difficulties:
  - Frame-by-frame repairing can not maintain temporal coherence
  - Single mosaic has limitations

Our approach

- Hole propagation
- Layer propagation
- Layered reference mosaic
- Image repairing
- Homography blending

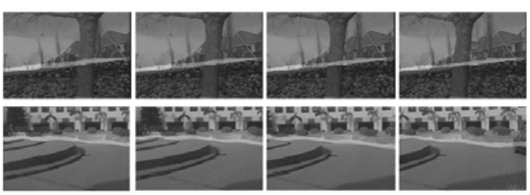
Hole Propagation

- Frame-by-frame segmentation
- Mean shift tracking



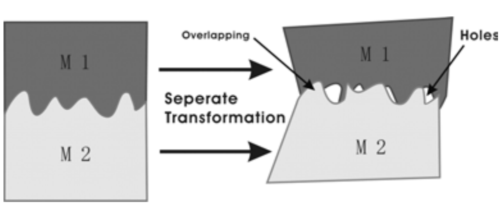
Layer Propagation

- Users only need to specify rough layer boundaries in key frames
- Optical Flow method to track the boundaries



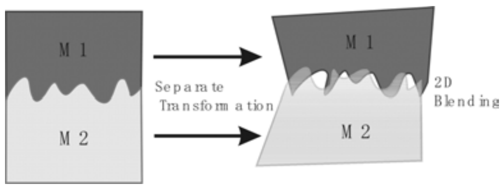
Homography Blending

- Overlapping and small holes are obtained by warping two layers, using their respective homographic transform



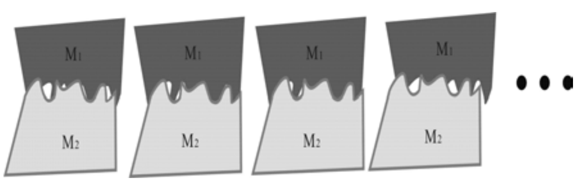
Homography Blending

- Normal 2-D blending



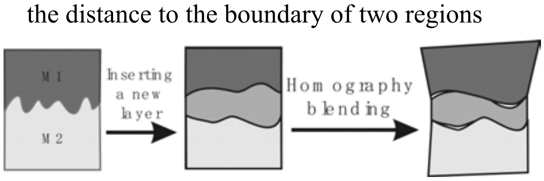
Homography Blending

- 2-D blending is performed on single frame
- 2-D blending can not solve the boundary shifting problem for a video sequence



Homography Blending

- Blending homography matrices
- Weighting function is calculated according to the distance to the boundary of two regions



Results


Overview

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

Motivation

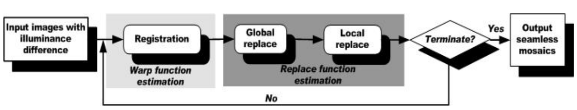
- Luminance inconsistency when registering images or frames
- Several factors
  - Exposure variance
  - White balance
  - Gamma correction
  - Vignetting
  - Digitizer parameters

Related Work

- Mosaic registration with exposure correction in the overlapping area
  - Block blending
  - Feather-based blending
  - Purpose: constructing natural transition
- Radiometric calibration
  - Construction high range image from several static images

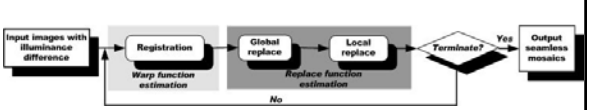
Our approach

- Warp function estimation
- Replacement function estimation
  - The function directly measures the color difference between images
- Modeless method



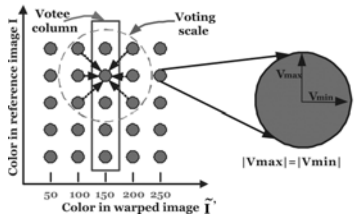
Our approach

- Global replace – globally map colors between images
- Local replace – estimate vignetting effect for each image
- Final replacement – global(local(.))



Global Replace

- Luminance Voting
- Voting space construction
  - Joint image
- Tensor encoding
  - 2D ball tensor for each sample
- Luminance voting
  - Exclusive voting



Monotonic Constraint

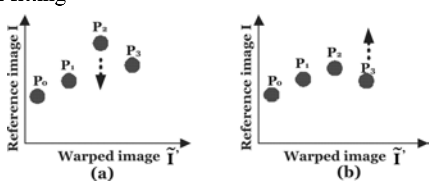
Monotonic constraint: Let (I',I) be the continuous joint image space.

If $I'(x_1'', y_1'') > I'(x_2'', y_2'')$, then $I(x_1, y_1) > I(x_2, y_2)$

if $(x_1, y_1) \leftrightarrow (x_2, y_2)$ and $(x_1'', y_1'') \leftrightarrow (x_2'', y_2'')$ are

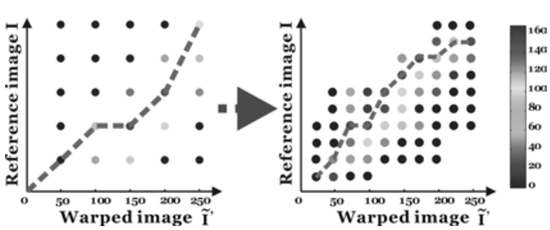
corresponding pixel pairs in overlapping area

- Local fitting



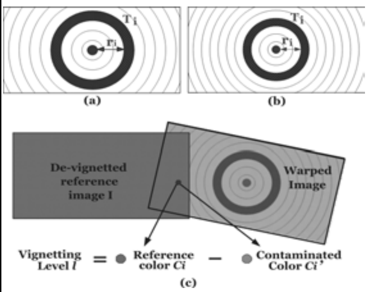
Multiscale

- Inferring the most-likely curve from noise
- Gaussian pyramid is constructed

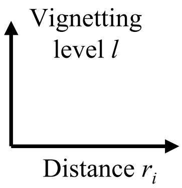


Local Replacement

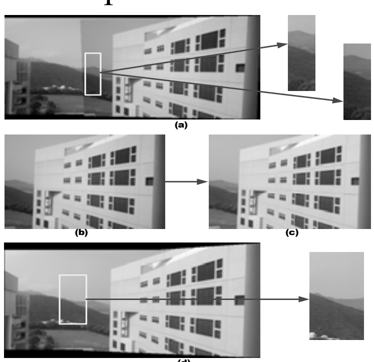
- Same process to estimate local replacement curve
- Vignetting is distance based optic phenomenon



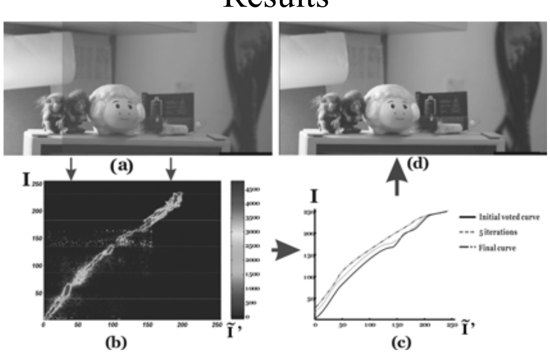
- Only difference on voting space



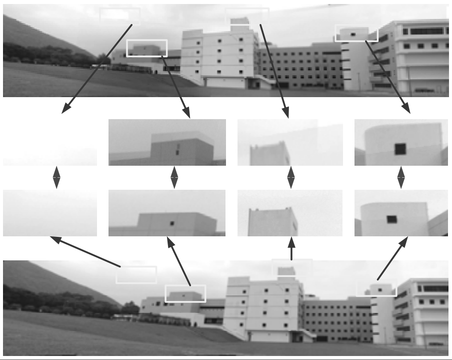
Local Replacement Results



Results



Results



Conclusion

- 2-D tensor voting method to connect curves
- Adaptive scale selection
- Geometric hole filling
- Layered background extraction and propagation
- Homography blending
- Luminance voting
- Global and local replacement function

Future work

- Movement registration
  - Mosaics with moving objects
  - Other image-based applications
- Generalized video repairing
  - Broader class of camera motions
  - Complex foreground

Overview

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

Contributions

- General, expandable computational framework
- Unified and rich representation for all potential types of structure and outliers
  - No hard decisions at early stages
- Model-free
- Very little initialization requirements

Contributions

- Efficient local information propagation through Tensor Voting
- Non-iterative
- Robust to noise
- Good results in large range of problems in Computer Vision and other fields

Future Work: Multiple Scales

- Scale affects
  - Desired level of smoothness
  - Noise robustness
  - Detail Preservation
  - Ability to fill in gaps
- A single scale may be insufficient in most cases
- Work has been done on multiple scale Tensor Voting but more is needed

Future Work

- Scale adaptation based on local criteria / automatic scale selection
- Integration of time into the framework
- Improvements in computational efficiency

Bibliography

- Book:
  - G. Medioni, M.S. Lee, and C.K Tang, "A Computational Framework for Feature Extraction and Segmentation", *Elsevier,* 2000
- Journal Articles
  - G. Guy and G. Medioni, "Inferring Global Perceptual Contours from Local Features", *IJCV*, vol 20, no. 1/2, pp. 113-133, 1996.
  - G. Guy and G. Medioni, "Inference of Surfaces, 3D Curves, and Junctions from Sparse, Noisy, 3-D Data", *IEEE Trans. on PAMI,* vol. 19, no.11, pp. 1265-1277, 1997

Bibliography

– M.S. Lee and G. Medioni, “Grouping ., -, ->, O-, into Regions, Curves, and Junctions”, *CVIU*, vol. 76, no. 1, pp. 54-69, 1999.
– M.S. Lee, G. Medioni, and P. Mordohai, “Inference of Segmented Overlapping Surfaces from Binocular Stereo”, *IEEE Trans. on PAMI*, vol. 24, no. 6, pp. 824-837, 2002
– M. Nicolescu and G. Medioni, "Layered 4-D Representation and Voting for Grouping from Motion", *IEEE Trans. on PAMI , vol. 25, no. 4, pages 492-501*, 2003

Bibliography

– C.K. Tang and G. Medioni, “Inference of Integrated Surface, Curve, and Junction Descriptions from Sparse 3-D Data", *IEEE Trans. on PAMI,* vol. 20, no. 11, pp. 1206-1223, 1998
– C.K. Tang, G. Medioni, and M.S. Lee, “N-Dimensional Tensor Voting, and Application to Epipolar Geometry Estimation," *IEEE Trans. on PAMI,* vol. 23, no.8, pp. 829-844, 2001
– C.K. Tang, G. Medioni, “Curvature-Augmented Tensorial Framework for Integrated Shape Inference from Noisy, 3-D Data”, *IEEE Trans. on PAMI,* vol. 24, no. 6, pp. 858-864, 2002