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

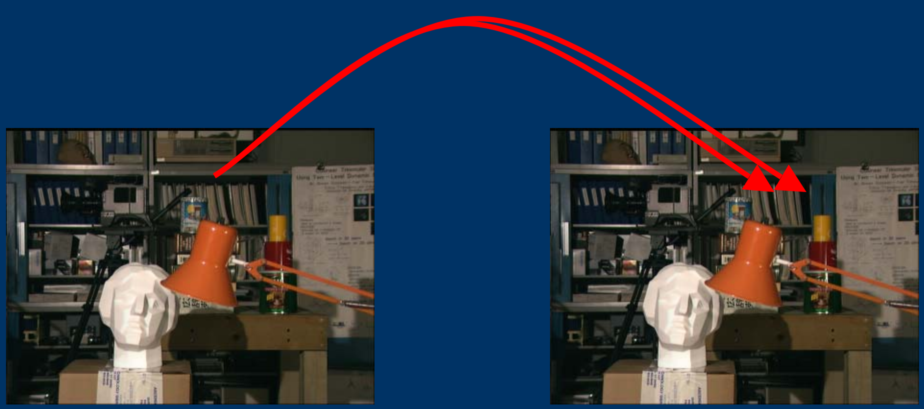
- So far, perceptual organization of tokens
- In real problems tokens represent image primitives:
  - Intensity
  - Color
  - Contrast
  - Disparity
  - Optical flow

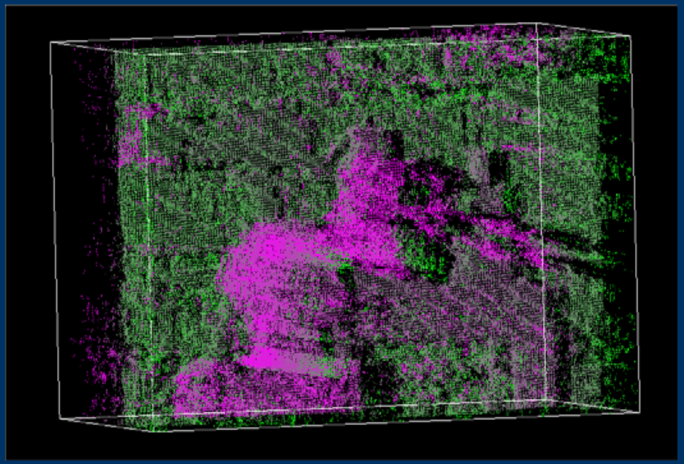
Token Generation from Pixel Correspondences

- 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



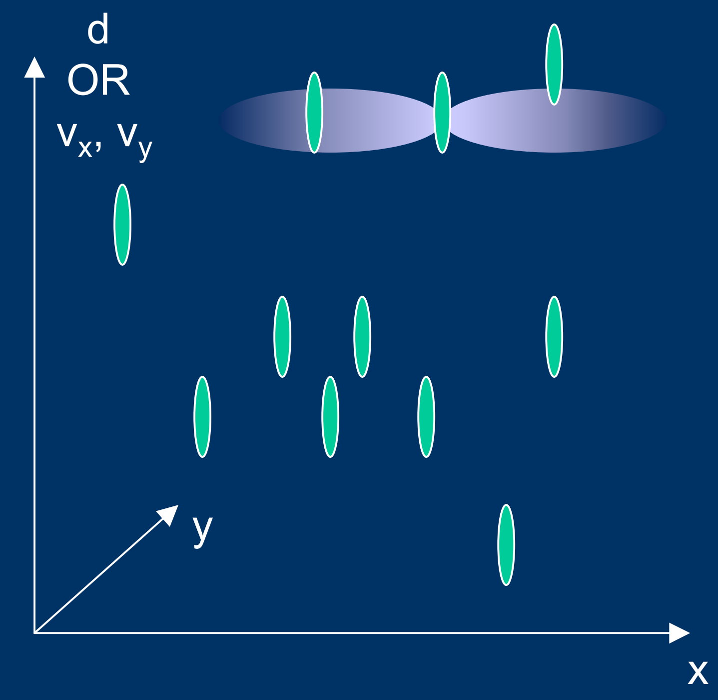


Initial Matching

- 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

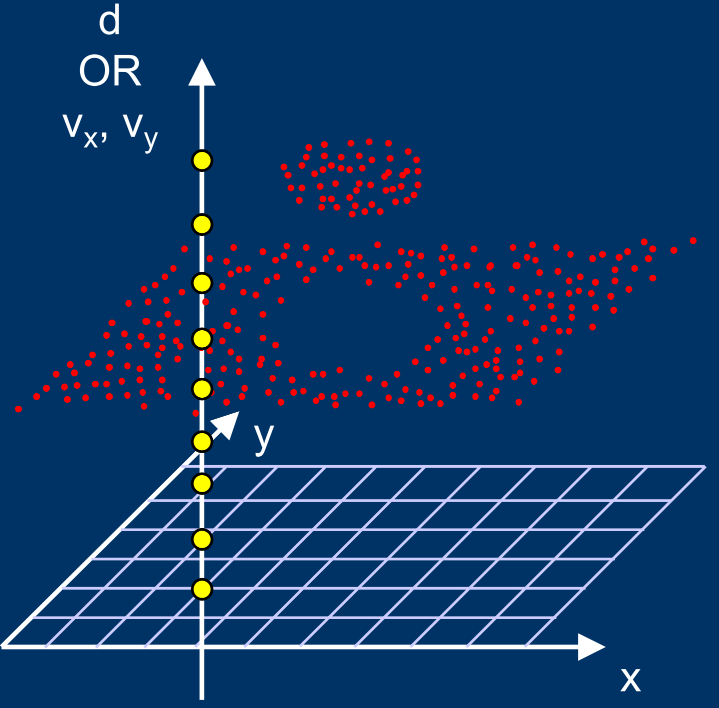
- Tokens are classified according to *saliency* and *polarity*
- Most salient token along each Line of Sight is retained
  - Disambiguation of initial matches
- Outliers rejected based on low saliency
- Other constraints can also be added here

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

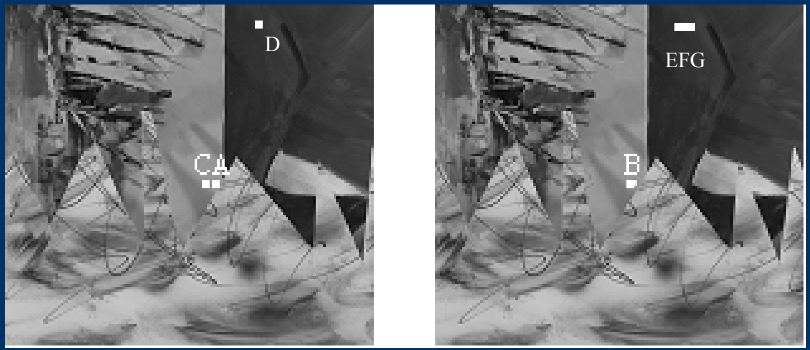
- Binocular stereo
- Multiple view stereo

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

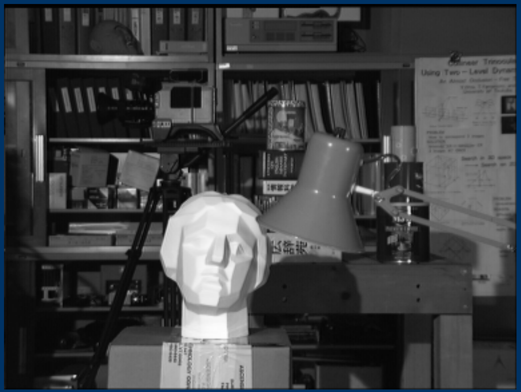
Challenges

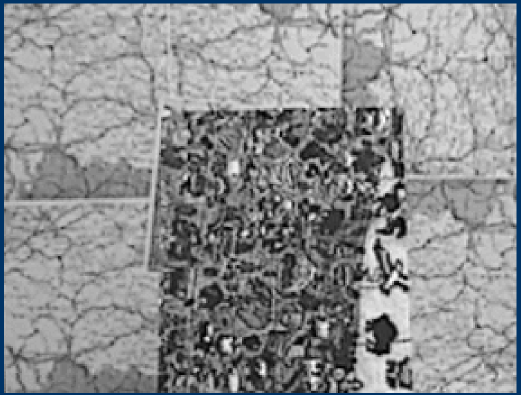
- Occluded pixels
  - Sometimes can generate higher matching scores than correct correspondences
- Textureless pixels
  - Ambiguous matching



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

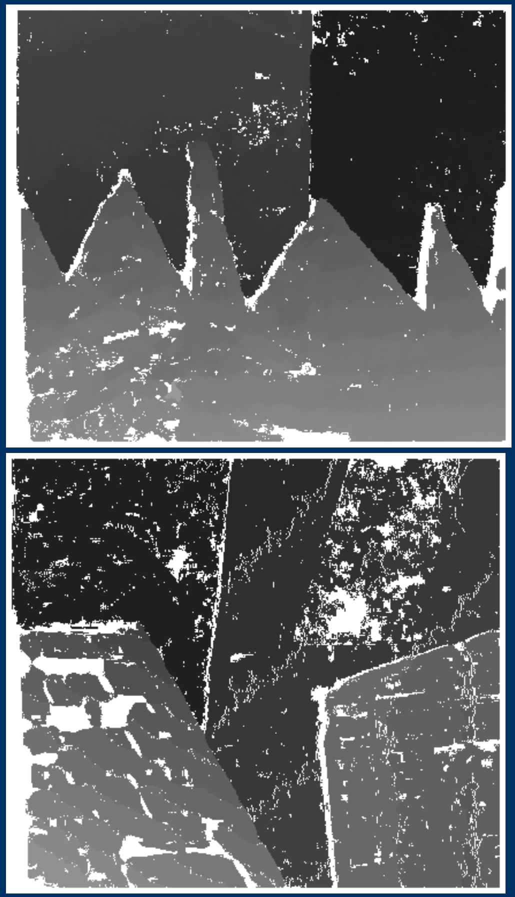



Grayscale versions used in all experiments

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)

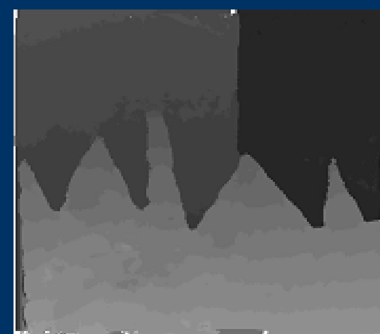
Discrete Densification

For each pixel without disparity estimate:

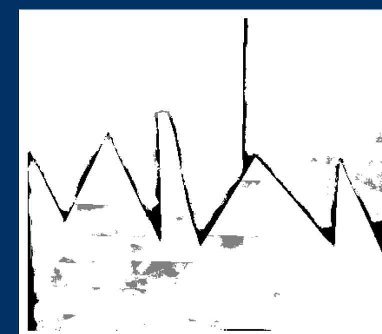
- Find disparity range from neighboring pixels
  - extend by a few disparity levels
- Generate candidate tokens for each potential disparity
- Collect votes at candidates
- Select most salient

Discrete Densification Resutls

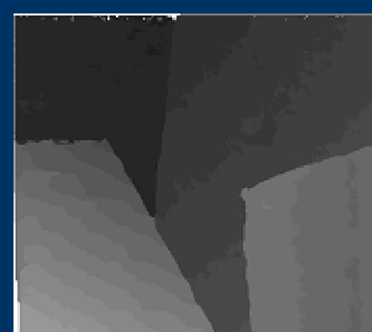
- Textureless pixels have been treated using smoothness of surfaces, since intensity is ambiguous
- Most errors in occluded pixels
  - Black: errors >1 disparity level
  - Gray: errors between 0.5 and 1 disparity level



Disparity map



Error map



Disparity map

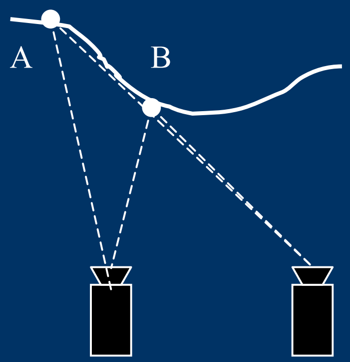
Error map

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

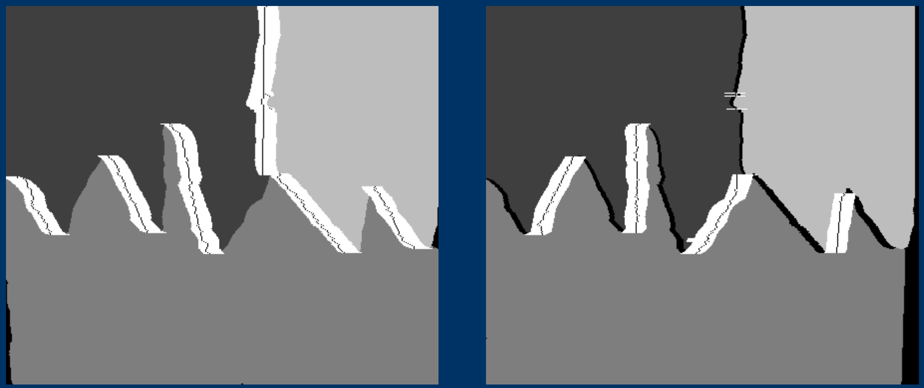
- Use depth discontinuities of dense disparity map to define “uncertainty zones” on both images
- Transition from high to low disparity defines an uncertainty zone in the left image
- Transition from low to high disparity defines an uncertainty zone in the right image
- Width proportional to disparity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



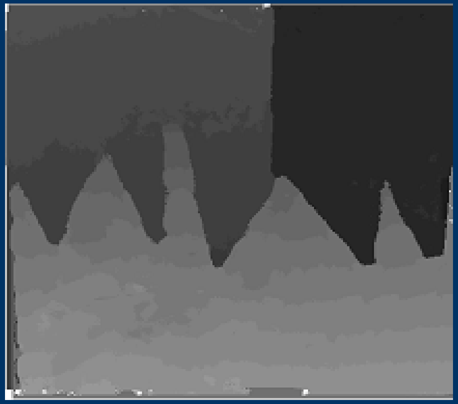
Left Image

Right Image

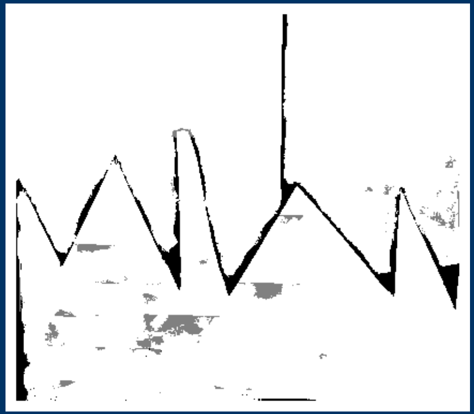
White: uncertainty zones

Black: no disparity estimate

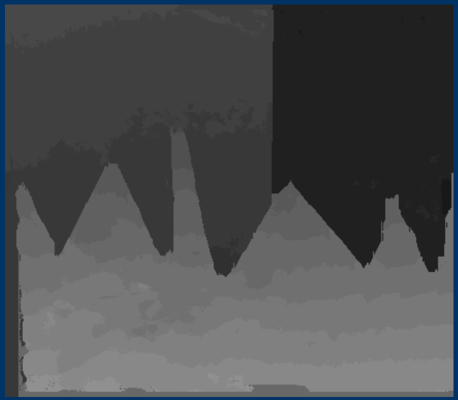
Use of Monocular Information: Depth Discontinuities (results)



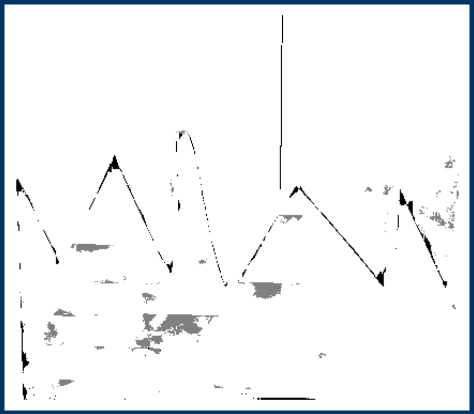
(a) Disparity map



(b) Error map

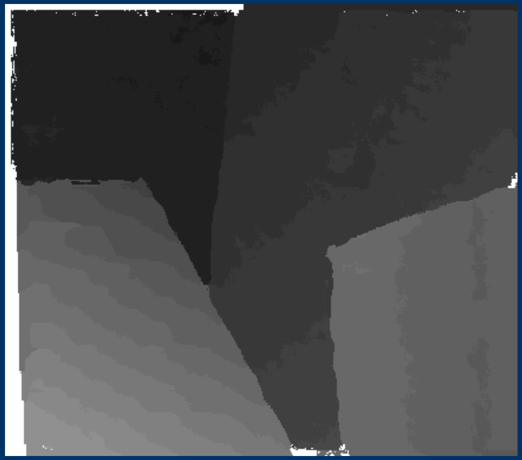


(c) Final disparity map



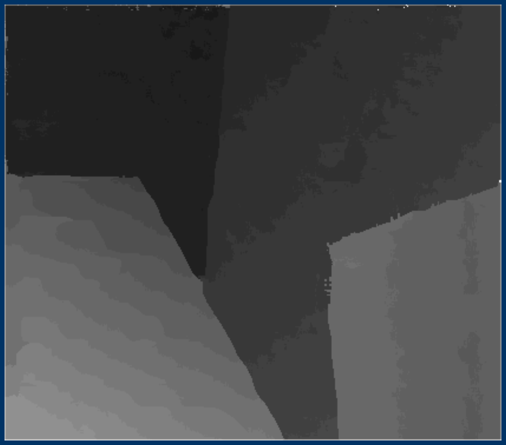
(d) Final error map

Venus



(a) Disparity map

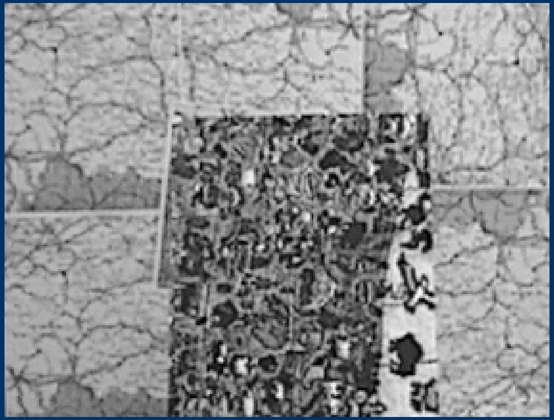
(b) Error map



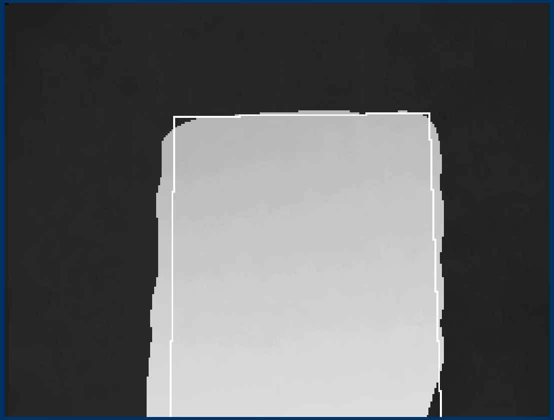
(c) Final disparity map

(d) Final error map

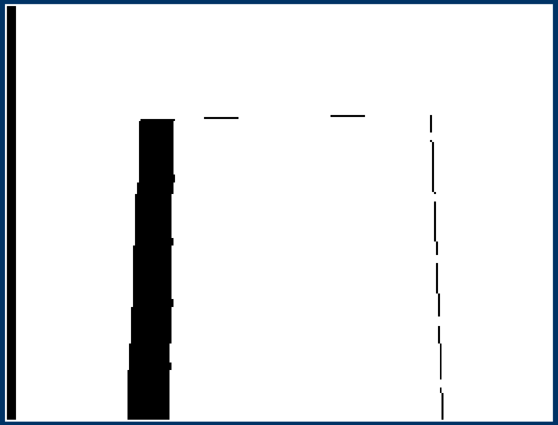
Map



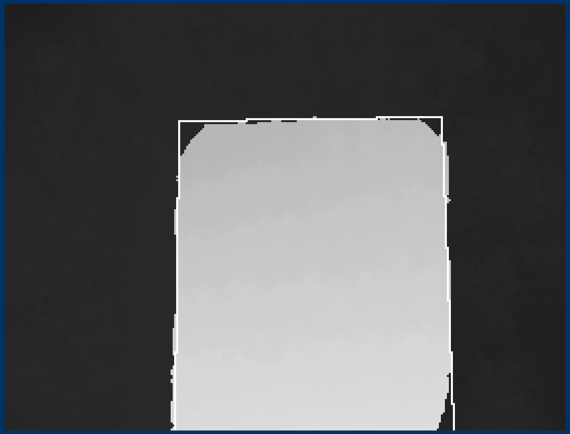
(a) Left image



(b) Disparity map with ground truth discontinuities

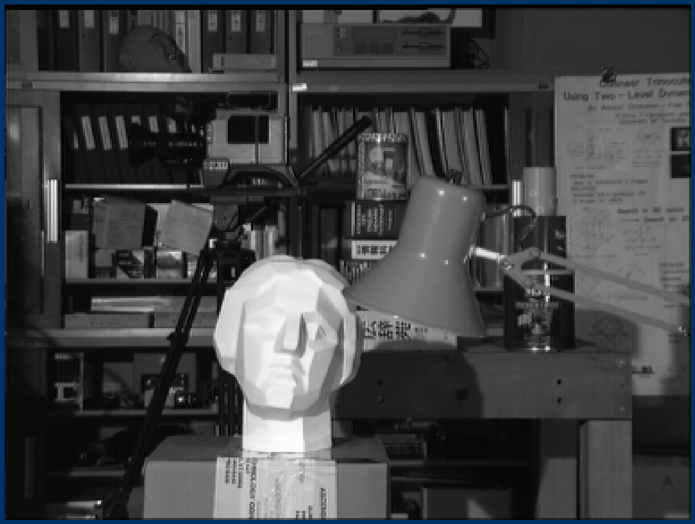


(c) True occlusion map



(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

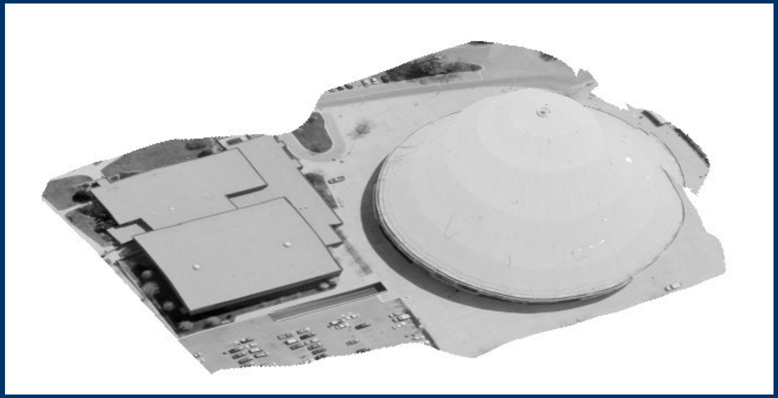
Scale	Error Rate
10	1.32%
20	1.27%
50	1.09%
100	0.97%
200	0.92%
500	0.93%
1000	1.06%
2000	1.10%

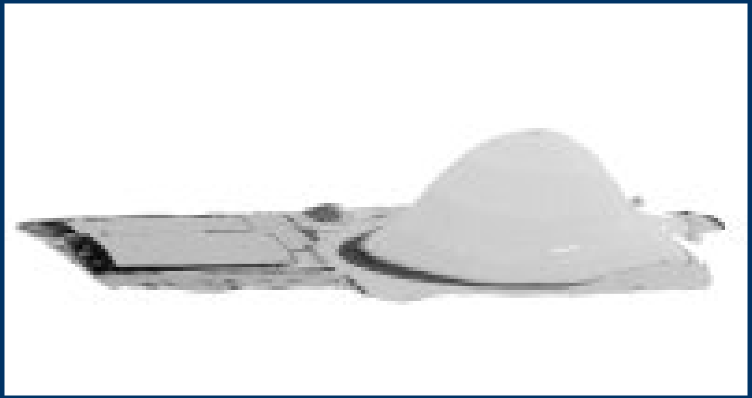
Arena

(non-parametric surfaces)

Left image

Right image





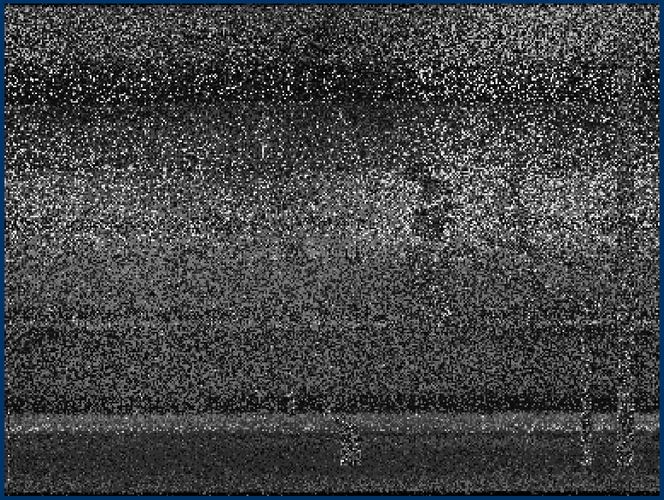
Views of reconstructed model in projective space

Random Texture

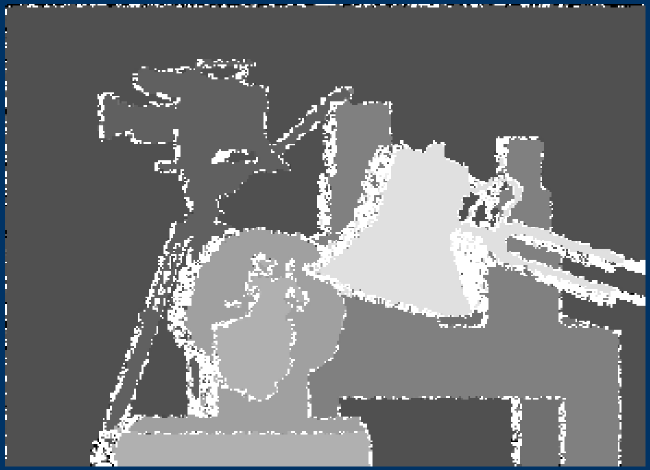
- Intensities randomly re-arranged in left image
- Ground truth disparities used to create right image
- Now, every pixel is textured

Random Texture

Left image



Right image



Results of sparse vote

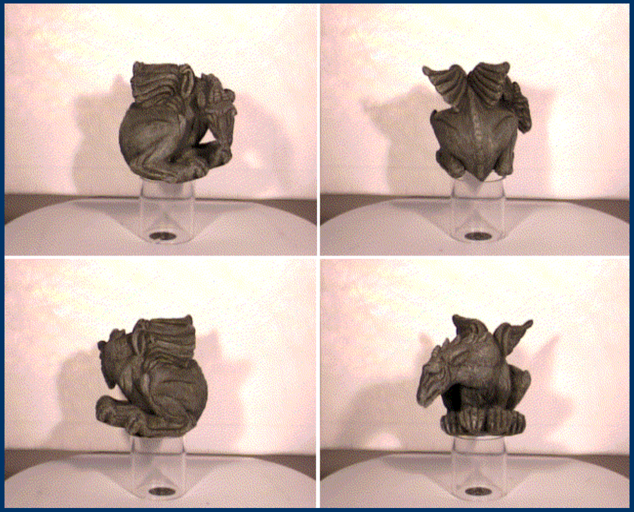
Error map

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

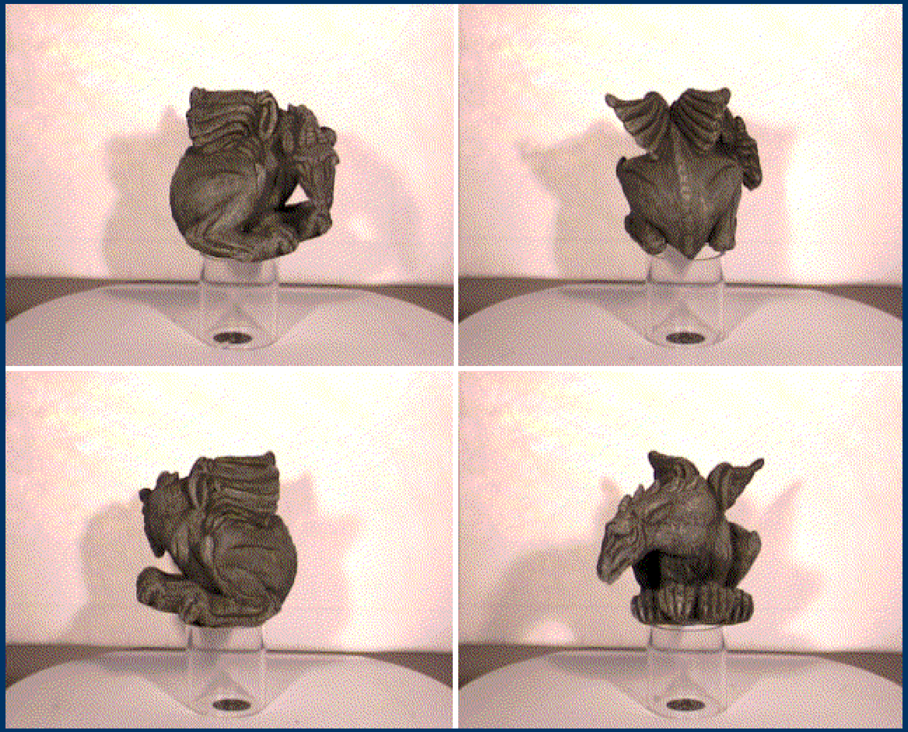
- Multiple images of complex scenes
- 360^o views
- No features visible in all views
- Non-planar surfaces
- Regions not equally rich in texture



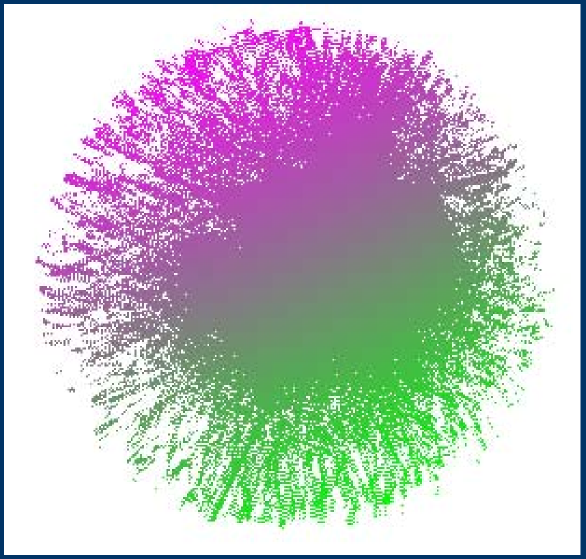
Approach

- 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

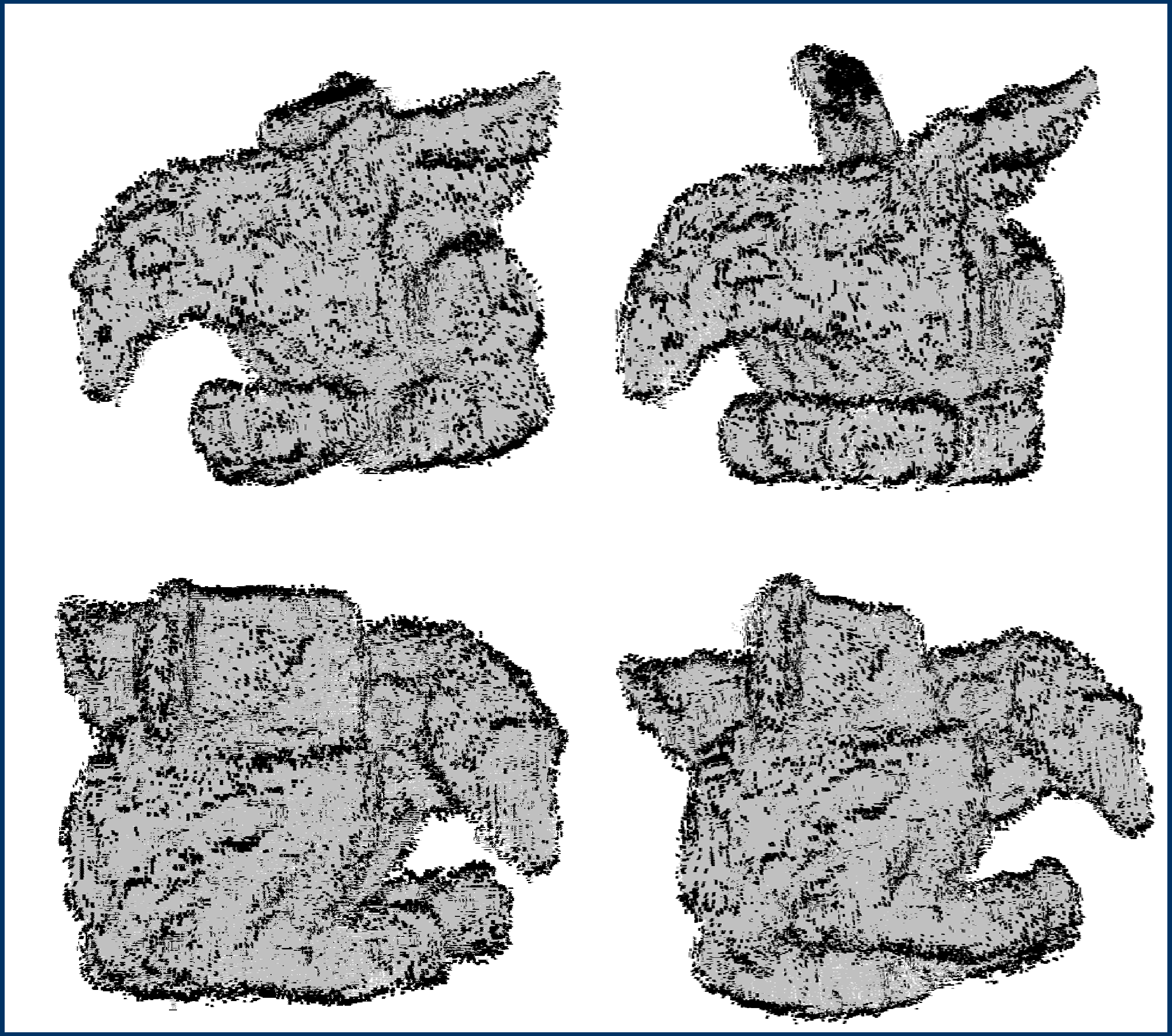


Four of the 36 input images



Initial matches (viewed from above)

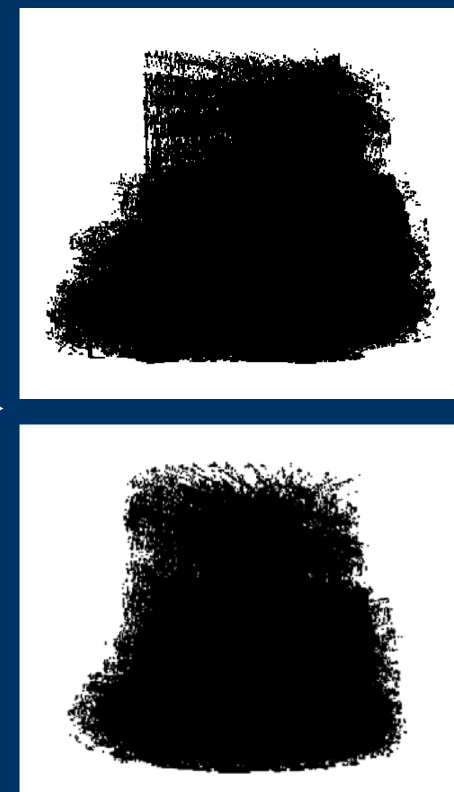
Dragon Output



Gray: surface inliers
Black: discontinuities

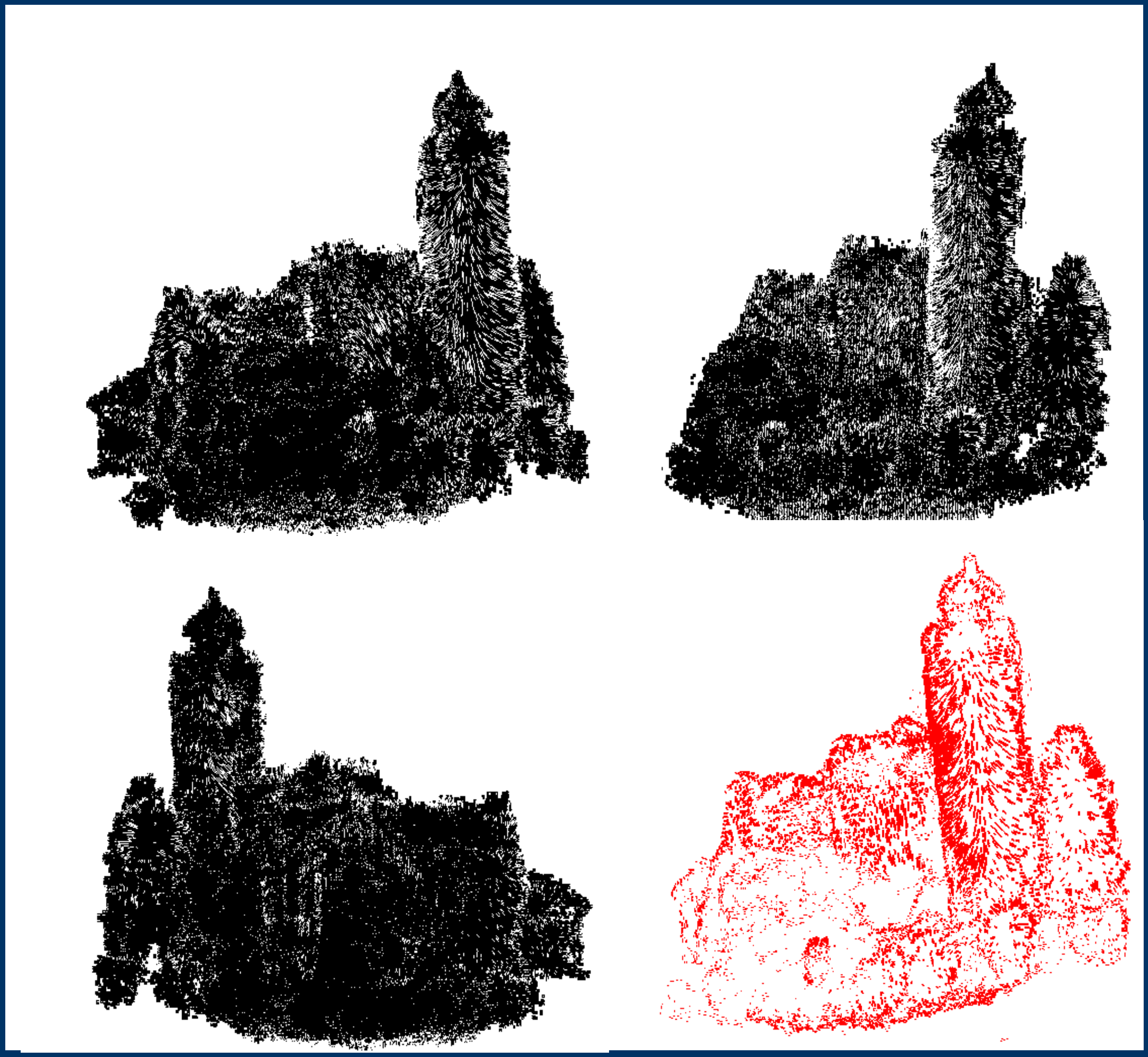
Lighthouse Input

Four of the 36 input images



Initial matches

Lighthouse Output



Black: surface inliers
Red: discontinuities