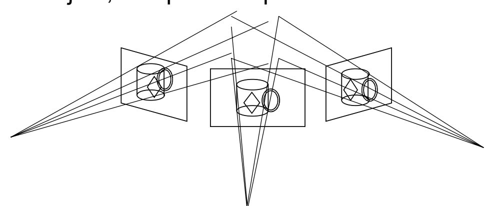
Stereo Matching

Computer Vision
CSE576, Spring 2005
Richard Szeliski

Stereo Matching

Given two or more images of the same scene or object, compute a representation of its shape



What are some possible applications?

Face modeling

From one stereo pair to a 3D head model

[Frederic Devernay, INRIA]

Z-keying: mix live and synthetic

Takeo Kanade, CMU (Stereo Machine)

Virtualized Reality™

[Takeo Kanade *et al.*, CMU]

- collect video from 50+ stream
- reconstruct 3D model sequences



- steerable version used for SuperBowl XXV “eye vision”

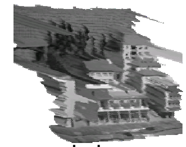
http://www.cs.cmu.edu/afs/cs/project/VirtualizedR/www/VirtualizedR.html

View Interpolation

Given two images with correspondences, *morph* (warp and cross-dissolve) between them [Chen & Williams, SIGGRAPH’93]


input

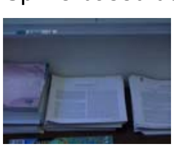

depth image


novel view

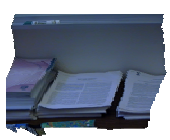
[Matthies,Szeliski,Kanade’88]

More view interpolation

Spline-based depth map

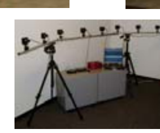

input


depth image


novel view

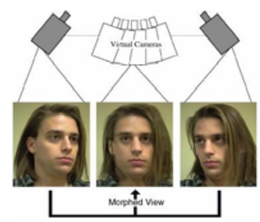
[Szeliski & Kang ‘95]

Video view interpolation



View Morphing

Morph between pair of images using epipolar geometry [Seitz & Dyer, SIGGRAPH'96]



Additional applications?

- Real-time people tracking (systems from Pt. Gray Research and SRI)
- "Gaze" correction for video conferencing [Ott,Lewis,Cox InterChi'93]
- Other ideas?

Stereo Matching

Given two or more images of the same scene or object, compute a representation of its shape

What are some possible representations?

- depth maps
- volumetric models
- 3D surface models
- planar (or offset) layers

Stereo Matching

What are some possible algorithms?

- match “features” and interpolate
- match edges and interpolate
- match all pixels with windows (coarse-fine)
- use optimization:
  - iterative updating
  - dynamic programming
  - energy minimization (regularization, stochastic)
  - graph algorithms

Outline (remainder of lecture)

Image rectification

Matching criteria

Local algorithms (aggregation)

- iterative updating

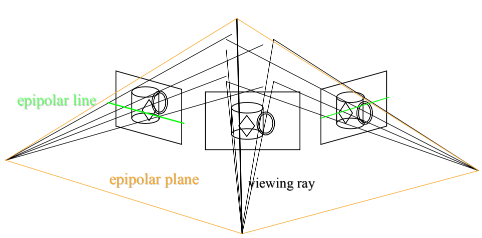
Optimization algorithms:

- energy (cost) formulation & Markov Random Fields
- mean-field, stochastic, and graph algorithms

Multi-View stereo & occlusions

Stereo: epipolar geometry

Match features along epipolar lines



Stereo: epipolar geometry

for *two* images (or images with collinear camera centers), can find epipolar lines

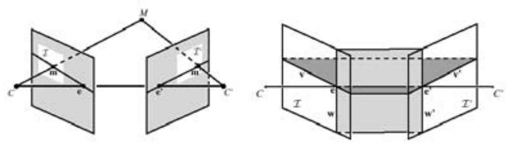
epipolar lines are the projection of the *pencil* of planes passing through the centers

Rectification: warping the input images (perspective transformation) so that epipolar lines are horizontal

Rectification

Project each image onto same plane, which is parallel to the epipole

Resample lines (and shear/stretch) to place lines in correspondence, and minimize distortion



[Zhang and Loop, MSR-TR-99-21]

Rectification

(a) Original image pair overlayed with several epipolar lines.

(b) Image pair transformed by the specialized projective mapping $\mathbf{H}_p$ and $\mathbf{H}'_p$. Note that the epipolar lines are now parallel to each other in each image.

BAD!

Rectification

(c) Image pair transformed by the similarity $\mathbf{H}_r$ and $\mathbf{H}'_r$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

(d) Final image rectification after shearing transform $\mathbf{H}_s$ and $\mathbf{H}'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

“Corner” like features [Zhang, …]

Edges [many people…]

Gradients [Seitz 89; Scharstein 94]

Rank statistics [Zabih & Woodfill 94]

Finding correspondences

apply feature matching criterion (e.g., correlation or Lucas-Kanade) at *all* pixels simultaneously

search only over epipolar lines (many fewer candidate positions)

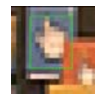
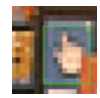

Image registration (revisited)

How do we determine correspondences?

- *block matching* or *SSD* (sum squared differences)

$$E(x,y;d) = \sum_{(x',y')\in N(x,y)} [I_L(x'+d,y') - I_R(x',y')]^2$$

d is the *disparity* (horizontal motion)

How big should the neighborhood be?

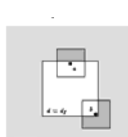
Neighborhood size

Smaller neighborhood: more details

Larger neighborhood: fewer isolated mistakes


w = 3

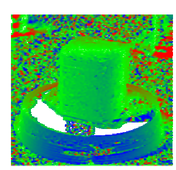

w = 20

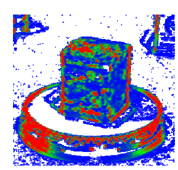


Stereo: certainty modeling

Compute certainty map from correlations

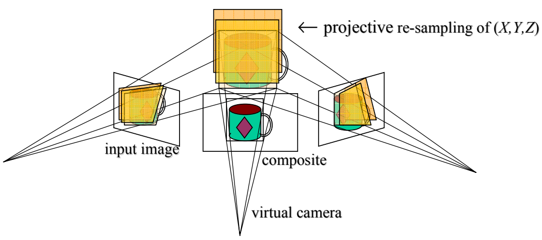

input


depth map


certainty map

Plane Sweep Stereo

Sweep family of planes through volume



- each plane defines an image ⇒ composite homography

Plane Sweep Stereo

For each depth plane

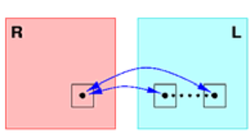
- compute composite (mosaic) image — *mean*

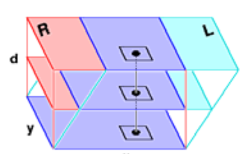
- compute error image — *variance*
- convert to confidence and aggregate spatially

Select winning depth at each pixel

Plane sweep stereo

Re-order (pixel / disparity) evaluation loops





for every pixel,
for every disparity
compute cost

for every disparity
for every pixel
compute cost

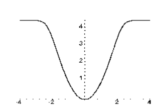
Stereo matching framework

1. For every disparity, compute *raw* matching costs

$$E_0(x, y; d) = \rho(I_L(x' + d, y') - I_R(x', y'))$$

Why use a robust function?

- occlusions, other outliers

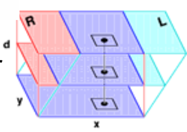


Can also use alternative match criteria

Stereo matching framework

2. Aggregate costs spatially

$$E(x,y;d) = \sum_{(x',y') \in N(x,y)} E_0(x',y',d)$$



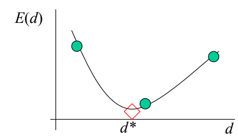
- Here, we are using a *box filter* (efficient moving average implementation)
- Can also use weighted average, [non-linear] diffusion…

Stereo matching framework

3. Choose winning disparity at each pixel

$$d(x,y) = \arg\min_d E(x,y;d)$$

4. Interpolate to *sub-pixel* accuracy



Traditional Stereo Matching

Advantages:
- gives detailed surface estimates
- fast algorithms based on moving averages
- sub-pixel disparity estimates and confidence

Limitations:
- narrow baseline $\Rightarrow$ noisy estimates
- fails in textureless areas
- gets confused near occlusion boundaries

Stereo with Non-Linear Diffusion

Problem with traditional approach:
- gets confused near discontinuities

New approach:
- use iterative (non-linear) aggregation to obtain better estimate
- provably equivalent to mean-field estimate of Markov Random Field

Linear diffusion

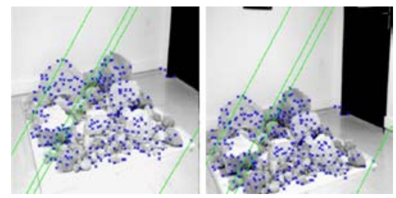
Average energy with neighbors + starting value

$$E(x,y,d) \leftarrow (1-4\lambda)E(x,y,d) + \lambda \sum_{(k,l)\in\mathcal{N}_4} E(x+k,y+l,d)$$
$$+\beta(E_0(x,y,d) - E(x,y,d))$$

window diffusion

Feature-based stereo

Match “corner” (interest) points



Interpolate complete solution

Data interpolation

Given a sparse set of 3D points, how do we *interpolate* to a full 3D surface?

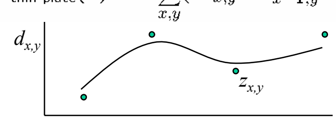
Scattered data interpolation [Nielson93]

- triangulate
- put onto a grid and fill (use pyramid?)
- place a *kernel function* over each data point
- minimize an energy function

Energy minimization

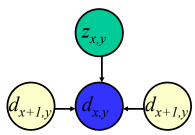
1-D example: approximating splines

$$\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{membrane}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - d_{x-1,y})^2 \\
E_{\text{thin plate}}(\mathbf{d}) &= \sum_{x,y} (2d_{x,y} - d_{x-1,y} - d_{x+1,y})^2
\end{aligned}$$



Relaxation

Iteratively improve a solution by locally minimizing the energy: *relax* to solution



Earliest application: WWII numerical simulations

Relaxation

How can we get the best solution?

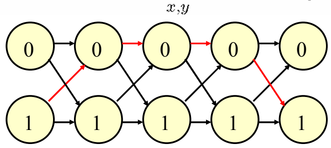
Differentiate energy function, set to 0

$$\begin{aligned}\frac{\partial E}{\partial d_{x,y}} &= 2(d_{x,y} - z_{x,y}) + \\ &\quad 2\lambda(2d_{x,y} - d_{x-1,y} - d_{x+1,y}) = 0 \\ d_{x,y} &\leftarrow \frac{1}{1+2\lambda}(z_{x,y} + d_{x-1,y} + d_{x+1,y})\end{aligned}$$

Dynamic programming

Evaluate best cumulative cost at each pixel

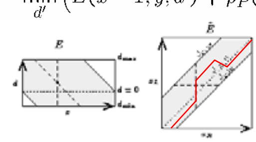
$$\begin{aligned}E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\ E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\ E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|\end{aligned}$$



Dynamic programming

1-D cost function

$$\begin{aligned}E(\mathbf{d}) &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x,y;d) \\ \tilde{E}(x,y,d) &= E_0(x,y;d) + \\ &\quad \min_{d'} \left(\tilde{E}(x-1,y,d') + \rho_P(d_{x,y} - d'_{x-1,y})\right)\end{aligned}$$



Dynamic programming

Disparity space image and min. cost path

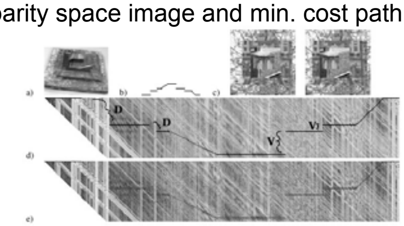


Fig. 4. This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation DSI_{xs} representation for the image pair in (c), and (e) the enhanced, cropped, correlation DSI for a noisy sloping wedding cake (SNR = 18 dB). In (d), the regions labeled "D" mark diagonal gaps in the matching path caused by regions occluded in the left image. The regions labeled "V" mark vertical jumps in the path caused by regions occluded in the right image.

Dynamic programming

Sample result
(note horizontal streaks)

[Intille & Bobick]

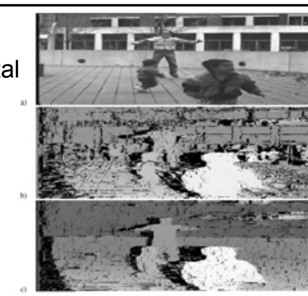
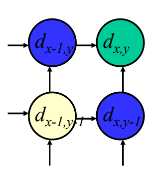


Fig. 12. Results of two stereo algorithms on Figure 1. (a) Original left image. (b) Cox *et al.* algorithm[14], and (c) the algorithm described in this paper.

Dynamic programming

Can we apply this trick in 2D as well?

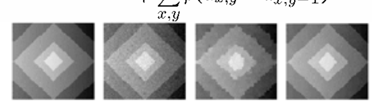


No: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

Graph cuts

Solution technique for general 2D problem

$$\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} f_{x,y}(d_{x,y}) \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} \rho(d_{x,y} - d_{x-1,y}) \\
&\quad + \sum_{x,y} \rho(d_{x,y} - d_{x,y-1})
\end{aligned}$$



(a) original image (b) observed image (c) local min w.r.t. standard moves (d) local min w.r.t. α-expansion moves

Graph cuts

α-β swap

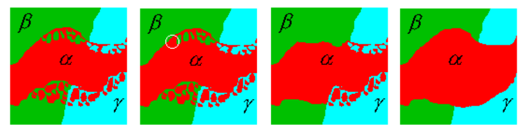
α expansion

modify smoothness penalty based on edges

compute best possible match within integer disparity

Graph cuts

Two different kinds of moves:



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Bayesian inference

Formulate as statistical inference problem

Prior model $p_P(\boldsymbol{d})$

Measurement model $p_M(I_L, I_R| \boldsymbol{d})$

Posterior model

$$p_M(\boldsymbol{d} \mid I_L, I_R) \propto p_P(\boldsymbol{d})\, p_M(I_L, I_R| \boldsymbol{d})$$

Maximum a Posteriori (MAP estimate):

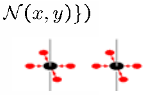
maximize $p_M(\boldsymbol{d} \mid I_L, I_R)$

Markov Random Field

Probability distribution on disparity field $d(x,y)$

$$p_P(d_{x,y}|\mathbf{d}) = p_P(d_{x,y}|\{d_{x',y'}, (x',y') \in \mathcal{N}(x,y)\})$$

$$p_P(\mathbf{d}) = \frac{1}{Z_P} e^{-E_P(\mathbf{d})}$$



$$E_P(\mathbf{d}) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y})$$

Enforces *smoothness* or *coherence* on field

Measurement model

Likelihood of intensity correspondence

$$p_M(I_L, I_R|\mathbf{d}) = \frac{1}{Z_M} e^{-E_0(x,y;d)}$$

$$E_0(x,y;d) = \rho(I_L(x'+d,y') - I_R(x',y'))$$

Corresponds to Gaussian noise for quadratic ρ

MAP estimate

Maximize posterior likelihood

$$\begin{aligned} E(\mathbf{d}) &= -\log p(\mathbf{d}|I_L, I_R) \\ &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y}) \\ &\quad + \sum_{x,y} \rho_M(I_L(x + d_{x,y}, y) - I_R(x,y)) \end{aligned}$$

Equivalent to *regularization* (energy minimization with smoothness constraints)

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

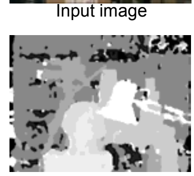
Admits a wider variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)
- mean-field optimization
- graph theoretic (actually deterministic) [Zabih]
- [loopy] belief propagation
- large stochastic flips [Swendsen-Wang]

Depth Map Results


Input image


Sum Abs Diff


Mean field


Graph cuts

Traditional stereo

Advantages:

- works very well in non-occluded regions

Disadvantages:

- restricted to two images (not)
- gets confused in occluded regions
- can't handle *mixed pixels*

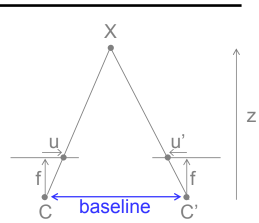
Multi-View Stereo

…rest of this material not covered in this lecture…

Stereo Reconstruction

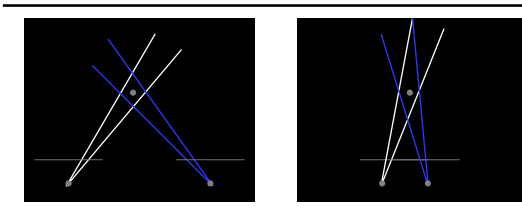
Steps

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth



$$disparity = u - u' = \frac{baseline * f}{z}$$

Choosing the Baseline



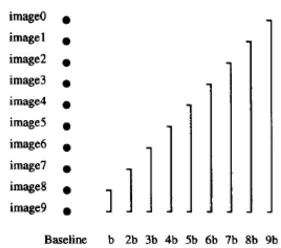
Large Baseline Small Baseline

What's the optimal baseline?

- Too small: large depth error
- Too large: difficult search problem

Effect of Baseline on Estimation

Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.



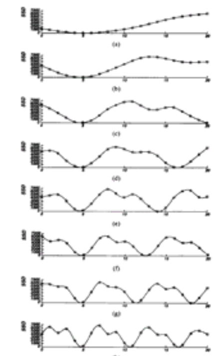


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

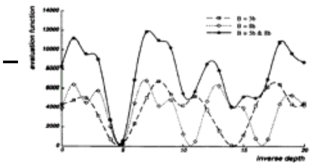


Fig. 6. Combining two stereo pairs with different baselines.

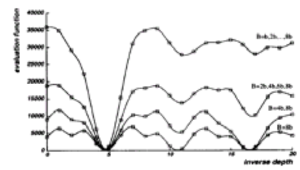


Fig. 7. Combining multiple baseline stereo pairs.

Multibaseline Stereo

Basic Approach

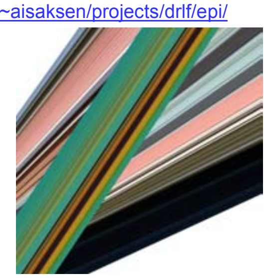
- Choose a reference view
- Use your favorite stereo algorithm BUT
  - replace two-view SSD with SSD over all baselines

Limitations

- Must choose a reference view
- Visibility: select which frames to match [Kang, Szeliski, Chai, CVPR'01]

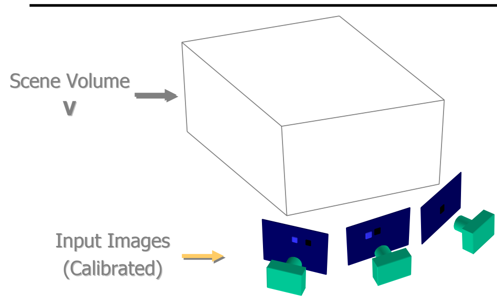
Epipolar-Plane Images [Bolles 87]

http://www.graphics.lcs.mit.edu/~aisaksen/projects/drlf/epi/



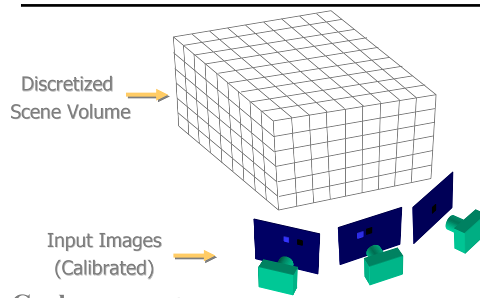
Lesson: Beware of ***occlusions***

Volumetric Stereo



Goal: **Determine transparency, radiance of points in V**

Voxel Coloring

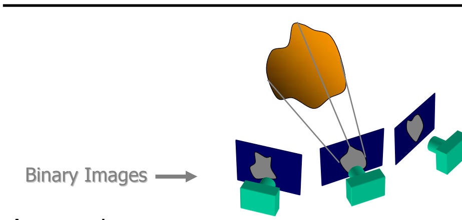


Goal: **Assign RGBA values to voxels in V**
photo-consistent **with images**

Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Martin 81, Szeliski 93]
2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. General Case
   - Space carving [Kutulakos & Seitz 98]

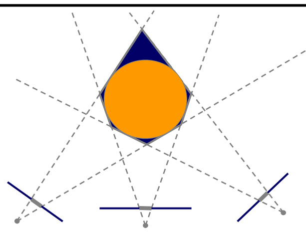
Reconstruction from Silhouettes



Approach:

- *Backproject* each silhouette
- Intersect backprojected volumes

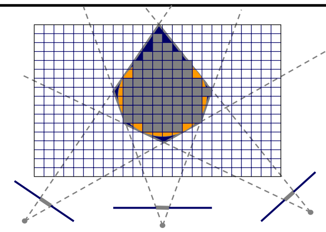
Volume Intersection



Reconstruction Contains the True Scene

- But is generally not the same
- In the limit get *visual hull*

Voxel Volume Intersection



Color voxel black if in silhouette in every image

- $O(MN^3)$, for M images, N^3 voxels
- Don't have to search 2^{N^3} possible scenes!

Properties of Volume Intersection

Pros

- Easy to implement, fast
- Accelerated via octrees [Szeliski 1993]

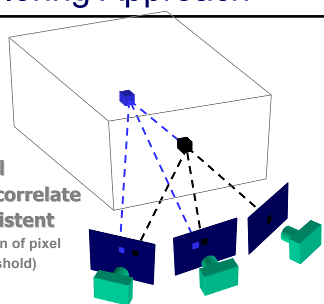
Cons

- No concavities
- Reconstruction is not photo-consistent
- Requires identification of silhouettes

Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Martin 81, Szeliski 93]
2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. General Case
   - Space carving [Kutulakos & Seitz 98]

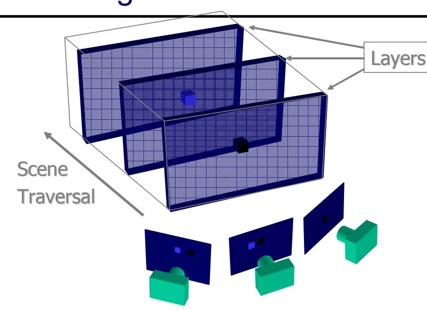
Voxel Coloring Approach



1. **Choose voxel**
2. **Project and correlate**
3. **Color if consistent**
   (standard deviation of pixel colors below threshold)

Visibility Problem: **in which images is each voxel visible?**

Depth Ordering: visit occluders first!



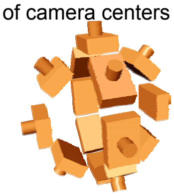
Condition: **depth order is *view-independent***

Compatible Camera Configurations

Depth-Order Constraint

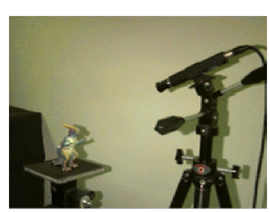
- Scene outside convex hull of camera centers

Inward-Looking
cameras above scene



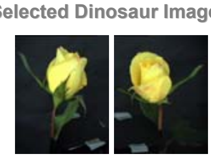
Outward-Looking
cameras inside scene

Calibrated Image Acquisition



Calibrated Turntable

Selected Dinosaur Images



Selected Flower Images

Voxel Coloring Results (Video)

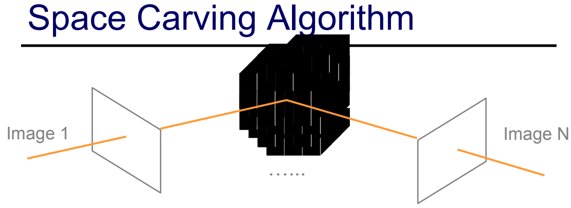
Dinosaur Reconstruction
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Flower Reconstruction
70 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Martin 81, Szeliski 93]
2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. General Case
   - Space carving [Kutulakos & Seitz 98]

Space Carving Algorithm



Space Carving Algorithm
- Choose a voxel on the current surface
- Initialize to a volume V containing the true scene
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence

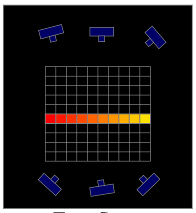
Space Carving Algorithm

The Basic Algorithm is Unwieldy
- Complex update procedure

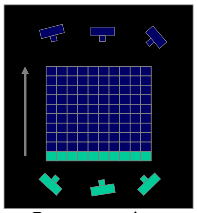
Alternative: Multi-Pass Plane Sweep
- Efficient, can use texture-mapping hardware
- Converges quickly in practice
- Easy to implement

Multi-Pass Plane Sweep

- Sweep plane in each of 6 principle directions
- Consider cameras on only one side of plane
- Repeat until convergence



True Scene



Reconstruction

Results: African Violet

Input Image (1 of 45)

Reconstruction

Reconstruction

Reconstruction

Results: Hand

Input Image
(1 of 100)

Views of Reconstruction

Other Approaches

Level-Set Methods ***[Faugeras & Keriven 1998]***

- Evolve implicit function by solving PDE's

Transparency and Matting ***[Szeliski & Golland 1998]***

- Compute voxels with alpha-channel

Max Flow/Min Cut ***[Roy & Cox 1998]***

- Graph theoretic formulation

Mesh-Based Stereo ***[Fua & Leclerc 95]***

- Mesh-based but similar consistency formulation

Virtualized Reality ***[Narayan, Rander, Kanade 1998]***

- Perform stereo 3 images at a time, merge results

Summary

Applications
Image rectification
Matching criteria
Local algorithms (aggregation & diffusion)
Optimization algorithms

- energy (cost) formulation & Markov Random Fields
- mean-field; dynamic programming; stochastic; graph algorithms

Multi-View stereo

- visibility, occlusion-ordered sweeps

Bibliography

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1):7-42, May 2002.

R. Szeliski. Stereo algorithms and representations for image-based rendering. In British Machine Vision Conference (BMVC'99), volume 2, pages 314-328, Nottingham, England, September 1999.

G. M. Nielson, Scattered Data Modeling, IEEE Computer Graphics and Applications, 13(1), January 1993, pp. 60-70.

S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In CVPR'2001, vol. I, pages 103-110, December 2001.

Y. Boykov, O. Veksler, and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, Unpublished manuscript, 2000.

A.F. Bobick and S.S. Intille. Large occlusion stereo. International Journal of Computer Vision, 33(3), September 1999. pp. 181-200

D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

Bibliography

Volume Intersection

- Martin & Aggarwal, “Volumetric description of objects from multiple views”, Trans. Pattern Analysis and Machine Intelligence, 5(2), 1991, pp. 150-158.
- Szeliski, “Rapid Octree Construction from Image Sequences”, Computer Vision, Graphics, and Image Processing: Image Understanding, 58(1), 1993, pp. 23-32.

Voxel Coloring and Space Carving

- Seitz & Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring”, Proc. Computer Vision and Pattern Recognition (CVPR), 1997, pp. 1067-1073.
- Seitz & Kutulakos, “Plenoptic Image Editing”, Proc. Int. Conf. on Computer Vision (ICCV), 1998, pp. 17-24.
- Kutulakos & Seitz, “A Theory of Shape by Space Carving”, Proc. ICCV, 1998, pp. 307-314.

Bibliography

Related References

- Bolles, Baker, and Marimont, “Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion”, International Journal of Computer Vision, vol 1, no 1, 1987, pp. 7-55.
- Faugeras & Keriven, “Variational principles, surface evolution, PDE's, level set methods and the stereo problem", IEEE Trans. on Image Processing, 7(3), 1998, pp. 336-344.
- Szeliski & Golland, “Stereo Matching with Transparency and Matting”, Proc. Int. Conf. on Computer Vision (ICCV), 1998, 517-524.
- Roy & Cox, “A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem”, Proc. ICCV, 1998, pp. 492-499.
- Fua & Leclerc, “Object-centered surface reconstruction: Combining multi-image stereo and shading", International Journal of Computer Vision, 16, 1995, pp. 35-56.
- Narayanan, Rander, & Kanade, “Constructing Virtual Worlds Using Dense Stereo”, Proc. ICCV, 1998, pp. 3-10.