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

Z-keying: mix live and synthetic

Takeo Kanade, CMU (Stereo Machine)
**View Interpolation**

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

**More view interpolation**

Spline-based depth map

![input depth image novel view](image)

[Szeliski & Kang '95]

**View Morphing**

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

**Video view interpolation**

![Video view interpolation](image)
Virtualized Reality™

[Takeo Kanade et al., CMU]
- collect video from 50+ stream
- reconstruct 3D model sequences

- steerable version used for SuperBowl XXV “eye vision”

Real-time stereo

Nomad robot searches for meteorites in Antarctica
http://www.frc.n.cmu.edu/projects/meteorobot/index.html

Used for robot navigation (and other tasks)
- Software-based real-time stereo techniques

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

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 image pair

Anaglyphs

http://www.rainbowsymphony.com/freestuff.html
(Wikipedia for images)

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

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

[Loop and Zhang, CVPR’99]
**Rectification**

**BAD!**

**GOOD!**

**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)

**Your basic stereo algorithm**

- For each epipolar line
  - For each pixel in the left image
    - compare with every pixel on same epipolar line in right image
    - pick pixel with minimum match cost
  - This should look familiar...

Improvement: match **windows**
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

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]

Stereo: certainty modeling

Compute certainty map from correlations
**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

\[ E_0(x, y; d) = \rho(I_L(x', y') - I_R(x', y')) \]

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 ⇒ 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

\[
E(x, y, d) \leftarrow (1 - 4\lambda)E(x, y, d) + \lambda \sum_{(k,l) \in N_4} E(x+k, y+l, d) + \beta(E_0(x, y, d) - E(x, y, d))
\]

**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

\[
\begin{align*}
E_{\text{total}}(d) &= E_{\text{data}}(d) + \lambda E_{\text{smoothness}}(d) \\
E_{\text{data}}(d) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{membrane}}(d) &= \sum_{x,y} (d_{x,y} - d_{x-1,y})^2 \\
E_{\text{thin plate}}(d) &= \sum_{x,y} (2d_{x,y} - d_{x-1,y} - d_{x+1,y})^2
\end{align*}
\]
Relaxation

How can we get the best solution?
Differentiate energy function, set to 0

\[ \frac{\partial E}{\partial d_{x,y}} = 2(d_{x,y} - z_{x,y}) + 2\lambda(2d_{x,y} - d_{x-1,y} - d_{x+1,y}) = 0 \]

\[ d_{x,y} = \frac{1}{1+2\lambda}(z_{x,y} + d_{x-1,y} + d_{x+1,y}) \]

Earliest application: WWII numerical simulations

Dynamic programming

Evaluate best cumulative cost at each pixel

\[ E_{\text{total}}(d) = E_{\text{data}}(d) + \lambda E_{\text{smoothness}}(d) \]

\[ E_{\text{data}}(d) = \sum_{x,y}(d_{x,y} - z_{x,y})^2 \]

\[ E_{\text{smoothness}}(d) = \sum_{x,y}|d_{x,y} - d_{x-1,y}| \]

Dynamic programming

1-D cost function

\[ E(d) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x, y; d) \]

\[ \tilde{E}(x, y, d) = E_0(x, y; d) + \min_{d'} \left( \tilde{E}(x-1, y, d') + \rho_P(d_{x,y} - d'_{x-1,y}) \right) \]
Dynamic programming

Disparity space image and min. cost path

![Diagram of disparity space image and min. cost path]

Dynamic programming

Sample result (note horizontal streaks)

[Intille & Bobick]

Dynamic programming

Can we apply this trick in 2D as well?

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

Graph cuts

Solution technique for general 2D problem

$$E_{\text{total}}(d) = E_{\text{data}}(d) + \lambda E_{\text{smoothness}}(d)$$

$$E_{\text{data}}(d) = \sum_{x,y} f_{x,y}(d_{x,y})$$

$$E_{\text{smoothness}}(d) = \sum_{x,y} \rho(d_{x,y} - d_{x-1,y}) + \sum_{x,y} \rho(d_{x,y} - d_{x,y-1})$$

![Graph cuts examples]

[a] original image  [b] observed image  [c] local min w.r.t. standard moves  [d] local min w.r.t. o-expansion moves
Graph cuts

\(\alpha - \beta\) swap
\(\alpha\) expansion
modify smoothness penalty based on edges
compute best possible match within integer disparity

---

Bayesian inference

Formulate as statistical inference problem
Prior model \(p_p(d)\)
Measurement model \(p_m(l_L, l_R \mid d)\)
Posterior model
\[ p_m(d \mid l_L, l_R) \propto p_p(d) \cdot p_m(l_L, l_R \mid d) \]
Maximum a Posteriori (MAP estimate):
maximize \(p_m(d \mid l_L, l_R)\)

---

Markov Random Field

Probability distribution on disparity field \(d(x,y)\)
\[ p_P(d_{x,y} \mid d) = p_P(d_{x,y} \mid \{d_{x',y'}, (x', y') \in N(x, y)\}) \]
\[ p_P(d) = \frac{1}{Z_P} e^{-E_P(d)} \]
\[ E_P(d) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y}) \]

Enforces smoothness or coherence on field
Measurement model

Likelihood of intensity correspondence

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

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

Corresponds to Gaussian noise for quadratic \( \rho \)

MAP estimate

Maximize posterior likelihood

\[ E(d) = -\log p(d|I_L, I_R) \]

\[ = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y}) \]

\[ + \sum_{x,y} \rho_M(I_L(x + d_{x,y}, y) - I_R(x, y)) \]

Equivalent to \textit{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
Stereo evaluation

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
Stereo Reconstruction

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

\[ \text{disparity} = u - u' = \frac{\text{baseline} \times f}{z} \]

Choosing the 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.
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/drfl/epi/

Lesson: Beware of occlusions

Active stereo with structured light

Camera 1
Projector
Camera 2
Project “structured” light patterns onto the object
- simplifies the correspondence problem

Spacetime Stereo

Li Zhang, Noah Snavely, Brian Curless, Steve Seitz

http://grail.cs.washington.edu/projects/stfaces/
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

Volume Intersection

Voxel Coloring and Space Carving