CS664 Computer Vision

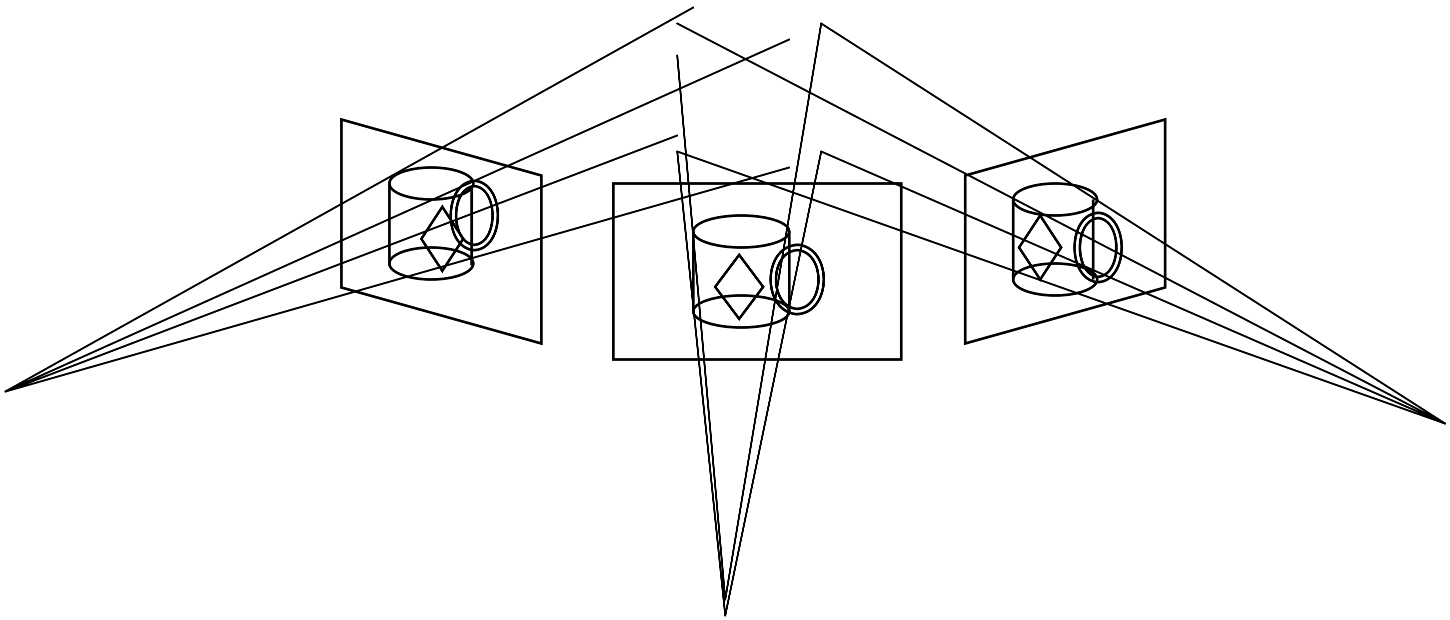
10. Stereo

Dan Huttenlocher

Cornell University
Faculty of Computing and Information Science

Stereo Matching

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



- Some applications

Face modeling

- From one stereo pair to a 3D head model

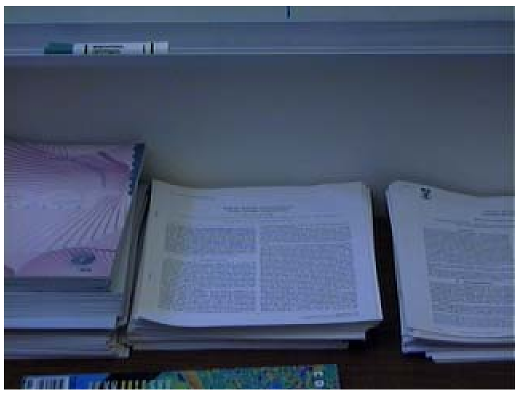
[Frederic Deverney, INRIA]

Z-keying: Mix Live and Synthetic

- Takeo Kanade, CMU (Stereo Machine)

View Interpolation

- Spline-based depth map



input

depth image

novel view

- [Szeliski & Kang ‘95]

Stereo Matching

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

- Some possible representations
  - Depth maps
  - Volumetric models
  - 3D surface models
  - Planar (or offset) layers

Stereo Matching

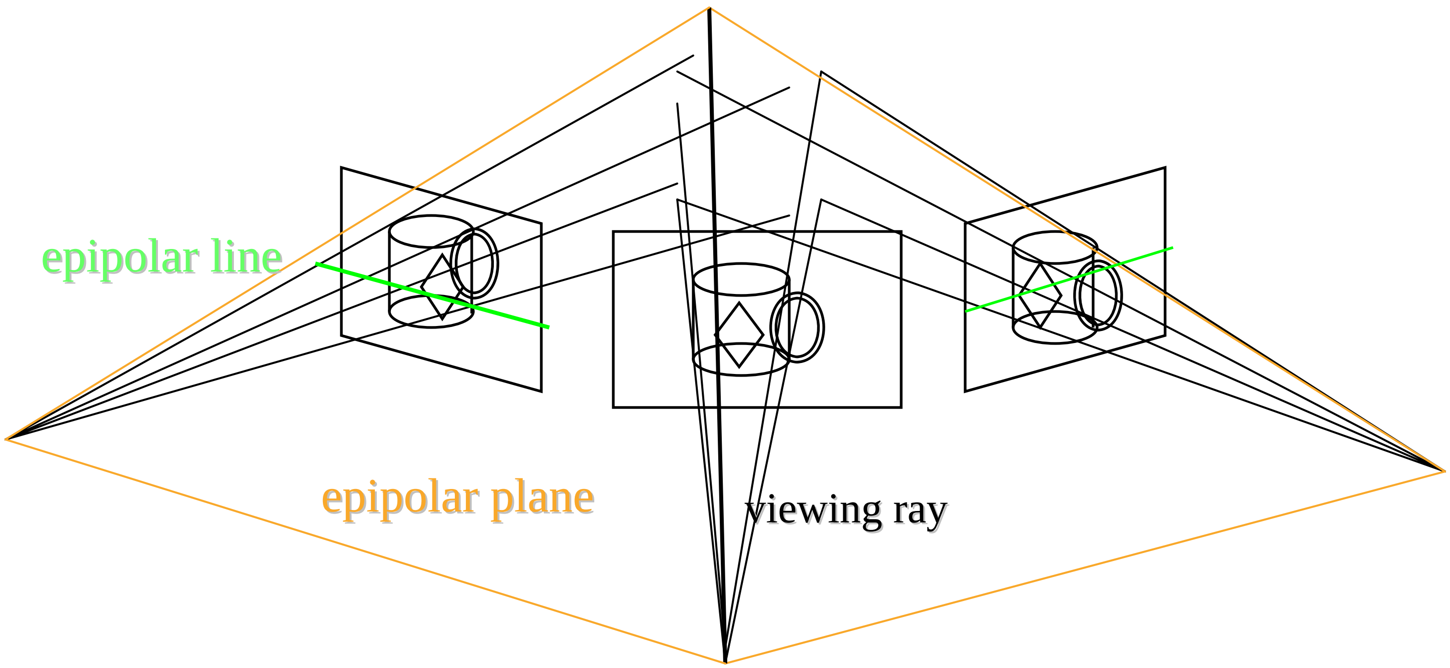
- Possible algorithms
  - Match “interest points” and interpolate
  - Match edges and interpolate
  - Match all pixels with windows (coarse-fine)
  - Optimization:
    - Iterative updating
    - Dynamic programming
    - Energy minimization (regularization, stochastic)
    - Graph algorithms

Outline

- Image rectification
- Matching criteria
- Local algorithms (aggregation)
  - Iterative updating
- Optimization algorithms:
  - Energy (cost) formulation & Markov Random Fields
  - Mean-field, stochastic, and graph algorithms

Stereo: epipolar geometry

- Match features along epipolar lines



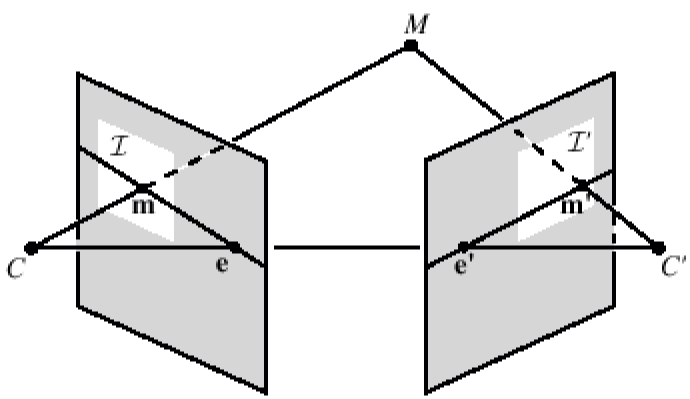
Stereo: Recall Epipolar geometry

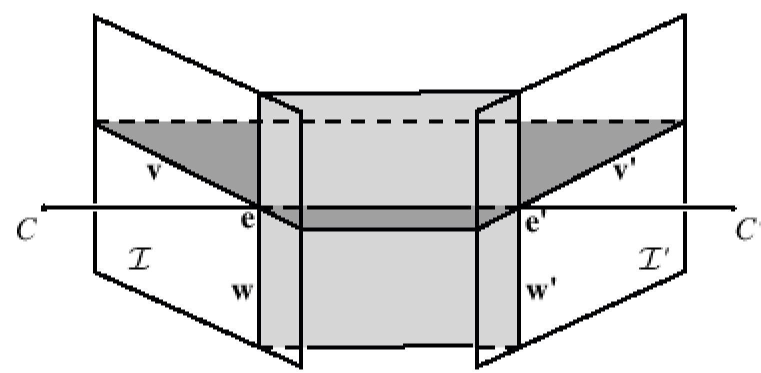
- For *two* images (or images with collinear camera centers), can find epipolar lines
- Epipolar lines are the projection of the *pencil* of planes passing through the centers

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

Rectification

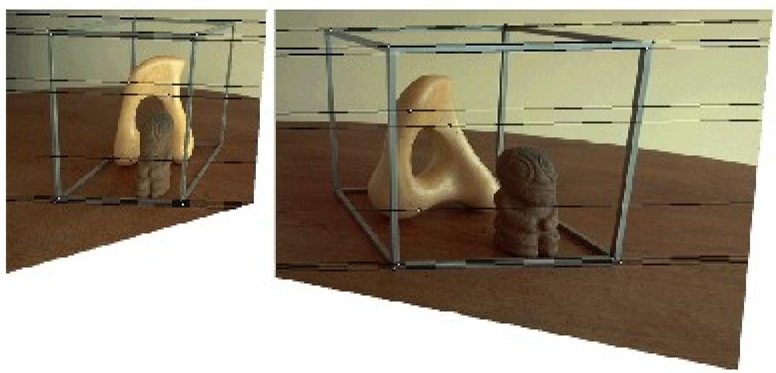
- Project each image onto same plane, which is parallel to the epipole
- Resample lines (and shear/stretch) to place lines in correspondence, and minimize distortion





Rectification

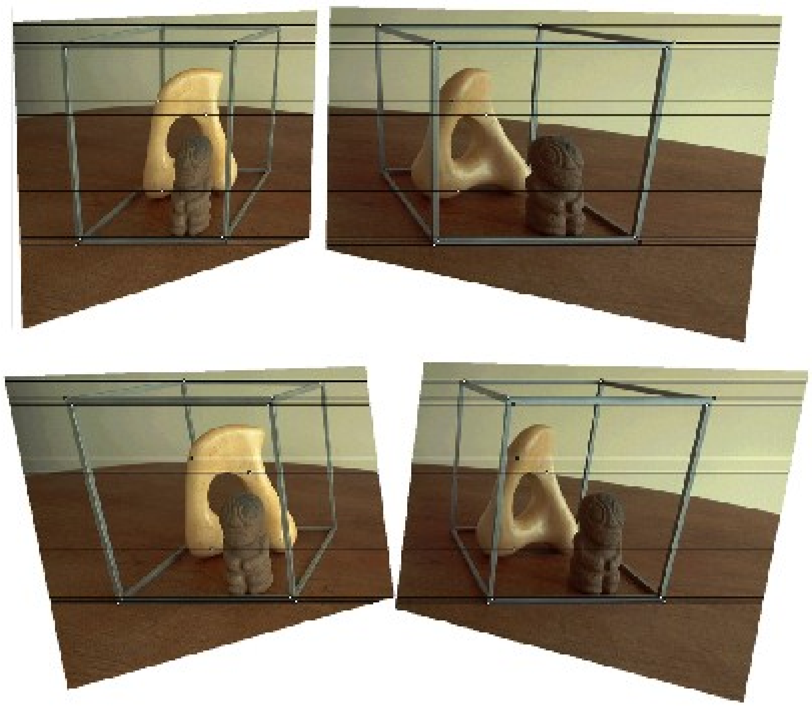
(a) Original image pair overlayed with several epipolar lines.



(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

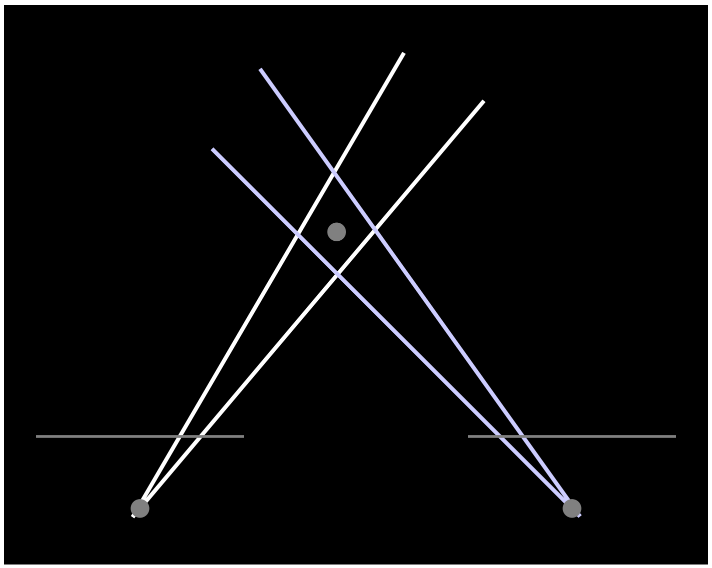


(c) Image pair transformed by the similarity $\mathbf{H}_r$ and $\mathbf{H}'_r$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

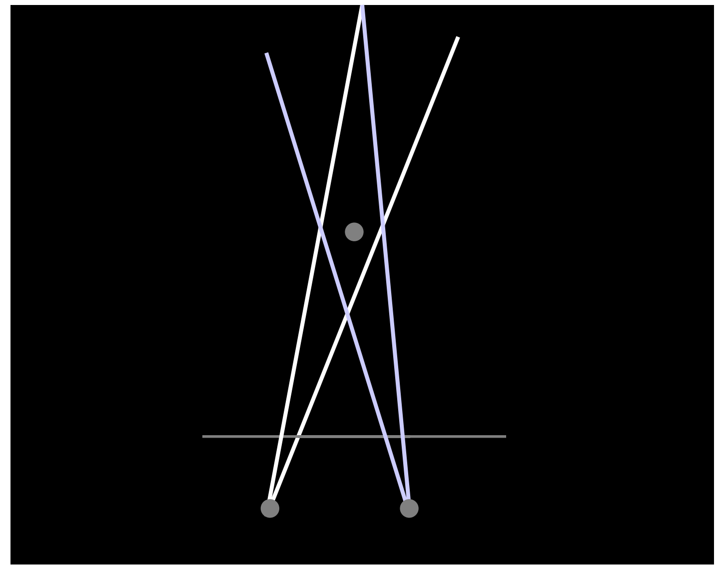
(d) Final image rectification after shearing transform $\mathbf{H}_s$ and $\mathbf{H}'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!

Choosing the Baseline



Large Baseline



Small Baseline

- What's the optimal baseline?
  - Too small: large depth error
  - Too large: difficult search problem

Matching Criteria

- Raw pixel values (correlation)
- Band-pass filtered images [Jones & Malik 92]
- “Corner” like features [Zhang, ...]
- Edges [Many 1980’s methods...]
- Gradients [Seitz 89; Scharstein 94]
- Rank statistics [Zabih & Woodfill 94]
- Slanted surfaces [Birchfield & Tomasi 99]

Finding Correspondences

- Apply feature matching criterion (e.g., correlation) at *all* pixels simultaneously
- Search only over epipolar lines (many fewer candidate positions)


Block Based Matching

- How to determine correspondences?
  - *Block matching* or *SSD* (sum squared differences)

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

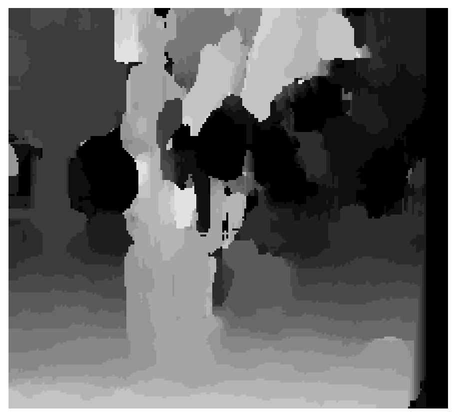
d is the *disparity* (horizontal motion)

- How big should neighborhood be?

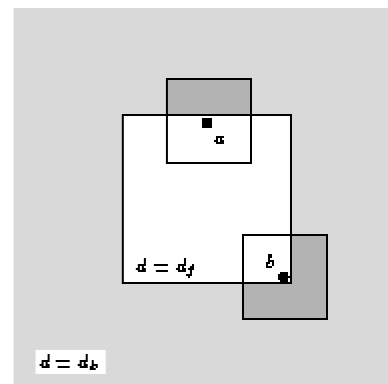
Effects of Block Size

- Smaller neighborhood: more details
- Larger neighborhood: fewer isolated mistakes

w = 3

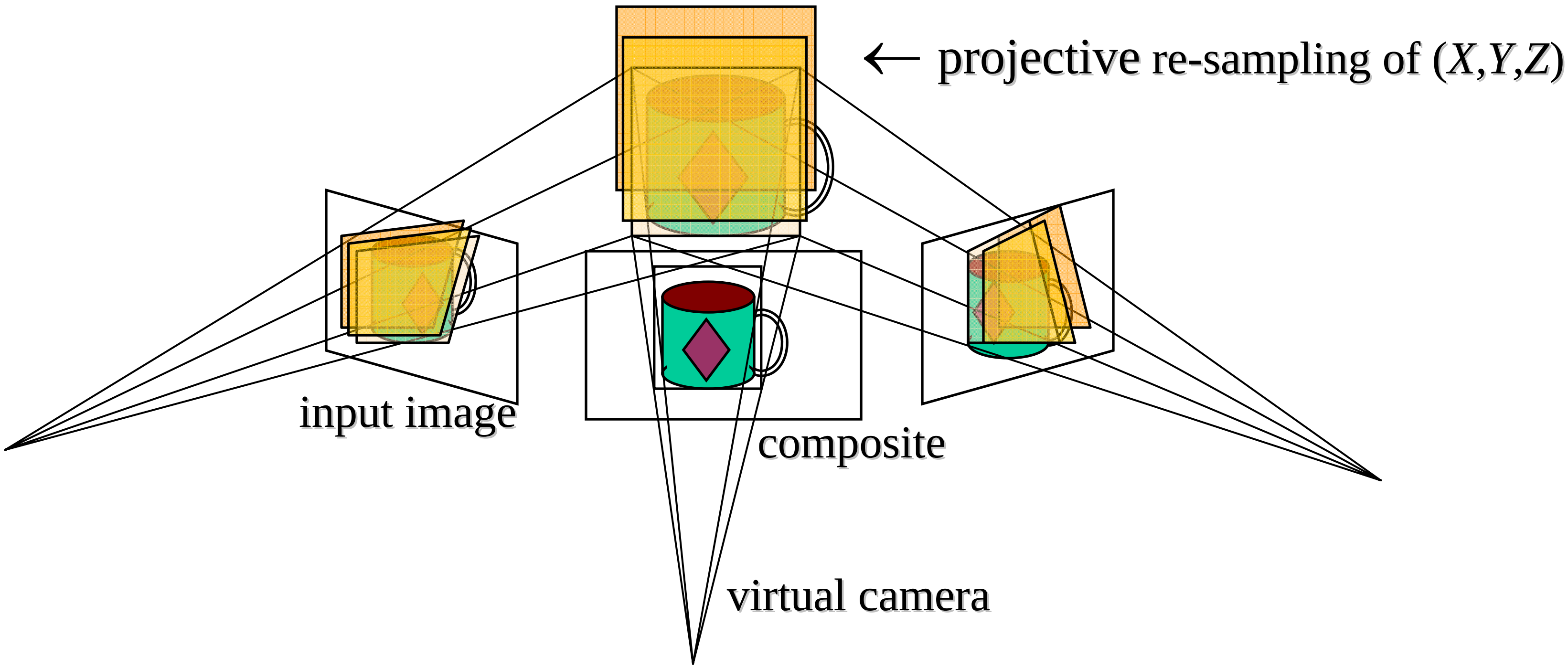


w = 20



Plane Sweep Stereo

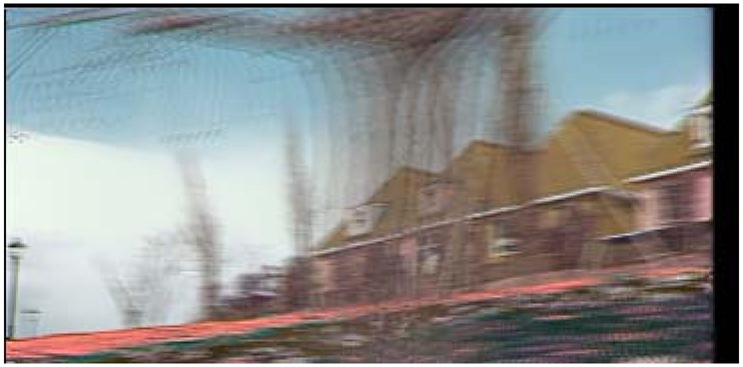
- Sweep family of planes through volume



  – each plane defines an image ⇒ composite homography

Plane Sweep Stereo

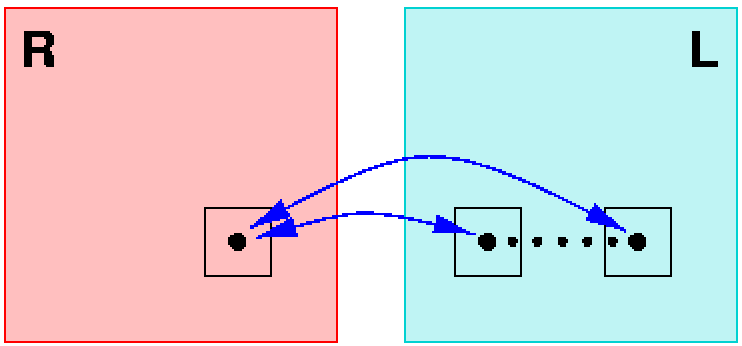
- For each depth plane
  - Compute composite (mosaic) image — *mean*

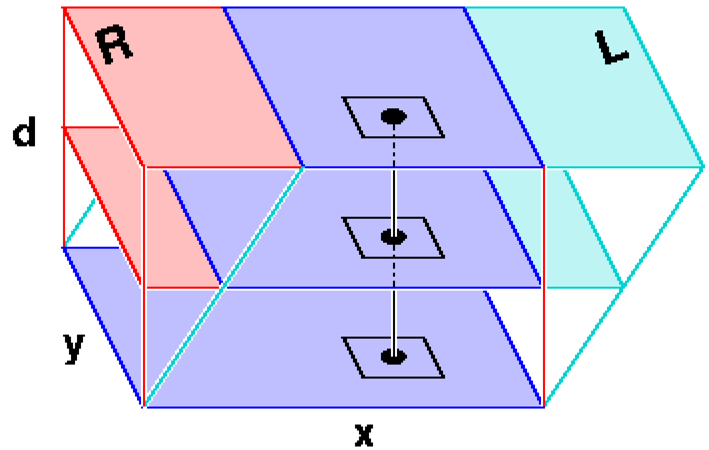


  - Compute error image — *variance*
  - Convert to confidence and aggregate spatially
- Select winning depth at each pixel

Plane Sweep Stereo

- Re-order (pixel / disparity) evaluation loops





for every pixel,
  for every disparity
    compute cost

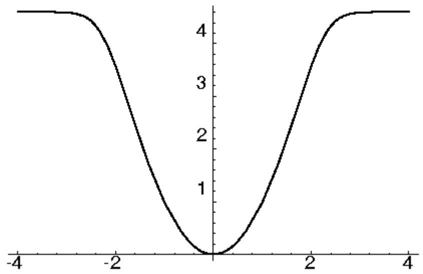
for every disparity
  for every pixel
    compute cost

Stereo Matching Framework

- For every disparity, compute *raw* matching costs

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

- Robust cost functions
  - Occlusions, other outliers



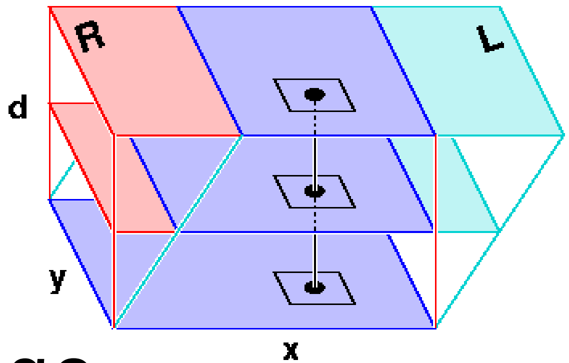
- Combine with spatial coherence or consistency

Stereo Matching Framework

- Aggregate costs spatially

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

- Can use *box filter* (efficient moving average implementation)



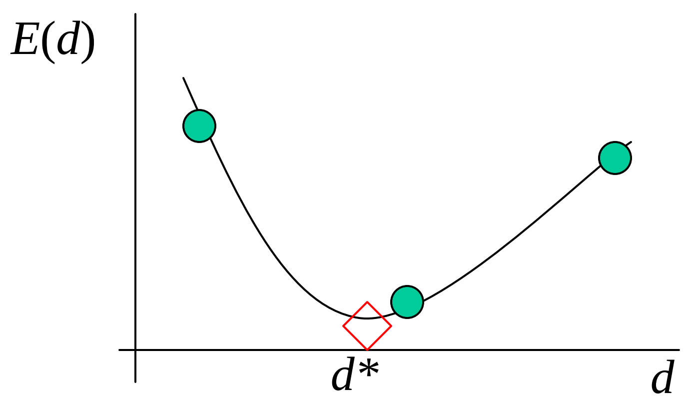
- Can also use weighted average, [non-linear] diffusion...

Stereo Matching Framework

- Choose winning disparity at each pixel

$$d(x,y) = \arg\min_{d} E(x,y;d)$$

- Interpolate to *sub-pixel* accuracy



Traditional Stereo Matching

- Advantages:
  - Detailed surface estimates
  - Fast algorithms using moving averages
  - Sub-pixel disparity estimates and confidence
- Limitations:
  - Narrow baseline $\Rightarrow$ noisy estimates
  - Fails in textureless areas
  - Gets confused near occlusion boundaries

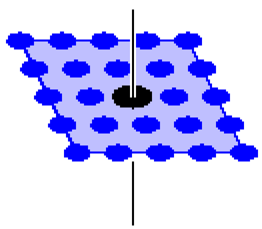
Stereo with Non-Linear Diffusion

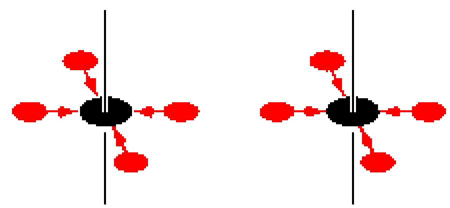
- Problem with traditional approach:
  – Gets confused near discontinuities
- Another approach:
  – Use iterative (non-linear) aggregation to obtain better estimate
  – Turns out to be provably equivalent to mean-field estimate of Markov Random Field

Linear Diffusion

- Average energy with neighbors + starting value

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

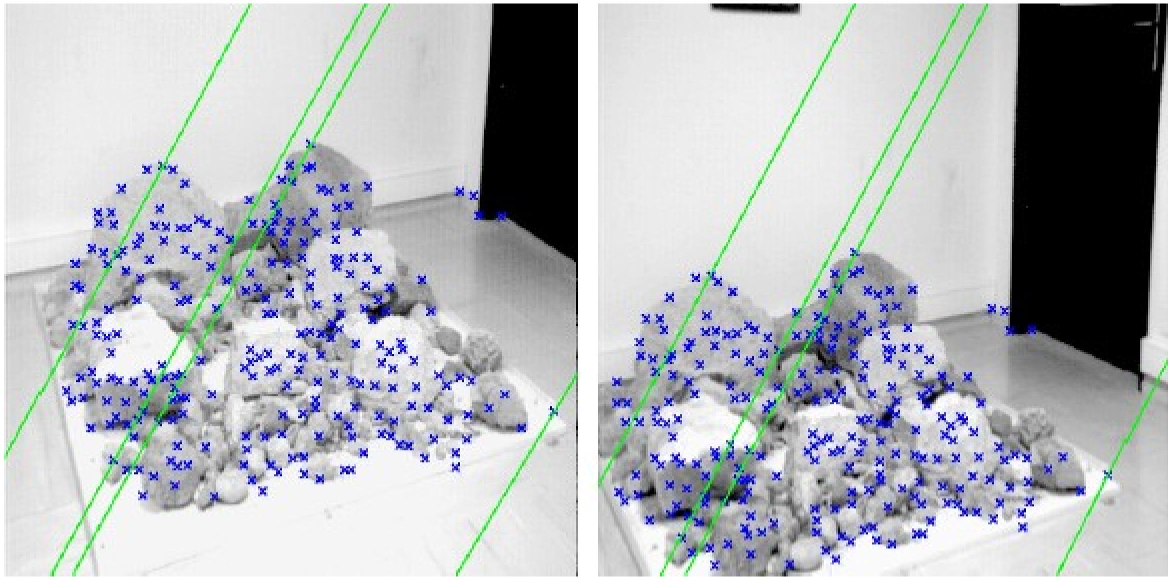




- window diffusion

Feature-Based Stereo

- Match “corner” (interest) points



- Interpolate complete solution

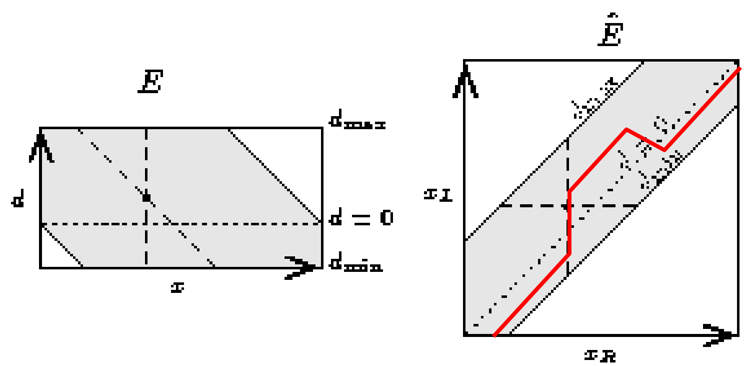
Data Interpolation

- 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

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

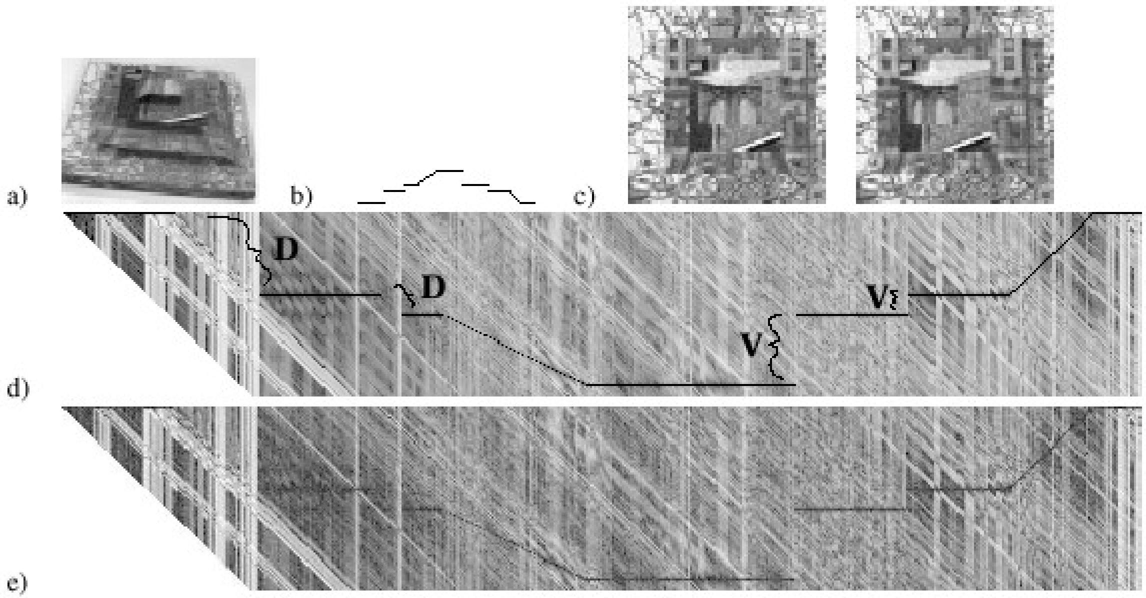


Fig. 4. This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation DSI representation for the image 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]

a)

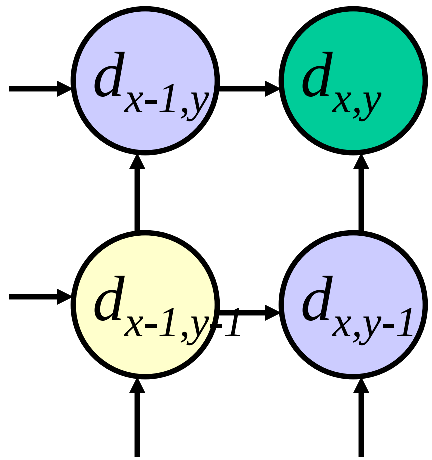
b)

c)

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?

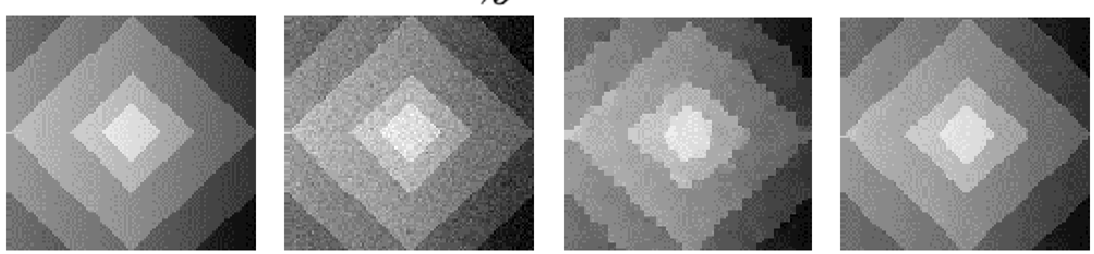


No: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

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

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):

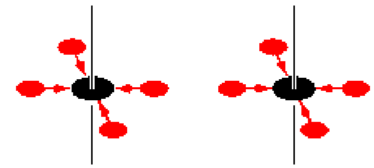
  maximize $p_M(\boldsymbol{d} \mid I_L, I_R)$

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