COSC160: Data Structures

Jeremy Bolton, PhD
Assistant Teaching Professor

GEORGETOWN UNIVERSITY

Outline

- Stereopsis
  - Motivation
  - Problems
  - Image rectification
- Matching criteria
  - Energy Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
    - Graph Cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

Outline

- Stereopsis
  - Motivation
  - Problems
  - Image rectification
- Matching criteria
  - Energy Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
    - Graph Cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

Goals of Stereo Matching

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

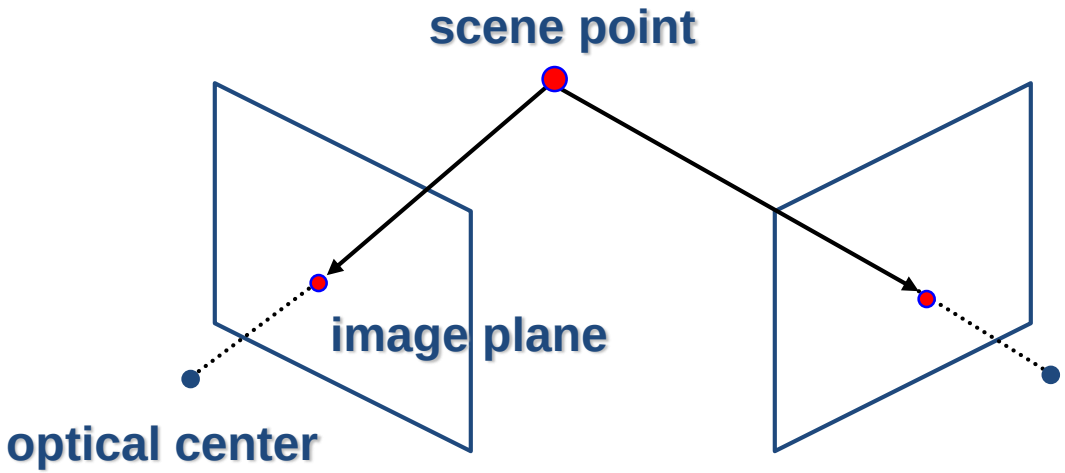
- What are some possible representations?
  - depth maps
  - volumetric models
  - 3D surface models

Stereo

- What is gained from stereo?
  - Assuming we know …
    1. Geometric details (extrinsic and intrinsic parameters), and
    2. Corresponding pixels in each image,

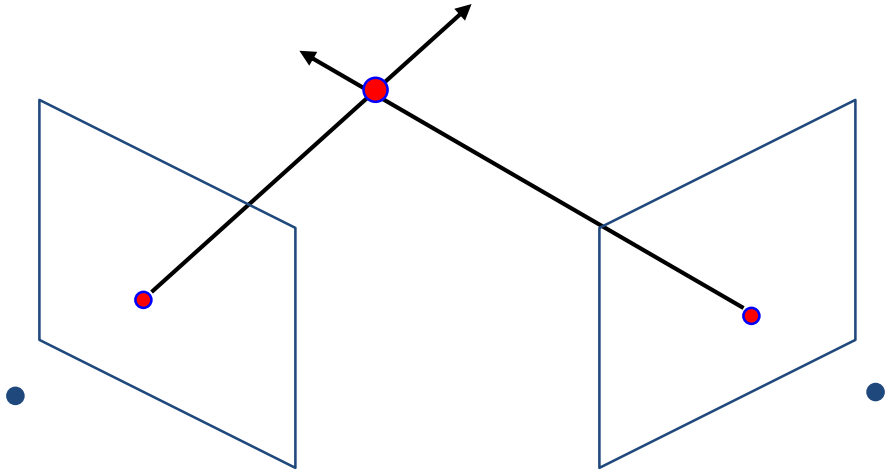
    We can directly extract depth information.

- What could go wrong?
  - Geometric details often unknown
  - Finding corresponding pixels is difficult!
  - Error: Vectors from corresponding image coordinates rarely intersect (perfectly).
    - There are many sources of unknowns and error.



Stereo

Binocular Reconstruction: Estimating depth from stereo



Common Approaches:

1. Simple Triangulation: Gives reconstruction as intersection of two rays
   - » Requires ***camera origins*** and pixel correspondences (enough information to characterize both rays)
2. Inverse Transformation Approach: Project both image coordinates to the 3-D scene using Each Cameras projection matrix
   - » Requires pixel correspondence and ***projection matrix*** of each camera (intrinsic and extrinsic parameters)

Triangulation

- Recall, camera projection matrix P, determines homography from point in 3-d scene to image plane.
- Conceptually The reverse map projects a point from the image plane, to a line in 3-d
- Since these vectors rarely intersect in practice, it is common to find the point nearest to the two lines

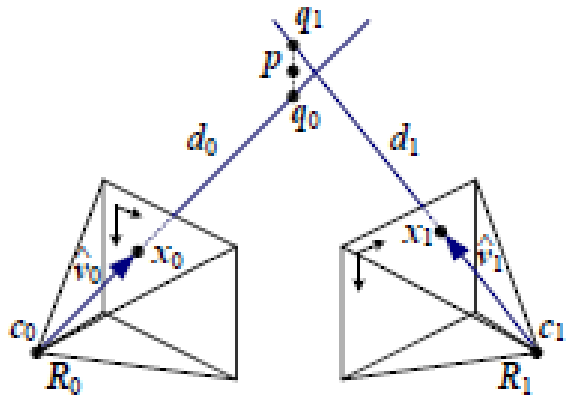
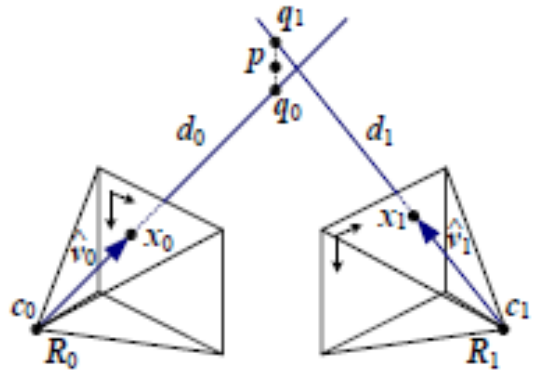


Figure 7.2 3D point triangulation by finding the point p that lies nearest to all of the optica rays $c_j + d_j \hat{v}_j$.

GEORGETOWN UNIVERSITY

Triangulation

- The ray v can be determined using the information contained in Camera Matrix or simply using two points c_j and x_j
- The nearest point to p on ray v_j is q_j
- This point can be characterized by a distance and directional vector d_j v_j, and minimizes the following “error” (deviation from true p).
- The squared error r_j can be computed, and thus value for p that minimizes this error can be computed using a least squares projection



$$\|\boldsymbol{c}_j + d_j\hat{\boldsymbol{v}}_j - \boldsymbol{p}\|^2,$$

Figure 7.2 3D point triangulation by finding the point $\boldsymbol{p}$ that lies nearest to all of the optical rays $\boldsymbol{c}_j + d_j\hat{\boldsymbol{v}}_j$.

$$r_j^2 = \|(\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)(\boldsymbol{p} - \boldsymbol{c}_j)\|^2$$

Minimum is when d_j is projection of v_j onto $(p\text{-}c_j)$

$$\boldsymbol{p} = \left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)\right]^{-1}\left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)\boldsymbol{c}_j\right]$$

GEORGETOWN UNIVERSITY

Simple Triangulation Solution

- Observe: Here we solved for p without using the projection matrix. Only using camera centers c and vectors (corresponding to points x) v.

$$\boldsymbol{p} = \left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j \hat{\boldsymbol{v}}_j^T)\right]^{-1} \left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j \hat{\boldsymbol{v}}_j^T)\boldsymbol{c}_j\right]$$

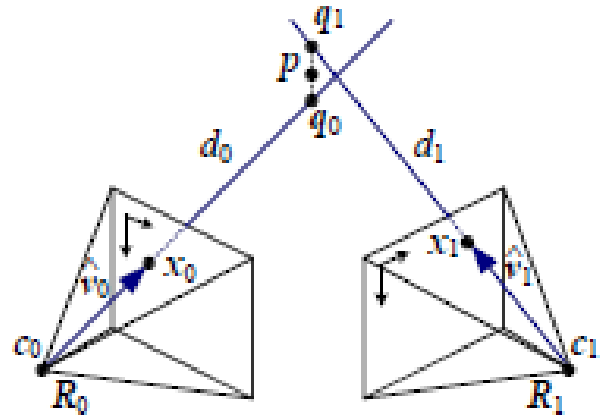


Figure 7.2 3D point triangulation by finding the point p that lies nearest to rays $c_j + d_j \hat{v}_j$.

GEORGETOWN UNIVERSITY

Algebraic Approach

- Algebraically, we can alternatively make use of the camera matrix P (if these parameters are known or have been learned during calibration)

- Given known pixel correspondences, (x,y) and (x,y) we can construct a series of linear equations and solve for the 3-D point (homogeneous) p = (X,Y,Z,W).

- Note there are 4 unknowns. How many images of the scene are needed to compute the location of point p?

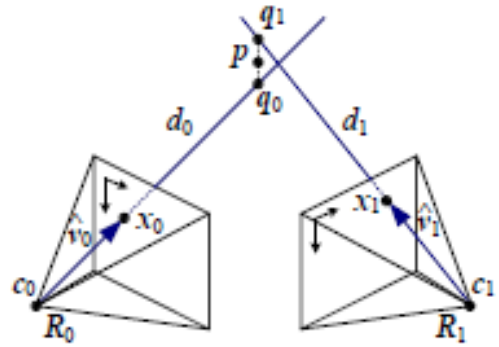


Figure 7.2 3D point triangulation by finding the point $\boldsymbol{p}$ that lies nearest to all of the op
rays $\boldsymbol{c}_j + d_j\hat{\boldsymbol{v}}_j$.

$$x_j = \frac{p_{00}^{(j)}X + p_{01}^{(j)}Y + p_{02}^{(j)}Z + p_{03}^{(j)}W}{p_{20}^{(j)}X + p_{21}^{(j)}Y + p_{22}^{(j)}Z + p_{23}^{(j)}W}$$

$$y_j = \frac{p_{10}^{(j)}X + p_{11}^{(j)}Y + p_{12}^{(j)}Z + p_{13}^{(j)}W}{p_{20}^{(j)}X + p_{21}^{(j)}Y + p_{22}^{(j)}Z + p_{23}^{(j)}W},$$

GEORGETOWN UNIVERSITY

Notes about reconstruction

- Requires information about cameras and pixel correspondences.

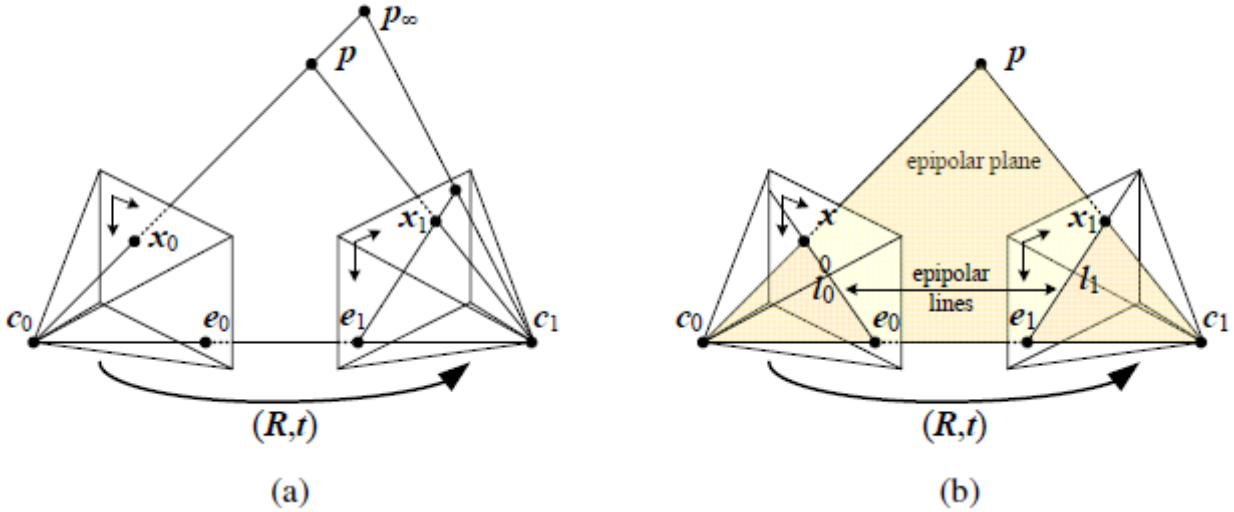
- Finding pixel correspondences is difficult in practice
  - Search over all pairs of pixels looking for similar pixel values?
  - Can we simplify this search?

Outline

- Stereopsis
  - Motivation
  - Problems
  - Image rectification
    - Essential Matrix
- Matching criteria
  - Energy Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
    - Graph Cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

Epipoles

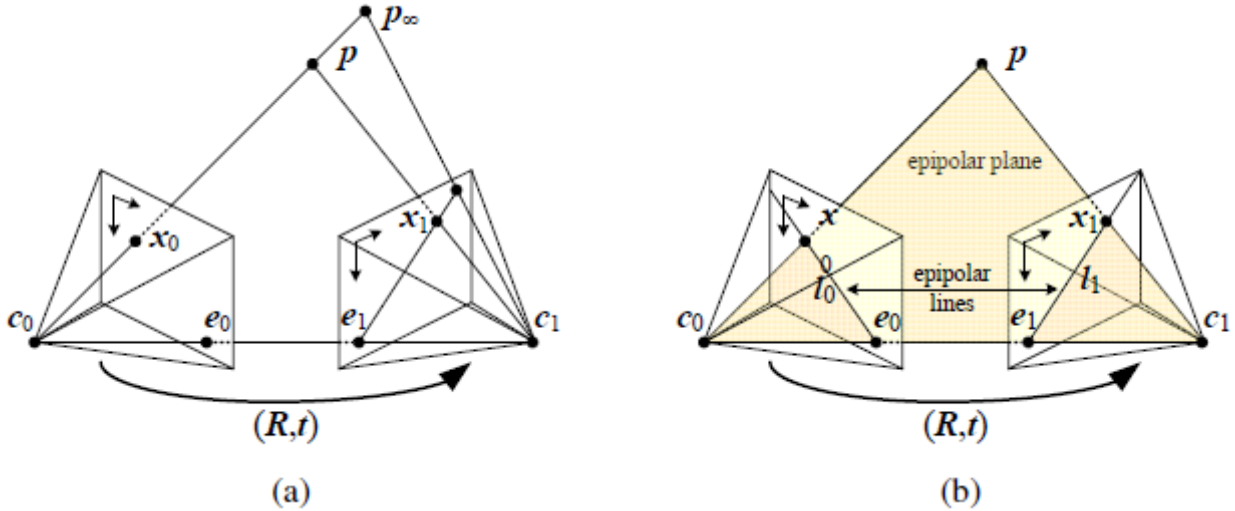
- Background
  - For *two* images (or images with collinear camera centers), can find epipolar lines
  - Epipolar lines are the projection of the *pencil* of planes passing through the centers



GEORGETOWN UNIVERSITY

Epipoles

- A pixel in one image x_0 exists at some point along a line in 3D.
- Thus its corresponding point in another image x_1 will be found somewhere along that lines projection into the second images plane (epipolar line segment).
  - Reduces search to find correspondence
- Points p, x_1, x_2, c_1, c_2, e_1, and e_2 lie on the epipolar plane.
  - x: coordinates in image plane
  - c: camera centers
  - e: epipoles



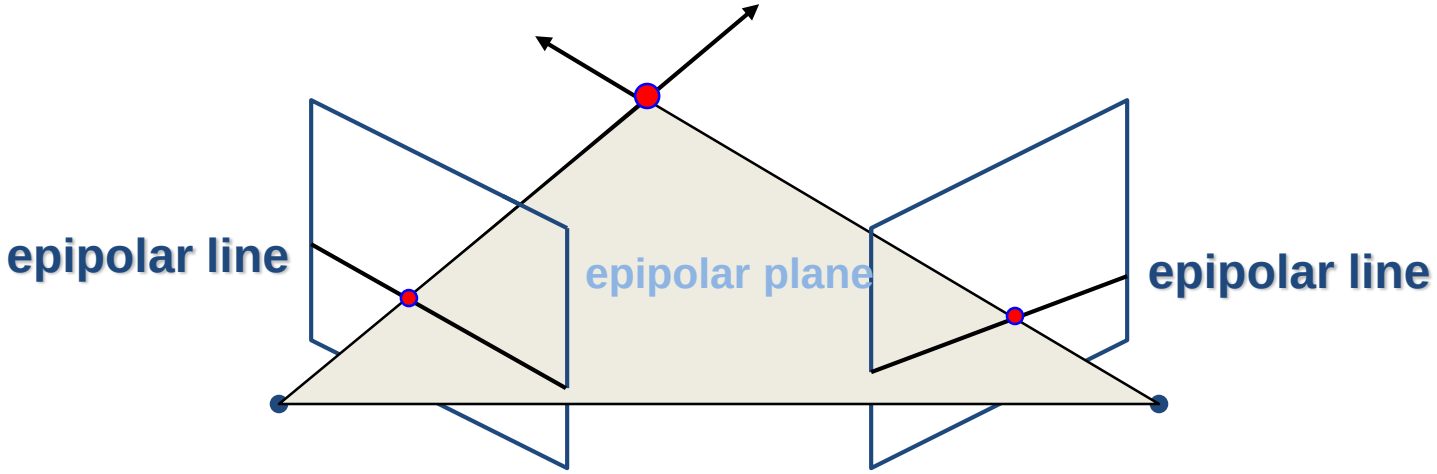
GEORGETOWN UNIVERSITY

Image Rectification

- **Rectification:** Warp the input images (perspective transformation) so that epipolar lines are horizontal
  - Simplifies stereo correspondence search along horizontal lines
  - Simplifies disparity (and therefore depth) computation

Stereo correspondence

- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point



Epipolar Constraint

- Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Image Warping for Image Rectification

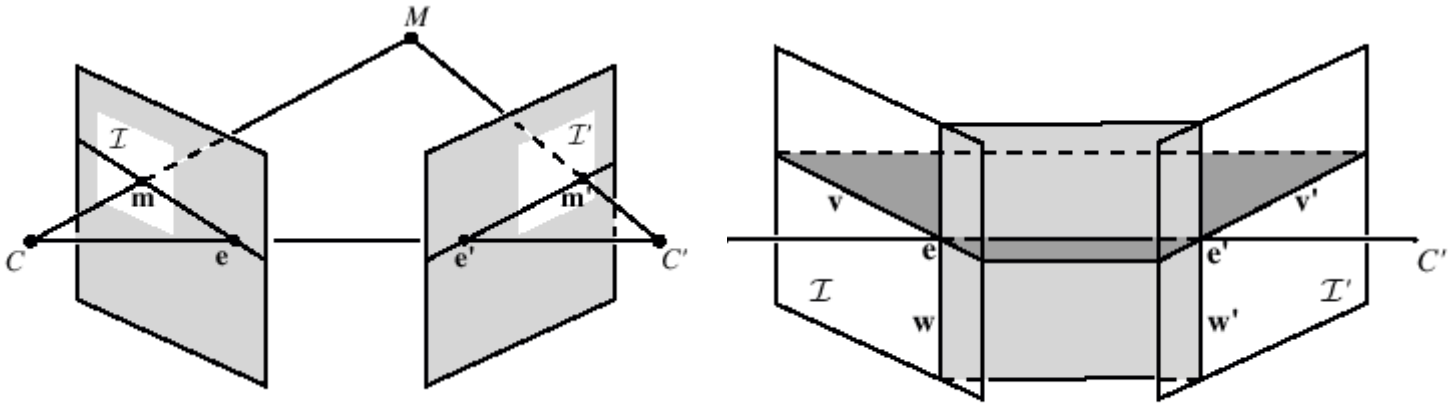
- Goal: Warp image(s) so corresponding epipolar lines are horizontal and/or co-align.
  1. Choose coordinate system to map both images … or for simplification… Choose one of the image coordinate systems (and warp the other image to this coordinate system, rather than warping both).
  2. Using knowledge of Camera Matrix, Homography, and Epipolar Constraint compute the appropriate Warping Matrix (Essential or Fundamental).
     1. Camera Motion Model: Instrinsics are known, working in normalized image space
     2. General model: accounts for intrinsics, scale, skew, …

     Simple Solution 1: Choose a plane parallel to the baseline.

     SS 2, Weak perspective assumption: Project onto planar scene
  3. Perform Warp.

Rectification

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



- [Loop and Zhang, CVPR'99]

Deriving the Epipolar Constraint

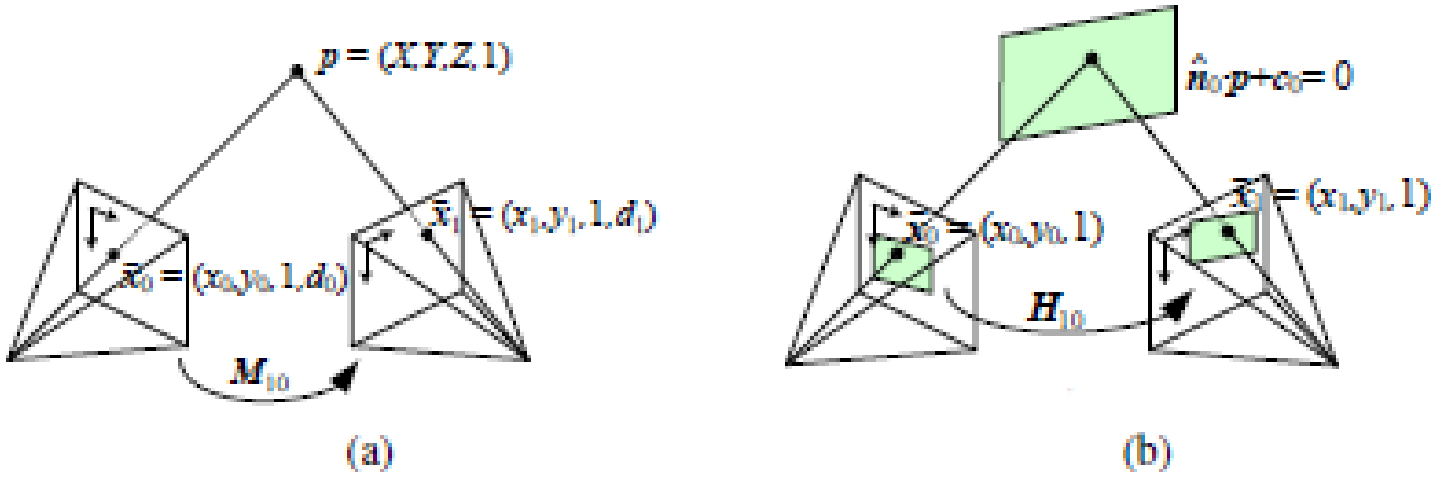
- Find projection matrix from one image to the other.

- Conceptually, given a camera matrix P, we can map a point to the image plane. The reverse transformation will map the image coordinate to a line (or point if we know the depth)

$$\bar{x}_0 \sim \tilde{K}_0 E_0 p = \tilde{P}_0 p.$$

- If the Camera matrix for the second image is known, we can then map to the second image

$$\bar{x}_1 \sim K_1 R_1 R_0^{-1} K_0^{-1} \bar{x}_0 = K_1 R_{10} K_0^{-1} \bar{x}_0$$



Deriving the Epipolar Constraint

- Assuming the relative camera locations can be encoded by a rotation and translation, if we re-orient the general coordinate system to coincide with that of one of the cameras, then this projection from one image to another is simplified.

$$d_1\hat{\boldsymbol{x}}_1 = \boldsymbol{p}_1 = \boldsymbol{R}\boldsymbol{p}_0 + \boldsymbol{t} = \boldsymbol{R}(d_0\hat{\boldsymbol{x}}_0) + \boldsymbol{t}, \qquad \hat{\boldsymbol{x}}_j = \boldsymbol{K}_j^{-1}\boldsymbol{x}_j$$

- Taking the cross product with t, then dot product with $\widehat{x_1}$ hat, yields

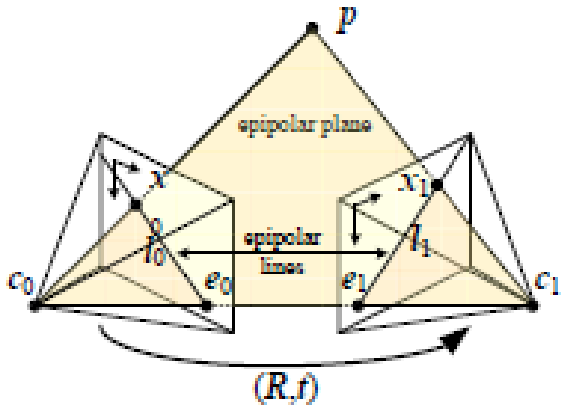
$$d_0\hat{\boldsymbol{x}}_1^T([\boldsymbol{t}]_\times\boldsymbol{R})\hat{\boldsymbol{x}}_0 = d_1\hat{\boldsymbol{x}}_1^T[\boldsymbol{t}]_\times\hat{\boldsymbol{x}}_1 = 0,$$

GEORGETOWN UNIVERSITY

Thus the Epipolar constraint

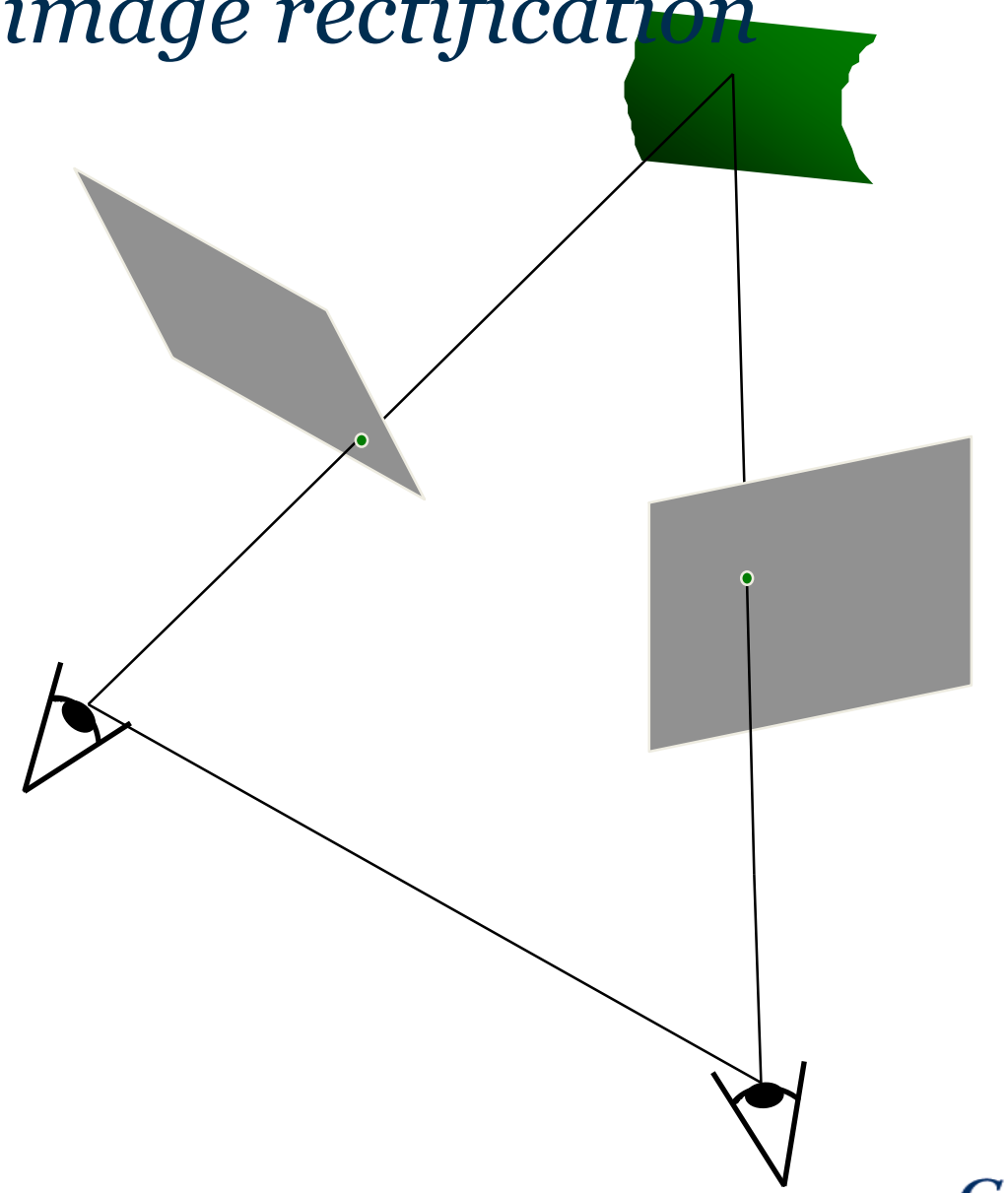
- E is referred to as the Essential Matrix

$$\hat{\boldsymbol{x}}_1^T \boldsymbol{E}\, \hat{\boldsymbol{x}}_0 = 0, \qquad \boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R}$$



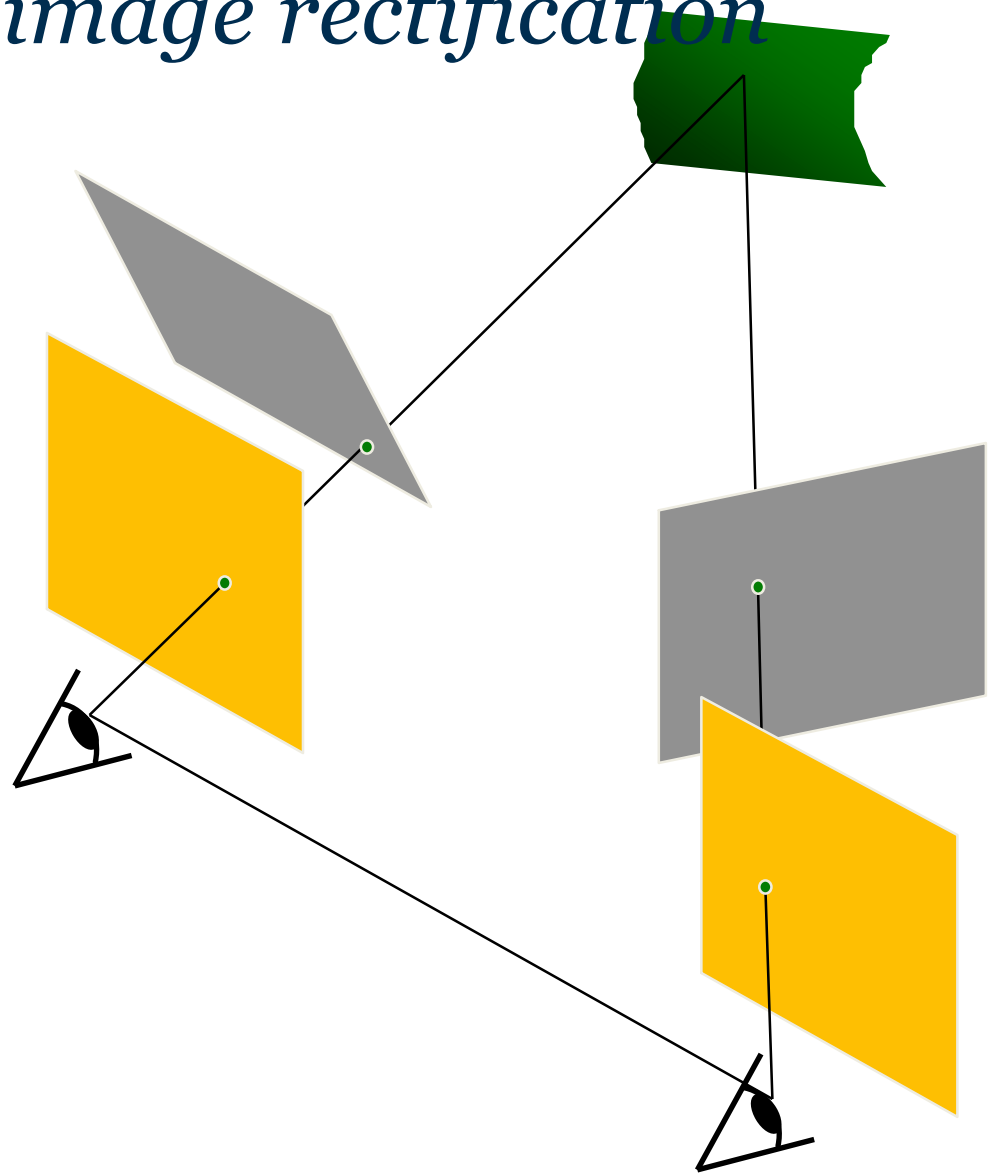
- Observe that E projects some point in image 1 to a line in image two. Give the equation above, E can be learned using point correspondences.

Stereo image rectification



Stereo image rectification

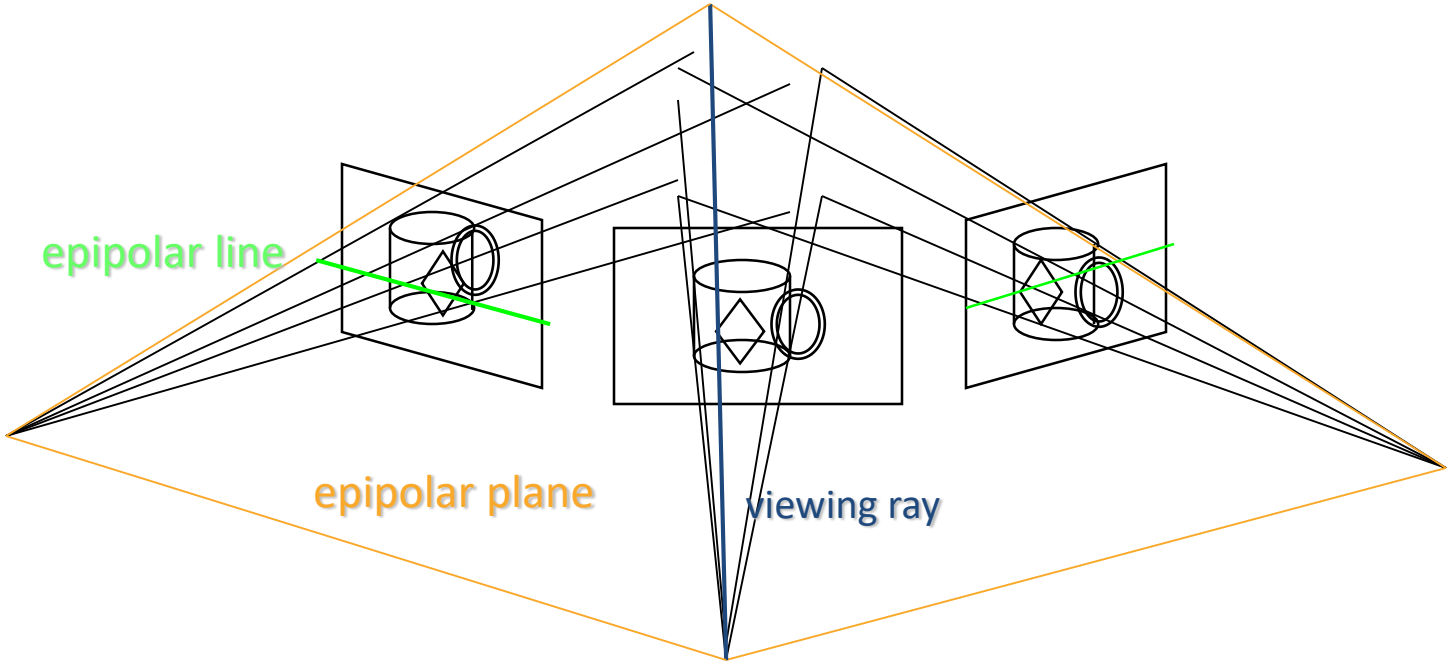
- Image projection
  - project image planes onto common (there are two degrees of freedom for this projection)
  - Goal: projection results in epipolar lines being horizontal image scans
    1. plane parallel to line between optical centers
  - a homography (3x3 transform) applied to both input images
  - pixel motion is horizontal after this transformation
  - C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



GEORGETOWN UNIVERSITY

Stereo: epipolar geometry

- Match features along epipolar lines



Rectification: Examples

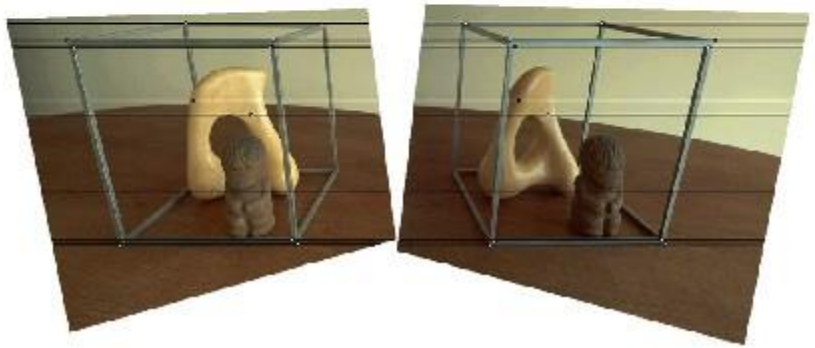
(a) Original image pair overlayed with several epipolar lines.

(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: Examples

(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 $\mathbf{H}_a$ and $\mathbf{H}'_a$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!

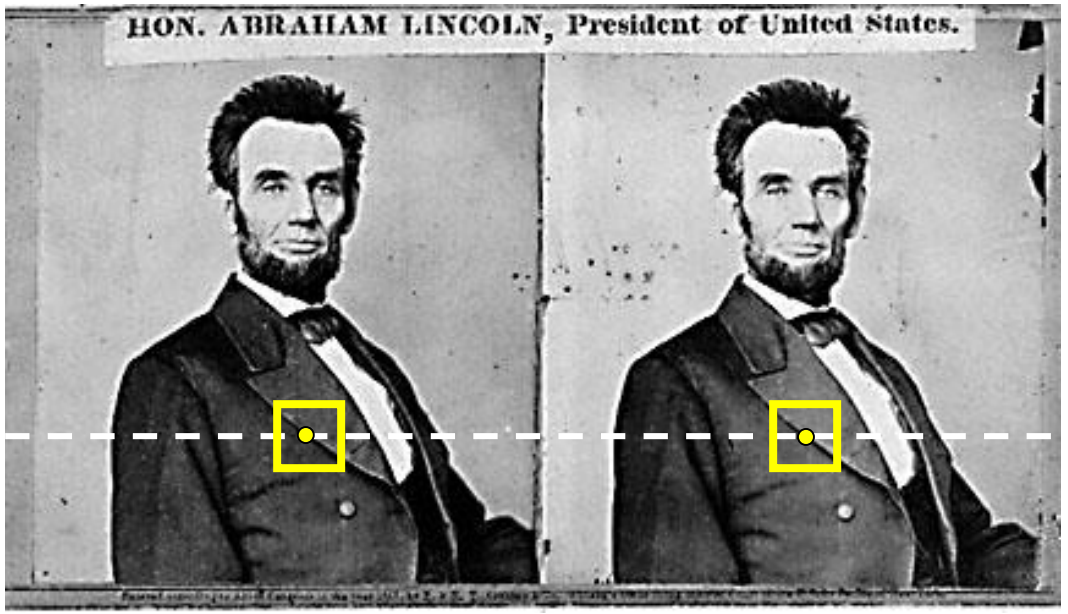
Outline

- Stereopsis
  - Motivation
  - Problems
  - Image rectification
- Matching criteria
  - Local Search (heuristic or brute force)
  - Energy Functional Minimization
    - Dynamic Programming
  - Graph Cuts
    - Min cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

Stereo Matching

- What are some possible algorithms?
  - match “features” and interpolate
  - match edges and interpolate
  - match all pixels with windows (coarse-fine)
  - Pose as optimization problem:
    - iterative updating
    - dynamic programming
    - energy minimization (regularization, stochastic)
    - graph algorithms

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

Improvement: match ***windows***

- Can use Lukas-Kanade (will discuss later!)

GEORGETOWN UNIVERSITY

Stereo matching algorithms

- Match Pixels in Conjugate Epipolar Lines
  - Assume brightness constancy
  - This is a tough problem
  - Numerous approaches
    - dynamic programming [Baker 81,Ohta 85]
    - smoothness functionals
    - more images (trinocular, N-ocular) [Okutomi 93]
    - graph cuts [Boykov 00]
  - A good survey and evaluation: http://www.middlebury.edu/stereo/

Matching criteria

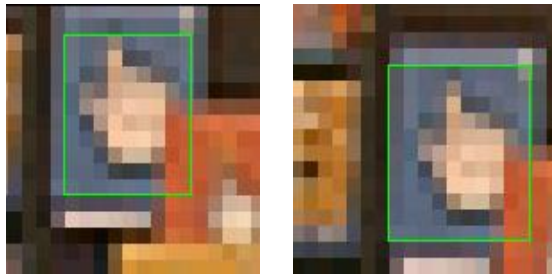
- Raw pixel values (correlation)
- Band-pass filtered images [Jones & Malik 92]
- “Corner” like features [Zhang, …]
- Edges [many people…]
- Gradients [Seitz 89; Scharstein 94]
- Rank statistics [Zabih & Woodfill 94]

Parameterize using disparity d

- 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?

Why is disparity d so important?

Stereo matching as energy minimization

- Matching Cost Formulated as Energy
  - "data" term penalizing bad matches

$$D(x, y, d) = \left|\mathbf{I}_L(x+d, y) - \mathbf{I}_\mathbf{R}(x, y)\right|$$

  - "neighborhood term" encouraging spatial smoothness

$$V(d_1, d_2) = \text{cost of adjacent pixels with labels d1 and d2}$$
$$= \left|d_1 - d_2\right| \quad (\text{or something similar})$$

$$E = \sum_{(x,y)} D(x, y, d_{x,y}) + \sum_{neighbors\ (x1,y1),(x2,y2)} V(d_{x1,y1}, d_{x2,y2})$$

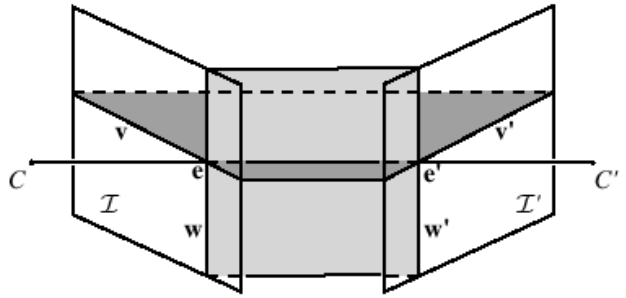
GEORGETOWN UNIVERSITY

Outline

- Stereopsis
  - Motivation
  - Problems
- Approaches
  - Image rectification
- Matching criteria
  - Energy Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
    - Graph Cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

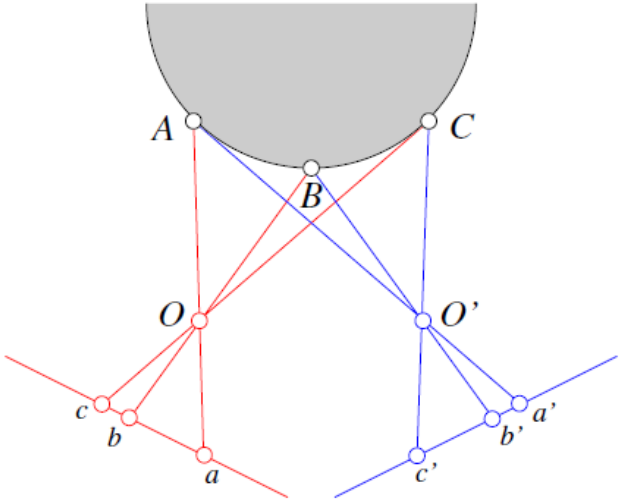
Dynamic Programming Solution

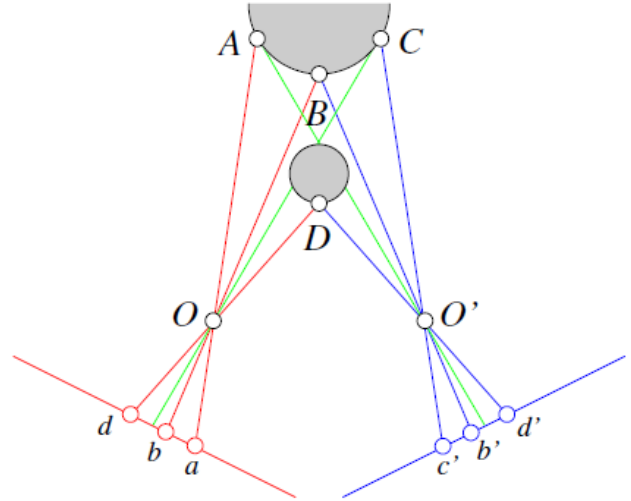
- Finding image pixel correspondences across epipolar lines is analogous to finding a min path across a graph where nodes are all possible pairwise correspondences.
- The path cost is the energy or error function.
- Smoothness constraints limit the nature of the path on the graph.
- Dynamic programming assumption:
  - Optimal path consists of optimal subpaths, assuming *ordering constraint* holds.



Ordering Constraint

- Ordering Constraint: Point correspondences should have the same relative order to other correspondences.
  - Not necessarily true in all cases, but assumed here.





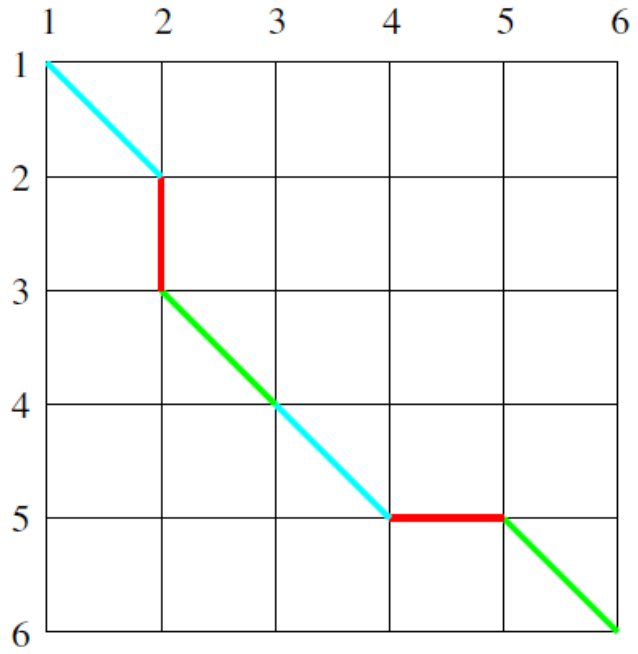
GEORGETOWN UNIVERSITY

Finding the optimal path

- Assume two images with epipolar lines with only 6 pixels (or 6 features). Find correspondences. Assume C(1,1) = 0, as boundary condition.

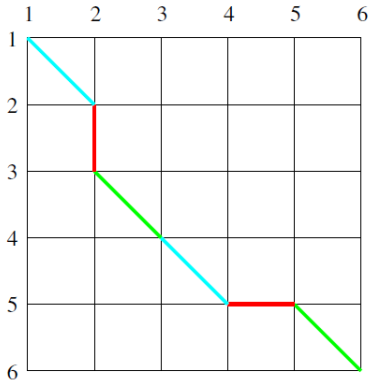
- Cost of adding an edge to the path
  - Pixel or feature difference (+ smoothness)

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



GEORGETOWN UNIVERSITY

Pseudo-code for DP Stereo Correspondence



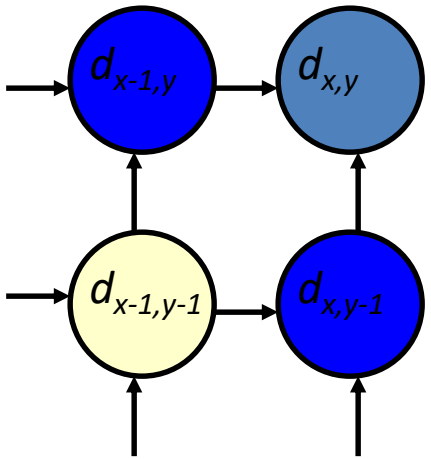
- Forward – Backward Algorithm
- Repeatedly, scan forward and find next correspondence based on cost/energy function.
- Keep track of back pointer and trace back to find optimal path.

```
% Loop over all nodes (k, l) in ascending order.
for k = 1 to m do
   for l = 1 to n do
     % Initialize optimal cost C(k, l) and backward pointer B(k, l).
     C(k, l) ← +∞; B(k, l) ← nil;
     % Loop over all inferior neighbors (i, j) of (k, l).
     for (i, j) ∈ Inferior-Neighbors(k, l) do
       % Compute new path cost and update backward pointer if necessary.
       d ← C(i, j) + Arc-Cost(i, j, k, l);
       if d < C(k, l) then C(k, l) ← d; B(k, l) ← (i, j) endif;
       endfor;
     endfor;
   endfor;
% Construct optimal path by following backward pointers from (m, n).
P ← {(m, n)}; (i, j) ← (m, n);
while B(i, j) ≠ nil do (i, j) ← B(i, j); P ← {(i, j)} ∪ P endwhile.
```

GEORGETOWN UNIVERSITY

Dynamic programming

- Can we apply this trick in 2D as well?



Not really: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

Outline

- Stereopsis
  - Motivation
  - Problems
- Approaches
  - Image rectification
- Matching criteria
  - Energy Functional Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
    - Graph Cuts
- Discussion: Statistical Approaches
- Other non-stereo schemes

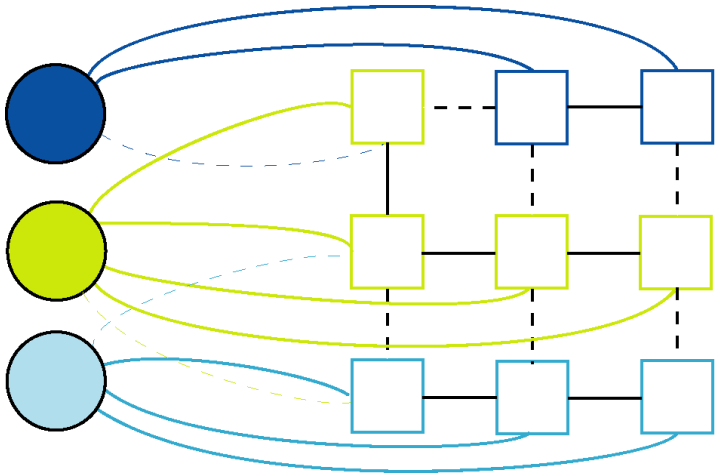
Graph cuts

- Solution technique

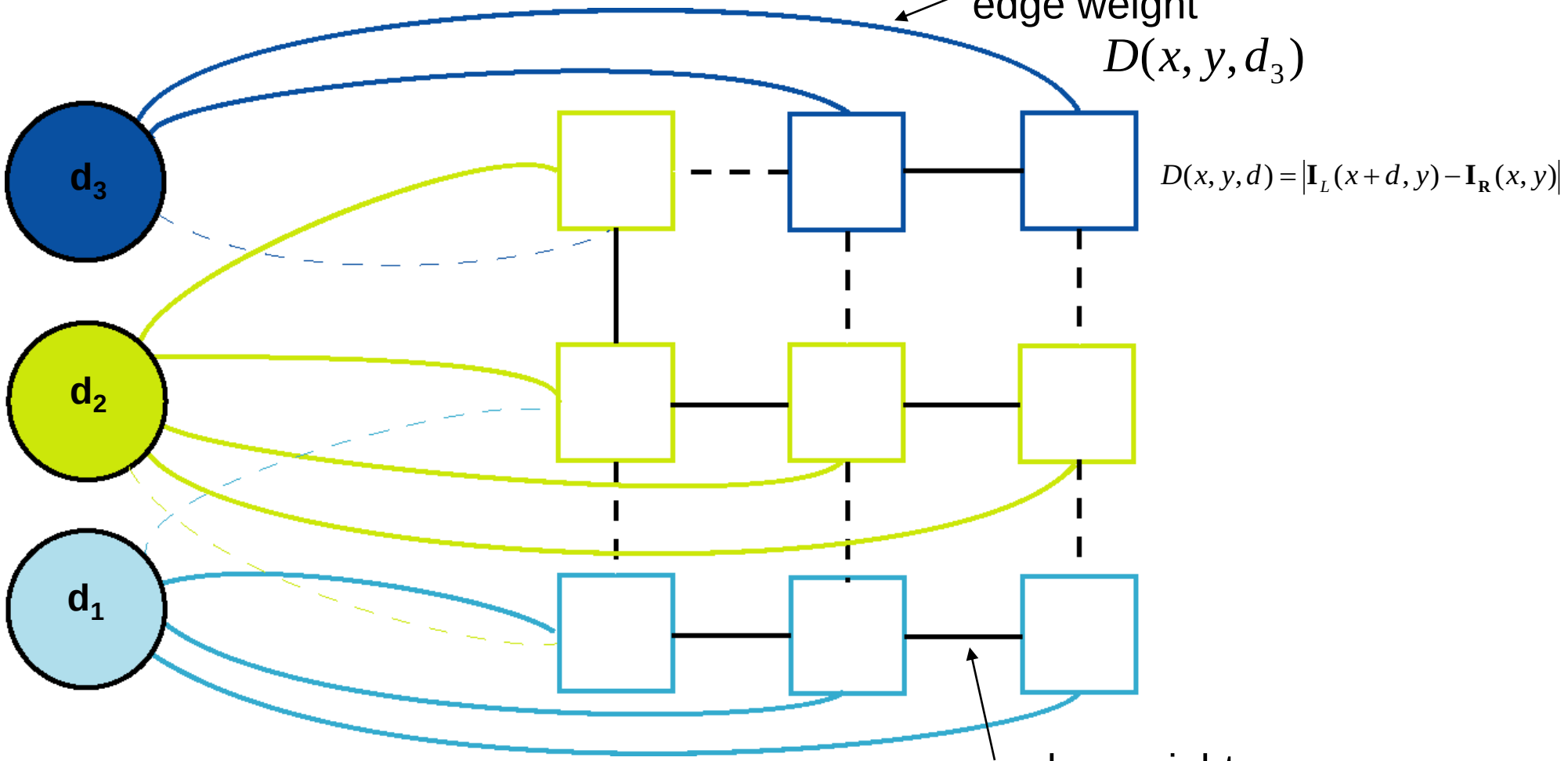
$$E_{\text{total}}(\mathbf{d}) = E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d})$$

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

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



Stereo as a graph problem *[Boykov, 1999]*



- **Pixels**

Labels

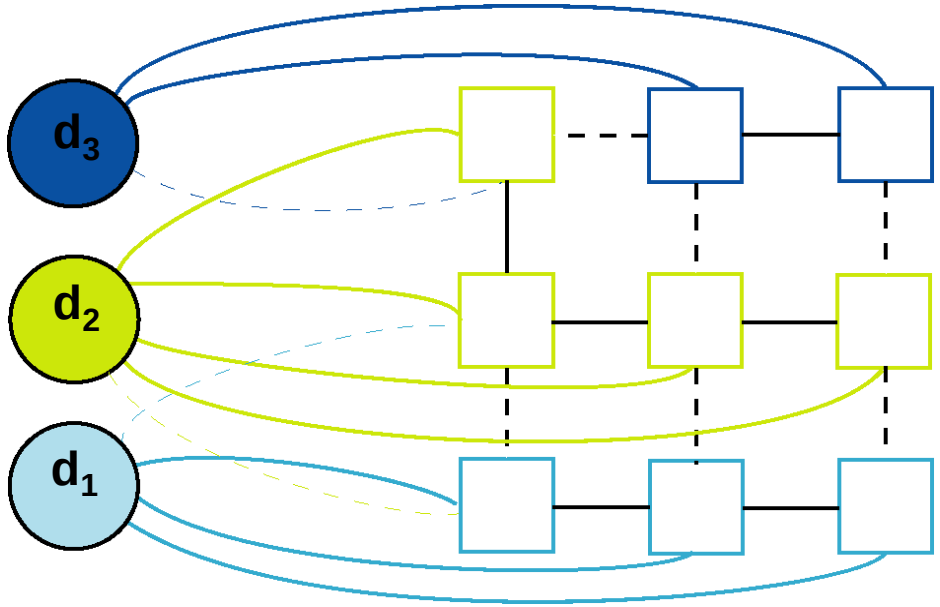
(disparities)

$V(d_i, d_{i-1})$ = cost of adjacent pixels with labels d1 and d2

$= |d_i - d_{i-1}|$ (or something similar)

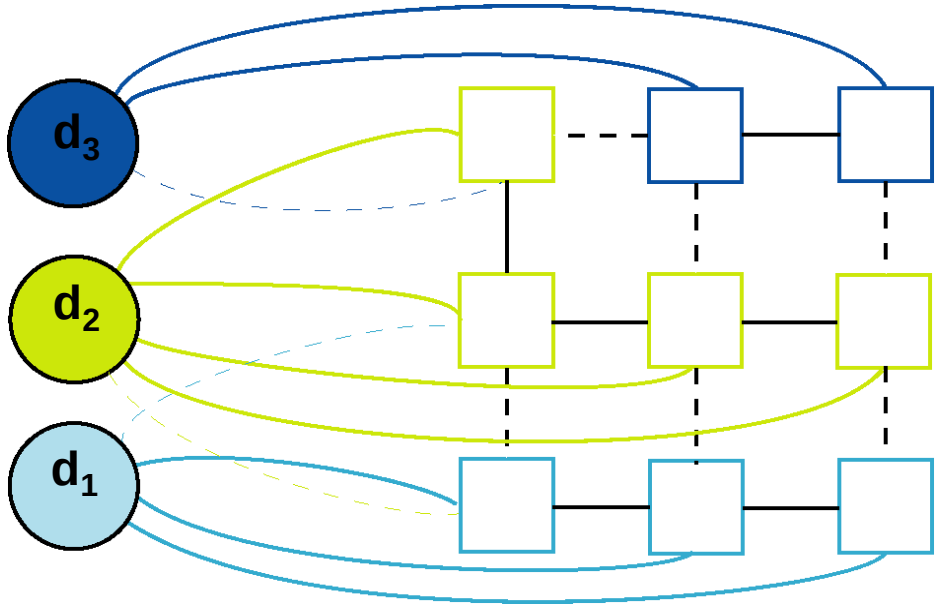
GEORGETOWN UNIVERSITY

Graph definition



- Initial state
  - Each pixel connected to it's immediate neighbors
  - Each disparity label connected to all of the pixels

Stereo matching by graph cuts



- Graph Cut
  - Delete enough edges so that
    - each pixel is (transitively) connected to exactly one label node
  - Cost of a cut: sum of deleted edge weights
  - Finding min cost cut equivalent to finding global minimum of the energy function

Computing a multiway cut

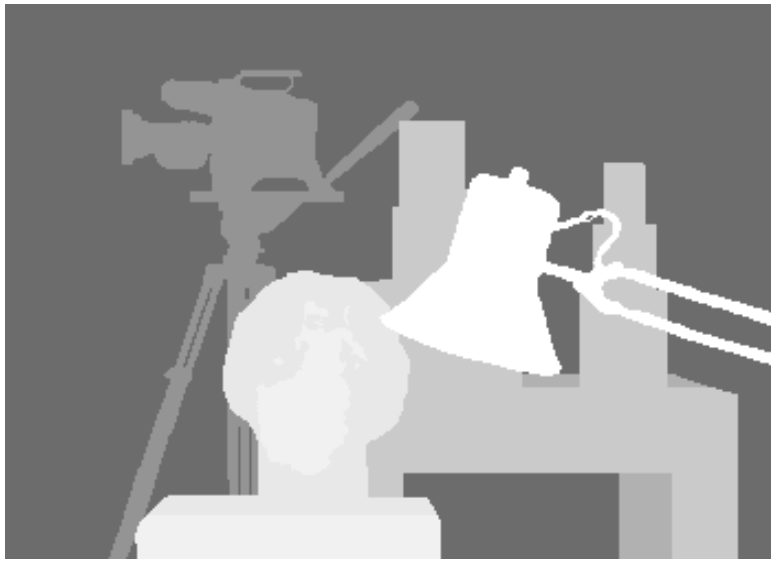
- With two labels: classical min-cut problem
  - Solvable by standard network flow algorithms

- More than 2 labels: NP-hard [Dahlhaus *et al.*, STOC ‘92]
  - But efficient approximation algorithms exist
    - Yuri Boykov, Olga Veksler and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.
  - Basic idea
    - reduce to a series of 2-way-cut sub-problems

Stereo results

- Data from University of Tsukuba
- Similar results on other images without ground truth

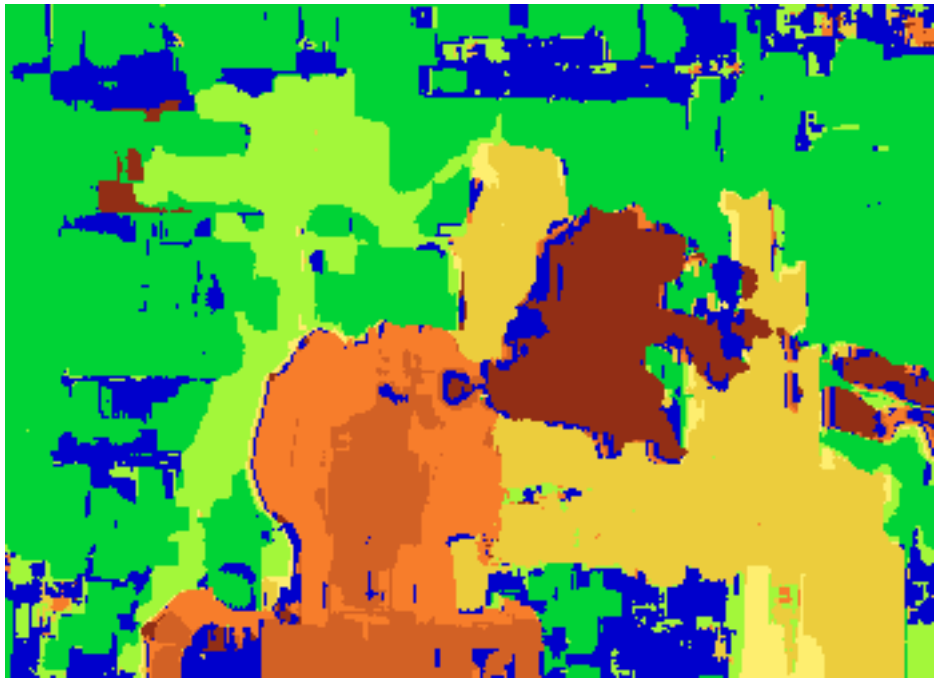
Scene



Ground truth

GEORGETOWN UNIVERSITY

Results with window search



Window-based matching
(best window size)

Ground truth

GEORGETOWN UNIVERSITY

Better methods exist…

State of the art method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

GEORGETOWN UNIVERSITY

Outline

- Stereopsis
  - Motivation
  - Problems
- Approaches
  - Image rectification
- Matching criteria
  - Energy Functional Minimization
    - Local Search (heuristic or brute force)
    - Dynamic Programming
  - Graph Cuts
    - Min cuts
- Discussion: Other Approaches

Other Approaches

- Variational Approach
  - Find function that optimizes energy functional
  - Common vision constraints are interpreted as constraints on kth derivatives of the function.
    - Regularization

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

- Probabilistic Approach
  - Energy minimization is analogous to maximizing probabilities

Example: Bayesian inference

- Formulate as statistical inference problem
- Prior model $p_P(\boldsymbol{d})$
- Measurement model $p_M(l_L, l_R | \boldsymbol{d})$
- Posterior model
  - $p_M(\boldsymbol{d} \mid l_L, l_R) \propto p_P(\boldsymbol{d})\, p_M(l_L, l_R | \boldsymbol{d})$
- Maximum a Posteriori (MAP estimate):
- maximize $p_M(\boldsymbol{d} \mid l_L, l_R)$

Measurement model

- Likelihood of intensity correspondence

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

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

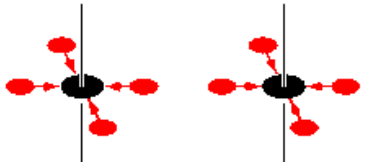
- Corresponds to Gaussian for quadratic ρ

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

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 probabilistic 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]

Other Methods for Extracting Depth Information

- When stereo imagery is not available, we can rely on other image characteristics to infer depth information

- Depth from
  - Texture
  - Shading
  - Motion
  - Focus
  - Visual Cues (and Scene Understanding)

Appendix

Jeremy Bolton, PhD
Assistant Teaching Professor

GEORGETOWN UNIVERSITY