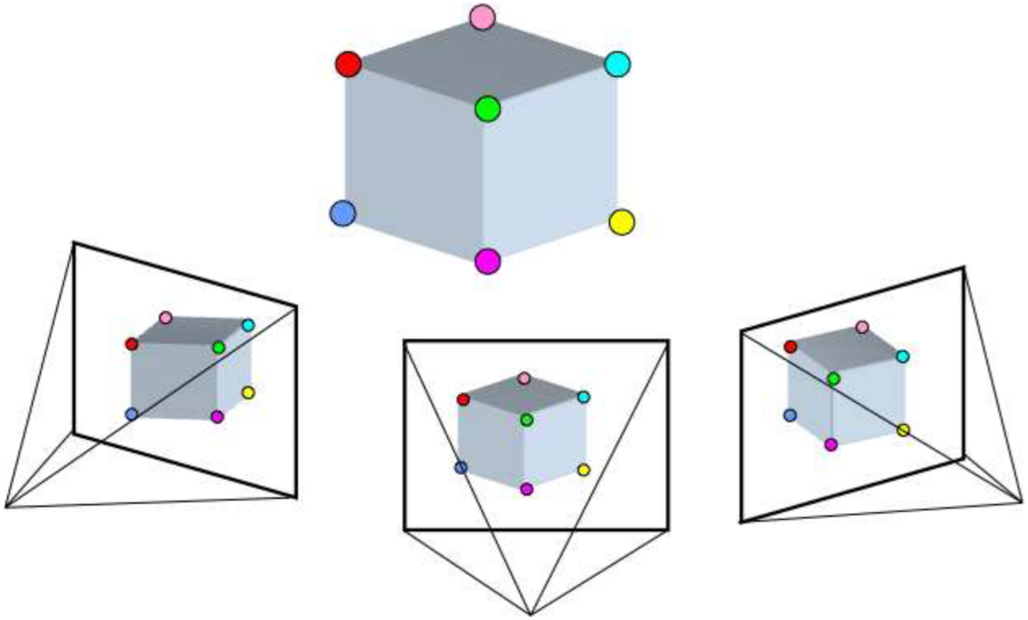
CSE578: Computer Vision

Spring’17

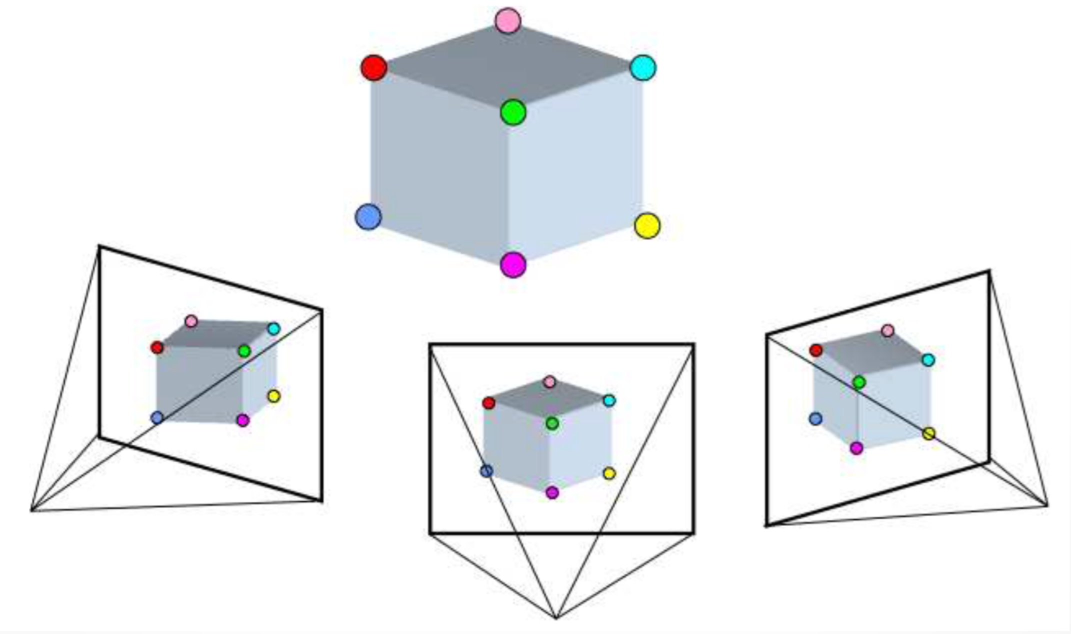
Multiple-View Structure Recovery



Anoop M. Namboodiri
Center for Visual Information Technology
IIIT Hyderabad, INDIA

Multiple Views of Points/Objects

- Given projections of a set of 3D points in two or more cameras, get their 3D coordinates.
- Each 3D point is identified in every camera view.
- What else is known? Camera matrices K_i, R_i, t_i? Only the intrinsic parameters K_i?



Variations of the Problem

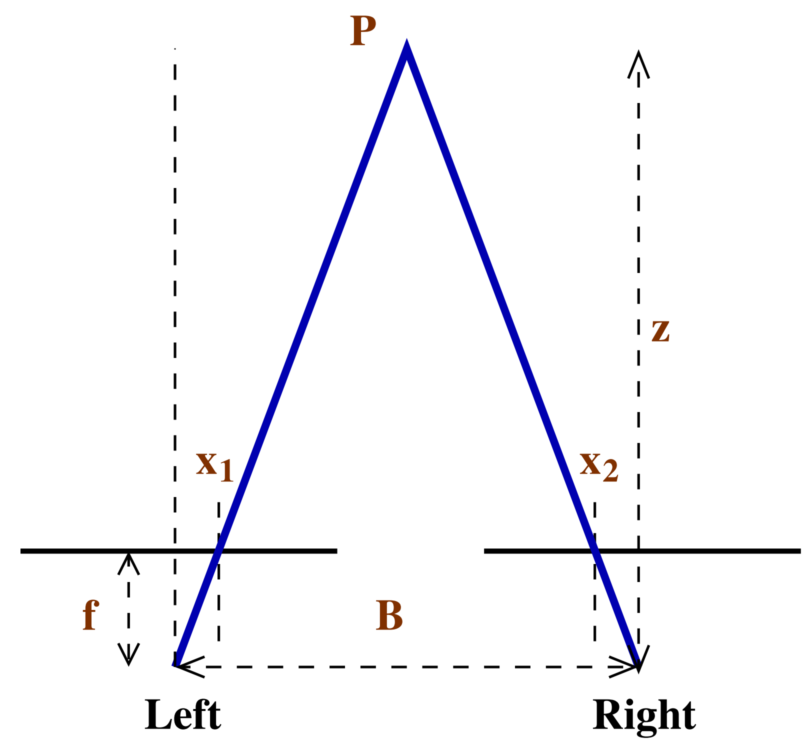
- **(Binocular) Stereo:** Two cameras with known intrinsic and extrinsic parameters.
- **Multiview Stereo:** Multiple known cameras
- **Structure-from-Motion:** Given m cameras and n points and projections $\boldsymbol{x}_{ij}$ of point $\boldsymbol{j}$ in camera $\boldsymbol{i}$, recover 3D points $\boldsymbol{X}_j$ and camera matrices $\boldsymbol{C}_i$
  - **Affine SfM:** For affine cameras
  - **Projective SfM:** For general projective cameras
- **Bundle Adjustment:** Directly recover $\boldsymbol{C}_i$, $\boldsymbol{X}_j$ from $\boldsymbol{x}_{ij}$

Binocular Stereo and Feature Correspondence

Geometry of Matching

- B: baseline, f: focal length, z: depth, x: image coords
- From similar triangles:

$$\frac{x_1 - x_2}{f} = \frac{B}{z}$$



- **Stereo Disparity or Parallax**: the "shift" between the left and right images. $\Delta = \frac{Bf}{z}$.
- Farther the point, smaller the disparity and vice versa
- A large baseline can give more reliable estimates of depth. But, matching may become harder
- Basic step: **Identify common points in camera views**

Identifying Common Points

- Find a world point in 2 or more views
- Appearance is the only clue to identify them
- Individual pixel colours are similar very often. Match is too noisy
- Match a (small) neighbourhood of colours from one image to a similar neighbourhood in others
- Will work if local surface is fronto-parallel and images have similar magnification
- Foreshortening can happen when viewing an oblique surface
- Many ambiguities. We need a lot of help!

Some Examples





Matching Patches

- Compare $m \times m$ patches from two views.
  Form vectors $\mathbf{v}$ and $\mathbf{v}'$ of length m^2 from them
- Matching scores between patches:
  - Sum of Absolute Difference (SAD): $||\mathbf{v} - \mathbf{v}'||_\mathbf{1}$
  - Sum of squared difference (SSD): $||\mathbf{v} - \mathbf{v}'||_\mathbf{2}$
  - Correlation: $\dfrac{\mathbf{v}'^\mathbf{T}\mathbf{v}}{\sqrt{\mathbf{v}^\mathbf{T}\mathbf{v}}\sqrt{\mathbf{v}'^\mathbf{T}\mathbf{v}'}}$
  - Normalized correlation: $\dfrac{\bar{\mathbf{v}}^\mathbf{T}\bar{\mathbf{v}}'}{\sqrt{\bar{\mathbf{v}}^\mathbf{T}\bar{\mathbf{v}}}\sqrt{\bar{\mathbf{v}}'^\mathbf{T}\bar{\mathbf{v}}'}}$. Range: $[-1, 1]$

    $\bar{\mathbf{v}}, \bar{\mathbf{v}}'$ are vectors with respective patch-mean colour subtracted.
  - Invariant to affine changes in intensity/colour.

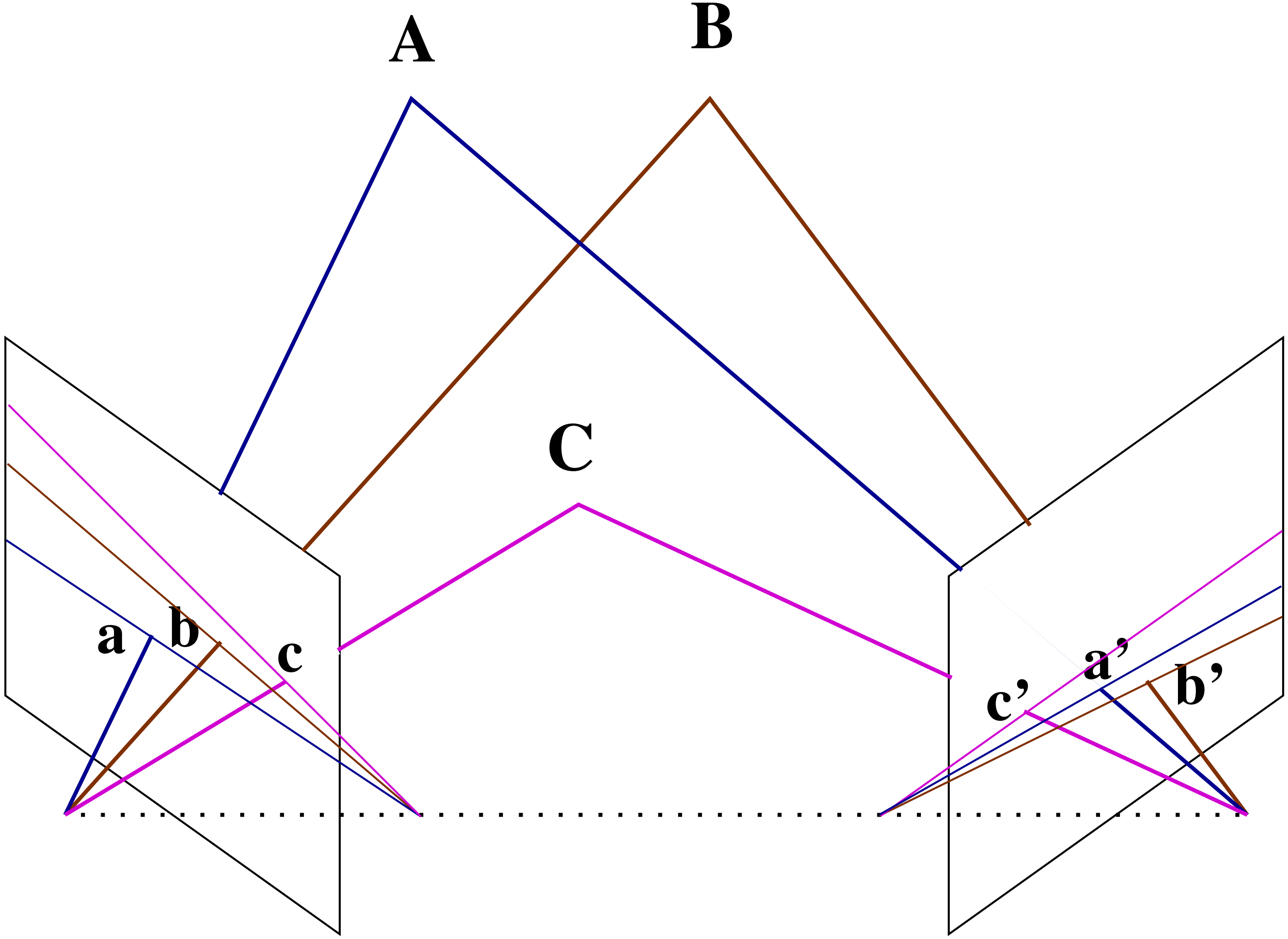
Constraints on Matching

- **Epipolar:** Match lies on the epipolar line of the pixel
- **Colour Constancy:** The appearance/colour does not change from one view to another
- **Uniqueness:** A point on left image can match with only one point on the right and vice versa
- **Ordering or Monotonicity:** If point A is to the left of B in view 1, it will to the left of B in view 2 also. (Violated if great difference in depth exists)
- **Continuity:** Disparity values vary smoothly (violated at occlusion boundares)

Sparse correspondence: only for good feature points

Dense correspondence: a match for every pixel

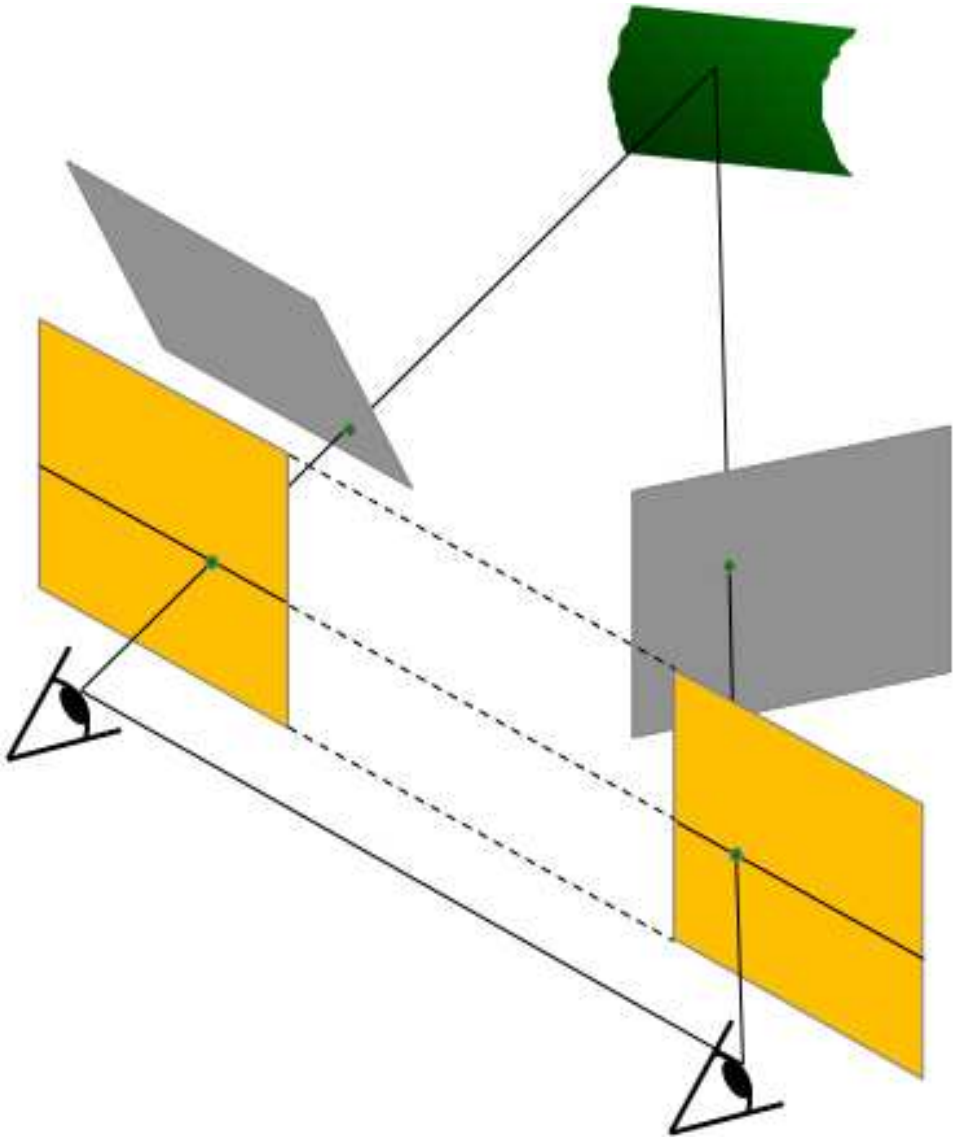
Epipolar Geometry



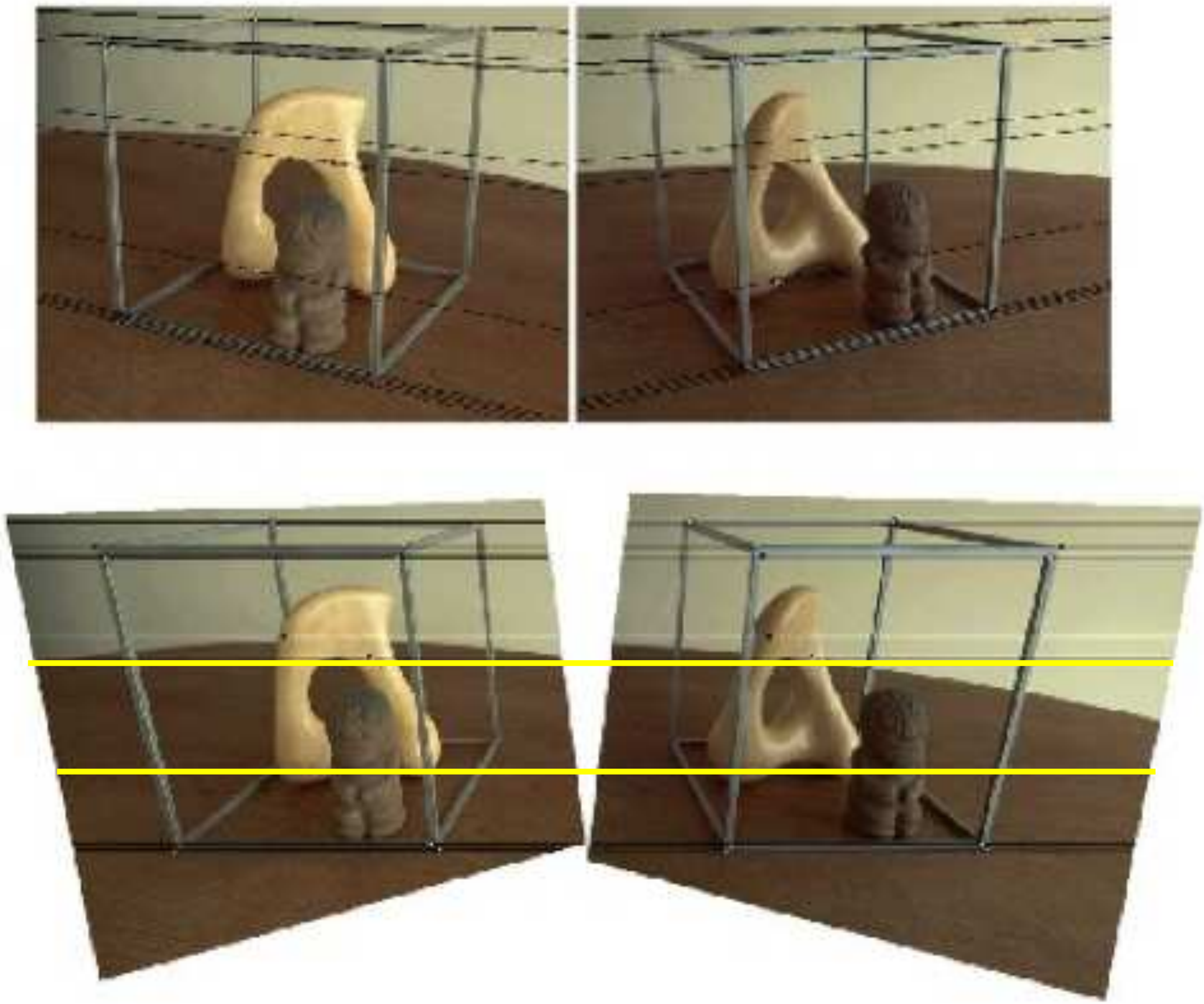
Reduced Search and Rectification

- The search is limited to a line if fundamental matrix known (i.e., **weakly calibrated**)
- Simplest if left and right cameras have same image plane and pure $\mathbf{X}$-translation between them.
- Fundamental matrix has a simple form.
  Epipolar constraint reduces to $y' = y$.
- Matches constrained to lie in the same scan line
- **Rectification:** A rotation of the camera (to make image planes parallel) and a change in $\mathbf{K}$ matrix (focal length, image center).
- Can be represented using a homography $\mathbf{H}$ to align one image plane to the other
  Or, homographies $\mathbf{H_1}, \mathbf{H_2}$ to align them to a third plane

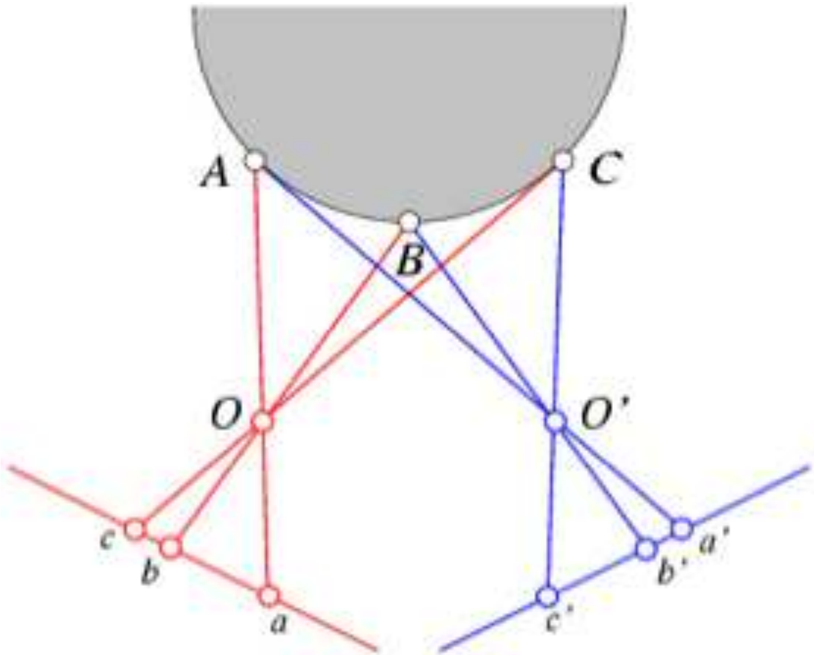
Rectification

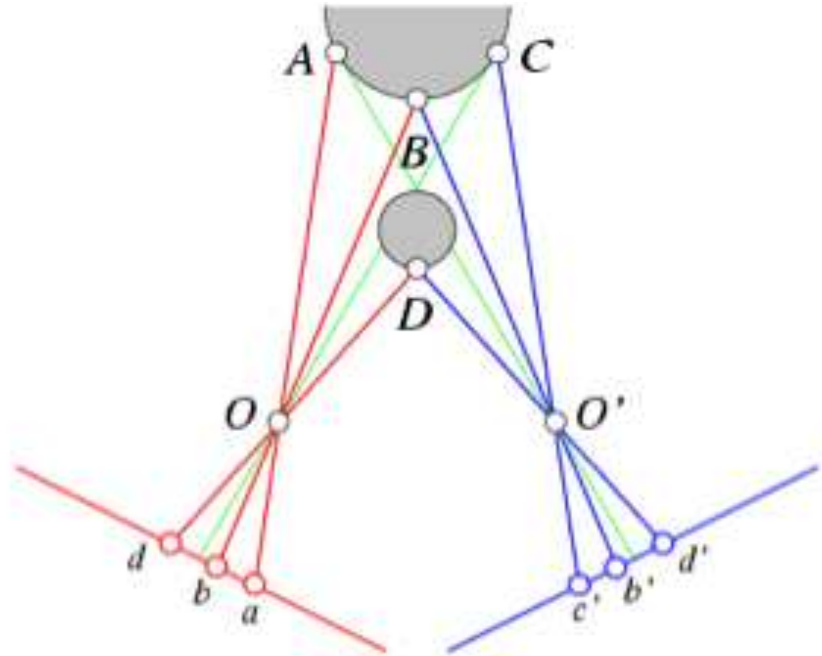


Rectification: Example



Ordering Constraint





Order of matches from left and right is ordinarily preserved
Violation may mean something else.

Various Situations

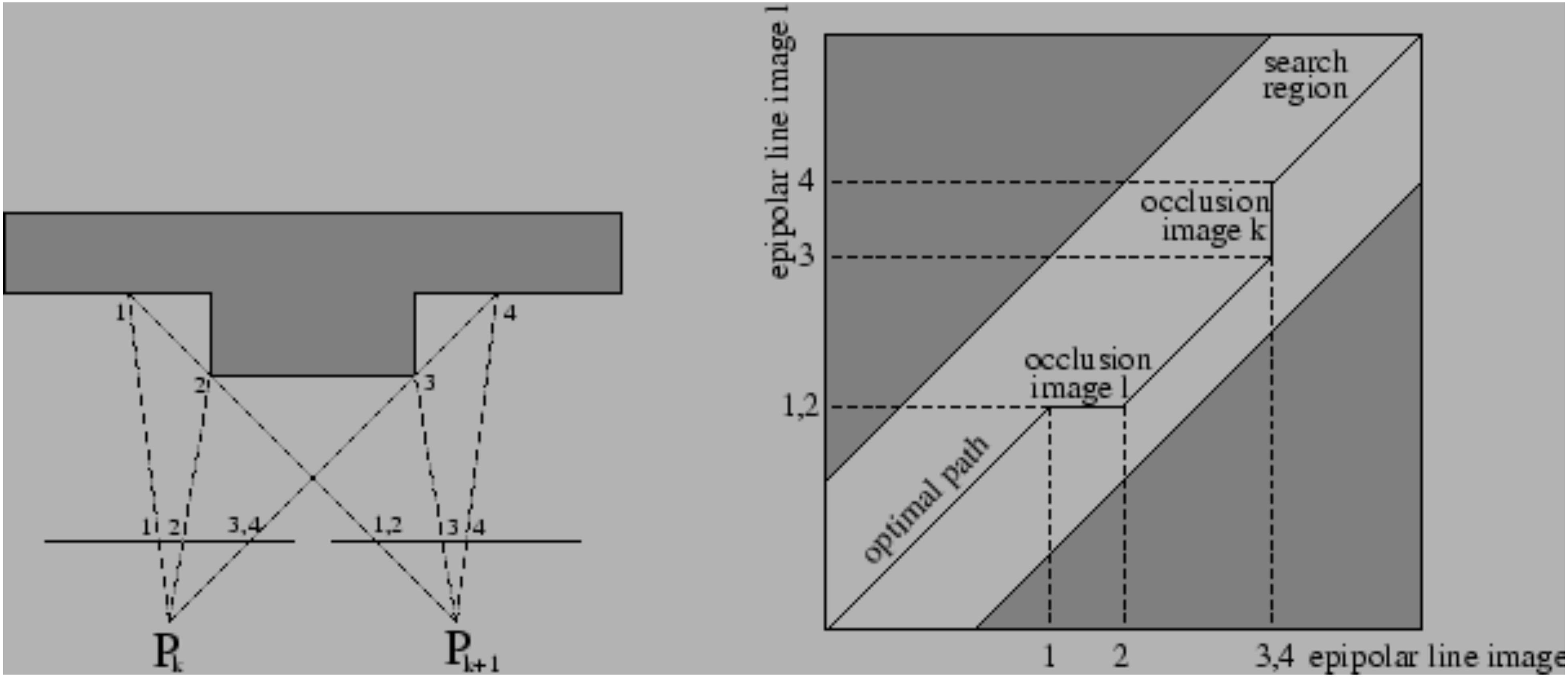


Image courtesy Marc Pollefeys

Shows the *epipolar line image* or *disparity space image* with different scenarios

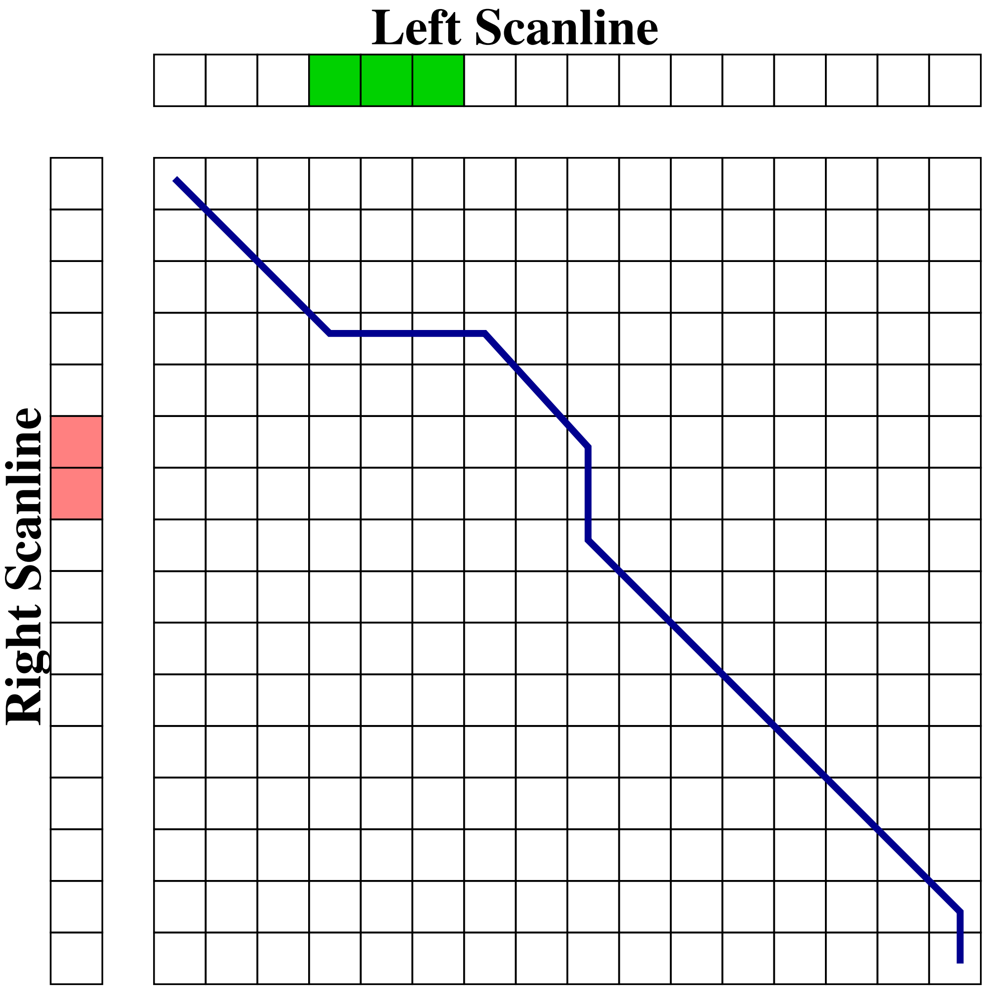
Scan-Line to Scan-Line Matching

- Disparity Line Image pits pixels of one row of the left image against the pixels of the matching epipolar line in the other.
- Several matching scenarios:
  - If left pixel $(i-1)$ matches with right pixel $(j-1)$, next pixel i can match pixel j, if the match is good
  - Otherwise, it may continue the match with $(j-1)$ with an occlusion cost (due to left occlusion)
  - Or, $(i-1)$ can match with j with another occlusion cost (due to right occlusion)
- Sub-pixel definitions may be needed when zoom is different

Dynamic Programming Solution

- Cost of matching: $C(i-1, j-1) + c(i, j)$ if pixels match, $C(i-1, j) + C_o$ if left occlusion, and $C(i, j-1) + C_o$ if right occlusion, where C_o is a high occlusion cost
- Select the minimum from those three and declare match or occlusion accordingly
- Can be setup nicely as a dynamic programming solution working in the i, j space, starting with leftmost pixel match
- Cost of matching: $O(N^2)$ where N is the number of pixels in each scanline.

Dynamic Programming Path



- Initialize first row and col to $i * C_o$
- Do for $i \in [0, N-1]$ and $j \in [0, N-1]$:

  Set $C(i,j)$ to min of $C(i-1, j-1) + c(i,j), C(i-1, j) + C_o, C(i, j-1) + C_o$
- Mark each as **M/L/R**
- Reconstruct from (N, N), by folllowing the **M** pixels and their connections.

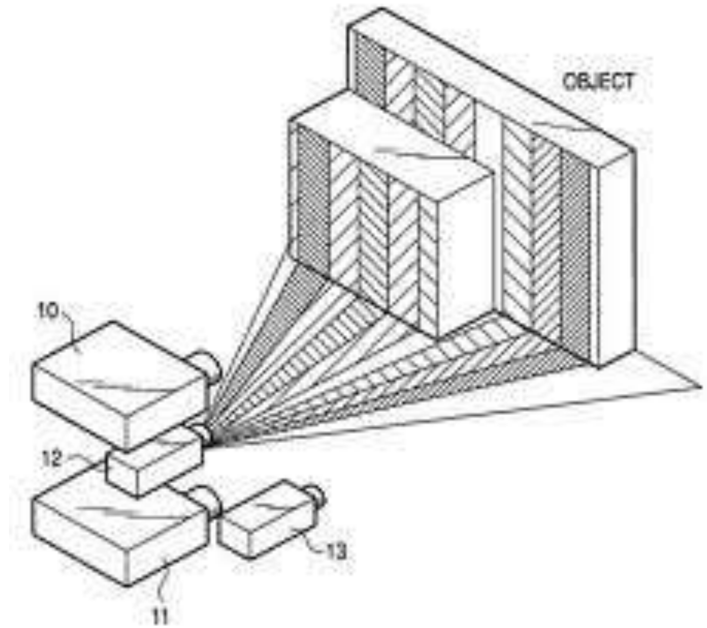
Globally Optimal Solution

- Provides a *globally optimal* match as opposed to the local matching done by search
- Provides dense correspondence: a match for every pixel
- Works well enough. And is a prototype for many global stereo matching approaches that followed
- Difficulty: assigning a cost for occlusions.
- Difficulty: maintaining consistency across scan lines

We will see another global solution using graphcuts later

Structured Lighting

- Finding correspondences is hard by itself
- Can we help it by projecting patterns onto the world? **Structured Lights!**
- Lightstrip range finders, etc.
- Combination of sinusoids sometimes to get dense matches
- *Active vision*, as it changes the appearance
- The light projected need not be in the visible spectrum



Xbox Kinect

IR-based range sensor for Xbox

- Aligned depth and RGB images at 640×480
- Original goal: Interact with games in full 3D
- Computer vision happy with real-time depth and image
  - Games, HCI, etc
  - Action recognition
  - Image based modelling of dynamic scenes
- Fastest selling electronic appliance ever!!
- Other products that use PrimeSense sensor